



HAL
open science

Towards a new approach of Robot as a Service (RaaS) in Cloud Computing paradigm

Labib Sadek Terrissa, Bouziane Radhia, Jean-François Brethé

► To cite this version:

Labib Sadek Terrissa, Bouziane Radhia, Jean-François Brethé. Towards a new approach of Robot as a Service (RaaS) in Cloud Computing paradigm . 5th. International Symposium ISKO-Maghreb (Knowledge Organization in the perspective of Digital Humanities: Researches and Applications), International Society for Knowledge Organization, Nov 2015, Hammamet Tunisia. hal-01245223

HAL Id: hal-01245223

<https://hal.science/hal-01245223>

Submitted on 11 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards a new approach of Robot as a Service (RaaS) in Cloud Computing paradigm

Labib Sadek Terrissa
Intelligent Computer Science
Laboratory – LINFI
Dept. of Computer Science
University of Biskra, Algeria\
terrissa@univ-biskra.dz

Radhia Bouziane
Intelligent Computer Science
Laboratory – LINFI
Dept. of Computer Science
University of Biskra, Algeria\
radia.bzne@gmail.com

Jean-François Brethé
GREAH Laboratory
University of LeHavre, France
jean-francois.brethe@univ-lehavre.fr

Abstract— For many years, robots appear in most fields. At times, we approach closer to create intelligent robots due to success in computer technology. Current developments in Cloud Computing introduce this concept in robotics field. Many projects have taken initiatives towards Cloud Robotics that allow the robots to outsource computing capabilities over the Cloud, like communication resources, storage, and computing power. The aim of our work is to define a kind of service in the direction of Cloud Robotics, which is ‘Robot as a Service’ (RaaS). This concept is based on the functions of SOA, and means the use of robots features in the Cloud Computing as a demand service.

Keywords— Robots, Cloud Computing, SOA, Cloud Robotics, Robot as a Service.

I. INTRODUCTION

Robotics is a field of science and technology dealing with the issues of design, control, applications and maintenance of robots [1]. Robots are now used widely in scientific research, manufacturing technology, civil engineering, as well as in medicine, underwater investigations and space research. An important challenge in this domain is to make robots able to operate and adapt with an autonomous way to their environment.

Cloud Computing is a generic term that involves delivering hosted services over the Internet. The name Cloud Computing was inspired by the Cloud symbol that is often used to represent the Internet [2].

Cloud Computing has potential to speed up many computationally-intensive robotics. In 2010, James Kuffner coined the term “Cloud Robotics” and described a number of potential benefits [9]. This approach would allow robots to offload compute-intensive tasks like image processing and voice recognition and even download new skills instantly [4]. This paper presents our research on the related concepts of the purpose of this work that is

‘Robot as a Service’ (RaaS), and an illustration of our design to implement this kind of service.

II. RELATED CONCEPTS

A. Cloud Computing

Cloud Computing is the latest paradigm that involves delivering hosted services over the Internet, based on a pay-as-you-go approach. According to NIST (National Institute of Standards and Technology), Cloud Computing is on-demand access to a shared pool of computing resource. It is an all-inclusive solution in which all computing resources (hardware, software, networking, storage, and so on) are provided rapidly to users as demand dictates [2].

The concept of Cloud services development encompasses several three different types of development: Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS) [5], and even X as a Service (XaaS), where X can be any Items such as: Robot as a Service (RaaS) [6].

B. Service-Oriented Architecture

Service-Oriented Architecture or SOA is way to design, implement, and assemble services to support or automate business functions [7]. An SOA is built using a collection of services that communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity [7]. Figure 1 shows the collaborations in a Service-Oriented Architecture. The collaborations follow the “find, bind and invoke” paradigm where a service consumer performs dynamic service location by querying the service registry for a service that matches its criteria. If the service exists, the registry provides the consumer with the endpoint address for the service [8].

