Theater robotics for human technology education

Mikko Laamanen
School of Computing
University of Eastern Finland
Joensuu, Finland
mlaamane@student.uef.fi

Ilkka Jormanainen
School of Computing
University of Eastern Finland
Joensuu, Finland
ilkka.jormanainen@uef.fi

Erkki Sutinen
Department of Information Technology
University of Turku
Turku, Finland
erkki.sutinen@utu.fi

ABSTRACT

Theater robotics allows its users to learn by transforming a story into a theatrical performance acted out by robots. A preliminary implementation uses affordable robotics kits controlled by a scripting environment that was designed iteratively by observing the users from the viewpoint of a given set of criteria. The end result facilitates a truly interdisciplinary learning experience that combines two types of artifacts: robots built with technology and engineering, and stories founded on humanities. Theater robotics can be applied not only in technology or robotics education, but also in computing education to teach specific skills such as computational thinking or basic concepts of programming with robots.

CCS Concepts

•Applied computing → Interactive learning environments; *Performing arts;* **•Computer systems organization →** *Robotics;*

Keywords

educational robotics, theater, technology education, curriculum

1. INTRODUCTION

Recent developments in computing education (CE) set new expectations for the learning environments and teachers' professional competencies. New school curricula all over the world propose that computing skills, including programming, should be taught at all school levels. There are remarkable national and international efforts to renew curricula to meet the requirements set by two interdependent trends: the so called 21st century human skills demanded in a knowledge economy and the rapid development of digital technology. For example, Australia has introduced a 'Digital Technologies' curriculum, which addresses computational thinking and the use of digital systems and data, as well as cultural impacts of technology. The specific learning objectives of the Digital Technologies curriculum are heavily geared towards algorithmic problem solving skills [5].

As another example, the new national curriculum in UK includes areas such as "computing" and "design and technology". UK's new curriculum will expose programming to pupils already during the first grades with the learning objectives to design, create, and debug simple programs. The new computing curriculum in England is divided into three stages, where the highest Stage 3 (grades 7-9 in the UK school system) includes rather specific computer science skills, such as sorting and searching algorithms, modularity, and decomposition [2].

Educational robotics forms a widely recognized learning environment for computing and technology education. Robots have been used in all education levels ranging from basic K-12 technology education to advanced university courses on artificial intelligence. However, much too often robotics learning environments emphasize solely the development of technical skills, such as engineering, physics, and mathematics, and pay little if any attention to the skills learned in humanities. Too rarely, there has been integration between technical and humanistic subjects or it has been rather superficial as its best.

To answer the challenge of twinning human and technical skills, this paper introduces the concept of theater robotics. In brief, theater robotics refers to an environment where robots act out a performance based on a given script, a set of behaviors, or interaction with the audience. Theater robotics is a novel framework for creating learning environments with natural collaboration between technical and humanistic school subjects, such as physics and history or literature: the plot comes usually from humanities, whereas the performance requires computing and engineering, which, again, is based on physical and mathematical modeling. This paper presents the technical fundamentals of the theater robotics framework with our initial views on educational value of the theater robotics as a learning environment for technology and computing education.

This paper is organized as follows. First, we position our research in the field of technology and engineering education. Then we introduce the concept of theater robotics. Furthermore, we analyze how our previous work in K-12 robotics education has guided us to develop the theater robotics framework. We present the prototype implementation of the framework in detail and conclude the paper with discussion for the future directions of development.

Koli Calling 2015, November 19-22, 2015, Koli, Finland
© 2015 ACM. ISBN 978-1-4503-4020-5/15/11…$15.00
DOI: http://dx.doi.org/10.1145/2828959.2828975

2. BACKGROUND

Computing education is a relatively young field that designs, implements and evaluates education-curricula, pedagogy, methods, materials, tools-in the five interrelated fields of Computer Science, Computer Engineering, Software Engineering, Information Systems and Information Technology. A key question of the computing education research (CER) community is the pedagogy that should be adopted when teaching computing skills for all students [5]. One of the rising pedagogical issues is contextualization of teaching computing. The computing educators are more and more demanded to bring real-life context into their teaching [9]. There is an ongoing debate whether additional, context-dependent knowledge distracts students when learning abstract computing concepts and context-independent skills. From another point of view, often the students are struggling and losing their motivation when they are not able to see how the decontextualized content would benefit them. Then, the context can provide relevance for teaching and learning [6].

