Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-30 (2016) 048–053

A Platform for Cloud Robotics [★]

Amadeu do Nascimento Jr. [*] **Eleri Cardozo** [*]
Ricardo S. Souza [*] **Eliane G. Guimarães** [*]

[*] *School of Electrical and Computer Engineering, University of Campinas, Campinas/SP, Brazil (e-mail: amadeujr@dca.fee.unicamp.br).*

Abstract: This paper presents the evolution of a software platform for supporting experimentation in mobile robotics as part of teaching and researching activities. Starting with Web-based laboratories (WebLabs) in the early 2000s the platform kept evolving according to the networking and distributed computing trends since then. In addition to the physical resources managed by the platform, the platform now is able to manage a pool of virtual machines as resources for experimentation. This new class of resources brings the processing power as required by many modern mobile robotics applications. Virtual machines can be widespread on a cluster of processors, on a private cloud computing infrastructure, or on a public cloud computing service. Like any other resource managed by the platform the access to the virtual machines is subjected to user authentication and authorization. A mechanism of user authentication and authorization based on federated identities (single-sign-on) allows the sharing resources maintained by different administrative domains. The paper emphasizes the current stage of the platform and a case study in mobile robotics localization. Localization, as many other mobile robotics algorithms, can employ parallelism at the cluster and cloud levels in order to improve speed, reliability, and scaling.

Keywords: Telerobotics, robot programming, mobile robots, telematics, cloud robotics.

1. INTRODUCTION

Mobile robotics attracts the interest of students and researchers due its multidisciplinary nature and wide spectrum of practical applications. Embedded systems, automatic control, artificial intelligence, human-machine interface, computer vision, and real-time systems are examples of areas with direct application in mobile robotics.

Teaching and researching in the field of mobile robotics require an infrastructure for supporting experimentation. This infrastructure can be integrally software-based where a simulator is employed to mimic mobile robots and their environments. A simulator is an important tool in case of multi-robot scenarios where tens or hundreds of cooperating mobile robots are employed or when experimenting with mobile robots that are not physically available. Even when mobile robots are available for experimentation, a simulator is convenient for debugging and tuning the robotic algorithms without the need to operate the physical equipment at an early stage of software development.

When the algorithms developed by students and researchers are ready to be tested in real world conditions, a concern is how to deploy this software on mobile robots. The more direct approach is to allow the software be installed in the mobile robot's embedded processors. As the embedded processors have limited processing power this solution is feasible only for less demanding applications such as teleoperation and obstacle avoidance.

A possible solution is to employ an adjunct processor, typically a notebook, where the software controlling the mobile robot runs. This processor can be placed on the top of the robot and to communicate with the mobile robot via a wired or wireless link. This solution also presents drawbacks as these processors must run all the software packages necessary to operate the mobile robot alongside with the operating systems, programming languages, and device drivers these packages require.

A more practical approach is to allow the interaction between the computer running the robotic application and the mobile robot be conducted over the network. We are developing a software platform to employ this approach since the mid of 1990s. This software platform, referred as the REALabs platform, is being continuously updated as the networking and distributed computing technologies evolve. We can identify some landmarks that motivated such updates. The first one is Web Services introduced in end of the 1990s. The second landmark is the HTTP/XML-based RPC (Remote Procedure Call) as a lightweight alternative to Web Services. The third landmark is the REST (Representational State Transfer) architectural style that replaces the concept of procedure call in RPC by the concept of object state manipulation. Finally, the last landmark is virtualization as an enabling technology for cluster and cloud computing. In fact, the current version of the REALabs platform employs REST as an interaction model with the mobile robots and virtual machines as processing elements where the platform services and the robotic applications run.

[★] This research was supported in part by FINEP and FAPESP.

2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.11.124

The paper is organized as follows. Section 2 describes the requirements of a software platform for support teaching and researching in mobile robotics as well as the motivating factors that drove the evolution of the REALabs platform developed at our Mobile Robotics Lab. Section 3 presents the actual stage of development of this platform. Section 4 presents a case study where the platform supports parallelization of mobile robotics algorithms. Section 5 presents some related work describing software platforms for supporting networked and cloud robotics. Finally, section 6 concludes the paper pointing out some evolutions already identified.

2. THE REALABS PLATFORM

This section describes the requirements of a software platform for experimentation in the field of mobile robotics that drove the evolution of the REALabs platform. The current stage of the platform in line with cloud robotics is presented in the next section.

