



LIBROB: An Autonomous Assistive Librarian

Costanza Di Veroli, Cao An Le, Thibaud Lemaire, Eliot Makabu, Abdullahi Nur, Vincent Ooi, Jee Yong Park, Federico Sanna, Rodrigo Chacon^(✉)^(iD), and Yiannis Demiris^(iD)

Imperial College London, South Kensington, London SW7 2AZ, UK
{costanza.di-veroli15, caoan.le15, thibaud.lemaire18, eliot.makabu15, abdullahi.nur15, vincent.ooi15, jee.park15, fs2215, r.chacon-quesada17, y.demiris}@imperial.ac.uk

Abstract. This study explores how new robotic systems can help library users efficiently locate the book they require. A survey conducted among Imperial College students has shown an absence of a time-efficient and organised method to find the books they are looking for in the college library. The solution implemented, LIBROB, is an automated assistive robot that gives guidance to the users in finding the book they are searching for in an interactive manner to deliver a more satisfactory experience. LIBROB is able to process a search request either by speech or by text and return a list of relevant books by author, subject or title. Once the user selects the book of interest, LIBROB guides them to the shelf containing the book, then returns to its base station on completion. Experimental results demonstrate that the robot reduces the time necessary to find a book by 47.4%, and left 80% of the users satisfied with their experience, proving that human-robot interactions can greatly improve the efficiency of basic activities within a library environment.

Keywords: Robot · Library · Human-robot interaction

1 Introduction

University libraries contain a wide range of written materials readily available for their users, however finding physical copies of these materials is not always easy. An obstacle that several library users face is the process of locating a specific book title within extensive book collections in a library. In university libraries for example, this process typically requires the user to manually type the book title on a computer using the library's search engine, which returns the library floor where the book is located and the book's identification number. The user is then expected to go find



Fig. 1. LIBROB: An autonomous assistive librarian.

the shelf that contains the book by themselves. We recently conducted a survey in which students at Imperial College London were asked about how much time they typically spend searching for a book. The results showed that 82% of students spent more than 5 min searching for a book, suggesting that the process by which books are located can be improved in order to give library-users easier access to the available resources.

In this paper we propose the use of an autonomous robot to guide the user directly to the book’s location in order to increase library-user’s satisfaction while reducing the time needed to locate a book inside the library. We called our solution LIBROB, a robot that receives a book title from the user, searches in the library’s database for the relevant books and autonomously guides the user directly to the correct shelf. We implemented an intuitive multi-language enabled graphical user interface (GUI) that is able to communicate with the user through visual feedback and speech synthesis for a better interaction with their users. Furthermore, the user can choose from two available options to input the book title: by text or speech. We tested our platform through experiments conducted in an actual library and involving 40 participants asked to find an specific book. An image showing LIBROB’s final design is shown in Fig. 1.

2 Related Work

Robots have already found their application in public libraries, and some significant advances have already been made. One of the most advanced examples is AuRoSS a robotic system capable of navigating a library, scanning the books on the shelves, analysing the result of the scanning process and generating a report based on the latest library database [1]. The scanning and the identification processes are both based on Radio Frequency Identification (RFID) technology. Even though many of the principles are relevant to our project (navigation, book identification, interaction and connection with the database), the aim and human interaction differ fundamentally. AuRoSS’s objective is to find misplaced books and send a report to the librarian, without any interaction with the actual users of the library. On the other hand, our study is focused on improving user experience through interactions with the user. Furthermore, AuRoSS is incapable of using the stairs or the elevator, which is an important development aspect for LIBROB as it is being designed to operate in a five-level library.

Another relevant development in this field is the Comprehensive Access to Printed Materials (CAPM) [2]. This is an autonomous mobile robotic library system developed to retrieve items from bookshelves and carry them to scanning stations located in the off-site shelving facility. Again there are similarities that can be used for this project but also crucial differences. The CAPM project does not directly interact with the user and operates in a off-site facility, while LIBROB aims to interact directly with the user and will operate in the actual library. Similar attempts for robotic applications in library environments have also been made in other universities. At Aberystwyth University, a group of students designed Hugh, a robot reported to help library-users find books in the

library by physically taking the user to any book’s location while interacting with speech [3]. Just as Aurora, Hugh also uses RFID to recognise books; but in contrast it aims at interacting fluently with the users, helping them locate the books in the library, similarly to what we are trying to achieve [4,5]. At University of Birmingham, a robot, Bob, has been assigned with the task of monitoring library spaces, while two robots, Vincent and Nancy were deployed in a public library in Westport, Connecticut, which were designed to teach computer programming skills to library users [3].