Beside contextualization, the renewed curricula emphasize interdisciplinary pedagogical approaches and projects. Various interdisciplinary teaching projects can be found from literature. For example, Lau et al. [8] report a study where programming was taught through fashion and design in a course of wearable computing. Other examples of combining arts and computing include Kim et al. [7] where the authors describe the development of the Artbotics curriculum, and Brunvand and Stout [3] that describe a cross-disciplinary course for art and computer science and engineering students. However, as Salgian et al. [10] conclude, there are few reported efforts on combining engineering with fundamentally different disciplines such as humanities or social sciences.

It is remarkable that these studies report courses and curriculum design for higher education institutions, but not for the primary school level. Our research focuses mostly on primary and secondary school, although the concept or technology do not limit the use of the proposed theater robotics technology in higher education.

Development of the theater robotics framework is based on our experiences on running SciKids' afterschool technology and robotics clubs since 2001 at the University of Eastern Finland. SciKids' lab environment is an open technology lab where a team of around ten K-12 students solves an open-ended problem with a few university students and researchers over a period of 4-8 months, gathering once or twice a month for 2-3 hours' sessions. Depending on the character of the problem, the club might invite other stakeholders, for instance from industry. The problems are solved following various creativity-enhancing approaches, techniques, and exercises and made tangible by robotics. Thus, our pedagogical approach has always combined the human and technical orientations. Robotics and programming as a part of the field of computing are rather easy to combine with the basics of physics, such as electronics and mechanics. The complicated part comes when the abovementioned skills are needed to combine with history, literature or religion studies. Experiences in sub-Saharan Africa for enriching learning environments, games, and animations with cultural heritage and storytelling traditions (see for example Bada, [1], and Duveskog, [4]) have shown that digital tools can be applied effectively in diverse contexts to cross inter-disciplinary borders in school and after-school settings.

While devising supplementary education with robotics for teachers within our technology club framework, we have clearly seen that teachers are in need for technology education projects that enable them to work naturally across subject borders. By the attribute natural, we mean that teachers should be able to take the best out of each subject and combine the skills taught in each subject seamlessly within a technology education project. Based on the literature review and our own experiences from running technology clubs, we argue that there is an obvious demand for a strong framework that combines computing and technology education with fundamentally different school subjects such as arts, literature and language studies, history, social sciences, and religion.

3. THEATER ROBOTICS FRAMEWORK

The concept of educational theater robotics is built on an idea to let students to design, build, and replay a narrative or a theater chapter with educational robotics familiar from technology and computing education. To complete a theater robotics project successfully, students are challenged to try and improve their skills in various school subjects within the same project.

The process description in Table 1 illustrates the multidisciplinary nature of a theater robotics project. Students can learn stories from books, Internet, or for example by approaching elderly people in their community to learn stories that are a part of the community's spoken heritage. This might be a good approach especially in cultures where the oral storytelling heritage is strong, such as in the countries of sub-Saharan Africa.

An initial experiment for testing the theatre robotics concept was carried out in the SciKids' technology club of University of Eastern Finland. A small group of 8-13-year-old children tried to present a story of Red Riding Hood with Lego Mindstorms EV3 robotics sets. Group gathered to work together once a week for three months. As a result, they were able to implement one chapter of the story with self-designed robots as actors.

