



A Collaborative Authoring Workspace and Script-Based Control Platform for Heterogeneous Robots

Jin-Ling Lin^{1,*} and Kao-Shing Hwang^{2,3}

¹Department of Information Management, Shih Hsin University, Taiwan

²Department of Electrical Engineering, National Sun Yat-sen University, Taiwan

³Department of Healthcare Administration and Medical Informatics, Kaohsiung Medical University, Taiwan

(Received 30 June 2015; Accepted 29 July 2015; Published on line 1 September 2015)

*Corresponding author mail: jllin@cc.shu.edu.tw

DOI: [10.5875/ausmt.v5i3.979](https://doi.org/10.5875/ausmt.v5i3.979)

Abstract: A collaborative script authoring and control platform for multiple heterogeneous robots is developed. The developed platform consists of a remote sever providing a graphic user interface (GUI) authoring tool, and a local control platform controlling a group of heterogeneous robots by interpreting the designated scripts. Designed to encourage impromptu creativity among programming students in class, this Web-based authoring tool fosters collaborative creativity in the authorship of command scripts. The architecture of the local controller is composed of an embedded system for controlling the movement of robotic puppets, and a management server for the synchronization of live shows. Post-activity assessment indicates that the proposed platform provides strong support and encouragement for interactive robot learning in curriculum training and evaluation.

Keywords: Collaborative authoring, robotic learning, web-based control platform, ubiquitous learning

Introduction

Robots are increasingly small and inexpensive to produce, leading to their use in a broader range of applications including military, industry, home service, and education [1][2]. Robots are traditionally manipulated through the use of computers or embedded systems. Changes in platforms require developers to modify source codes to adapt control of the robots to different operating systems. End users must also re-install the control system for various devices. Moreover, the operation of drama-based robots is still requires considerable expertise. The requiring highly specialized code generally discourages students from learning robotics or its applications.

The importance of hands-on experiences in understanding high-level theoretical concepts and in-depth integrated problems is well recognized [3][4]. Many robotics educators have developed a variety of curricular tools to support the study of robotics. Some of these tools attempt to simplify control instruction through the use of

programming tools, allowing students to learn about the required characteristics of robotics, such as controllability, autonomy, and intelligent interaction with users [5]. To overcome the constraints of interdisciplinary training, graphic programming tools for robotic control have been developed to allow students easier to control robots [6]. Meanwhile interactive systems are also used in educational robots to allow students to gain practical experience with accessory operations [7], [8]. Internet technologies provide users with remote access to robots, and have driven the development of web-based tools for robot operation [9]. Web-based authoring tools overcome limitations of time and distance, allowing designers to create and perform scripts anywhere, at any time. One of the objectives of this work is to develop a cloud-computing-based robotic puppet authoring platform as a supplementary tool for an Introduction to Robotics course. Students are encouraged to form teams and create short puppet shows using the proposed authoring platform, letting the robots perform their roles via the equipped embedded system.

Students learning to build robots should be



familiarized with how all of the parts of a complex system (including mechanics, computing devices, and sensors) interact and depend on each other. The interdisciplinary nature of robotics means that, when students learn to engineer robots, they will inevitably be encouraged to learn about the many other disciplines upon which robotics depends [10]. This could be a profound and difficult lesson for electrical engineering and computer science students who ultimately need to create software that operates inside much larger systems [11]. It has been found that high-school students had significantly increased their comfort level with technology after involvement in hands-on robotics projects [12]. It has also been noted that such projects are more conducive than other learning methods for fostering good teamwork.

Combining robotics and education creates exciting opportunities to motivate, challenge, and inspire youth to learn about science and technology [13]. Over the past several years, a task force at our university has been developing a pedagogical method to embody embedded systems within the context of robotics. To alleviate the loading of prospective students of the robotics curriculum in their junior and senior years, a rapid prototyping platform for robotics applications was first proposed [14]. The prototyping platform was then enhanced by adding cloud computing and real time interaction with the proposed authoring tool and control platform [15].

To facilitate the evaluation of student learning, it is important to classify learning progress into a set of learning themes for curriculum design and achievement evaluation [16]-[19]. Three learning themes were set for this interdisciplinary course: (1) professional skills in dynamics-kinetics, communication, operating environment, and programming, especially regarding embedded systems; (2) capabilities related to electronic design, robotics, problem solving, presentations, peer cooperation, and division of work; (3) self-motivation and

self-confidence. Professional skills and capabilities encompass lessons learned from direct involvement in the building and programming of robots. The remaining two themes represent important additional opportunities for students to achieve broader learning.

