CloudIoT: Robot as a Service in Cloud Computing

[1]K.Swathi, [2]Dr A.V.Krishna prasad
Assistant professor, Associate professor
Computer Science and Engineering, Computer Science and Engineering
GPREC (Kurnool), MVSR (Hyderabad)

ABSTRACT: Cloud computing and Internet of Things (IoT) are two very different technologies that are already part of our life. Their adoption and use are expected to be more and more pervasive, making them important components of the Future Internet. A novel paradigm where Cloud and IoT are combined together is foreseen as innovative and as an enabler of a large number of application scenarios.
Cloud computing is shaping the cyber world and evolves as a key computing and service platform for sharing resources including platforms, software applications and everything in the form of services. This is known as "X as a Service". The application of cloud computing is still limited currently in the cyberspace due to the cloud services can only reside in cloud instead of our daily life environment. In fact, there is still a plethora of physical position based on-site service demands that cloud computing could help little due to the "cyber limitation". In this paper, we aim to integrate the cyber world and the physical world by bringing up the idea of "Robot Cloud" to bridge the power of robotics and cloud computing. To make it possible, we design a novel Robot Cloud stack to support our idea and adopt the Service-Oriented Architecture (SOA) to make the functional modules in the Robot Cloud more flexible, extensible and reusable.

Keywords: Cloud computing, Internet of Things, Robot Cloud, Robot as a Service (RaaS).

I. INTRODUCTION AND MOTIVATION

The Internet of Things (IoT) has rapidly evolved in the last years as an umbrella term envisioning that every single object on Earth (the so-called things like robots) can be identified, controlled, and monitored through the Internet. These heterogeneous smart objects are typically endowed with sensing and actuation capabilities, thus being able to capture physical phenomena and translate them into data streams (there by providing information about the environment where they are inserted into), as well as to affect the physical realm as a response to various stimuli. Furthermore, they can seamlessly collaborate with other physical and/or virtual resources also available in the Internet to provide value-added information and functionalities for end-users and applications with minimum human intervention. [**1**]

The dissemination of the IoT paradigm can produce a significant impact of the daily lives of human beings with the emergence of new applications and systems in several real-world domains. From the point of view of a user, IoT will play a leading role in application scenarios such as robotics, smart homes, healthcare, and enhanced learning. From the perspective of business users, the most apparent effects will be similarly visible in fields such as logistics, industry, energy, agriculture, and retail. Ultimately IoT can foster the development of wide-scope applications in the contexts of smart cities, environmental monitoring etc.

Recently, the Cloud Computing paradigm has been advocated as a promising solution to meet some of the requirements of IoT. Cloud Computing can be defined as a model that allows accessing a set of shared and configurable computing resources (e.g., networks, servers, storage facilities, applications) offered as services. These resources can be rapidly provisioned and released with minimum management effort or interaction with the service provider and are offered on-demand, so that users pay only the amount of the effective use of a resource, i.e., in a pay-per-use model. In addition, Cloud Computing promises reduced upfront investment, high availability, fault-tolerance, virtually infinite scalability, etc., characteristics that have notably attracted the attention from both academia and industry. These features are appealing to IoT as they will allow any device to be a simple terminal without the need of possessing large computational resources since cloud services can be transparently, pervasively accessed via Internet. Envisioning the potential benefits brought by Cloud Computing to IoT, major cloud vendors such as Amazon and Google have recently started offering cloud services aiming to support IoT devices and applications in terms of computing capabilities, data analytics, resource elasticity, and scalability. [**1**]

The alignment of IoT and Cloud Computing can take place through two different ways, namely (i) bringing the cloud to the things and (ii) bringing the things to the cloud. Bringing the cloud to the things refers to take advantage of the main features offered by cloud services to compensate technological constraints of IoT in terms of storage, processing, and energy, as well as to enhance the capabilities of IoT infrastructures and applications. As an example, most of the things in IoT are devices with low processing power and storage due to their limited energetic capabilities, a contrast to the several complex processing tasks that need to be performed. To overcome this limitation, such devices could play the role of simple data providers and send data to be processed and stored directly on the cloud, i.e., externally to the device itself. On the other side of the spectrum, bringing the things to the cloud stands for leveraging IoT-based capabilities and offering them as pay-per-use services on the cloud, leading to what has been sometimes referred to as Cloud of Things (CoT). This relies on the notion of **Everything as a Service** (XaaS or *aaS), a flexible, customizable model that envisions offering not only computing resources, but also anything that can be consumed as utility cloud services. An example of this approach is the concept of Sensing as a Service, a model to virtualize, share, and reuse sensing devices and their respective data to be ubiquitously consumed from the cloud. In this perspective, the cloud becomes an intermediate layer between the smart things and users/applications, thus hiding the inherent complexity of the former aiming to foster the development of the latter.