The objective of the experiment was not to assess students' learning or motivation, but to analyze the feasibility of Lego Mindstorms robotics set for theater robotics. The analysis was guided by the evaluation criteria proposed by Weiss and Overcast [11]. We investigated especially flexibility, expandability, durability, and the total cost of the learning environment. Furthermore, we wanted to explore the features of hardware and software that would be needed when implementing the theater robotics framework. The experiment supported our initial idea to create a dedicated robotics platform to implement the theater robotics framework. Especially flexibility and total cost of the Lego Mindstorms are problematic in theater robotics projects. However, Lego robots' durability and accessibility are excellent, and these factors need to be taken into account when conceptualising and developing our own platform.

4. IMPLEMENTATION

The proposed theater robotics platform is designed and implemented with a modular approach. The technical framework is divided into four main modules: a scripting environment, robot actors, a theater stage, and a Central Director

Table 1: A theater robotics process

Phase	Subjects	Remarks
1. Students choose a story or a theater play to implement	History, literature, religion	
2. Information collection and story scripting	Literature, articulacy, information retrieval and media literacy	IT skills include use of search engines and online resources
3. Robot design and building	Arts and design, IT, engineering, technology and handicraft, mathematic, physics	
4. Programming / defining scripts for the robot actors	Computer science, logic	IT skills focus to computer science core skills, such as programming fundamentals
5. Theater performance	Arts, sport, social skills, technology	

Unit (CDU). Each module can be developed and maintained separately allowing flexible modifications to the system. For example, it is possible to replace robot hardware without modifying the rest of the framework.

An overview of the framework and relations between the modules are depicted in Figure 1. The theater robotics setup can consist of one or more scripting environments and robots on one stage. Each setup has one CDU module. CDU and stage modules are logically unified to implement an indoor positioning system as explained later in this paper.

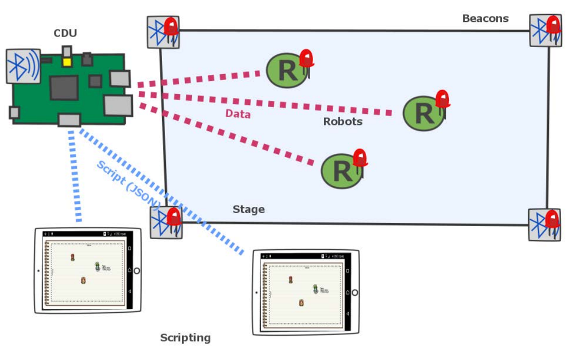

Figure 1: Platform design

4.1 RobotStory scripting environment

RobotStory scripting environment is a Android smart device application, which can be used to build scripts that define a theater play (phase 4 in Table 1). Users are adopting a role of a playwright or a scriptwriter when they are using the environment, rather than a traditional role of a programmer. Scripting can, indeed, be understood as a programming of the robots, but more accurately users are defining the rules how robots should behave during the theater play. We are using tablet devices to make the set as flexible and affordable as possible.

RobotStory uses a storyboard to present the theater play. The play is divided to a set of pages each presenting small sketches which visualize events on a stage. For example, one page could present a robot moving on the stage to ”meet” another robot. The next page could then present another interaction between the robots when they have ”met”. We have adopted the storyboard presentation to split the theater play into small and reasonable sized blocks (pages). Each page can be edited individually and together they form a complete (and hopefully logical!) story. This is similar to what we teach about program design for example in CS1 courses.

Pages in RobotStory are created by defining actions for the actor robots with a graphical user interface. The approach is similar for example to Lego Mindstorms programming environments where graphical blocks represent robot's actions and these blocks are combined together to create logic of the script. Actions are defined by combining primitive functions, such as drive forward, stop, and turn left. User-defined actions can be complex and they form more abstract presentation than scripting with lower-level robot API functions. When the user wants to run the play, the graphical script will be converted into JSON (JavaScript Object Notation) format with necessary metadata about the robot and sent to the CDU server for further processing.

RobotStory environment is currently under development. We envisage functionality that allows learners to move and work in different levels of abstraction to learn how a high-level and abstract robot behaviours such as ”Escape from robbers” is constructed. This would potentially help learners to understand computational principles and fundamentals of programming, such as sequential execution order, loops, and conditional statements.