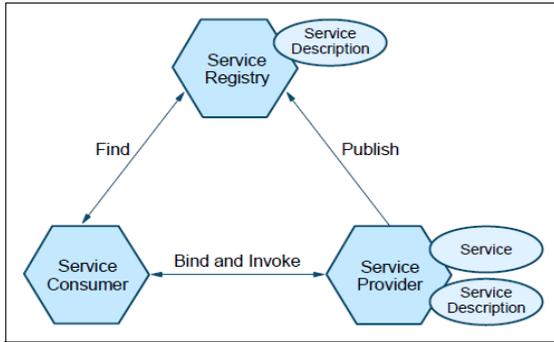


Figure 1. Collaborations in a Service-Oriented Architecture [8]

Cloud Computing extends the scope of SOA to include the development platform and the execution infrastructure [6]. The services provided in the Cloud come from multiple organizations. These organizations are referred to *Cloud providers*. As a result, the Cloud has multiple services provided by multiple Cloud providers [7].

C. Robot Operating System (ROS)

ROS is an open-source, meta-operating system for your robot. It provides the services you would expect from an operating system, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management [15]. With the aid of ROS, it is possible to command the robot to walk from place A to place B without thinking about which walking algorithm will be used. This meta operating system is extendable with modules, called stacks. There are modules for many robots like Nao, TurtleBot or PR2 [11].

In ROS, processes are called nodes which are organized in a Peer-to-Peer network without a central point of communication. The master node provides name registration and lookup for all custom nodes. Without the master node, it is not possible to communicate with nodes, services, messages, and others [11].

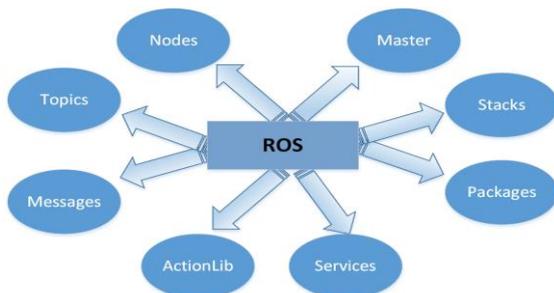


Figure 2. Core concepts of ROS [11]

D. Cloud Robotics

In 2010, James Kuffner has introduced the topic “Cloud Robotics” [9]. This idea combines the computational power of the Cloud Computing and the availability of internet-connected devices in robotics [10]. The advantage of this concept is that it enables robots to carry just a minimum set of hardware resources, and access all knowledge and resources available on the Cloud.

Cloud Robotics pushes the idea of Inaba “remote brained robots” (1997) [3] with the aid of unbounded computation power on demand [11].

III. RELATED WORK

The following projects are some currently working on the topic of Cloud Robotics.

A. RoboEarth

RoboEarth represents a World Wide Web for robots [12]. Through RoboEarth robots are able to share data and learn from each other [3].

This project is part of the Cognitive Systems and Robotics Initiative founded by the European Union. It provides a distributed database to store information like object descriptions, maps and task specifications [11] created by humans in a format that is readable by the robots.

RoboEarth offers a Cloud Robotics infrastructure, which includes everything needed to close the loop from robot to the Cloud and back to the robot.

B. Rapytua

Rapytua is an open source Platform-as-a-Service (Paas) framework designed specifically for robotics applications [12]. It is based on an elastic computing model that dynamically allocates secure computing environments for robots [13]. Furthermore, this project is also called “the RoboEarth Cloud Engine” because Rapytua’s computing environments allow robots to easily access the RoboEarth knowledge repository [12].

Figure 3 shows that by extending the RoboEarth database, robotic specific data and processing units are both settled in the Cloud [11].

