Design of Therapeutic Training Sequences for Infants Using a Visual Approach

Eugen Richter, Luiza Mici(✉), Norman Hendrich, and Jianwei Zhang

Computer Science Department, University of Hamburg,
Vogt-Koelln-Str. 30, 22527 Hamburg, Germany
{erichter,mici,hendrich,zhang}@informatik.uni-hamburg.de

Abstract. In this paper we present and discuss the design, implementation and evaluation of a visual programming environment. The proposed application allows authoring of therapeutic training sequences for the CareToy system, a hardware system developed for the purpose of in-home rehabilitation training for infants, aiming to improve the motor skills of preterm infants diagnosed with neurological conditions.

Keywords: CareToy · Tele-rehabilitation · Preterm infant stimulation · Early intervention · Motor skills · Visual programming

1 Introduction

Stroke and other neurological conditions affect the population of preterm infants to a non-marginal extent. Literature suggests the application of treatment at an early stage in form of rehabilitative care to improve the development process of the infant [1,2]. Healthcare costs and the shortage of necessary healthcare professionals however, are limiting factors to a wide-spread availability of this type of care. To approach these problems and improve the development of motor skills in neurologically affected preterm infants the CareToy hardware system [3], a gym-based solution equipped with visible and audible stimuli as well as sensorized toys, has been developed.

The CareToy system presents a solution allowing the administration of patient-tailored care that is remotely monitored, evaluated and adjusted by a healthcare professional. The aim of the CareToy project is to exploit new technologies to promote early intervention using a portable low-cost instrumented system [4], tele-monitored by healthcare professionals administering focused therapy that is usually performed in medical institutions [5]. Figure 1 outlines the general setup. The CareToy system has been designed with the idea to allow families with infants requiring rehabilitative care to bring the system into their home and let the infant playfully perform a set of specific tasks tailored to its individual needs.

One key element of the CareToy system is its modular design. It is composed of several smart, sensorized hardware components that allow the system to stimulate the infant as well as monitor and measure its progress in a non-invasive

P. Cipresso et al. (Eds.): MindCare 2014, LNICST 100, pp. 145–154, 2014.
DOI: 10.1007/978-3-319-11564-1_15

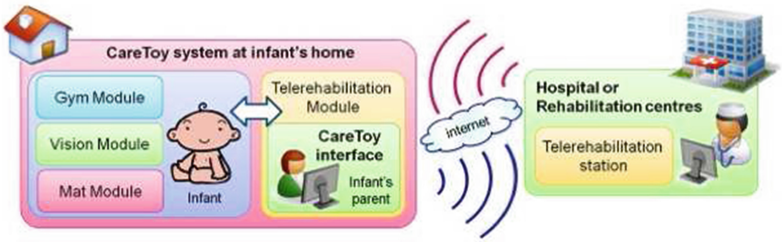

Fig. 1. The general setup of the CareToy system. The system used at home (left) is remotely connected to the medical site (right) with healthcare professionals monitoring the progress of the infant and providing individual therapeutic training sequences.

and transparent way. The primary objectives of the CareToy system are the measurement and stimulation of three main functions:

- **Grasping forces**
- **Gaze**
- **Postural control**

A gym-based structure (gym module) equipped with embedded visible and audible stimuli sources, sensors and cameras is used as the basis. Mechatronic toys containing sensors for force and pressure are used for grasp evaluation and feedback. A vision module consisting of four cameras is used for the recording of video data during the execution of training sequences serving as the basis for visual performance evaluation. A tactile mat (mat module) is used to measure and promote postural control. The tele-rehabilitation module completes the system offering remote communication capabilities for secure data transmission.

Each component can be used on its own or in combination with others depending on the designed therapeutic training sequences. However, despite being very end user friendly the system's flexibility makes it non-trivial with regard to programmability. Initial study of requirements obtained from healthcare professionals has shown the need for tools that offer a high level of hardware abstraction, yet are flexible enough for the professional to apply her/his expert knowledge towards the creation of programmed behavior tailored to the infant's needs.