3 System Description

3.1 Hardware Design

An illustration of LIBROB and it’s hardware components is shown in Fig. 2. We selected a PeopleBot mobile robot [6] to be LIBROB’s base due to its body dimensions which allow it to move around freely through narrow space between shelves in libraries. In order to interconnect the electronic devices used to operate the robot, a wireless router was installed onto the moving base. This router connects to the library’s wireless network and gives LIBROB internet and local network access to allow all the system devices to communicate using ROS [7]. A tablet sits on top of the robot and plays the role of GUI as well as representing LIBROB’s head. A Raspberry Pi 3b + works as motor control unit. In addition, this unit runs all ROS nodes but the Navigation node, which runs on a remote laptop due to the relatively low memory performance of the Raspberry Pi. One SICK LMS200 rangefinder is used as Lidar sensor to scan the robot’s surroundings. Finally, a microphone coupled with an external sound card is used as audio recording device to help with the speech processing task.

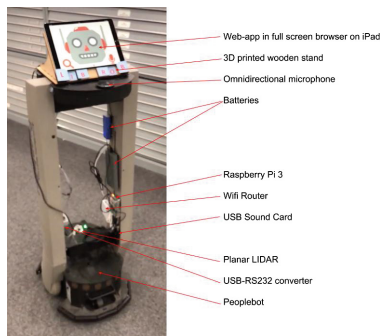


Fig. 2. LIBROB’s hardware components.

3.2 Software Components

The high-level design of LIBROB’s software system is depicted in Fig. 3. The whole system operates around a central ROS node called **Behaviour**. All other nodes in the system primarily communicate with this node, which is in charge of taking the major decisions in the process.

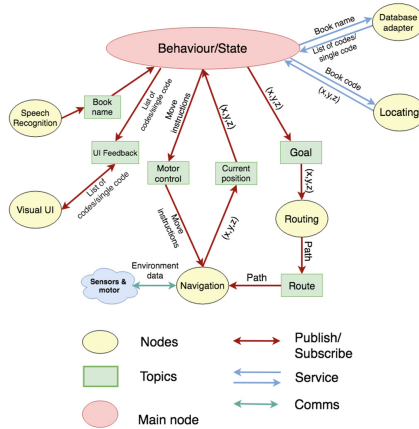


Fig. 3. Software framework diagram.

User Interface. The GUI was implemented as a web application, motivated by the idea of having a cross-platform application that would run on many different devices. The fundamental purpose of the Visual UI node is to make LIBROB more user-friendly, and allow the user to interact with the robot more intuitively.

In a typical interaction between LIBROB and a user, first the user approaches to LIBROB and selects one of the two available options to input a book title (see Fig. 4a): the user can either press the microphone icon and ask for a book by talking to the robot, or press the search icon to type in the book title. A drop-down menu is also available at the bottom of the page to allow for different language options. Indeed, LIBROB is able to communicate in different languages (LIBROB currently support English, French, and Italian). Once LIBROB receives the book title, a list of the books returned by the database is displayed as shown in Fig. 4b. Relevant information about the title, author, floor number, availability and identification code of each book is also provided. The user is able to scroll through the list, find and select the desired book by pressing on the icon next to it (see Fig. 4c). After pressing this icon, LIBROB will start moving towards the destination shelf where the book is located. At this point, LIBROB provides an audio-visual message saying “let’s go!” to tell the user to follow him. Once LIBROB reaches the book’s location, it will announce to the user “We have arrived at your shelf !” (see Fig. 4d) and the operation is terminated.