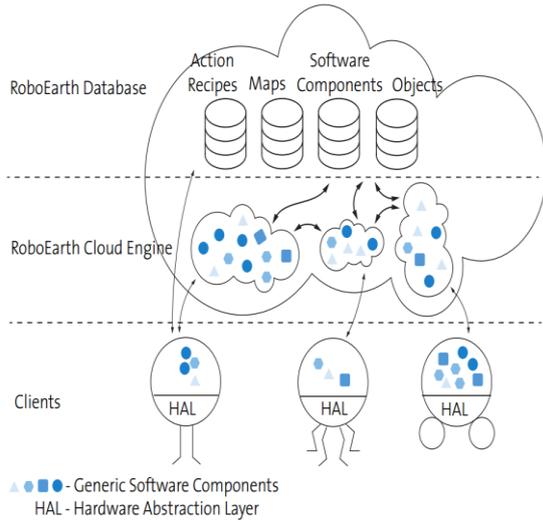


Figure 3. The Cloud Engine “Rapyuta” [11]

C. Cloud-based robot grasping

It is a project which goal has been to determine the optimal way to grasp a newly encountered object for robots [3], with the aid of the Google object recognition engine, which is established by the Google Goggles [11], a free image recognition service for mobile devices [3].

The project is structured into two phases [14]:

- The offline phase which aims to train the recognition server by storing semantic information about objects.
- The online phase where robots upload images about objects to receive a possible grasping policy by matching sensor data against 3D CAD models in an online database.

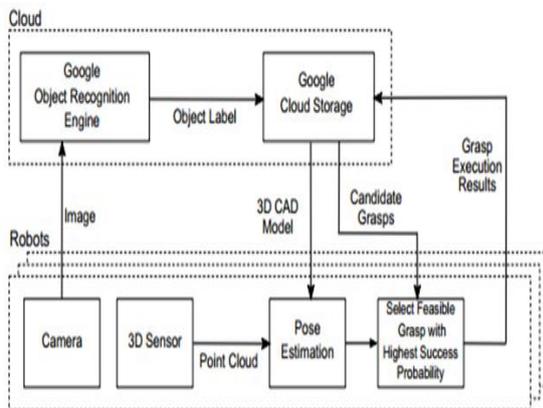


Figure 4. Online phase of Cloud-based robot grasping [14]

IV. DESIGN OF RaaS

In this section, we will clarify a global view of our design to implement the RaaS, the use of robots

features on demand, as a service, by the research of available services in different Clouds, where a robot would be able to accomplish a specific task with a simple connection to the Cloud, and regardless of hardware or software.

The general architecture of our system is illustrated as follows in Figure 5.

To determine the composition of a RaaS system, we have identified four components in this architecture:

1. The Cloud Administrator: A Cloud side administrator, used to deploy new services (executable packages) in the Cloud, available to a Client.
2. The Client: The Client RaaS benefits of the characteristics of Cloud Computing. It represents two main actors:

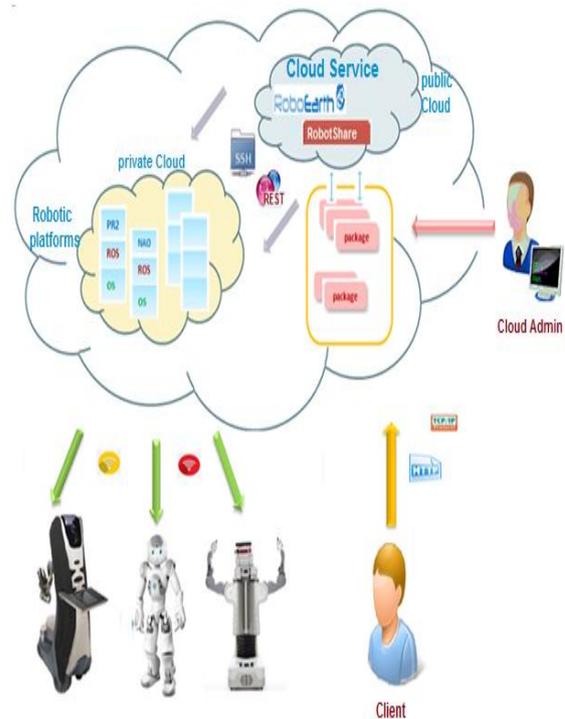


Figure 5. System Architecture

- Client Administrator: He intervenes in the client side, he is responsible of the management and the local robots configuration through a Web interface that provides the following operations :

1. Adding a new robot: by adding its identifier, physical logical address, the robot type, the system used packages and

Cloud resources to be allocated for the robot ('RAM, HDD and CPU').