The rest of this paper is organized as follows. The background and basic concepts are presented in Section II. We describe the Robot as a Service Section III. We Explain Robot Cloud Architecture in Section IV. We present the Robot cloud Prototype in Section V and conclude this paper in Section VI.

II. BACKGROUND AND BASIC CONCEPTS

In this section we recall the basics of IoT and Cloud and overview the characteristics essential for their integration.

Internet of Things:

The next wave in the era of computing is predicted to be outside the realm of traditional desktop. In line with this observation, a novel paradigm called Internet of Things rapidly gained ground in the last few years. IoT refers to ‘‘a world-wide network of interconnected objects uniquely addressable, based on standard communication protocols.’’ whose point of convergence is the Internet. The basic idea behind it is the pervasive presence around people of things, able to measure, infer, understand, and even modify the environment. IoT is fueled by the recent advances of a variety of devices and communication technologies, but things included in IoT are not only complex devices such as mobile phones, but they also comprise everyday objects such as food, clothing, furniture, paper, landmarks, monuments, works of art, etc. These objects, acting as sensors or actuators, are able to interact with each other in order to reach a common goal. In the following we describe a few important aspects related to IoT. [**1, 2, 3**]

RFID: In IoT scenario, a key role is played by Radio-Frequency Identification (RFID) systems, composed of one or more readers and several tags. These technologies help in automatic identification of anything they are attached to, and allow objects to be assigned unique digital identities, to be integrated into a network, and to be associated with digital information and services. In a typical usage scenario, readers trigger the tag transmission by generating an appropriate signal, querying for possible presence of objects uniquely identified by tags. RFID tags are usually passive (they do not need on-board power supply), but there are also tags powered from batteries.

(Wireless) sensor networks: Another key component in IoT environments is represented by sensor networks. For example, they can cooperate with RFID systems to better track the status of things, getting information about position, movement, temperature, etc.Sensor networks are typically composed of a potentially high number of sensing nodes, communicating in a wireless multi-hop fashion. Special nodes (sinks) are usually employed to gather results. Wireless sensor networks (WSNs) may provide various useful data and are being utilized in several areas like healthcare, government and environmental services (natural disaster relief), defense (military target tracking and surveillance), hazardous environment exploration, seismic sensing, etc. However, sensor networks have to face many issues regarding their communications (short communication range, security and privacy, reliability, mobility, etc.) and resources (power considerations, storage capacity, processing capabilities, bandwidth availability, etc.). [**1**]

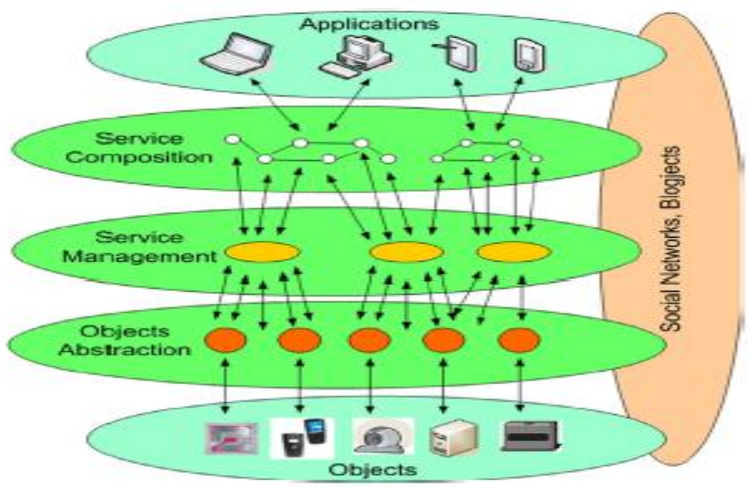


Fig 1. IoT paradigm: an overall view

Besides, WSN has its own resource and design constraints (that are application- and environment- specific) and that heavily depend on the size of the monitoring environment. The scientific community deeply addressed several issues related to sensor networks at different layers (e.g., energy efficiency, reliability, robustness, scalability, etc.).

Addressing: Thanks to wireless technologies such as RFID and Wi-Fi, IoT paradigm is transforming the Internet into a fully integrated Future Internet. While Internet evolution led to an unprecedented interconnection of people, current trend is leading to the interconnection of objects, to create a smart environment. In this context, the ability to uniquely identify things is critical for the success of IoT since this allows uniquely addressing a huge number of devices and controlling them through the Internet. Uniqueness, reliability, persistence, and scalability represent critical features related to the creation of a unique addressing schema. Unique identification issues may be addressed by IPv4 to an extent (usually a group of cohabiting sensor devices can be identified geographically, but not individually). IPv6, with its Internet Mobility attributes, can mitigate some of the device identification problems and is expected to play an important role in this field.