Most robotics courses use laboratory assignments or projects to convey kinematics, dynamic control, and robot programming concepts to students. In light of growing interest, the use of laboratory equipment always introduces problems. The learning robots used in teaching laboratories (e.g., RHINO, PIONEER and Lego NXT) are quite limited in terms of the range of functions provided and in the programming environment offered. Therefore, some universities use commercial or self-developed simulators as complementary learning platforms for robotics courses. From the viewpoint of robotic system design, unlike in the past when computers bundled motor drivers in a large module, current embedded systems with real-time operating systems provide the advantage of rich functional source operating systems which can reduce programming and hardware design requirements. This inspires the motivation to design a robotic puppet authoring platform to allow students to gain hands-on practice.

System Architecture

The proposed cloud computing based robotic puppet authoring platform emphasizes screenplay development and dramatic performance, rather than the hardware—the robotic puppets. Therefore, the authoring platform can be described in two parts – the proscenium and backstage, as shown in Figure 1. The proscenium is responsible for displaying the webpage interface, so it is set up as a web server. The proscenium also requires a database to save the screenplays created for later robotic puppet shows. The backstage is responsible for administering the connection between server and robotic puppets. It manages the transmission of instructions (i.e., to which robotic puppet the instructions should be transmitted and to which webpage the responses from the robotic puppet should be transmitted) so it is set up as a management server. In other words, the authoring tool is composed of a web server, a management server, and a database which contains the scripts and components to create scripts. Figure 2 shows the relationship of the server, proscenium, backstage, and robotic puppets for three actors.

The robotic puppets authoring platform provides a convenient way to create screenplays for different roles based on cloud services, and Wi-Fi based drama shows via smartphones. Designers can create or edit their scripts using smartphones, personal digital assistants (PDAs), or

Jin-Ling Lin received the master and Ph.D. degrees in computer science from the University of Oklahoma, Norman, OK, USA, in 1989 and 1993, respectively. She is currently a Professor with the Department of Information Management, Shih Hsin University, Taipei, Taiwan. Her current research interests include object-oriented programming, data science, data mining, information retrieval, intelligent system, intelligent network routing, and path planning in logistics.

Kao-Shing Hwang is a professor of Electrical Engineering Department at National Sun Yat-sen University and an adjoint professor of the Department of Healthcare Administration and Medical Informatics, Kaohsiung Medical University, Taiwan. He received the M.M.E. and Ph.D. degrees in Electrical and Computer Engineering from Northwestern University, Evanston, IL, U.S.A., in 1989 and 1993, respectively. Since August 1993, he has been with National Chung Cheng University in Taiwan. He was the chairman of the Electrical Engineering Department (2003~2006) and the director of the Optomechatronics Institute of the university (2010~2011). He has been a member of IEEE since 1993. His research interest includes methodologies and analysis for various intelligent robot systems, machine learning, embedded system design, and ASIC design for robotic applications.



computers to access the cloud service—authoring platform. The end users can then use their smartphones to control the robotic puppets' performance, which are downloaded from the cloud. The system architecture is illustrated shown in Figure 3.

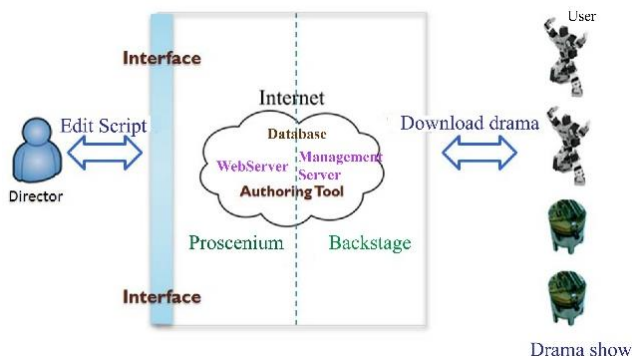


Figure 1. Two parts of the authoring platform: proscenium and backstage.

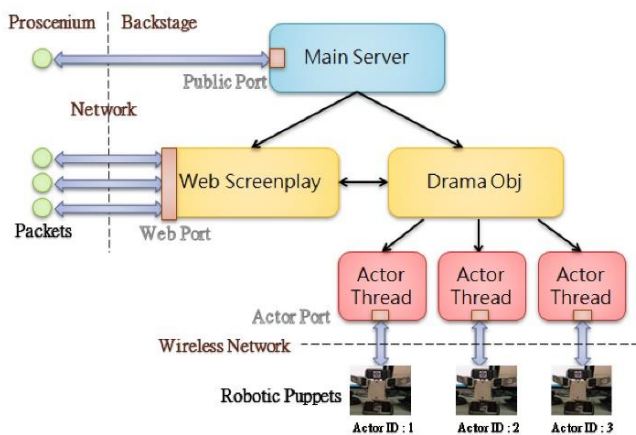


Figure 2. Example of the relationship between the server, proscenium, backstage, and robotic puppets.