The visual programming paradigm is a valid approach to development of such tools, providing better organization and a more explicit presentation of information [14]. Despite the lack of a completely visual programming language and the lack of an exhaustive body of empirical data for the validation of proposed advantages of the visual programming approach [10,12], literature indicates that users without technical background knowledge generally benefit from the use of a visual programming environment [9,14].

In order to make the execution of programming tasks and the understanding of programmed system behavior easier for the non-technical healthcare professional a new visual programming environment was developed for the CareToy system. Feedback received from healthcare professionals has shown the developed application to provide the necessary level of abstraction and flexibility allowing development of programmed behavior of the system without any technical expert assistance.

Section 2 of this paper presents background information, a brief overview of related work and context information related to the CareToy project as the platform for the developed visual programming environment. Section 3 describes the features and the design and implementation specifics of the proposed visual programming solution. Section 4 presents the evaluation methodology and discusses the obtained experiment results. Section 5 provides the conclusions.

2 Background

The field of visual programming is a well researched one with a variety of existing domain and task specific solutions. *LabVIEW* (G language) represents a well known visual programming language (VPL) with interfaces for instrumentation hardware. *Scratch* describes a multimedia oriented VPE for educational purposes [13]. Due to the end user friendliness, use of VPLs is gaining traction in the field of robotics as well. Microsoft's *Robotics Developer Studio* includes a VPL as an option for robotic applications using a dataflow-based approach. *Ruru* is proposed as an interactive VPL geared towards robotics novices [8]. RoboStudio is being used as a VPE for service authoring on personal service robots [6]. Unrelated to the field of robotics *Officeaid* represents a VPE aiming at reducing development effort, providing end users with an easy-to-use icon-based visual option for self-programming of office tasks [7]. A more recent VPE for the purpose of various processing tasks related to 'Big Data' by non-technical end users is presented in [11].

All controllable components within the CareToy hardware system (shown in Fig. 2) are able to produce visible and audible stimuli (LEDs, video and sound playback) in order to stimulate the infant's perception and gain its attention. Additionally, toy devices are equipped with sensors that can be configured to acquire force, pressure or activation feedback from the infant. Hence, the visual programming environment needed to provide abstract interfaces for each stimulating property of a component, for the definition of feedback events and for runtime related specification. Due to their domain and task focused design none of the available products or proposed solutions was found capable to provide an out-of-the-box solution.

The initial idea to integrate a simplified textual programming language was discarded based on the finding that with increasing complexity of the training sequence, recognition and alteration of programmed training sequences quickly became unmanageable for healthcare professionals.

The developed visual programming environment follows the icon-based visualization principle representing each hardware component as a fully configurable block. Complex blocks were introduced allowing to establish hierarchy within the sequence. An automatic layout feature was implemented to round off the visual

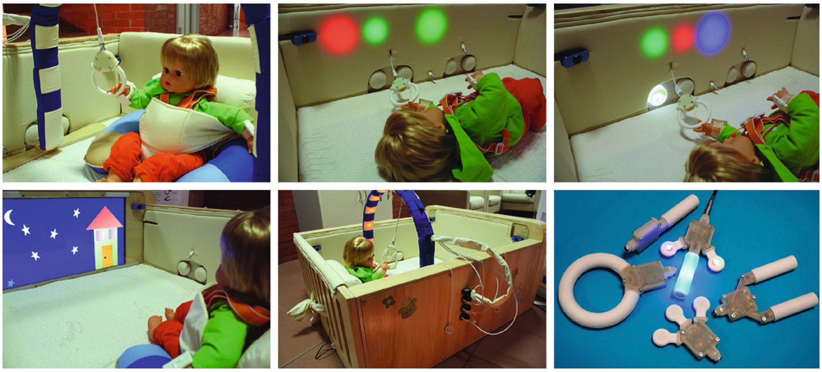

Fig. 2. Components of the CareToy hardware system being used within training sequences. (1) Interaction with the ring toy in sitting position, (2+3) interaction with the ring toy with visual stimuli from the left wall, (4) focus on the screen wall in sitting position, (5) interaction with the ring toy with visual stimuli from the arch. (6) The sensorized toy devices.