Middleware: Due to the heterogeneity of the participating objects, to their limited storage and processing capabilities and to the huge variety of applications involved, a key role is played by the middleware between the things and the application layer, whose main goal is the abstraction of the functionalities and communication capabilities of the devices. The middleware can be divided in a set of layers (see Fig. 1): Object Abstraction, Service Management, Service Composition, and Application.

Cloud Computing:

The essential aspects of Cloud computing have been reported in the definition provided by the National Institute of Standard and Technologies (NIST). ‘‘Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction’’.

Layered architecture and service models. The architecture of Cloud can be split into four layers: datacenter (hardware), infrastructure, platform, and application. Each of them can be seen as a service for the layer above and as a consumer for the layer below. In practice, Cloud services can be grouped in three main categories: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). SaaS refers to the provisioning of applications running on Cloud environments. Applications are typically accessible through a thin client or a web browser. PaaS refers to platform-layer resources (e.g., operating system support, software development frameworks, etc.). IaaS refers to providing processing, storage, and network resources, allowing the consumer to control the operating system, storage and applications even X as a Service (XaaS) where X can be any item. [**6, 7, 8**]

Zhang et.al has already presented a reusable and customizable cloud computing architecture called Cloud Computing Open Architecture (CCOA). It also depicts several principles for building the Cloud Computing systems and applications as well as for adopting the service-oriented architecture (SOA) that could make the CCOA more flexible, extensible and reusable. [**6**]

Mobile Cloud computing becomes a hot topic lately. Mobile Cloud and Robot Cloud share similarity to some extent since Robots typically interact with other services in the Cloud through wireless communication. But the difference between them is prominent. Mobile Cloud provides the rich computation and data resources in the Cloud platform to mobile users. It works by, for example, offloading the workloads submitted to mobile devices to run on the Cloud platform. In Robot Cloud, however, the objective is to integrate the robots into the Cloud platform.

Service-Oriented Architecture:

Service-Oriented Architecture (SOA) is not a revolutionary shift of paradigm. It evolves from the distributed object and component architecture. In an SOA environment, end users request an IT service (or an integrated collection of such services) at the desired function, quality, and capacity level, and receive it either at the time as it is requested or at a later time specified by the users. Service discovery, interoperability, and reliability are important functionalities that need to be supported in SOA.

The key idea behind SOA is that different companies offer their services in the form of modular and loosely coupled software components, which exchange data among the companies over HTTP (Hypertext Transfer Protocol) on the Internet with no human intervention. In other words, services can be replaced on the fly without the interruption of using the services. SOA has a number of outstanding characteristics, such as reusability, substitutability, extensibility, scalability, customizability and composability. [**9**]

Cloud-enabled Robotics:

Professor James Kuffner at Carnegie Mellon University has described the possibility of embracing the cloud in robots. CPU-intensive tasks in robots can be offloaded from smaller and less power-consuming onboard computers to remote cloud servers. A robot can improve its ability by integrating with cloud-based services. An example of this is the cloud-enabled robot with the object recognizing function such as Google Goggles. Imagine a robot finds an object that it never saw or used before, for instance a box of cornflakes. The robot can be programmed to send an image of the box to the cloud. The object recognition algorithm is run in the cloud and returns the robot the object's name, a 3-D model, and the instructions of how to pour it. Other functions of a cloud-enabled robot include navigating environments and operating tools.

Professor Kuffner aims to develop the robots as an effective terminal which is able to offload most of its complex computational and storage work to the cloud, rather than intensify the ability of the robots themselves. In the robotics level, robots are isolated, which means that there are little direct interactions between robots. This is because in this architecture a cloud-enabled robot depends on, therefore interacts with, the cloud rather than its peers.

However, different from professor J. Kuffner's intention to make robots "lighter, cheaper and smarter", Robot Cloud aims to shape the robots as a part of the cloud computing service which leads to the conception "Robot as a Service". [**9**] On one hand, in the Robot Cloud, service-oriented architecture does not have to be implemented over the Web. Thus, the standard interface can be simpler and faster. Services can also be migrated onto a local machine to reduce the communication cost. Besides, the Robot Cloud is not the simple extension of the cloud computing technologies or the robotics. It has the features of both cloud computing and robotics as well as its own distinctive features of the Robot Cloud itself. Especially, on the robotics level, robots in Robot Cloud can contact each other rather than act as isolated units that are only allowed to exchange data with the remote servers. As a result, the Robot Cloud will make the best use of most excellent existing scientific achievements in the realm of robotics.