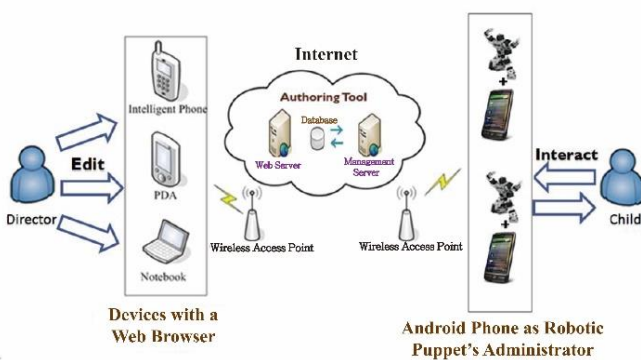


Figure 3. Architecture of the web based robotic puppet authoring platform.

The proposed authoring platform uses cloud-based servers to store scripts and screenplays and to run authoring tools (i.e., web and management servers).

Script components include emotions, audio, and actions, which can be also obtained by a communication function, deployed on both the client and server. This function reads the rotary angles of the robotic motors on the client's side. Perform a dramatic show requires a smartphone and robotic puppets. Android allows developers to call its application framework which provides numerous functions including access to hardware controls, initiation of an activity life cycle, or location detection. Android provides abundant APIs for system developers to design their own applications, making it a good choice for administering the robotic puppet shows. At the physical level, an Android phone connects to a robotic puppet via Bluetooth and communicates with the management server by Wi-Fi. The kernel space is based on the Linux operating system. The Android runtime environment is use to provide complete control over the robotic puppet shows.

Web Based Robotic Puppet Authoring Platform

The dramatic script used for robotic puppets differs from scripts used by human performances. Humans can read the script and imagine the scenario, but robotic puppets do not have this capacity. The proposed authoring platform attempts to design a drama script based on step-by-step timestamps similar to animation, to overcome the lack of imagination of robotic puppets.

The actions of the robotic puppets include emotions, gestures, and sounds. These actions, based on timestamps, are used as the building blocks for constructing screenplays. Robotic puppets may move around a small area and perform certain simple gestures. The gestures of a robotic puppet can be controlled by setting the parameters (i.e., the position and speed of the motors). They can also follow pre-defined instructions and talk like humans to create emotional building blocks which allow them to express anger, happiness or sadness. The repository of pre-designed emotional expressions can be modified and expanded by teachers or students.

In Figure 4, the icon on the left represents the emotion of happiness, which can be downloaded from local or public repositories, the picture on the right represents the puppet's gesture, and the upper part displays the text. Emotions, gestures, and dialogue change over time during the play. A user selects an icon and applies it to the design by assigning the emotion with a particular timestamp. Users can also edit the positions and speeds of the puppet's motors to create different gestures. The speech content is also editable in a drama script, and designers can select from a range of background music and puppet voices.



Figure 4. Sketch of a robotic puppet—gesture, emotion, and text.

The authoring platform allows users to design or manually arrange various actions based on one-second timestamps. To reach the goal of WYSIWYG (What You See Is What You Get), the platform also provides a play function which allows users to see the robot's gestures and expressions, and hear their speech at any given moment, via the performance interface. The GUI of the proposed Object Editor is shown in Figure 5. Designers can not only edit the gestures, emotions, or text at a particular timestamp, but can also choose robots and preview the performance during the editing process. The dashed purple rectangle in the middle of Figure 5 denotes the timestamp. The action of an actor at a particular timestamp is shown in the middle left of the object editor, marked by the dotted green rectangle.

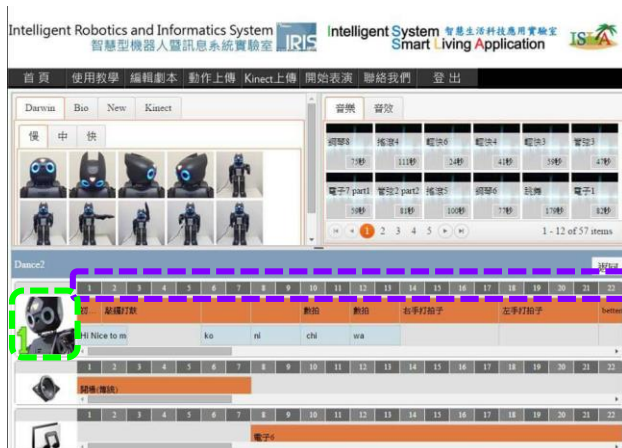


Figure 5. GUI of the proposed Object Editor.

