Proceeding of the 2015 IEEE
International Conference on Information and Automation
Lijiang, China, August 2015

CRALA: Towards A Domain Specific Language of Architecture-Centric Cloud Robotics

Huaxi (Yulin) Zhang
MIS, INSSET
Universit de Picardie Jules Verne
33, rue Saint Leu, 80039 Amiens
Email: yulin.zhang@u-picardie.fr

Lei Zhang
State Key Laboratory of Synthetical
Automation for Process Industries
Northeastern University, Shenyang, China
Email: zl.org.cn@gmail.com

Zheng Fang
State Key Laboratory of Synthetical
Automation for Process Industries
Northeastern University, Shenyang, China
Email: fangzheng@mail.neu.edu.cn

Harold Trannois
MIS, INSSET
Universit de Picardie Jules Verne
33, rue Saint Leu, 80039 Amiens
Email: harold.trannois@u-picardie.fr

Marianne Huchard
LIRMM, UMR 5506
CNRS et Universit Montpellier 2
161 rue Ada, 34392 Montpellier, France
Email: huchard@lirmm.fr

René Zapata
LIRMM, UMR 5506
CNRS et Universit Montpellier 2
161 rue Ada, 34392 Montpellier, France
Email: zapata@lirmm.fr

***Abstract*—Cloud robotic system is a mono- or multi- robot system that profits one or more services of Cloud Computing. In a few short years, Cloud robotics as a newly emerged field has already received much research and industrial attention. The use of the Cloud for robotics and automation brings some potential benefits largely ameliorating the performance of robotic systems. However, there are also some challenges. First of all , from the viewpoint of architecture, how to model and describe the architectures of Cloud robotic systems? How to deploy these architectures in Clouds? Merely a language could explicitly describe or model the architecture of Cloud robotic systems. In this paper, we present an architecture approach to support design and implementation of Cloud robotic system.**

I. INTRODUCTION

In a few short years, Cloud robotics as a newly emerged field has already received much research and industrial attention.

A brief history Cloud Robotics was firsly introduced by James Kuffner [1]. This broader term was quickly be been accepted and adopted by many researchers including the organizers of this Special Issue of the IEEE Transactions on Automation Science and Engineering [2]. In 2009, the RoboEarth project was announced. It envisioned a World Wide Web for robots: a giant network and database repository where robots can share information and learn from each other about their behavior and environment [3]. Under a major European Union grant, the RoboEarch research team developed a series of system architectures for service robotics [4], [5], developing Cloud networking [6], [7], and computing resources [8] to generate 3D models of environments, speech recognition, and face recognition [9].

In August 2014, a US project "RoboBrain" initiated by Ashutosh Saxena announced his goal to build a RoboBrain: a large-scale computational system that learns from publicly available Internet resources, computer simulations, and reallife robot trials [10], [11].

Our vision. Cloud robotics is often broadly defined as: "*Any robot or automation system that relies on either data or code from a network to support its operation, i.e., where not all sensing, computation, and memory is integrated into a single standalone system [2].*" From the rigorous definition of Cloud Robotics, it is not tele or remote control of robots, or networked robotics. Networked robotics can be considered as an evolutionary step towards Cloud robotics [6]. The objective of Cloud Robotics is to overcome the limitations of networked robotics with elastic resources offered by a ubiquitous Cloud infrastructure. Thus, we define Cloud Robotics as : "*Cloud robotic system is a mono or multi robot system that profits one or more services of Cloud Computing.*"

The use of Cloud computing for robotics and automation brings some potential benefits largely ameliorating the performance of robotic systems. Due to the limited capacities of on-board processing, storage and battery capacities, robotic devices are constrained to numerous limitations. It not only solves the problems of robotic system problem, such as onboard computation and storage limitation, asynchronization communition, compability problem of multi robot systems [8], but also make possibility of different directions or enhance their performance, such as remote brain, big data and shared knwoledge-base, collective learning and intelligent behavior[12]. Furthermore, Qureshi and Koubaa [12] has summarized five traits of performance enhancement by using Cloud robotics: 1) Robot brain, 2) Big Data, 3) Collective Learning, 4) Intelligence and Behavior, and 5) Cloud architecture.