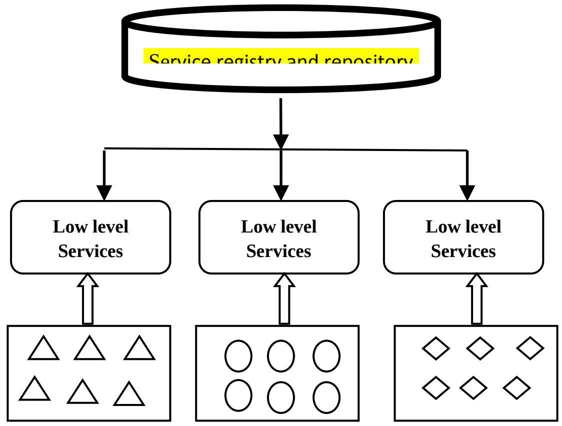
III. ROBOT AS A SERVICE

As SOA becomes more and more popular, this new architecture style has been applied to the development of robotics application. Chen applied the SOA concepts to develop re-composable embedded systems and robotics applications, and implemented a prototype of the system. Researchers encapsulate the functions of every part of robot and the robotics applications as well-defined services. Programmers are then able to assemble new robotics applications using these services. In this way, the whole systems are loosely coupled, more flexible to adjust. More importantly, new applications can be developed more efficiently just by composing every useful component and customizing them. [**5**]

The SOA-based robot architecture relies on extending cloud computing to the field of robotics RaaS (Robot as a Service in Cloud Computing). There are a large number of robot units in a RaaS system. All these robot units provide services to the consumers. A RaaS system also has complete functions of SOA, that is, as service provider, as a service broker, and as a service client. [**10, 11**]

A RaaS unit is a service broker: A client can look up the services and applications available in the unit's directory. A client can search and discover the applications and services deployed in the robot by browsing the directory. The services and applications can be organized in a hierarchy or classes to facilitate the discovery. [**9**]

Fig.3 illustrates the service layers and the hierarchy of service registry and discovery. There are three types of elementary services, which are also called low-level service, including "Basic Hardware Services", "Application Services" and "Common Services" (represented by different shapes at the low level services layer in Fig.3). These low-level can be composed to provide high level services (the second layer in Fig.3). All services, no matter high-level or low-level, are published through the service registry and repository.



Basic hardware services Application services Common services

Fig.3.Illustration of service layers registry and repository

Basic Hardware Services: This type of services plays a role of basic hardware resources. The following are some exemplar basic hardware services: a Core Processor Service runs as the core of the system, performs basic diagnosis and controls various services in order to finish specific tasks; a Sonar Sensor Service initializes the sonar sensor driver and calculates the distance of the sonar senor from an obstacle in front of it; a Touch Sensor Service generates an event when the touch sensor is touched or has encountered an obstacle; a Motor Driver Service keeps reading a managed command pool and retrieves the command from the command list and executes it; a Compass Sensor Service keeps reading incoming data from the corresponding port, calculates the compass value, and turns around according to the compass value; a Video Camera Service pops up a window displaying a real-time video captured by the camera mounted on the robot.
Application Services: Application services registered in the service broker are mainly task-related services, such as Patrol Service (driving the robot to perform the patrolling and object detection task autonomously), Intruder Detection Service (helping the robot detect the presence of intruders), Bomb Detection Service (assisting the robot to detect potential bombs in a dangerous environment), and so on.
Common Services: Common Services represent several classic algorithms and commonly used functions. For example, a maze navigation algorithm is required by most mobile robots, and a human face recognition algorithm is also used by various robotics applications which need to recognize different people from their faces. Since the service broker provides a method for robotics application developers to retrieve existing usable services, developers are able to reuse the appropriate components and combine them according to the requirements, which greatly facilitate the application development.

IV. ROBOT CLOUD ARCHITECTURE

Target Business Model

A good understanding of the target business model greatly helps us design the architecture of the Robot Cloud. Generally, there are four parties in the target business model of Robot Cloud, which are Robot Cloud Service Provider, Cloud Computing /Robotics Provider, Value-added Service Provider, and also the Consumer. The relationship among all the roles can be illustrated by Fig. 4.

The Robot Cloud is owned by the Robot Cloud Service Provider, who is responsible for the management, government, execution and maintenance of the Robot Cloud. Acting as a one stop shop in front of the consumers, the Robot Cloud Service Provider receives consumer's requests and provides the Robot-Cloud services to the consumers. Cloud Computing Provider and Robotics Provider act as the Infrastructure Provider in the Robot Cloud, who provides cloud services primarily or various types of robotics. The Value-added Service Provider offers services that existing cloud does not own.