After the actions for each robotic puppet have been designed, a script can be created via the proposed Director Panel. A dramatic script, which has a tree-like structure, includes several actors and is formatted according to a pre-defined XML schema. In the authoring tool, a robotic puppet, as an actor, symbolizes an object. With the object-oriented conception, the designer may add, edit, or delete an actor dynamically and conveniently via the proposed Object Editor. In other words, an actor is viewed as a single object when users design these scripts.

The authoring tool supports the synchronous performing and pausing of robotic puppet shows via the Director Panel. All of these isolated actors follow the same timeline when the robotic puppets perform a drama. An example of a dramatic script tree structure is shown in Figure 6.

Robotic puppets have the capacity to receive instructions, judge information, and respond with correct actions. The puppets also have hands to clap and wheels for movement. A smartphone functions as the puppet's head. The body is composed of two hands and two wheels with four AX-12 motors. Each puppet has a unique identifier in the authoring platform, allowing them to be distinguished from each other. The authoring platform can send script instructions to the specified puppet and have the puppet perform the instructions. While the show is being performed, a management module is responsible for receiving instructions from the authoring platform and assigning them to the corresponding robotic puppets.

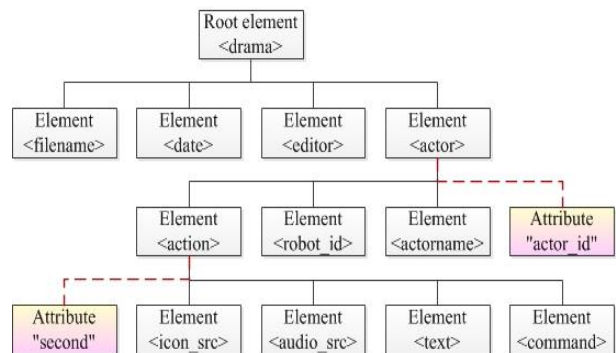


Figure 6. Example of an XML-based script structure.

There are two performance modes in the robotic puppet shows—single and synchronous. The single performance mode requires the robot to respond to each action-instruction. The counter on the Web Screenplay counts the time over which the puppet must respond. If the response time exceeds a time unit in the timestamp, the management server stops receiving the response and prepares to accept the next action-instruction. The processing flow of a single performance mode is provided in Figure 7.

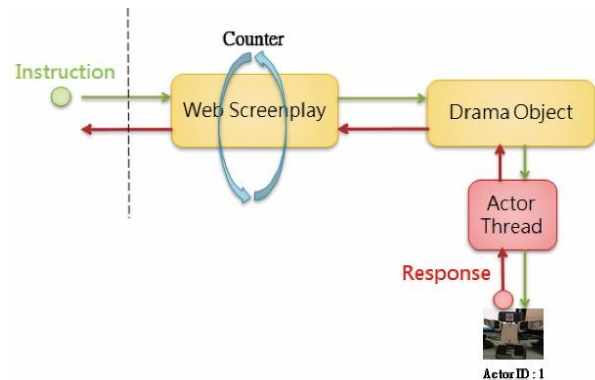


Figure 7. Processing flow of a single performance mode.

The web server sends a synchronous performance signal, determined by the drama's composition, to the management server to activate the synchronous performance mode. All of the actors perform in the show in synchronous performance mode. The processing flow of the synchronous performance mode is shown in Figure 8. To improve the efficiency of instruction delivery and the accuracy of the synchronization, the proscenium of the synchronous performance mode does not need to receive responses and the backstage is supported by an instruction gate in the Drama Object. The instruction gate collects the instructions for the current timestamp. When the gate obtains all the actors' instructions for the current timestamp, it will send them simultaneously to the robotic puppets. The counter in the Drama Object counts the time needed to collect all of the actors' instructions. If the time exceeds one time unit, the instruction gate abandons these collected instructions and waits for the next timestamp's packets.

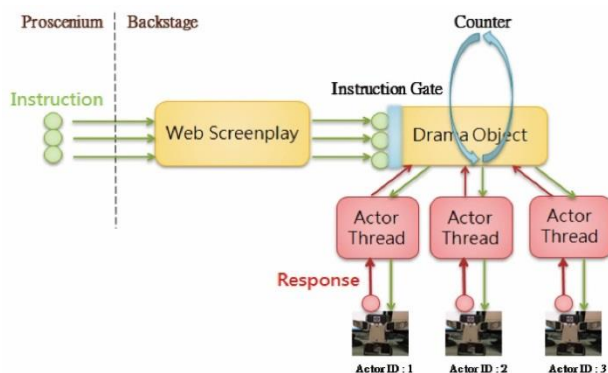


Figure 8. The processing flow of the synchronous performance mode.