4.2 Robot actors

Robots are considered as actors in the theater robotics framework. Robots receive commands from the CDU and follow the commands to execute actions. After completing an action, a robot reports its state to the CDU and starts to wait the next commands to perform. This means that robots do not have a logical reasoning for the theater play but they rely on commands they receive from the CDU. When driving, a robot's position and heading is checked constantly and the CDU module calculates and commands necessary corrections to the robot's driving path immediately.

Robot actors are based on Arduino prototyping boards with an embedded computing unit (http://arduino.cc). Arduino is a popular, open-source platform for creating various kinds of robots, automation solutions, and other electronic assets. The Arduino board has a built-in microcontroller, which is capable enough to handle robot actors' actions. There are various models of Arduino boards with different specifications available. For example, Arduino Uno has Atmega328 microcontroller with 14 digital input/output pins and 6 analog inputs. Arduino boards can be programmed with Arduino's integrated development environment (IDE) using C programming language, but it is also possible to use another languages and environments, such as assembler.

An Arduino board does not contain out-of-box any sensors or other electronics that are needed to build a robot.

We use simple DC motors with wheels to allow robots to move freely on the stage. Other electronics needed include motor controllers, resistors, LEDs, an ultrasonic and a gyroscopic sensor, and batteries. Also a Bluetooth module is needed for connecting the CDU with the robot and special infrared LEDs are needed for positioning the robots with stage module's machine vision system.

The theater robot set is supplied with necessary parts for building a solid base for robots. The base models of robots need to be extended with other material available to make robots resemble the characters of the theater play. This emphasises multidisclipinary nature of theater robotics projects when art, design, and handicrafts are combined with physics, electronics, and ICT (Phase 3 in Table 1).

4.3 Stage

The theater stage can have an arbitrary size depending on the requirements of the play to be implemented. The only requirement is that the stage is shaped as a rectangle so that positions of robots on the stage can be detected reliably. The stage can be equipped with necessary props to bring the milieu of a theater play explicitly visible.

The stage is equipped with a robot positioning system implemented using a machine vision approach. The system uses infrared (IR) LEDs to track objects on the stage, and a web camera connected to the Raspberry Pi computer in the CDU module. The camera is modified with a visible light filter so that only IR light passes to the camera lens. In this way, camera image can be easily processed with machine vision algorithm and IR light sources can be found from the image and their relative positions can be calculated. Figure 3 presents debugging output of the machine vision system while the CDU is tracking and commanding a robot to move around on the stage.

Small beacons with Bluetooth connection and IR LEDs are used to mark stage's corners. Bluetooth connection enables switching the beacons on and off, allowing finding individual corners. When positions for each of the corners in the image and real size of the stage are known, it is possible to calibrate the stage in the algorithm. After this, position of any detected IR light source within the calibrated stage area can be calculated. Actor robots also have IR LEDs, which are switched on when a robot needs to be positioned or tracked. Figure 2 shows testing and development stage with corner beacons and one robot on the stage. Camera is located above the stage so that objects and their IR light sources can be seen as well as possible. It is not, however, necessary to locate the camera precisely above the center of the stage. The machine vision algorithm makes perspective transformation to the picture so that object positions can be calculated correctly (Figure 3). This gives freedom to use various solutions to place the camera, such as attaching it to ceiling or wall or using a free-standing tripod with necessary extensions. This allows locating a theater robotics set in a school to a place where it serves the joint projects between different subjects as well as possible.

When the CDU is aware of each robot's position on the stage, it is possible to use abstract actions in the scripting environment, such as commanding a character to move to "meet and greet" another character. Preliminary results for the accuracy of the machine vision based positioning are promising. Our system is capable to detect objects with the accuracy of less than ten centimeters. This should be a sufficient accuracy for the complete implementation of the theater robotics platform.