System Architecture Overview

Designing a novel architecture is a very complex project, which needs to consider many factors including reliability, scalability, modularity, interoperability, interface, QoS, etc. [**4**]

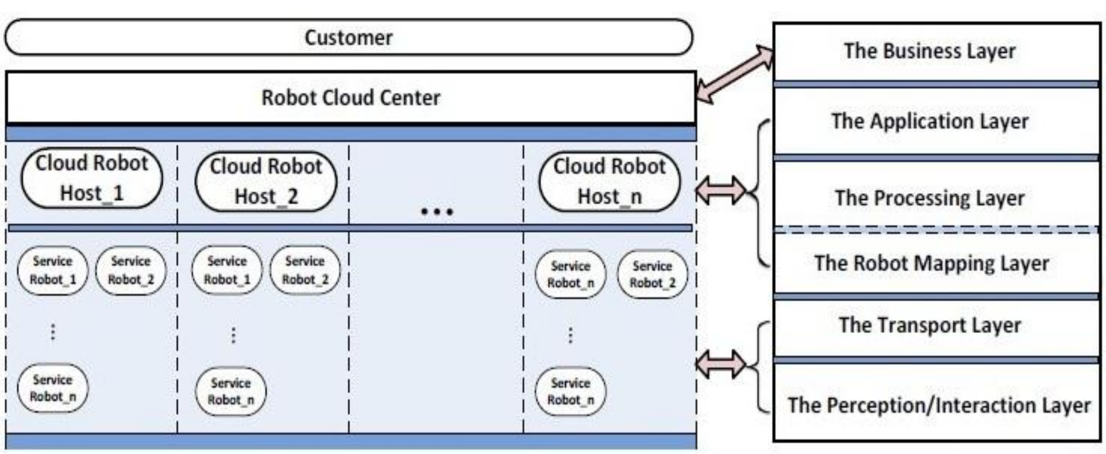


Fig 4. System Architecture Overview

As a whole, the architecture of Robot Cloud includes three parties, which are the Robot Cloud Center, Cloud Robot Host and Service Robot, from top to bottom. Robot Cloud Center is the core of the Robot Cloud which is in charge of performing all the computing or storage intensive tasks. A Robot Cloud Center contains a number of Cloud Robot Hosts which are deployed in different physical environments that could be different geographical places in a city. Robot Hosts meet the requirement of interacting with the physical world in Robot Cloud. A Robot Host may contain several Service Robots, which form the Perception/Interaction Layer of the Robot Cloud.

From another perspective, the Robot Cloud is also a six-layer architecture system consisting of the Perception/Interaction Layer, the Transport Layer, the Mapping Layer, the Processing Layer, the Application Layer, and the Business Layer.

The Perception/Interaction Layer is like the facial skin and the five senses of the Robot Cloud, which is designed to perceive the physical properties of the objects (such as the location, surrounding environments, etc.) by various sensors (such as sonar sensors, GPS, 2-D barcode, etc.), and to convert the information to digital signals so that it can be transmitted over network more conveniently. Moreover, the Perception/Interaction Layer is responsible for the interaction between human and robots, through interactive devices.

The Transport Layer is responsible for transmitting data received from the Perception/Interaction Layer to the upper layer mostly through wireless networks (such as WiFi, 3G networks, etc.).

Both the Mapping Layer and the Processing Layer rely on computing and storing abilities. The Former focuses on the tasks such as robots scheduling and brings the great convenience to robot management. The later processes large quantities of tasks and the huge amount of information that the Robot Cloud carries through the cloud computing technologies.

Based on the Processing Layer, we build the Application Layer which is formed by diverse application services (such as Robots' Authentication and Access, Location service, etc.). As a result, the Robot Cloud can provide different clients with different applications.

Finally, the Business Layer acts like the manager of the Robot Cloud, whose functions include managing the applications, relevant business models and other businesses. The business Layer manages not only the release of various applications and the charging for using them, but also the research on business models and profit models, which benefit the long-term development of the Robot Cloud. The reason why we build the Robot Cloud based on the service-oriented architecture is partly because of the requirement of the business model.

Robot Cloud Center

Robot Cloud Center Architecture

Robot Cloud Center is the core of the Robot Cloud which is in charge of performing all the computing or storage intensive tasks. It can be a data center built with a large number of devices in a specified physical location. Cloud computing is the primary technology of the Robot Cloud Center.

Employing the cloud computing technology in the robotics is still challenging. Three objectives help address the challenges of defining an open Robot Cloud Center architecture.

1) Articulating a reusable way of creating scalable and configurable provisioning platform for Cloud Computing.

2) Proposing a set of common and shared services to build the Robot Cloud services (or other Cloud offerings) for Cloud customers in a unified approach.

3) Maximizing the potential business value of a Robot Cloud based on an extensible infrastructure and management system. This will attribute to the value added services of business cloud, through combining the power of SOA and cloud computing. [**4, 6, 10**]

Thus, we propose a reference model of the Robot Cloud Center, which is shown in Fig. 4.1.