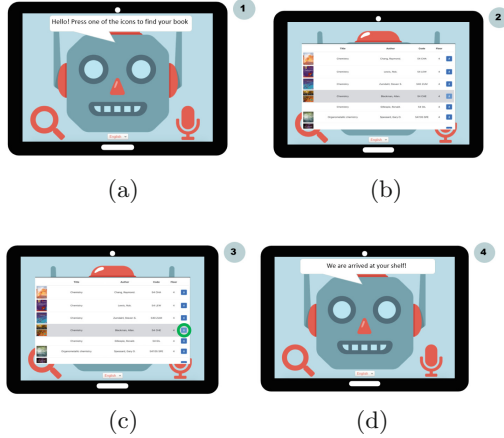


Fig. 4. LIBROB’s Graphical user interface.

These specific messages are used for a standard scenario with no issues. However, other types of messages have also been designed to handle different situations, such as when the robot has not found a book, or when the user is trying to interact with it while it is moving. Furthermore, all these messages are available in all supported languages.

Speech Processing Node. With the aim of ensuring the smoothest human-centered interaction, much effort was allocated to the **Speech node**. In order to allow LIBROB to communicate with humans naturally, we made use of a combination of three technologies within the field of Speech Processing, namely real-time speech recognition, language translation, and natural language processing. The **Speech node**’s initial goal is to make the speech-to-text conversion of what the user says through speech recognition, and to publish the book request on a ROS topic that would be fed to the **Database Adapter node**. This task was made robust by ensuring it would operate well even in noisy environments and in environments where noise levels may vary. Furthermore, to allow for a more natural communication between LIBROB and their users, we added a natural language processing layer to the design, with the use of the *Snips NLU* package. This package is based on a machine learning model trained with many examples of how humans could ask for a book orally [8]. The objective of this layer is to extract semantic information on the presence of a book title or an author’s name in the transcribed string, in which case the relevant information is sent as a request. Following the implementation of this natural language processing layer, LIBROB was able to understand sentences such as: “*Could you help me find a book on Organic Chemistry written by Alan Turner please?*” or “*I really need a book to study for my Digital Signal Processing exam tomorrow!*”.

Moreover, we decided to add real-time language translation to the **Speech node** to allow users to talk to LIBROB in their native language. Thus, depending

on whether or not the user requests for a book in English, a language translation to English is performed in the background using the speech translation package from the Python Package Index [9] and the translated string is fed to the natural language processing layer. The same English-trained model is used for any language as it only has to operate with the output of the translation layer. A full work flow of the speech node can be found in Fig. 5. In addition, the **Speech node** is also designed to handle the case where the user has to take the lift to get to the right floor. In this scenario, a background listener is triggered as soon as LIBROB enters the lift to listen continuously until it hears the floor number corresponding to the floor where the book is located.

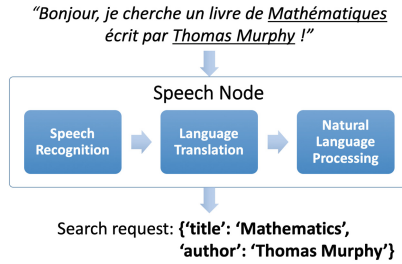


Fig. 5. Work flow of the Speech node. The example input is a French sentence translated as: “Hello, I am looking for a Mathematics book written by Thomas Murphy!”

Database Adapter Node. Determining exactly which book the user is looking for, based on the limited amount of information input during the search process, is crucial to the correct operation of LIBROB. The logical solution to achieve this is to connect the robot with a database where all the relevant information of every book in the library is stored. Due to the large amount of books generally available in a library, storing such a database locally on LIBROB is impractical as it would require a tremendous amount of memory space. The method that is currently used in LIBROB is to run an HTTP request script to the library’s database Application Programming Interface (API) which will only return information based on the search performed by the user.

The **Database Adapter node** script can perform the search with title information, author information or both. By adding arguments to the search, it is possible to perform a more specific search. The most important argument added is to limit the search to printed books so that it excludes any journals, articles or e-books. Furthermore, additions or subtractions to the library are recorded and the availability of books is tracked by the library and visible on the API’s response. This information can therefore be exploited by LIBROB to enhance the system. The data returned by the search request is then filtered in order to extract the book’s title, author, code, floor, availability and thumbnail.