2. Updating the robot configuration.
 3. Delete a robot.
 4. Control of robots: (Stop/Start robots ...etc).
 5. Robot monitoring: supervising the robot behavior and the data processed.
 6. Purchase: This option allows the purchase of new resources or packages (packages shows the operation or discipline of the robot such that, the client can buy a package manufacturing boxes, or a package for house cleaning, or harvest agriculture package ... etc.).
- Robot: this is the main actor. It benefits of all the services necessary to perform its required task.

The relationship between these two components is inspired by SOA (Figure 6).

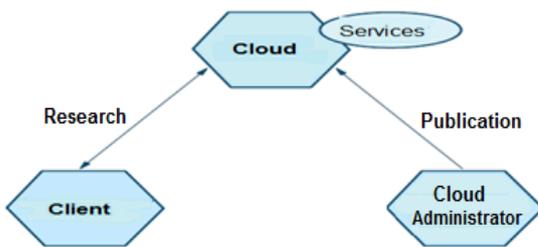


Figure 6. SOA and System components

3. The Cloud: For Cloud Robotics solution, the Cloud is the part that provides the software, and the infrastructure necessary for the applications of a robot. In our system, we propose here the storage of executable packages, and the use of infrastructure. We have defined a virtual robotic layer which contains virtual robot systems (VRS). The VRS is a robotic operating system which is virtually executed in the cloud environment. In our experiment it is composed by an operating system (Ubuntu), a robot operating system (ROS) and stacks. It represents a robotic platform.
4. The network: The set of protocols and technics used to ensure the connection and the communication between the Client and the Cloud.

The idea of a RaaS system is to use an executable plan as a service from internal or external repertories in the Cloud. If the plan does not exist in the local plan, we start a research task in the other Clouds.

V. SCENARIO

We can propose a several scenario for our approach but the most typical scenario is when the robot needs packages to perform a specific task, it will be able to connect to the Cloud and retrieves these packages. In the Cloud side, the provider should offer the requested package. The functioning of the system is described in Figure 7.

- An administrator Client can send a request to a Cloud service for a package (already published by the Cloud Administrator) compatible with a particular robot.
- The only requirement is to have a connection so that the robot can execute the associated program in its virtual machine.

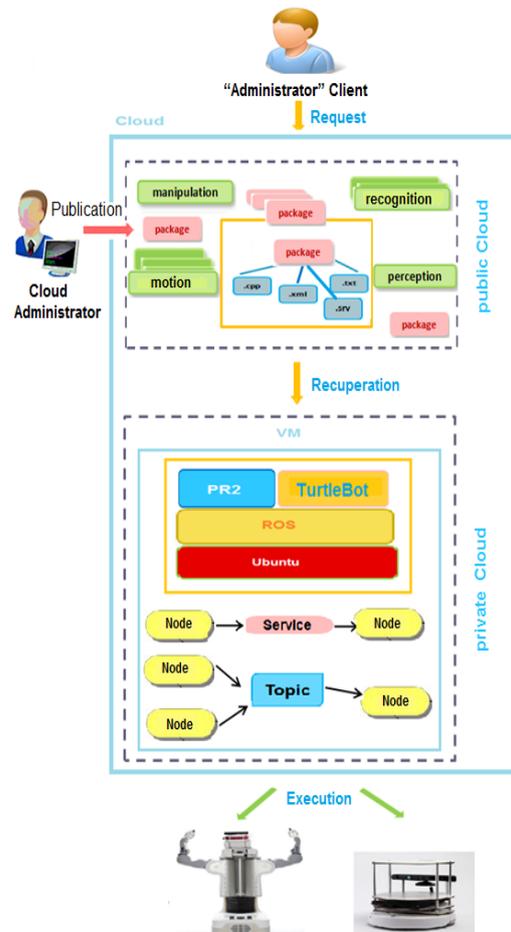


Figure 7. Functioning of the system

- The program manages a set of Nodes, and also the publication of Messages in associated Topics. These notions of ROS can perform a set of operations on the robot.