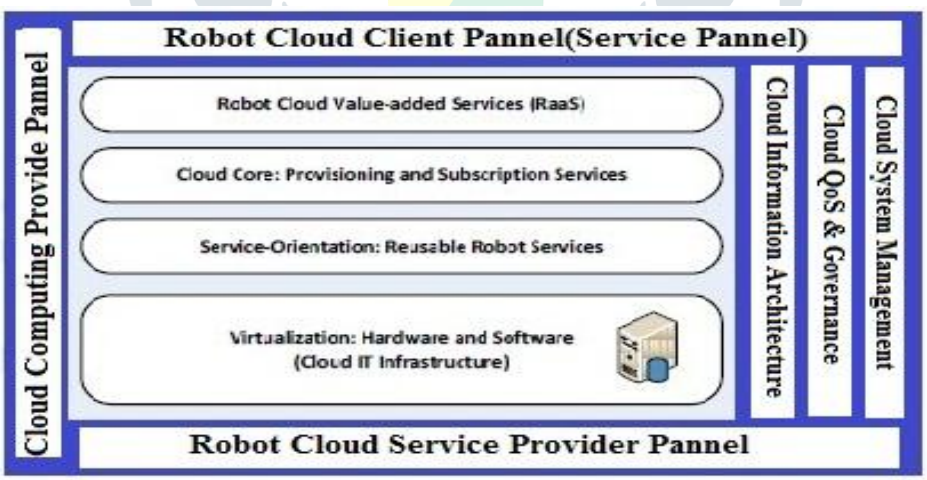


Fig.4.1.The architecture of a Robot Cloud Center

Here, we have to clarify three points:

First, the Robot Cloud Client Panel is the interface through which the customers access the Robot Cloud and request the services that satisfy their needs. In other words, the Robot Cloud Client Panel is the interaction platform between the service provider and the customers of a Robot Cloud. Technically, the Robot Cloud Client Panel can be a front-end for web access. Second, "since most Cloud vendors do not work alone anymore, they need to collaborate with their partners", it is helpful to design a Cloud Computing Provider Panel to integrate the management of the Robot Cloud. Third, even though X as a Service (XaaS) is not a strange concept to IT professionals, the new concept of Robot as a Service (RaaS).

Robot Cloud Panels:
People (i.e. customers, Robot Cloud provider, and also cloud computing provider) access the Robot Cloud through three separate Robot Cloud Panels. However, according to the life cycle reference of SOA, the functions of all these Panels can be classified into four categories, i.e., modeling, assembling, deploying, and managing and analyzing the robotics applications.
1) Modeling: It enables the platform to gather and analyze user requirements. The developers of robotics applications will design stimulate and optimize them according to user requirements recorded by the system.
2) Assembling: It is used to compose new robotics applications using existing services published in the service broker. MRDS (Microsoft Robotics Developer Studio) is a windows-based environment for developers to easily create robotics applications. Because MRDS with various useful toolkits provides powerful functionalities, we have ported MRDS and its graphic language, VPL (Visual Programming Language) into our model and assembled the development phases of robotics applications.
3) Deploying: If robotics applications are deployed and invoked on a “Robot Cloud Panel”, applications can be replaced and updated without stopping the robots. However, it requires frequent communication between the “Robot Cloud Panel” and robots. As the result, it is difficult to guarantee the real-time response. In our work, we implement a service auto-installer to synchronize newly developed services or applications and target robots through wireless network.
4)Managing & Analyzing: This function is composed of four important sub-components: scheduler(scheduling all requests received by the robots), robot manager (adding new robots or substituting existing ones, managing the robot mapping, etc.), robot monitor (monitoring the robot status, service running status and analyzing the logs), and service broker manager (the CRUD operation of the service broker).

Service Robot
Service Robot Architecture
Generally speaking, Service Robots in the Robot Cloud are robotics provided by Robotics Providers, but are built with specially designed architecture in order to integrate with the cloud computing resources. Indeed, Service Robots play the most fundamental role in the Robot Cloud system because they are the entities which provide services for the end users. Service Robots need to communicate with other parts of the system (i.e. the Robot Host), and they are also required to interact with each other.

As we have discussed, Service Robots form the Perception / Interaction Layer of the Robot Cloud. Service Robots are able to perceive information in the surrounding world and also provide different services directly to customers.

The physical architecture of a Service Robot is shown in Fig.4.2. A typical service robot’s components should include (but not limited to) a position module, a mechanical module, a communication module, the computation module and battery. To improve the service quality of the robotics, Providers may add other modules onto the robot including Alert system, QoS module and Safety mechanism. Note that the controlling function is not implemented in a single module in the architecture. The controlling function consists of two parts. One part performs certain physical actions. This function is implemented in the mechanical module, which needs the assistance of other modules, such as position modules. The other verifies who is authorized to issue the instructions. This function is implemented in the safety module.