end-user experience. The created application was developed with a generic interface for the specification of hardware block information using manually written configuration files containing the specification data of the represented hardware component. Use of the XML standard allows full modification of the specification data, offering potential reusability of the developed software solution.

3 Design and Implementation

In order to offer an easy and visual way for programming of therapeutic training sequences to healthcare professionals without programming experience, the initial approach taken was to restrict the domain of the visual notation. The main element of the visual notation architecture is the block, which corresponds to a configurable hardware entity of the system. Geared towards the CareToy hardware system, following categories and instances of blocks were included:

- **Toys** (Small ring, big ring, Mickey, U-shape)
- **Walls** (Left wall, right wall)
- **Arch**
- **Screen**

Each included block instance contains configuration options for the stimuli of the corresponding hardware entity (e.g. different types of colored lights, button lights, etc.). Additionally each of the blocks has various internal control parameters related to the block itself as well as the stimuli (e.g. activation time, duration interval, color, size, volume, etc.), which can be easily set through the corresponding configuration window of the block in question. Furthermore, the configuration window offers the option to enable an event-action mechanism for

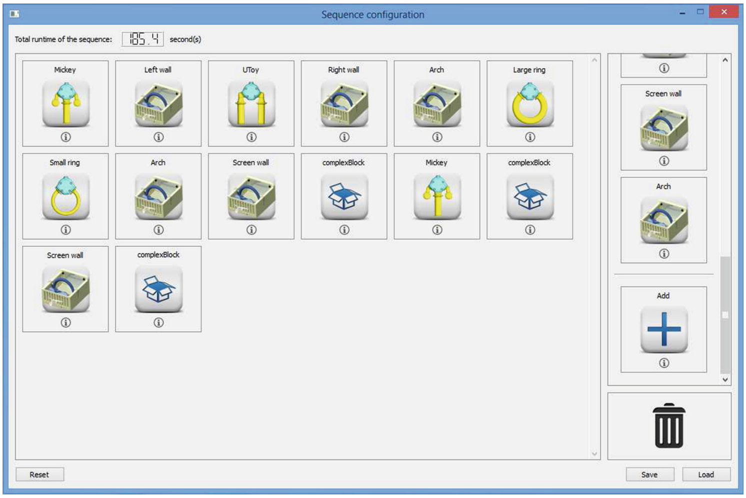

Fig. 3. Main view of the visual programming environment with a therapeutic training sequence defined on the main canvas. The blocks palette (right side) contains a list of blocks corresponding to hardware system entities together with user-defined complex blocks.

the acquisition and real-time processing of the infant's feedback, with actions corresponding to activation of stimuli.

In addition to the previously described blocks the type of a complex block was added to the visual notation architecture, allowing to introduce hierarchy into the programmed sequence.

A training sequence is authored as a sequence of blocks, with the aim to produce an output of stimuli gaining the infant's attention and thereby motivating coordinated motion leading to an improvement of motor skills. After the evaluation of the progress of an infant's motor development (e.g. through the exhaustive set of data obtained from the CareToy trial program) these training sequences can be further tailored to the infant's individual needs.

In order to allow the result data obtained during the CareToy trial program to be used for an easier tracking of the infant's progress, performance comparability and the inherent validation of the developed system, the application has been restricted to the design of sequential programmed behavior.