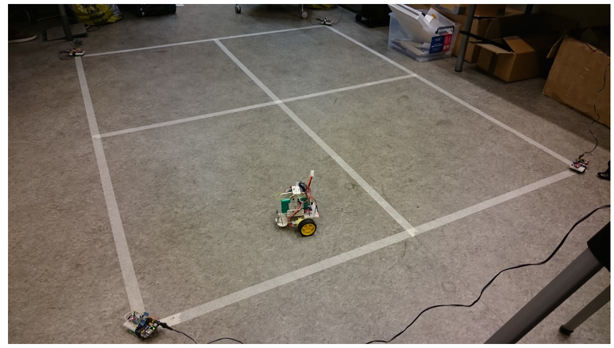

Figure 2: A testing stage

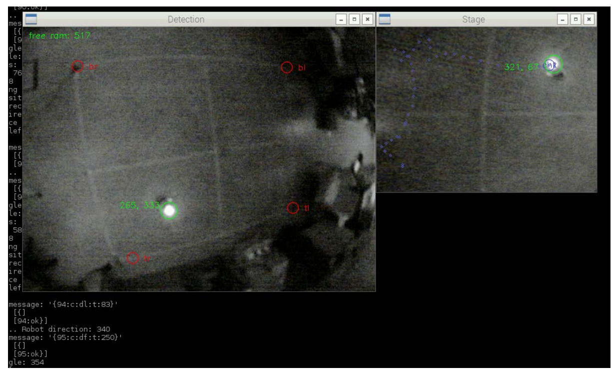

Figure 3: Machine vision tracking a robot on a stage

4.4 Central Director Unit CDU

The CDU module is responsible for maintaining communication between different parts of the system. It receives scripts that the users have created, and schedules and dispatches commands to the robots in an appropriate time and order. The CDU module can be seen as the director of the theater play who communicates with the actors and controls the execution of the play.

The CDU module is implemented with Raspberry Pi 2 (http://www.raspberrypi.org), which is a credit-card sized low-cost computer. Depending on the model, A Raspberry Pi 2 costs of about USD30. It features 900MHz ARM CPU and 1GB of RAM (Raspberry Pi 2 model B). The prize-power ratio of is ideal for the theater robotics framework, keeping in mind the aim to produce an affordable set of tools. Raspberry Pi computers are popular amongst electronics hobbyists and there is a strong and vibrant user and developer community around it. This leads to the easy access of examples and documents allowing fast development and problem solving. Raspberry Pi 2 does not have a built-in Bluetooth or WLAN, so external adapters are needed to establish communication between the CDU, scripting environment and robots.

The CDU module runs a server program written in C++. The server is capable of receiving scripts from RobotStory scripting environments. Robots automatically establish a connection to the CDU server when they are turned on.

When the CDU server receives a script, it creates a plan for executing it with the associated robot actors. The CDU server converts the script to a set of primitive commands that are defined in the theatre robot application programming interface (API). When all robots of the play are connected to the CDU, the play is ready to be performed.

5. DISCUSSION

Our experiments and experiences with Lego Mindstorms informed us that there is a need to design and implement a dedicated set of hardware for the theater robotics framework. Lego robotics is too expensive for deployment, keeping in mind that our target is to develop a system that would be affordable for schools also in emerging economies. More importantly, existing programming solutions for educational robotics are geared towards traditional procedural or object-oriented programming approach and they do not support the inter-disciplinary motivation behind theater robotics: to combine storytelling with the learning of technology. The open source nature of Arduino gives advantages to experiment with different technical solutions to build a flexible and expandable system, in contrast to the relatively closed nature of the Lego controlling unit.

To make the robot actors behave properly, there is an absolute need to detect their position on the stage. The experiment and our prototyping efforts this far, alongside with research literature have shown that one of the biggest bottlenecks in the implementation is a reliable and accurate indoor positioning system. The machine vision approach has given promising results, but the prototype needs still more development. The scripting environment provides lot of possibilities as an educational tool. The storyboard approach described in this paper is one possible solution for scripting theater performances. Other approaches would be an event driven approach or interactive performance with the audience. In the event driven solution, users would script a set of rules for a play and the CDU would react to events on the stage and direct the play based on the rules. The interactive approach would allow audience to give their input (with their own smart devices) to the live performance. Adding different approaches for constructing and playing a theater chapter would no doubt broaden the scope of theater robotics as a learning environment.