Implementation and Experiments

Before using the authoring platform, students need to design and assemble their own robots [20], [21]. The smart motors provided to students are the AX series of DYNAMIXEL Servos [22]. Other required items include a smartphone for each group, a number of RC-servo motors, mechanical frames, I. C., and circuits. In addition, the platform requires students to provide primitive motions by commanding the motors and recording relative parameters of these motions in the web database.

The proposed authoring platform used Linux, Apache, MySQL, and PHP/Perl/Python (LAMP) to set up the web server, as LAMP is a free open source and can be used to set up extremely robust websites [23]. The management server used Java as development tool and Java virtual machine as the execution environment, because of Java's excellent network support and cross-platform capability.

Android smartphone were used as the puppet

heads and serve as the main controllers of the robotic puppets. Each smartphone needs to be set up in advance. The Android robotic puppet show application provides an identifier, IP, port, script ID, actor ID, and MAC to the management server. The action instructions are sent to a robotic puppet during the performance and contained information about the account, script ID, actor ID, timestamp, emotion, actor's lines, and motor parameters. The account, script ID, and actor ID are used to ensure the correctness of the packet. The timestamp, emotion, actor's lines, and motor parameters are actions performed by the puppets. When an Android phone receives an instruction from the management server, the application software decomposes the instruction and transmits the appropriate action to the robotic puppet.

Android phones transmit the commands to activate the AX-12 motors [22]. The motors can work in series, but they operate in parallel so as to increase the fault tolerance of the robotic puppets. With a parallel design, the entire system could still work correctly, even if one motor was disabled. To make things simpler for the students, the puppets had few motors and a limited degrees of freedom for the hands and wheels. This also helped reduce development costs and achieve maximum fault tolerance.

An example of a robotic puppet show, created and performed with the proposed authoring platform, is shown in Figure 9. Two robots wore traditional puppet costumes and enacted the "Annoying Orange", based on a popular Youtube video [24]. The conversation in the original video of "Annoying Orange" is rapid and chatty. The robotic puppets created by the proposed authoring tool not only spoke the actor's lines, but also acted with spirited gestures [20].



Figure 9. Example of a robotic puppet show in synchronous performance mode – Annoying Orange [20].

Another example of synchronous performance is shown in Figure 10 and Figure 11. Two robotic puppets imitated Jack and Rose's parts in the "You trust me" and "I am flying" scenes from the movie *Titanic*. Figure 9 to Figure 11 show that the robotic puppets have the ability to perform simple movements, present emotions, and read actors' lines as cued by the timestamp. The online videos of these two examples can be found in [20].

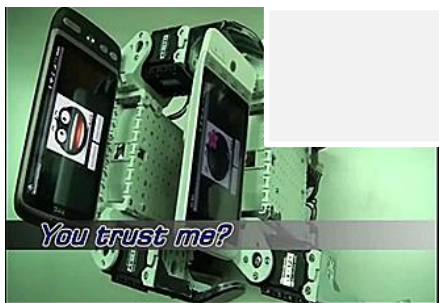


Figure 10. Example of a robotic puppet show in synchronous performance mode – the "You trust me" scene from the movie *Titanic* [20].



Figure 11. Example of a robotic puppet show in synchronous performance mode – the "I am flying" scene from the movie *Titanic* [20].

Curriculum Training and Evaluation

The demo robots, which were the outcome of the term project for the Introduction to Robotics course, were constructed by junior and senior students with backgrounds in Computer Science, Electrical Engineering, and Mechanical Engineering. Students made their own controllers and communication modules from scratch to practice skills in circuitry and electronics. The instructors only provided the necessary RC servo motors and assembly kits rather than comprehensive kits from commercial vendors such as iRobot Create or Biologic robots.

The assessment mainly focused on knowledge of robotics and related soft-ware/hardware skills. In addition to grading, a teaching survey and course questionnaire was administered by the university's academic affairs office at the end of the semester to evaluate the effectiveness of students' learning based on the proposed platform. Among other concerns, the first 14 questions of the survey cover students' expectations for the course and their study plans, regardless of their technological backgrounds. The remaining parts cover whether students have learned the core themes and content of the course, while also asking about the course's learning methods. In addition to the official survey, students were asked to complete monthly written surveys reflecting on the prior week's activities. Students rated their teams' performance and recorded minutes of group meetings. These minutes focused on problems encountered and

suggested solutions along with specific course activities.