3.1 User Interface

The proposed visual programming environment displayed to the healthcare professional is illustrated in Fig. 3 and is composed of the following components:

- **The main canvas**: The canvas is the central part of the VPE where the configuration blocks dragged from the palette can be dropped onto and then

moved around freely or copied until the desired sequence of blocks is completed. The sequence is interpreted from left to right and top to bottom. A double click on any of the blocks will open the block configuration window.
- **The blocks palette**: The palette displays a list of blocks corresponding to hardware system entities together with user-defined complex blocks. A special *Add* block is also included for the purpose of creation of user-defined complex blocks.
- **The trash can**: Dragging blocks from the main canvas onto the trash can icon leads to the deletion of the block from the current sequence.

The block configuration window illustrated in Fig. 4 allows configuration of the duration of the block, the number of block repetitions, the stimuli to be used during the execution of the block and the handling of feedback provided by the infant. Furthermore, a summarizing description of the block can be inserted into its dedicated text field. That description can later be found as the block's tool-tip text when returning to the main canvas, facilitating an easier comprehension.

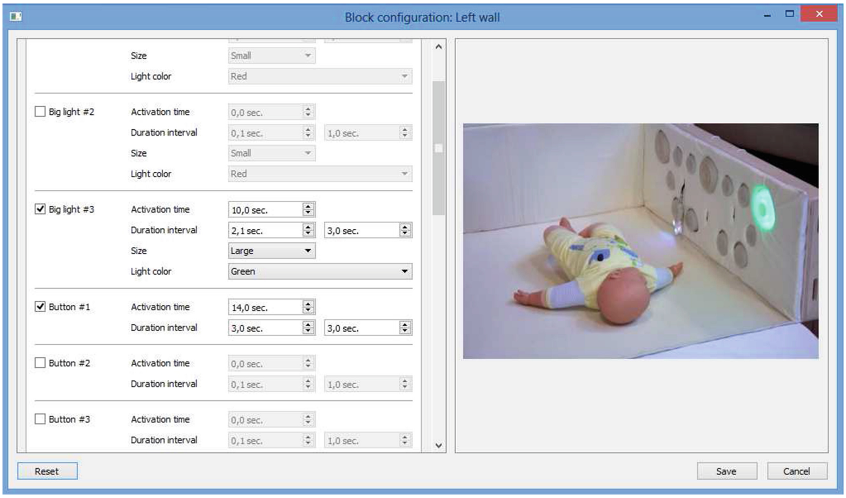

Fig. 4. Block configuration window (left wall).

Aside from the basic blocks, the user can easily create custom complex blocks using the *Add* block functionality. Once saved, a complex block becomes available as a block element inside the blocks palette. With the help of complex blocks the healthcare professional can program subsequences and reuse them within the same training sequence or across multiple training sequences, thereby effectively simulating subroutines or control flow behavior (e.g. loops). Complex blocks do not only increase programming flexibility but also the readability of the sequence, leading to easier comprehension and recognition. To round-off the presentation an overall duration of the defined sequence is displayed at the top of the canvas.

Upon saving of the defined training sequence, an XML-formatted output file is generated containing the description of the sequence as it is used by the execution engine developed for the control of the hardware components.

3.2 Patient-Tailored Training Sessions

Daily therapeutic training sessions are defined as a combination of multiple training sequences scheduled for execution on specific days. For that purpose an additional software module, which encapsulates the VPE, has been implemented in close cooperation with the healthcare professionals based on their requirements and needs. The main aim of that software module is to allow a flexible and less time-consuming creation and management of training sessions tailored to the infant's individual needs. Furthermore, initially defined sessions can be adapted and customized for each infant as a consequence of the evaluation of gradually acquired daily training session results representing the infant's progress.