6. CONCLUSION AND FUTURE WORK

We have presented fundamental ideas and the current status of development of a theater robotics framework to enable a natural integration of technical and humanistic school subjects. Theater robotics requires various features from robotics hardware and software that cannot be found from existing educational robotics kits, such as Lego Mindstorms. In order to examine these features, we devised an evaluation scheme for assessing the needed technical and conceptual aspects of robotics sets. This evaluation scheme was used in an experiment where a group of school students built the Little Red Riding Hood theater chapter with Lego robots. The evaluation results guided us on designing a theater robotics set based on robots with Arduino microcontrollers and a tablet-based scripting environment. The prototype implementation has shown us challenges on conceptualizing abstract concepts, such as how a dialogue in a theater play could be presented with robots. The finalized implementation of theater robotics framework needs to be tested carefully in real-life school contexts during further studies. Questions in the these studies will be connected to pedagogical value of theater robotics as a multidisclipinary learning environment, and also to more specific questions such as how theater robotics could be used in scaffolding when teaching basic programming concepts.

7. REFERENCES

[1] Bada, J. K., & Suhonen, J. Netaids: Digital learning environment for hiv/aids education in ugandan schools. *Journal of Health Informatics in Developing Countries*, 5(2), 2011.

[2] Brown, N.C.C, Sentance, S., Crick, T., & Humphreys, S. Restart: The resurgence of computer science in uk schools. *Trans. Comput. Educ.*, 14(2), June 2014.

[3] Brunvand, E., & Stout, P. . Kinetic art and embedded systems: a natural collaboration. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, pages 323–328. ACM, 2011.

[4] Duveskog, M., & Sutinen, E. Enriching student hiv awareness by digital storytelling. *Journal of Educational Multimedia and Hypermedia*, 22(4):383–406, 2013.

[5] Falkner, K., Vivian, R., & Falkner, N. . The australian digital technologies curriculum: challenge and opportunity. In *Proceedings of the Sixteenth Australasian Computing Education Conference - Volume 148 (ACE '14)*, volume 148, pages 3–12, Darlinghurst, Australia, Australia, 2014. Australian Computer Society, Inc.

[6] Guzdial, M. . Does contextualized computing education help? *ACM Inroads*, 1(4):4–6, 2010.

[7] Kim, J.H., Coluntino, D., Martin, F. G., Silka, L., & Yanco, H.A. . Artbotics: community-based collaborative art and technology education. In *ACM SIGGRAPH 2007 educators program (SIGGRAPH '07).*, New York, NY, USA, 2007. ACM.

[8] Lau, W.Y., Ngai, G., Chan, S.C.F, & Cheung, J.C.Y. . Learning programming through fashion and design: a pilot summer course in wearable computing for middle school students. In *Proceedings of the 40th ACM technical symposium on Computer science education (SIGCSE '09)*, New York, NY, USA, 2009. ACM.

[9] Rick, D., Morisse, M., & Schirmer, I. Bringing contexts into the classroom: a design-based approach. In *Proceedings of the 7th Workshop in Primary and Secondary Computing Education*, pages 105–115. ACM, 2012.

[10] Salgian, A., Nakra, T. M., Ault, C., & Wang, Y. . Conducting robotsâĂŤbridging the gap between science, technology and the arts in the undergraduate curriculum. In *Integrated STEM Education Conference (ISEC), 2013 IEEE*, pages 1–4. IEEE, 2013.

[11] Weiss, R. & Overcast, I. . Finding your bot-mate: criteria for evaluating robot kits for use in undergraduate computer science education. *J. Comput. Sci. Coll.*, 24(2):43–49, December 2008.