The evaluation is presented in terms of the support of the learning process with regards to curriculum themes for students specializing in different backgrounds, namely, Computer Science, Electrical Engineering, and Mechanical Engineering, as shown in Figure 12. Another assessment, shown in Figure 13, focuses on the effectiveness of learning methods on completion/motivation, professional skills, cooperation, and problem solving for students taking the Introduction to Robotics course. The numbers in the figures are the averaged accumulated scores of the courses over six successive years and hundreds of students. In Figure 12 and Figure 13, points are assigned on a scale of one to three. A three indicates a strong confirmation that the proposed platform does help students master the desired skills for completing interactive robot applications. A two indicates a secondary positive confirmation, and a one represents students regarded the proposed platform as irrelevant to the application of interactions among robots.

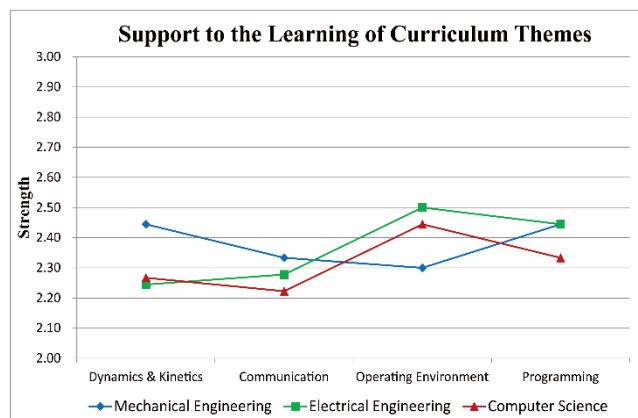


Figure 12. Support to the learning of curriculum themes – points are given based on a strong three to a weak one.

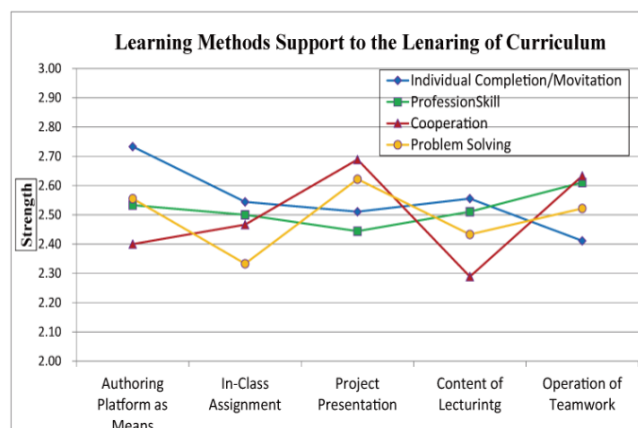


Figure 13. Learning methods' support to the learning of curriculum – points are given based on a strong three to a weak one.

This course emphasizes active learning, allowing students to immediately apply concepts learned in lectures to robots, and cooperative project-based learning,



where students work on projects in teams. The final project may potentially include interdisciplinary outreach, for example, collaborating with art students. On the other hand, the main purpose of the hands-on platform is to make students aware of the current capability of robot software and hardware. The effectiveness of the learning outcomes is supported by the increased number of more students who indicated intent to continue studying robotics in graduate school (Table 1).

		# of students entering graduate school	% of students further studying robotics
Without using the proposed platform	2006	22	22.72
	2007	16	18.75
Using the proposed platform	2008	18	33.33
	2009	19	36.84
	2010	17	41.17
	2011	15	40.00
	2012	15	40.00
	2013	15	40.00

Table 1. Students Studying Robotics in Graduate School

The most important tools provided by the proposed platform provide abstractions of the programming interface and modularity in the hardware, such that the students do not need to delve into details when programming each individual robot. The former encourages the capability to capture and generalize idioms, often in black box entities with simple interfaces. Modularity means the ability to compose systems out of reusable mix-and-match parts. Using the proposed platform as a curricular supplement has several advantages. By integrating open-source operating systems into the programming layers, low-level programming chores are encapsulated through built-in methods. Although students lack the relevant knowledge, they can overlook details of higher level programming tasks thanks to the rich library of functions provided by the operating systems. Furthermore, the graphic programming interface tool significantly reduces the cost of solving problems in prototype development. This interface allows skilled programming students to contribute well-designed functions (programs) in the form of icons to a shared pool. Moreover, the design of the multi-tier system allows large-scale integration into a single unit that is tailored to the problems of specific domains.

Results indicate that the proposed platform is comparable to other commercial authoring platforms in terms of engaging students' interest and encouraging intention to further study robotics. Table 2 compares the

capabilities and the editing interface of Choregraphe by Aldebaran Robotics [25], RoboPlus Task by Robotis O, and the proposed platform. Only the proposed platform includes features such as web-based architecture, integration of facial expressions, synchronization between robots, and manipulation between multiple robots in script design. RoboPlus Task can create a customized program for every ROBOTIS product – Darwin and Bioloid, but only allows for command line editing.