This information is then made available to other LIBROB’s nodes as required. Figure 6 is an example of one search response returned by the **Database Adapter node** when a search request is given to LIBROB.

```
{
  "title": "Harry Potter and the Order of the Phoenix",
  "author": "J.K. Rowling",
  "code": "800.R0W",
  "floor": "1",
  "availability": "True",
  "thumbnail": "https://proxy-eu.hosted.exlibrisgroup.comexl"
}
```

Fig. 6. Example of one search response returned by the Database Adapter node for a search request with title key “Harry Potter”.

Locator Node. The purpose of the Locator node is to provide the book’s location inside the library given it’s code. When an user selects a book (see Fig. 4c) the correspondent code obtained by the **Database Adapter node** is passed to the **Locator node**. The **Locator node** then searches through the list of shelves and responds with the book’s coordinates that are used as a goal for the **Navigation node**. The relation between book-codes and their positions in the library is made using a file in which only the code of the first book in every shelf together with their corresponding location is stored. Just with this information, a range of all the books contained in an specific shelf can be obtained, giving LIBROB all the information needed to take the user to the correct location.

Navigation Node. As the layout of the library does not change often, a map of the library is assumed to be available. We mapped all five floors of Imperial College’s central library by constructing the map while driving the robot manually using the on-board Lidar [10]. Maps were manually corrected and normalised such that lifts overlap in different floor maps. An example of the maps used by LIBROB is shown in Fig. 7. The robot is localised using ROS’s navigation stack [11, 12]. In addition, the navigation module also uses the ActionLib library [13] to notify the user that LIBROB has reached the book shelf.

Behaviour Node. Figure 8a shows the design of the state machine implemented in the **Behaviour node** and that determines the actions to be taken by LIBROB based on received inputs and measurable goals. During the *Idle* state LIBROB is waiting for a user to initiate interaction and request a book title. In *Display list of Books* state the database request results are displayed to the user. *Navigate to Book* state uses the assigned shelf coordinates for the user-selected book as a navigation goal. Then, in *Indicate book* state LIBROB notifies the user where the book is located. Finally, in the *Return to station* state LIBROB uses the station coordinates to navigate back to its original location.



Fig. 7. Map of Imperial College Library’s 4th Floor.

To implement the steps taken to navigate to a book on a different floor, the *Navigate to Book* state is separated into multiple sub-states as shown in Fig. 8b, and include the following: in *Move to Lift* state LIBROB navigates to the front of the lift doors. Then, in *Wait for Lift* state LIBROB waits for the indication that the lift doors have opened. In *Enter Lift* state LIBROB enters the lift and finally in *Move to book* state the requested shelf location becomes the new navigation goal.

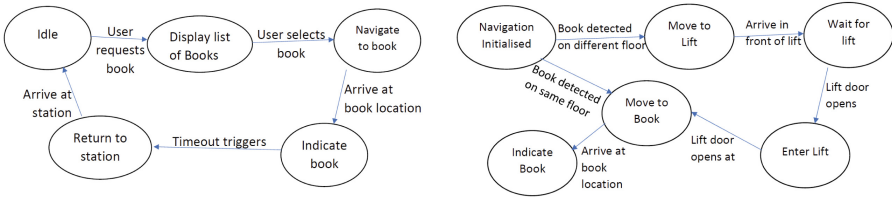


Fig. 8. State machines of the Behaviour node. (a) Main state machine. (b) Navigation state machine.

4 Experimental Procedure and Evaluation

4.1 Experimental Setup

We conducted experiments at Imperial College’s Central Library where 40 volunteers were asked to search for a given book. All participants were students from Imperial College London. Participants were divided in two groups: the control group and the LIBROB group. The control group only had access to the existing library’s search system (described in Sect. 1) to search for a book, while the LIBROB group was only allowed to use LIBROB. We compared these two groups in terms of the average time needed to find a book. In addition, we

retrieved students’ satisfaction and preferred method (speech or text) of interaction when using LIBROB, the successful against failed attempts of LIBROB to guide a user to requested book and the number of times the user required help to use LIBROB. Consent was obtained from the participants followed by a demographics questionnaire.

To start, participants were asked to find a particular book within the fourth floor of the library. Participants from the control group were given with a laptop and the library’s search system ready to use at the same start position that the robot was placed for participants of the LIBROB group. No training sessions or further instructions were given. Then, participants filled a final questionnaire. Participants from the control group were asked if they would like to see improvements on the current library’s search system. Participants from the LIBROB group were presented with four questions regarding (1) perceived usefulness and (2) how satisfied were they with the robot, (3) how likely they were to use LIBROB again and (4) how intuitive it was for them to use it. All these questions were rated using a scale from 1 to 5. Participants reporting not finding LIBROB useful were asked to further describe the reasons why.