Among these the potential directions and the research challenges of the field, we focus on architecture-centric Cloud robotics. How to construct the architectures of Cloud robotic systems? How to model and describe the architectures of Cloud robotic systems? How to reuse existing components or services in system's architectures? How to deploy these architectures in Clouds?

Lei ZHANG to whom all correspondence should be addressed.

This work was supported by the National Natural Science Foundation of China under Grant No. 61300020, the Scientific Research Funds for Introduced Talents of Northeastern University under Grant No. 28720524.

In this paper, the main contributions are following:

- an architecture-centric design process for Cloud robotic systems,
- a domain specific powerful language for architecture-centric Cloud robotic systems called CRALA, whic not only covers the description of robots, components/web services, but also Cloud modelisation (IaaS, PaaS levels).

The rest of the paper is organized as follows: We begin with an introduction of related concepts, background and related works of Architecture-centric Cloud robotics. We then present an overview of the architecture-centric design process for Cloud robotic systems. We then describe the metamodel of CRALA. Afterwards, we illustrate an example by using CRALA to model a Cloud robotic system. We finish with a discussion of architectures for Cloud robotics and future works.

II. Background and Related works

A. Related concepts

Architecture-centric Cloud robotics is a methodology of developing robotics systems on Clouds using architecture-centric development techniques.

Cloud computing Cloud computing is defined by the National Institute of Standards and Technology (NIST) as: "Cloud computing as 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 [13]".

Clouds offer services that can be grouped into three categories: software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS) [14].

1) Infrastructure as a Service: IaaS refers to on-demand provisioning of infrastructural resources, usually in terms of VMs. The cloud owner who offers IaaS is called an IaaS provider.
2) Platform as a Service: PaaS refers to providing platform layer resources, including operating system support and software development frameworks.
3) Software as a Service: SaaS refers to providing on-demand applications over the Internet.

System/Software architectures. Traditionally, software architecture is a collection of models that capture a software systems principal design decisions in the form of components (foci of system computation and data management), connectors (foci of component interaction), and configurations (specific arrangements of components and connectors intended to solve specific problems) [15]. Generally speaking, a software system architecture [16] gathers design decisions on the system. As the development of computer science, a system is more and more complex than before with the integration of "Internet of Things", "Cloud Computing" and "Robotics" etc.

Architectures Modeling language. Architecture models are often expressed using ADLs (Architecture Description Language) which, in most cases, provides information on the structure of the software system listing the components/services and connectors the system is composed of. A system architecture could cover different abstraction levels, such as specification, configuration and assembly [17] and from different viewpoints [18]. A good architecture model from different views not only facilitate the development of robotic systems bu also could be used to validate the non-functional properties of systems, such as security etc.

B. Related Works

The description of Cloud robotics systems should cover robot description, web services/component description and cloud robotic system global architecture description.

Robot description. Robot description languages provide models of a robot and then design and implement software components that work on the model components rather then the particular robot instance.

The representative example of robot description language is the Unified Robot Description Format (URDF) [19], which can be used to specify the kinematics and dynamics, the visual representation and the collision model of a robot. However, URDF is not designed for specifying robot components such as sensors, actuators, and control programs.

COLLADA [20] is an XML Schema designed for describing 3D objects including their kinematics. It mainly focuses on modeling information about scenes, geometry, physics, animations, and effects. But similar to URDF, it lacks elements for describing sensors, actuators and software. SRDL [21] focus on modelling robot components, i.e. sensors, actuators and control programs, especially via capabilities to actions.

Many works try to develop an OWL ontology to describe robots, such as [22], [23] in specific domains or [24], [25] focusing on sensor ontology.

Web service description. Web service description in robotics often serves to match the capabilities with robot components, such as PHOSPHORUS [26], Larks [27], OWL-S [28] and SRDL[21] In general, the term capability matchmaking refers to the process of matching an advertisement of a capability with a request.

Cloud Robotic description. All above works cover a part description of Cloud robotic systems, referring to robot description or web service description. However, it misses a language which fully cover the different descriptions of Cloud robotic architecture, including the description of Clouds, robots and components/web services.