Platforms	Chore-graph for NAO	RoboPlus Task for Darwin, Bioloid	Proposed Platform
Functions			
Web-Based authoring	No	No	Yes
Vocal Expression	Yes	Yes	Yes
Facial Expr.	No	No	Yes
Multiple Robots Manipulation	No	No	Yes
Programming Interface	GUI	Command Line	GUI
Synchronization between Robots	No	No	Yes

Table 2. Comparison of authoring tools and performance platforms for robotic puppets show.

Conclusion

The proposed collaborative platform provides a rapid learning experience with regards to interactive robot applications and has several advantages over commercial training kits for dedicated humanoid robot control systems, including an open source operating system and information hiding programming skills. Therefore, the proposed system can be used to achieve independence of the hardware and the operating system. Moreover, the completed platform can be packed into a reusable stub like bricks for a layered wall to facilitate rapid prototyping for interactive robotic puppet performances. By using the proposed platform, students can complete interactive robot applications without having to master dynamics, kinetics, communication, the operating environment, and programming.

To increase learner engagement, screenwriters can create appropriate learning objects via the proposed web-based authoring platform, which standardizes robotic devices so that the robot control system can be independently run on different operating systems. Designing the motions of humanoid robots and interfacing reactions with the platform helps students practice the skills and concepts of mechanics and



dynamics learned from their robotics courses. The creation and performance of a live robotic puppet show attracts students' attention more than dedicated robot programming. Live robotic puppet shows provide an engaging, learner-centered mode of learning robotics.