2.1 Software Requirements

A software platform for experimentation in mobile robotics differs from robotic frameworks such as ROS (Robot Operation System), Quigley et al. (2009). Although it is possible to develop robotic applications directly over a robotic framework, a software platform offers many added facilities as long as some requirements are fulfilled. In short, robotic frameworks offer facilities to manipulate the robot's hardware while software platforms offer facilities that favor software integration.

The first requirement of a software platform for mobile robotics is its independence of operating systems, programming languages, and software architectures. This demands interaction mechanisms with the mobile robots at a higher level than those offered by the robotic frameworks. The platform must rely only on well established and accepted open standards and protocols.

The second requirement is the secure access to the resources managed by the platform. Authentication and authorization mechanisms as those provided by open standards such as OpenID and OAuth, Siriwardena (2014), are able to provide secure access to the resources even if resources are widespread across multiple administrative domains.

The third requirement is the availability of easy to use management interfaces to deploy and configure the platform for a particular need, e.g., a course in mobile robotics, a research project in outdoor navigation, and so on. Such management interfaces can rely on server pages and databases for storing users, resources, reservations, and active sessions.

Finally, as a fourth requirement, the platform must offer simple and extensible programming interfaces to the services it provides. Interfaces based on RPC composed of hundreds of operations as those found on robotics frameworks are not adequate for software platforms. Instead, interfaces combining few operations and an elaborated resource representation (e.g., as provided by ontologies) is a clear trend in software platforms. REST-based (RESTful) programming interfaces, as adopted by the today's major internet corporations, are examples of this trend.

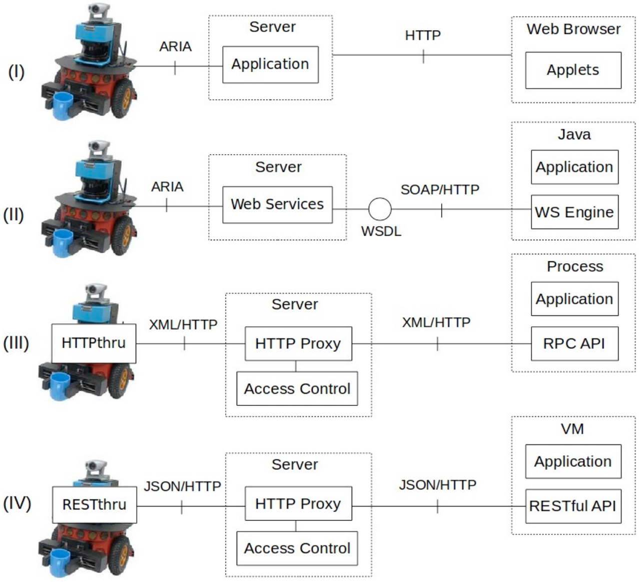

Fig. 1. Previous versions of the REALabs platform

2.2 REALabs Platform Versions

The four previous versions of the REALabs platform are depicted in Fig. 1 and described in the sequence.

Java Version. This early version of the REALabs platform (circa 2000) took advantage of the then recently introduced Java programming language and its related technologies such as Java applets, Java servlets, and Javascript. The idea was to offer interfaces in a Web browser where certain operations on the mobile robot can be performed, e.g., teleoperation, sensor readings, and uploading of code written in C to control the robot.

One drawback of this version of the platform is that it requires the programming of the uploaded code in C using a low level robotic framework provided as by the mobile robot's manufacturer. Moreover, no security mechanism exists to avoid concurrent access to the mobile robot or to avoid damages to the robot due to the execution of a faulty uploaded code. Fig. 1(I) presents this version of the REALabs platform.

Web Services Version. By the mid 2000s Web Services was considered a key technology for developing distributed applications over the internet. An XML-based language for service description, WSDL (Web Service Description Language), was introduced for specifying services accessible via the SOAP (Simple Object Access Protocol) protocol. This version of the REALabs platform defined a set of WSDL interfaces for locomotion, actuation, and sensing operations on the mobile robots. Applications can generate stubs in different programming languages from the WSDL specifications as well as to combine Web services in workflows through the WS-BPEL (Business Process Execution Language) language.

Although the solution adopted in this version of the platform is elegant, several drawbacks were identified. Firstly,