This module is part of the tele-rehabilitation architecture which has been designed and implemented for the purpose of providing interoperability between the clinical environment and the in-home system, based on secure encrypted communication channels and a reliable data transport. The main tasks handled by the software architecture can be summarized as follows:

1. Programming and management of training sequences
2. Download of training sequences for in-home training
3. Execution of sequences and result data acquisition
4. Preprocessing and upload of the result data
5. Evaluation of obtained training session results

4 Evaluation

The visual programming environment, as presented in this paper, is currently being used for the design of therapeutic training sequences for infants enrolled into the trial program of the CareToy project. The usability evaluation was performed through measurements of efficiency, effectiveness and user satisfaction. For the measurement of effectiveness and user satisfaction seven healthcare professionals working within the CareToy project were used as the target group. Overall a set of experiments and questionnaires were performed in order to measure the usability of the VPE in the intended context. Given the small pool of participants no statistical generalization of the experiment results is claimed, but the feedback received during the trial program of the CareToy project has been of great importance for the evaluation and improvement of the proposed approach.

4.1 Efficiency

Independent experiments were carried out in order to obtain data for the assessment of the efficiency of the VPE. For this purpose the metrics for ease of comprehension described in [10] were applied. Even though the results of such experiments are sensitive to the background of the participants, a subjective measurement of this aspect can be obtained as described in the following.

Two groups of four participants each were initially introduced to the proposed visual programming environment and a textual pseudo-language defined for testing purposes. They were then asked to design a set of training sequences using both approaches. Participants of the first group had programming experience, participants of the second group did not.

An adapted questionnaire based on the ease of comprehension metrics was given to the participants upon completion of the programming tasks they were assigned. Their feedback regarding the visual approach compared to the textual approach is presented in Table 1. It is immediately apparent that for the second group had a smaller difficulty changing between the programming approaches. However this was not the case for the first group with non-technical background.

Table 1. Ease of comprehension results.

	Group 1	Group 2	Overall
Programming experience	No	Yes	-
Rating	(5) Much easier	(3) About the same	(4) Moderately easier

Participants using the visual programming approach were found not to require any additional information other than the task description to successfully complete the task. On the other hand, using the pseudo-language approach they were found to require the reference document and parameter specification data for correct entry of parameter values - a circumstance directly associated with a higher learning curve of the textual approach.

4.2 Effectiveness

Apart from measurements conducted to obtain an indication regarding the efficiency of the proposed visual approach, its effectiveness was measured. During the first months of usage, a rapid growth of the number of defined training sequences was identified (122 sequences) even though the number of users did not change. Initially technical assistance was required due to limitations posed by the hardware components themselves. However, after a short period of growing more familiar with the CareToy system, no further assistance was necessary. Longer training sequences were gradually refined using/reusing complex blocks. Complete control of system parameters offered by the VPE allowed a highly flexible definition and adjustment of programmed behavior of the system to the individual needs of the infant.

4.3 User Satisfaction

In order to measure user satisfaction the guidelines outlined by Lewis [15] were followed. A questionnaire with 19 closed-ended questions was prepared and delivered to all healthcare professionals working with the visual programming environment. The highest score for each question is 1 and the lowest is 7. The results are reported as mean values in Fig. 5. Furthermore, comments provided for each point of the questionnaire were considered in order to improve the overall usability of the proposed VPE.

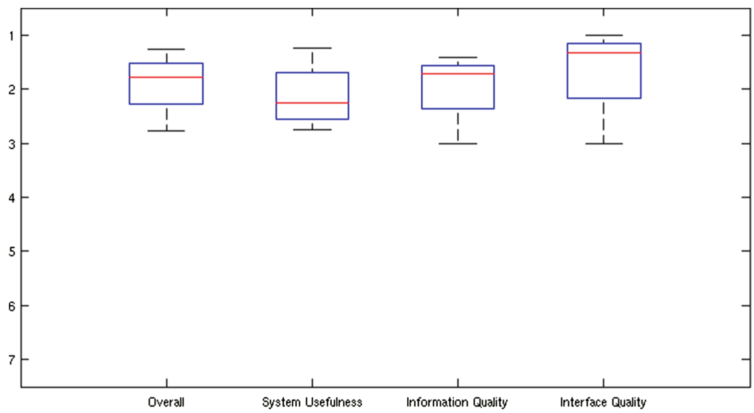

Fig. 5. Mean values obtained from user satisfaction questionnaire results. Left to right: Overall satisfaction, System usefulness, Information quality, Interface quality

5 Conclusions

The work presented in this paper focused on a visual programming environment giving healthcare professionals without technical background knowledge an effective tool for authoring of therapeutic training sequences as programmed behavior of the CareToy system.