III. Architecture Modeling for Cloud robotics

The architecture design for Cloud robotic system is different with traditional software design, as it concerns two special aspects: Cloud-based systems and robotic systems. We identify the architecture-centric development process for Cloud robotic systems into three main phases, as shown in Figure 1.

1) *Specification design.* Architect or robotics engineers should choose a robotics architecture pattern for system according the models of robots (hardware) and its functional tasks (objective), for example, a pioneer robot with a task of path planning.

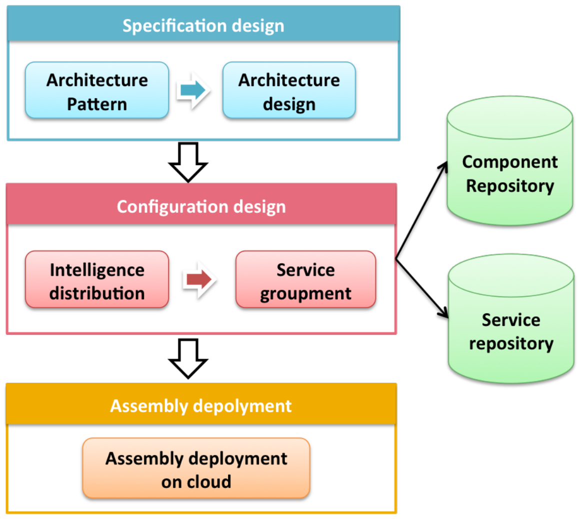

Fig. 1. Cloud robotic system architecture design

2) *Configuration design.*
   - First of all, architect should consider how to distribute intelligence of robots. That means, which components should be placed on robot itself and which services should be placed on Cloud. How to choose the appropriate component or service from component or service repository [1]? This design decision should consider different factors, including robot capacities, system non-functional properties such as real-time, security etc.
   - Secondly, architect should choose operating system for their services, as in robotics domain, there exists some operating systems which are widely used, such as ROS[29]. How to distribute these services in different virtual machines.
3) *Assembly deployment.* At last, how to deploy this architecture model in Clouds, automatically or not? How to reflect and supervise a runtime model to prevent VMs failure etc?

During the process, five factors affect architecture decisions *robot models, tasks, intelligence, non-functional properties*, and *Clouds*, as shown in Figure 2.

- *Robot model* describes the hardware model of robots consisting of sensors and actuators etc.
- *Task* is objective realized by robots.
- *Intelligence distribution* defines how to distribute the intelligence to robots and Clouds.
- *Non-functional properties* are non-functional requirements required to be exposed by Cloud robotic systems, such as security, realtime, safety etc.

[1]As the language CRALA is under a Cloud robotics project, one of which objectives is robot brain based on existing robot components and services. Thus here, we accentuate the search and selection from component or service repository.

- *Clouds* represent Cloud infrastructures (IaaS) used to deploy robotics services. Clouds can be mono-cloud or multi-Cloud.

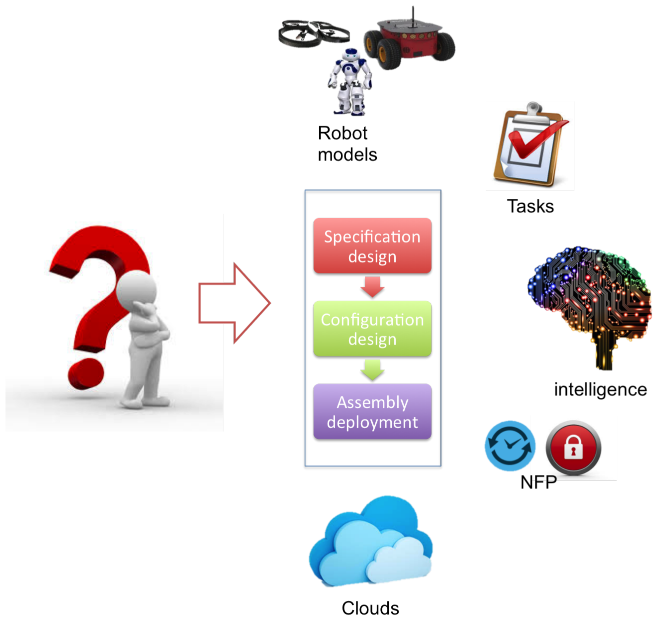

Fig. 2. Cloud robotic system architecture design

IV. CRALA: A Domain Specific Language

CRALA is a domain specific language for architecture-centric Cloud robotics, and it is also an architecture description language. CRALA models architectures at three separate abstraction levels, each of which is designed in different development phase as shown in Fig. 1. For now, the first version of CRALA presented in this paper mainly focuses on modeling essential elements of architectures and their basic properties, as the design concept of CRALA is to auto-develop and enrich the language by experimentation and real use cases. The three levels are as following:

1) *Specification* defines the abstract architecture specification. It defines which functionality should be supplied by robotic systems. All the constituents of this architectural models are abstract and without any consideration of Cloud etc.
2) *Configuration* defines the sets of component or service implementations (classes) by searching and selecting from the component/service repository and defines how to group services and components in different virtual or physical machines by consideration of system requirements.
3) *Assembly* depicts how configuration is deployed on Clouds. This architecture model exactly depicts the current state of the running Cloud robotic system.

A. Architecture Specification

Architecture specification is composed by *component roles* and *connections*. Component roles describe the roles that components should play in the system. A component role lists the minimum list of interfaces (both required and provided)

TABLE I. DECISION DECISIONS MADE IN EACH ARCHITECTURE ABSTRACTION MODEL

Architecture	Defined Aspects
Architecture specification	1) Functionalities of the system, 2) system non-functional properties
Architecture configuration	1) Component/service selection (for reuse) or implementation (for from scratch), 2) Component/service groupment, and 3) Operating system selection
System assembly	1) Cloud deployment, 2) Running state

the component/service (will be selected or implemented in configuration level) should expose. One side, as they define the requirements of the architect (its ideal view) to guide the search for corresponding concrete components (or service) in component (or service) repository, component roles are abstract and partial component (or service) representations. Another side, they can be used as the design specification for implementing new components or services from scratch. The metamodel of specification is illustrated in Fig. 3.

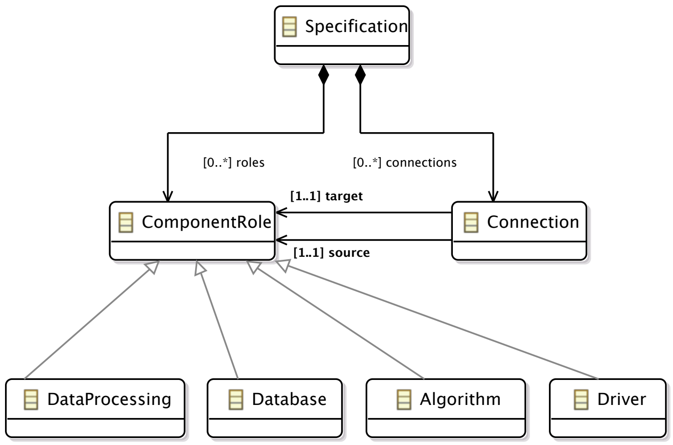

Fig. 3. Architecture specification Metamodel

We could find that here we principally specify four sub-types of component roles: Database, Data Processing, Algorithm and Driver, defined as following.

- Database: specifies databases that should included in the system.
- Data processing: specifies the components used to treat and filtre data.
- Algorithm: robotic algorithm.
- Driver: hardware diver for robots.

B. Architecture Configuration

Architecture configurations are the second level of system architecture descriptions. They result from the search and selection of real component classes (or web services) in a component (or service) repository. The mdetamodel is shown in Fig. 4.

In configuration description, firstly component role will be implemented by software component or web service.