The following sequence diagram shows the interaction between the system components.

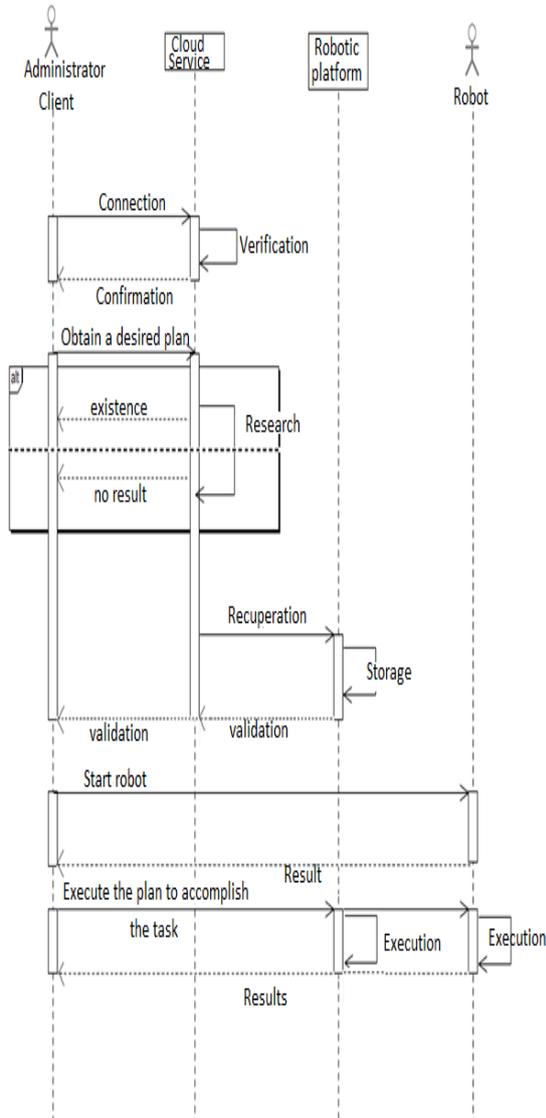


Figure 8. Sequence diagram

The “Administrator” Client, after an authentication, sends a request to the Cloud service, asking for an executable plan which delegates the commands to the particular robot. After exploring the database the plan will be stored, if exist, in the private robotics platform for execution.

VI. IMPLEMENTATION

To test our approach we are simulated the robot service oriented architecture. Figure 9 is a descriptive scheme of the application.

The idea is to recover an executable package compatible with ROS from the machine 2 to run it on the machine 1, so that the robot connected to this latest machine could accomplish its assigned task, that simulates the behavior in a Cloud Computing architecture, where the package is a service that is described in the registry, and is generally in the Cloud.

This reflects the concept of 'Robot as a Service' (RaaS) that we have defined.

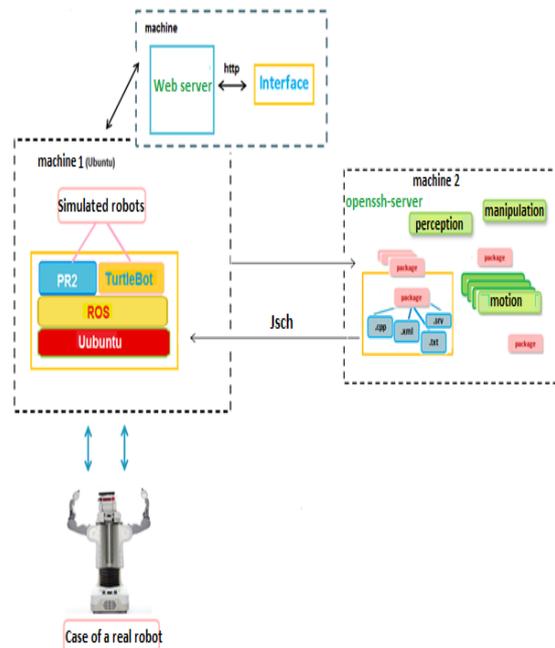


Figure 9. Illustration of the application

We have also developed two interfaces for the PC and smart-phone users (under android). These interfaces are hosted in the Web server of free virtual machine offered by the provider. Through these interfaces the customer is able to manage its robots (add, delete or change a robot). He can also supervise and monitor these robots. After authentication, the client can get the desirable package (See Figure 10).