References

- [1] L. Major, T. Kyiacou, and O. Brereton, "Systematic literature review: teaching novices programming using robots," *The Institution of Engineering and Technology Software*, vol. 6, no. 6, pp. 502-513, 2012.
doi: [10.1049/iet-sen.2011.0125](https://doi.org/10.1049/iet-sen.2011.0125)
- [2] E. Garcia, M. Jimenez, P. Santos, and M. Armada, "The evolution of robotics research," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 90-103, 2007.
doi: [10.1109/MRA.2007.339608](https://doi.org/10.1109/MRA.2007.339608)
- [3] M. Hernando, R. Galan, I. Navarro, and D. Rodriguez-Losada, "Ten Years of Cybertech: The Educational Benefits of Bullfighting Robotics," *IEEE Transactions on Education*, vol. 54, no. 4, pp. 569-575, 2011.
doi: [10.1109/TE.2010.2095014](https://doi.org/10.1109/TE.2010.2095014)
- [4] M. Yilmaz, S. Ozelik, N. Yilmazer, and R. Nekovei, "Design-Oriented Enhanced Robotics Curriculum," *IEEE Transactions on Education*, vol. 56, no. 1, pp. 137-144, 2013.
doi: [10.1109/TE.2012.2220775](https://doi.org/10.1109/TE.2012.2220775)
- [5] L. Riek, "Embodied Computation: An Active-Learning Approach to Mobile Robotics Education," *IEEE Transactions on Education*, vol. 56, no. 1, pp. 67-72, 2013.
doi: [10.1109/TE.2012.2221716](https://doi.org/10.1109/TE.2012.2221716)
- [6] M. Jou, C.-P. Chuang, D.-W. Wu, and S.-C. Yang, "Learning robotics in interactive Web-based environments by PBL," in proceeding of *ARSO IEEE Workshop on Advanced Robotics and Its Social Impacts*, Aug. 23-25, 2008, Taipei, Taiwan.
doi: [10.1109/ARSO.2008.4653615](https://doi.org/10.1109/ARSO.2008.4653615)
- [7] Z. Shiller, "A Bottom-Up Approach to Teaching Robotics and Mechatronics to Mechanical Engineers," *IEEE Transactions on Education*, vol. 56, no. 1, pp. 103-109, 2013.
doi: [10.1109/TE.2012.2226176](https://doi.org/10.1109/TE.2012.2226176)
- [8] S. Lalle, U. Pattacini, S. Lemaignan, and A. Lenz, "Towards a Platform-Independent Cooperative Human Robot Interaction System: III An Architecture for Learning and Executing Actions and Shared Plans," *IEEE Transactions on Autonomous Mental Development*, vol. 4, no. 3, pp. 239-253, 2012.
doi: [10.1109/TAMD.2012.2199754](https://doi.org/10.1109/TAMD.2012.2199754)
- [9] M. Kulich, J. Chudoba, K. Kosnar, and T. Krajnik, "SyRoTek—Distance Teaching of Mobile Robotics," *IEEE Transactions on Education*, vol. 56, no. 1, pp. 18-23, 2013.
doi: [10.1109/TE.2012.2224867](https://doi.org/10.1109/TE.2012.2224867)
- [10] C. Rogers and M. Portsmore, "Bringing Engineering to Elementary School," *Journal of STEM Education*, vol. 5, no. 3, 4, pp. 17-28, 2004.
- [11] R. Beer, H. Chiel, and R. Drushel, "Using Robotics to Teach Science and Engineering," *Comm. of the ACM*, vol. 42, no. 6, pp. 85-92, 1999.
doi: [10.1145/303849.303866](https://doi.org/10.1145/303849.303866)
- [12] I. Nourbakhsh, K. Crowley, A. Bhave, E. Hamner, T. Hsiao, A. Perez-Bergquist, S. Richards, and K. Wilkinson, "The Robotic Autonomy Mobile Robots Course: Robot Design, Curriculum Design, and Educational Assessment," *Autonomous Robots*, vol. 18, no. 1, pp. 103-127, 2005.
doi: [10.1023/B:AURO.0000047303.20624.02](https://doi.org/10.1023/B:AURO.0000047303.20624.02)
- [13] S.-H. Kim and J.-W. Jeon, "Introduction for Freshmen to Embedded Systems Using LEGO Mindstorms," *IEEE Transactions on Education*, vol. 52, no. 1, pp. 99-108, 2009.
doi: [10.1109/TE.2008.919809](https://doi.org/10.1109/TE.2008.919809)
- [14] K.-S. Hwang, W.-H. Hsiao, G.-T. Shing, and K.-J. Chen, "Rapid Prototyping Platform for Robotics Applications," *IEEE Transactions on Education*, vol. 54, no. 2, pp. 236-246, 2011.
doi: [10.1109/TE.2010.2049359](https://doi.org/10.1109/TE.2010.2049359)
- [15] K.-S. Hwang and T.-L. Kuo, "Authoring Tool for Robotic Puppet Play," in proceeding of *International Conference on System Science and Engineering*, June 8-10, 2011, Macau, China.
doi: [10.1109/ICSSE.2011.5961968](https://doi.org/10.1109/ICSSE.2011.5961968)
- [16] W. White and D. Niemann, "The Presentation of Lagrange's Equations in Introductory Robotics Courses," *IEEE Transactions on Education*, vol. 32, no. 1, pp. 39-46, 1989.
doi: [10.1109/13.21161](https://doi.org/10.1109/13.21161)
- [17] N. Vira and E. Tunstel, "Use of symbolic computation in robotics education," *IEEE Transactions on Education*, vol. 35, no. 1, pp. 18-30, 1992.
doi: [10.1109/13.123413](https://doi.org/10.1109/13.123413)
- [18] F. Mrad and G. Deeb, "Extending the utility of the RHINO educational robot," *IEEE Transactions on Education*, vol. 40, no. 3, pp. 184-189, 1997.
doi: [10.1109/13.618028](https://doi.org/10.1109/13.618028)
- [19] R. M. Murray, S. Waydo, L. B. Cremean, and H. Mabuchi, "A New Approach to Teaching Feedback," *IEEE Control Systems Magazine*, pp. 38-42, 2004.
doi: [10.1109/MCS.2004.1337856](https://doi.org/10.1109/MCS.2004.1337856)
- [20] IRIS, "The Drama of Puppet Robots," National Chung Cheng University, 6 Jan. 2013. [Online]. Available: <http://youtu.be/Z6OlnrlpP8I>. [Accessed



- 06/30/2015]
- [21] IRIS, "An Editorial Platform for Screenplay of Interactive Robotic Puppet Shows", National Sun Yat-sen University, 27 Feb. 2014. [Online]. <https://www.youtube.com/watch?v=NpwmK9MMGys>. [Accessed 06/30/2015]
- [22] Trossen Robotics, "Trossen Robotics," AmeriCommerce, 2013. [Online]. Available: <http://www.trossenrobotics.com/p/bioloid-premium-robot-kit.aspx>. [Accessed 06/30/2015]
- [23] J. Lee and B. Ware, *Open Source Web Development with LAMP: Using Linux, Apache, MySQL, Perl, and PHP*, Boston, MA: Addison Wesley, 2002.
- [24] Daneboe, "Annoying Orange," GAGFILMS.COM, 9 Oct. 2009. [Online]. Available: http://www.youtube.com/watch?v=ZN5PoW7_kdA. [Accessed 04/01/2015]
- [25] Aldebaran, "NAO Documentation v1.14.5," Aldebaran, 2012. [Online]. Available: <http://doc.aldebaran.com/1-14/index.html>. [Accessed 07/20/2015]
- Robotis, *RoboPlus v 1.0.31.0 User Manual*, Seoul, Korea: Robotis, 2013.