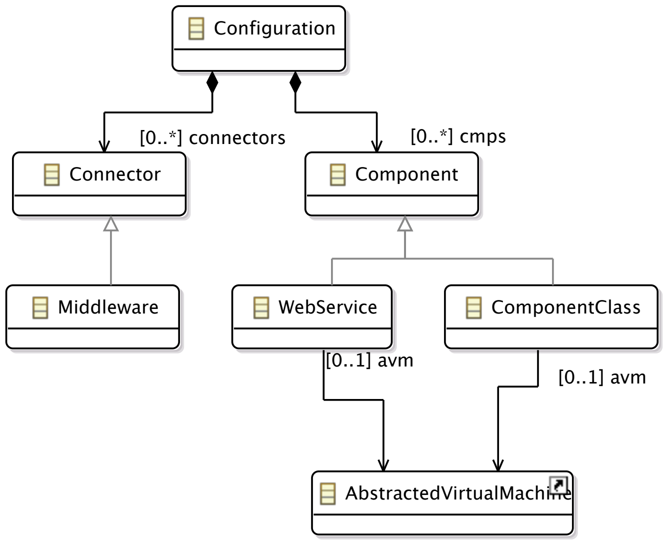

Fig. 4. Architecture Configuration Metamodel

- Software component: A software component often can be characterized as: attributes, component interface, behaviors and properties.
- Web service: A more formal and extended definition is the one offered by the W3C Web Services working group[30]:A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

Secondly, components and services will be placed in robots or virtual machines. In the first version of CRALA, we ignore the possibility that besides virtual machines, they could also be placed in physical machines directly.

C. System assembly

System assemblies are the third level of system architecture descriptions. They result from the instantiation of the component classes and the deployment of the web services from a configuration. They provide a description of runtime software systems including the Cloud deployment information.

The metamodel of Cloud is illustrated in Fig. 5. Each component or service are depicted clearly which VMs they are deployed and each VM is depicted with which physical machine of which Cloud it is deployed.

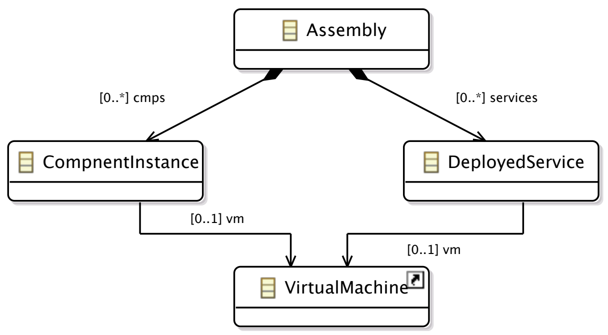

Fig. 5. Assembly Metamodel

D. Cloud Metamodel

In CRALA, it has a sub-metamodel of Cloud which enable to model architecture of Cloud from IaaS and PaaS levels as presented in Section II-A. The metamodel of Cloud is illustrated in Fig. 6.

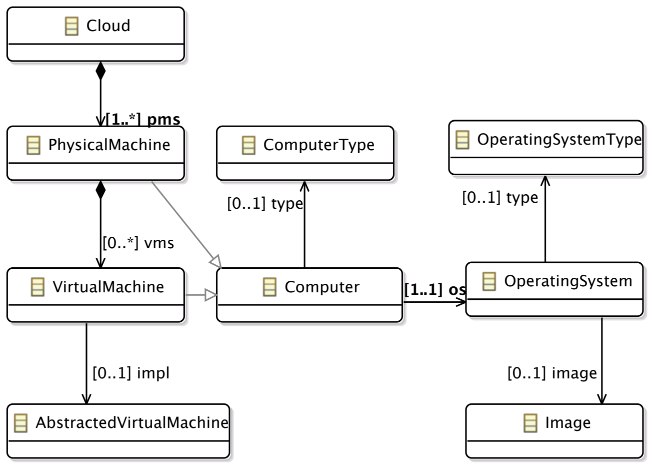

Fig. 6. Cloud IaaS Metamodel

E. Robot Model

In our CRALA, we concentrate on the modelling of component/service and cloud part, thus the hardware of robots are not our important contribution.

Furthermore, great efforts for providing a standard for sensor specifications are undertaken by the W3C Semantic Sensor Network Incubator Group [31]. As sensors are an important aspect within robotics, we fristly base on our previous work [32] and then we use directly works [24], [25], [21] and the Semantic Sensor Network Incubator Group.

V. Illustrating Example

In this section, we use an illustrating example to explain the main concepts in CRALA. We first present an example of components in three levels, as shown in Fig. 8. We use localization algorithm component as an example. It shows the

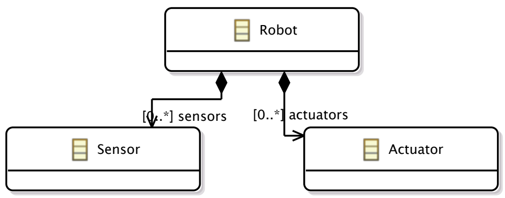

Fig. 7. Robot Metamodel

relationship of different component forms in three levels: component roles (specification), component/service (configuration) and component or service instance (assembly).