Statistical analysis is performed using two group comparison test (i.e. T-Test). Normally distributed values are represented as means and standard deviations.

Table 1. Demographics of the participants from the experiment ($N = 40$).

Factor	n	%
<i>Gender</i>		
Female	11	27.5
Male	29	72.5
<i>Age (years)</i>		
18–24	37	92.5
25–34	3	7.5
<i>Group</i>		
LibRob	20	50
Control	20	50

4.2 Results

The demographics of the participants are shown in Table 1. Every participant successfully found the book they were asked to find. However, the LIBROB group spent 47.4% less time in doing so, resulting in a reduction of almost half the time as shown in Fig. 9. The average time for the LIBROB group was 91.9 s (SD = 24 s), while for the control group was 174.9 s (SD = 40 s). A T-Test showed that there was a significant difference in the time to find a book ($p < 8.191e-09$) between both groups. Furthermore, it should be noted that LIBROB also reduces

the variance of the search time. This is an expected result as once the user inputs their query, the robot guides them to the right shelf, and the variance in time is only introduced when users need to locate the book within the shelf. Such results confirm that LIBROB helps users to find books faster.

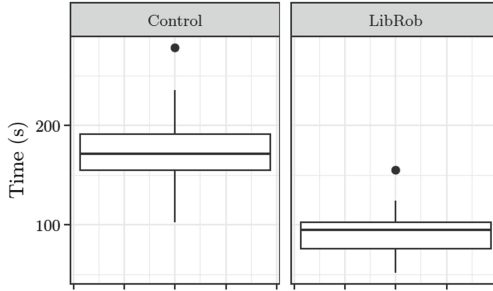


Fig. 9. Experiment results. Time to find a book for each group.

When control group participants were asked if they would like to see improvements on the current library’s search system, 85% of participants in the control group stated that they would like to have a better system to find books in the library, suggesting a considerable need for a better solution to locate books in the library. The final questionnaire results for the LIBROB group are summarised in Fig. 10. The results allow us to conclude that LIBROB not only can reduce the time users spend searching a book in the library, but also that, even in cases where the time spent looking for a book was not significantly reduced, it still ensures a positive experience as evidenced by the high rates reported by participants: satisfaction - 80%, usefulness - 70%, willingness to use the robot again - 85% and intuitiveness - 90%. Only 10% of the participants were not immediately sure about how to operate LIBROB, further supporting how intuitive it is to use. Overall, the interactions between the users and LIBROB proved to optimise the book location time and provide a satisfactory experience for the library users.

When comparing the use of text-typing and speech-to-text, only 20% of users used text-typing to search for a book title, suggesting that speech-to-text is the preferred method of interaction with LIBROB. Only three participants reported not being satisfied with the robot. Two of them claimed that LIBROB was too slow or took too long. This was primarily due to the nature of the robot base itself as it has a maximum forward speed of 0.8 m/s, which could feel slow for some users if they are covering considerable distances. The third one was not satisfied with the precision of the indications received, he found that identifying the right shelf was not sufficient, and wanted LIBROB to point precisely at the book. Nevertheless, none of these participants found issues while interacting with the robot or reported LIBROB bringing them to the wrong shelf.

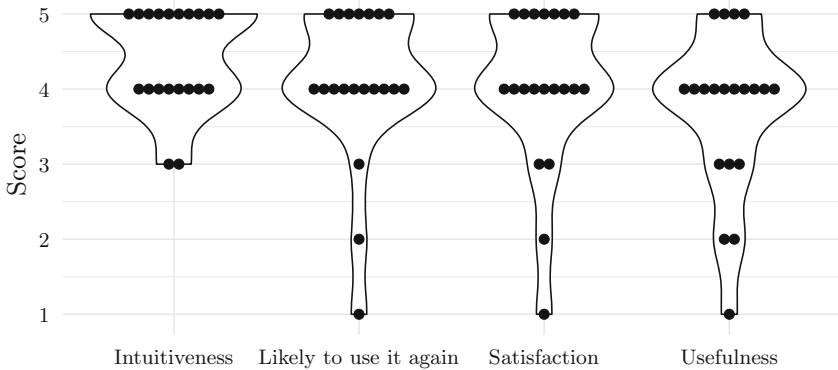


Fig. 10. Responses to the survey presented to the LIBROB group.