Evaluation of the visual programming environment was approached from three directions. Efficiency and effectiveness measurements and a user satisfaction questionnaire were performed. Efficiency measurements confirmed users without technical background to perform better using the proposed visual programming environment. Additionally, effectiveness measurements and results from the user satisfaction questionnaire have shown an overall approval of the proposed VPE.

Further work will focus on the introduction of control flow functionality such as loops, in order to reduce the effort associated with programming of repetitive subsequences.

Acknowledgments. The work presented in this paper is part of the CareToy project (www.caretoy.eu), funded by the European Union (ICT-2011.5.1-287932).

References

1. Hadders-Algra, M.: Early brain damage and the development of motor behavior in children: clues for therapeutic intervention? Neural Plast. **8**(1–2), 31–49 (2001)
2. Blauw-Hospers, C.H., Hadders-Algra, M.: A systematic review of the effects of early intervention on motor development. Dev. Med. Child Neurol. **47**(6), 421–432 (2005)
3. CareToy: A modular smart system for infants rehabilitation at home based on mechatronic toys, 05 March 2014. http://www.caretoy.eu
4. Cecchi, F., Serio, S.M., Del Maestro, M., Laschi, C., Sgandurra, G., Cioni, G., Dario, P.: Design and development of biomechatronic gym for early detection of neurological disorders in infants. In: 2010 Annual International Conference of the IEEE on Engineering in Medicine and Biology Society (EMBC), pp. 3414–3417 (2010)
5. Cono de la Cuerda, R., Munoz-Hellin, E., Alguacil-Diego, I.M., Molina-Rueda, F.: Telerehabilitation and neurology. Revista de Neurologia **51**(1), 49–56 (2010)
6. Datta, C., Jayawardena, C., Kuo, I., MacDonald, B.: RoboStudio: a visual programming environment for rapid authoring and customization of complex services on a personal service robot. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 2352–2357 (2012)
7. Di Felice, P., Lochovsky, F.L., Mosser, T.: Officeaid VPE: a visual programming with examples system for specifying routine office tasks. J. Vis. Lang. Comput. **2**(3), 275–296 (1991)
8. Diprose, J., MacDonald, B., Hosking, J.: Ruru: a spatial and interactive visual programming language for novice robot programming. In: 2011 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 25–32 (2011)
9. Figl, K., Derntl, M., Caeiro Rodriguez, M., Botturi, L.: Cognitive effectiveness of visual instructional design languages. J. Vis. Lang. Comput. **21**(6), 359–373 (2010)
10. Kiper, J.D., Howard, E., Ames, C.: Criteria for evaluation of visual programming languages. J. Vis. Lang. Comput. **8**(2), 175–192 (1997)
11. Mason, D.: Data programming for non-programmers. Procedia Comput. Sci. **21**, 68–74 (2013)
12. Myers, B.A.: Taxonomies of visual programming and program visualization. J. Vis. Lang. Comput. **1**(1), 97–123 (1990)
13. Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y.: Scratch: programming for all. Commun. ACM **52**(11), 60–67 (2009)
14. Whitley, K.N.: Visual programming languages and the empirical evidence for and against. J. Vis. Lang. Comput. **8**(1), 109–142 (1997)
15. Lewis, J.R.: IBM Computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. Int. J. Hum.-Comput. Interact. **7**(1), 57–78 (1995)