Figure 10. Main interface

The following Figure (Figure 11) shows the result of the execution of a recovered package (a package of detection and avoidance of obstacles) for the PR2 robot.

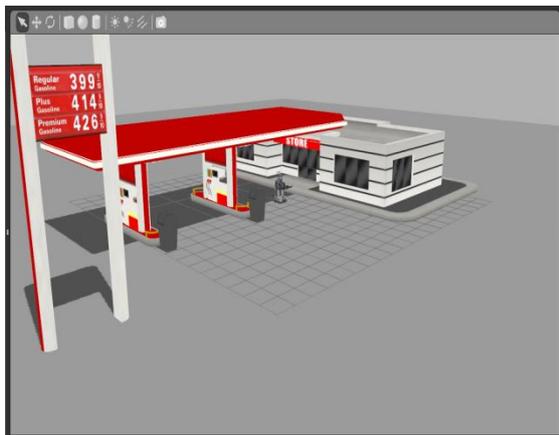


Figure 11. The behavior of PR2 in gas station

VII. CONCLUSION

In this paper, we have defined a new approach in Cloud Robotics. It's Robot as a Service. It's an architecture based on SOA paradigm and ROS as a robotic middleware. This solution allows robots to perform several tasks by connecting to Clouds and do not care about robots features or hardware. We have also tested our approach using PR2 and Care-O-Bot robots with different environment for a first validation. The next step of this work will be to test our architecture on real Cloud infrastructure like Amazon EC2 (public Cloud) with a real client (e.g., industrial client).

REFERENCES

- [1] David Jovovic, "Development and some examples of robots in agriculture", Annals of the ORADEA University, September 2013.
- [2] Zaigham Mahmood and Richard Hill, "Cloud Computing for Enterprise Architectures", Springer, 2011.
- [3] Ben Kehoe, Sachin Patil, Pieter Abbeel and Ken Goldberg, "A Survey of Research on Cloud Robotics and Automation", IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, 2015.
- [4] P. Kopacek OVE, "Development trends in robotics", University of Technology, Institute for Mechanics & Mechatronics, Intelligent Handling and Robotics (IHRT), 2013.
- [5] Christoph Fehling, et al, "Cloud Computing Patterns : Fundamentals to Design, Build, and Manage Cloud Applications", Springer-Verlag Wien, 2014.
- [6] Zhihui Du, et al, "Design of a Robot Cloud Center", Tenth International Symposium on Autonomous Decentralized Systems, 2011.
- [7] Douglas K. Barry, David Dick, "Web Services, Service-Oriented Architectures, and Cloud Computing", 2ed ed Elsevier, 2013.
- [8] Mark Endrei, et al, "Patterns: Service-Oriented Architecture and Web Services", International Technical Support Organization, April 2004.
- [9] J. Kuffner, "Cloud-Enabled Robots," in IEEE-RAS International Conference on Humanoid Robots, 2010.
- [10] Daniel Lorencik, Martina Tarhanicova, and Peter Sincak, "Cloud-Based Object Recognition: A System Proposal", 2014.
- [11] Florian JohannBen, "Nao Robots in the Cloud", Master thesis submitted in terms of the degree course Computer Science at the Department of Computer Science, Hamburg, 2014.
- [12] RoboEarth Project, <http://www.roboearth.org/>
- [13] Rapyuta: The RoboEarth Cloud Engine
- [14] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, K. Goldberg, "Cloud-Based Robot Grasping with the Google Object Recognition Engine", 2013.
- [15] ROS, <http://wiki.ros.org/ROS/Introduction>