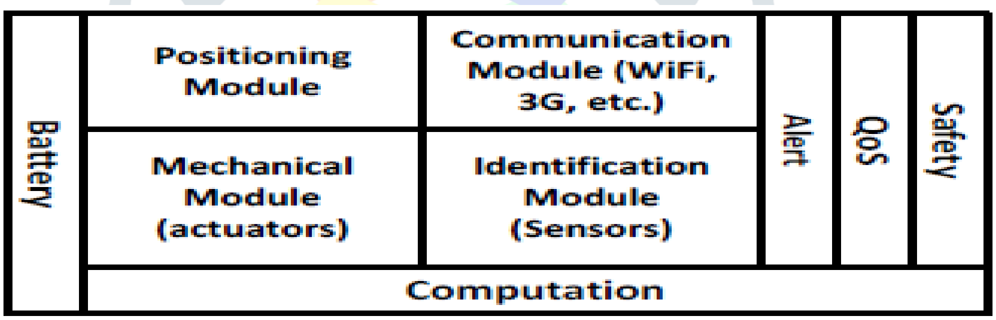


Fig 4.2. Service robot architecture

V .THE CLOUD ROBOT PROTOTYPE

A robot is a good combination of the digital cyber world and the real world. Most of the robots controlling programs are written in object oriented languages such as Java and C++.The libraries that most robot manufacturer used to program the robot actions are similar. With these libraries, it is easy for the users to develop their own controlling algorithms and make the robots perform desired actions.

We have implemented a prototype of the robot cloud based on the infrastructure and components described in previous sections. An Application Programming Interface (API) is implemented using the object oriented programming languages (detailed in the subsections below) in the PaaS layer. The API describes a set of robot actions provided by the Robot Cloud in the behavior level. [9, 10]

Accessing robots from the cloud programs
Although programming a robot is easy nowadays, the interoperability is still poor compared to the highly developed software industry. With the development of SOA (Service Oriented Architecture) and related technologies, the interoperability of software, especially those based on Internet, have advanced to a very high level. For example, a program can now access another program’s data or functions through some common programs called middleware. The cloud computing technology has one of the many ways to implement SOA software, which draws more and more attention in the industry Cloud service providers have also made a lot of efforts to improve the interoperability between the cloud computing platform and the software running on top. A typical solution is to build up a library which makes it easy for the programmers to access the platform’s features. Software on the platform can also have access to the public data (e.g. user information, contact, schedule etc).The advantage of the Software industry in interoperability can be introduced to the robots, if we can merge the robots into the modern software architecture such as cloud computing. The robots can be regarded as another resource in the cloud computing infrastructure, just like computing capacity and storage.

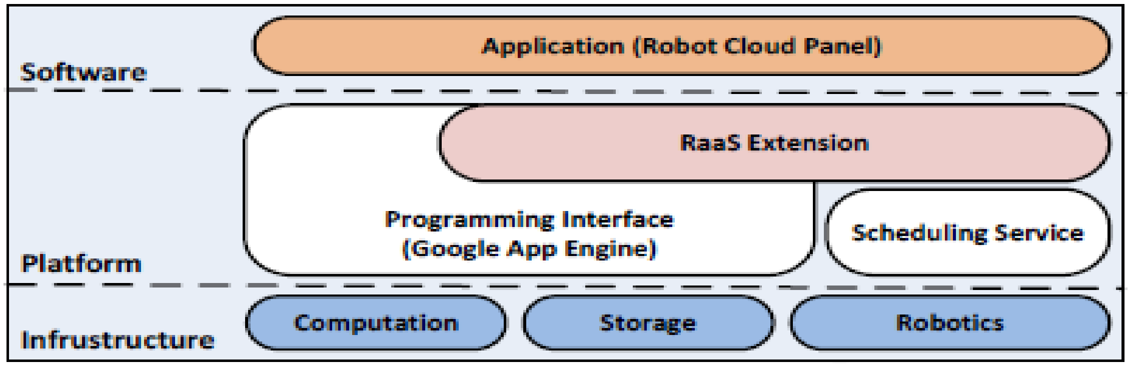


Fig 5. System architecture of the prototype system

The extension takes the advantage of the Google App Engine (GAE), which stores and analyzes the data through the interface provided by the Cloud platform. The extension provides the developers the extra classes that can access the robots.

Controlling robots from the cloud server

All the data generated by or related to the robots are stored as persistent objects in the Google App Engine Platform. Robot instances can share these data through the scheduling service to cooperate and provide more excellent services.

The Scheduling service is a standard SOA programming interface, which is implemented in Java and hosted by the Google App Engine. The service contains two main functions: Upload the data submitted by the robots to the cloud storage and accept the requests committed by the applications. The Scheduling service is tightly coupled with the Google App Engine. When new data arrive from the working robots, it will be transformed immediately into the instances of the corresponding data type and stored persistently through the JDO (Java Data Object) interface provided by the Google App Engine. The data in the cloud storage is clearly classified so that both applications and working robots can access the data by certain search requests. On the other hand, when users submit new requests to the system through the application, they will be placed in the command queue located in the cloud storage. The scheduling service then rearranges the commands and sends them to certain robots for execution. Since the robot resource is also limited, the rearranging procedure is very important in optimizing the allocation of robot resources, just as how the operation system arranges the processes to make full use of the computing and storage capacity.