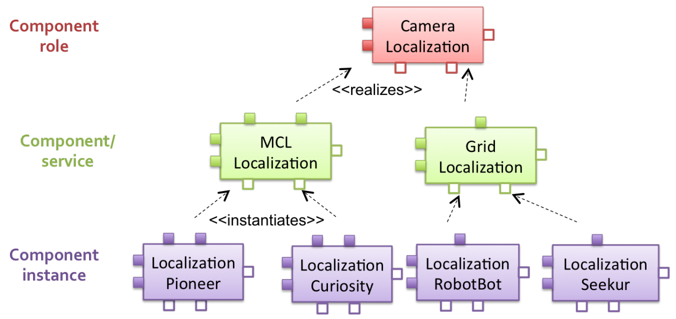

Fig. 8. The Localization component role, some subtypes, some possible concrete realizations and some of their instantiations

Secondly, we illustrate an example of a part architecture of a Cloud robotic system, as shown in Fig. 9. We can find that in this figure, the architecture description is the third level: assembly. The and differences with other languages are as follows:

1) Cloud IaaS model is presented,
2) Cloud PaaS model is described,
3) It explicitly defines component and service as two types of component.

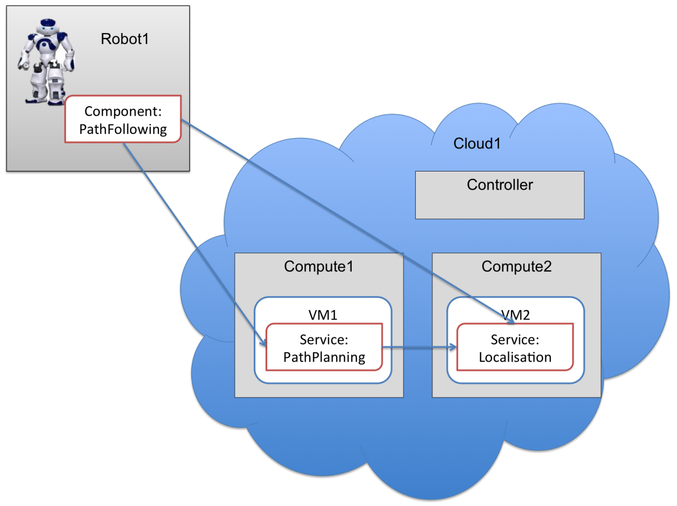

Fig. 9. Cloud robotic architecture example

VI. Conclusion and Future Works

In this paper we investigate architecture design process for Cloud robotic systems and propose a domain-specific architecture description language for architecture-centric Cloud robotics. We present CRALA for describing Cloud robotic architectures, and show that linking architecture descriptions with Cloud deployment aspect allows to explicitly master and control Cloud robotic systems. The proposed language is implemented by EMF and we use a use case to illustrate CRALA.

In future work, we aim to extend CRALA in several ways. We will develop mechanisms to support the automatically developing process of architecture-centric Cloud robotic systems. First of all, how to search the correspondent and appropriate components or services in repository to construct architecture configuration automatically. Secondly, how to deploy the configuration on Cloud automatically. Then how to reorganize the system on Clouds when service failure. Our overall goal is to construct an intelligent development environment to construct Cloud robotic systems.

References

[1] J. J. Kuffner, "Cloud-enabled robots," in *IEEE-RAS International Conference on Humanoid Robotics, Nashville, TN*, 2010.

[2] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," 2015.

[3] O. Zweigle, R. van de Molengraft, R. d'Andrea, and K. Häussermann, "Roboearth: connecting robots worldwide," in *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*. ACM, 2009, pp. 184–191.

[4] R. Arumugam, V. R. Enti, L. Bingbing, W. Xiaojun, K. Baskaran, F. F. Kong, A. S. Kumar, K. D. Meng, and G. W. Kit, "Davinci: A cloud computing framework for service robots," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 3084–3089.