5 Conclusions and Future Work

The current version of LIBROB allows the users to be successfully guided towards the book they are looking for. Nevertheless, further improvements are still possible, some of which were already implemented as separate functions. A face recognition algorithm was developed to detect faces of the users through the tablet's webcam to track their face position and allow LIBROB's eyes to follow them. This functionality has the potential to capture the users' attention and make them feel even more engaged when interacting with LIBROB. Another feature that was not tested in this first prototype is the use of elevators. However, in its current version LIBROB is able to understand when it has arrived to the correct floor as well as entering the elevator and updating the floor map once reaching a new floor. The reason why it was not evaluated in our experiment is due to the lack of an internet connection in the elevator, which is fundamental for the speech recognition package to operate. `Pocketsphinx`, an offline speech recognition package, was also tested, however the accuracy of the speech recognition was considerably degraded. Finally, some participants suggested the inclusion of a tool to point to the queried book in order to further reduce the time spent looking for it once the shelf has been reached. In order to do so, a method to recognise the book codes on the shelf together with an orientation-controlled laser, the robot could be able to point at the right book.

In conclusion, LIBROB has been successfully implemented and tested. With the results obtained, we confirmed that the time taken for students to locate a book was significantly reduced when they were using LIBROB. Furthermore, the users were overall more satisfied with their experience. In addition, the feedback obtained indicates that many users were happy with the GUI, found it intuitive and felt it enhanced their experience with the library system. While test participants were all Imperial College students and not representative of the general public, we have hopefully demonstrated sufficient evidence that our approach is an efficient solution for this type of application.

References

1. Li, R., Huang, Z., Kurniawan, E., Ho, C.K.: AuRoSS: an autonomous robotic shelf scanning system. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6100–6105. IEEE (2015)
2. Suthakorn, J., Lee, S., Zhou, Y., Thomas, R., Choudhury, S., Chirikjian, G.S.: A robotic library system for an off-site shelving facility. In: 2002 Proceedings of IEEE International Conference on Robotics and Automation, ICRA 2002, vol. 4, pp. 3589–3594. IEEE (2002)
3. Phillips, D.: Robots in the Library: gauging attitudes towards developments in robotics and AI, and the potential implications for library services. Dissertation C (2017)
4. Meet Hugh, the robot librarian. <https://www.aber.ac.uk/en/news/archive/2016/02/title-181095-en.html>. Accessed 27 Oct 2018
5. Robot librarian designed by Aberystwyth University students. <https://www.timeshighereducation.com/news/robot-librarian-designed-aberystwyth-university-students>. Accessed 27 Oct 2018
6. Activmedia Robotics, L.L.C.: Pioneer 2/PeopleBot™. Operations Manual. <http://www8.cs.umu.se/research/ifor/dl/Product%20info/ActiveMedia/Pioneer%20%20%20PeopleBot%20Operation%20Manual.pdf>. Accessed 7 Feb 2019
7. Quigley, M., et al.: ROS: an open-source Robot Operating System. In: ICRA workshop on open source software, vol. 3, no. 3.2 (2009)
8. Coucke, A., et al.: Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. arXiv preprint [arXiv:1805.10190](https://arxiv.org/abs/1805.10190), 25 May 2018
9. PyPI. <https://pypi.org/project/translation/>. Accessed 5 Dec 2018
10. Kohlbrecher, S., Von Stryk, O., Meyer, J., Klingauf, U.: A flexible and scalable slam system with full 3D motion estimation. In: 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), pp. 155–160. IEEE (2011)
11. Navigation. <http://wiki.ros.org/navigation>. Accessed 8 Feb 2019
12. Zheng, K.: ROS navigation tuning guide. arXiv preprint [arXiv:1706.09068](https://arxiv.org/abs/1706.09068). Accessed 27 Jun 2017
13. Actionlib. <http://wiki.ros.org/actionlib>. Accessed 8 Feb 2019