The controlling system of each robot is implemented in an Intel Atom process based x86 embedded system. Since the robot system shares the same architecture with the widely used PC platform, the system can get full support of the PC software and hardware. In this prototype, a Microsoft Windows XP is installed in each robot as the operation system and a WIFI adapter is installed in the system to provide IEEE 802.11g wireless Internet connection to the system. Some motion control units (PhidgetMotorControl HC, Phidget, and Calgary, Canada) are connected to the universal serial bus (USB) to control the movement of the robots. A local controlling service is also implemented by the C# programming language. The API provided by the manufacturer of the USB motion controller makes it possible for the local controlling service to control the robot programmatically. The service obtains the latest commands by sending the HTTP requests periodically to the scheduling service and receiving their commands in the XML format as the response, in a similar way as what most SOA applications do.

Application developers can control the robot simply by invoking the methods provided by the RaaS Application Programming Interface in the level of behavior instead of in the low level of motion control. This makes it much easier for the robots to interact with other robots or even with people.

CONCLUSION AND FUTURE WORK

We have developed a way of building a Robot Cloud in order to combine the cyber (virtual) world and the physical world. The Robot Cloud has the great potential to serve the diverse and large amount of location-based on-site demands, which cannot be handled by the existing cloud systems. The prototype implemented in this work show that 1) the virtualization of robot resources is feasible, 2) the virtualization can improve the profit of the robot center, and 3) one large robot center is better than multiple independent small centers.

There are still some open issues for robot cloud research. First, in order to reduce the total cost of robot cloud, we need a clever strategy to select the position of the robot center. A promising solution could take into account the distribution of requests and the traveling cost, and maintaining cost of different positions to build the optimization model. Finally, the implementation and deployment of robot services may need to be tailored or adapted in different application domains in order to achieve the optimal performance. It is critical to investigate the conditions and factors that have biggest impact on the performance of Robot Cloud. Unmanned aerial vehicles, which have been employed to provide different services at specific locations for people, can be regarded as a special kind of robot.

REFERENCES

[1] Mohammad Aazam et al, "Cloud of Things: Integrating Internet of Things and Cloud Computing and the Issues Involved," in Proceedings of 2014 11th International Bhurban Conference on Applied Sciences & Technology (IBCAST), Islamabad, 2014.

[2] Manuel Díaz et al, "State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing," Journal of Network and Computer Applications, pp. 99-117, 25/09/2015.

[3] Rao, B. P., Saluia, P., Sharma, N., Mittal, A., Sharma, S. V., 2012. Cloud computing for Internet of Things & sensing based applications. In: Sensing technology (ICST), 2012 Sixth International Conference on. IEEE, pp.374–380.

[4] Jiehan Zhou et al, Cloud Things: a Common Architecture for Integrating the Internet of Things with Cloud Computing», in Huazhong University of Science and Technology, Wuhan, 2013.

[5] R. Arumugam, V. Enti, L. Bingbing et al, DAvinCi: A Cloud Computing Framework for Service Robots, In proceedings of 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, May, pp. 3084-3089.

[6] L. Zhang and Q. Zhou: CCOA: Cloud Computing Open Architecture. In Proceedings of the 2009 IEEE International Conference on Web Services, Los Angeles, CA, July 06-July 10, pp. 607-616.

[7] D.Huang,“Mobile cloud computing,” IEEE COMSOCMultimediaCommunicationsTechnicalCommittee(MMTC) E-Letter, vol. 6, no. 10, pp. 27–31, 2011.

[8] Rao, B.P.Saluia, P.Sharma, N.Mittal, A.Sharma, S.V.2012. Cloud computing for Internet of Things &sensing based applications. In: Sensing technology (ICST), 2012 Sixth International Conference on. IEEE, pp.374–380.

[9] Chen, Yinong, Zhihui Du, and Marcos García-Acosta. "Robot as a service in cloud computing."In Service Oriented System Engineering (SOSE), 2010 Fifth IEEE International Symposium on,pp. 151-158. IEEE, 2010.

[10] Vlad M.Trifa, ChristopherM.Cianci and Dominique Guinard: Dynamic control of a robotic swarm using a serviced-oriented architecture. In the proceedings of the 13th International Symposium on Artificial Life and Robotics (AROB 2008), Beppu, Japan, January 31 February 2,2008.

[11] Yinong Chen, S.Abhyankar, L.Xu, W.T.Tsai, and Marcos Garcia-Acosta: Developing a security robot in service-oriented architecture. In the proceedings of 12th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS’ 08), Kunming, China. Oct. 2008, pp.106-111.

[12] W.Hao, T.Gao, I.Yen, Y.Chen, R.Paul: An Infrastructure for Web Services Migration for Real-Time Applications, 2nd IEEE International Symposium on Service-Oriented System Engineering, Shanghai, 2006, pp.41- 48.