[5] Z. Du, W. Yang, Y. Chen, X. Sun, X. Wang, and C. Xu, "Design of a robot cloud center," in *Autonomous Decentralized Systems (ISADS), 2011 10th International Symposium on*. IEEE, 2011, pp. 269–275.

[6] G. Hu, W. P. Tay, and Y. Wen, "Cloud robotics: architecture, challenges and applications," *Network, IEEE*, vol. 26, no. 3, pp. 21–28, 2012.

[7] K. Kamei, S. Nishio, N. Hagita, and M. Sato, "Cloud networked robotics," *Network, IEEE*, vol. 26, no. 3, pp. 28–34, 2012.

[8] G. Mohanarajah, D. Hunziker, R. D'Andrea, and M. Waibel, "Rapyuta: A cloud robotics platform," 2014.

[9] M. Tenorth, A. C. Perzylo, R. Lafrenz, and M. Beetz, "The roboearth language: Representing and exchanging knowledge about actions, objects, and environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1284–1289.

[10] A. Saxena, A. Jain, O. Sener, A. Jami, D. K. Misra, and H. S. Koppula, "Robobrain: Large-scale knowledge engine for robots," *arXiv preprint arXiv:1412.0691*, 2014.

[11] G. Kho, C. Hung, and H. Cunningham, "Robo brain: Massive knowledge base for robots," 2014.

[12] B. Qureshi and A. Koubâa, "Five traits of performance enhancement using cloud robotics: A survey," *Procedia Computer Science*, vol. 37, pp. 220–227, 2014.

[13] P. Mell and T. Grance, "The nist definition of cloud computing," 2011.

[14] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of internet services and applications*, vol. 1, no. 1, pp. 7–18, 2010.

[15] D. E. Perry and A. L. Wolf, "Foundations for the study of software architecture," *ACM SIGSOFT Software Engineering Notes*, vol. 17, no. 4, pp. 40–52, 1992.

[16] R. N. Taylor, N. Medvidovic, and E. M. Dashofy, *Software architecture: foundations, theory, and practice*. Wiley Publishing, 2009.

[17] H. Y. Zhang, C. Urtado, and S. Vauttier, "Architecture-centric component-based development needs a three-level adl," in *Software Architecture*. Springer, 2010, pp. 295–310.

[18] P. Clements, D. Garlan, L. Bass, J. Stafford, R. Nord, J. Ivers, and R. Little, *Documenting software architectures: views and beyond*. Pearson Education, 2002.

[19] Unified robot description format. [Online]. Available: http://www.ros.org/wiki/urdf

[20] R. Diankov, R. Ueda, K. Okada, and H. Saito, "Collada: An open standard for robot file formats," in *Proceedings of the 29th Annual Conference of the Robotics Society of Japan, AC2Q1–5*, 2011.

[21] L. Kunze, T. Roehm, and M. Beetz, "Towards semantic robot description languages," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5589–5595.

[22] C. Schlenoff and E. Messina, "A robot ontology for urban search and rescue," in *Proceedings of the 2005 ACM workshop on Research in knowledge representation for autonomous systems*. ACM, 2005, pp. 27–34.

[23] R. Chatterjee and F. Matsuno, "Robot description ontology and disaster scene description ontology: analysis of necessity and scope in rescue infrastructure context," *Advanced Robotics*, vol. 19, no. 8, pp. 839–859, 2005.

[24] M. Compton, C. Henson, L. Lefort, H. Neuhaus, and A. P. Sheth, "A survey of the semantic specification of sensors," 2009.

[25] M. Compton, H. Neuhaus, K. Taylor, and K.-N. Tran, "Reasoning about sensors and compositions." in *SSN*. Citeseer, 2009, pp. 33–48.

[26] Y. Gil and S. Ramachandran, "Phosphorus: A task-based agent matchmaker," in *Proceedings of the fifth international conference on Autonomous agents*. ACM, 2001, pp. 110–111.

[27] K. Sycara, S. Widoff, M. Klusch, and J. Lu, "Larks: Dynamic matchmaking among heterogeneous software agents in cyberspace," *Autonomous agents and multi-agent systems*, vol. 5, no. 2, pp. 173–203, 2002.

[28] D. Martin, M. Burstein, D. Mcdermott, S. Mcilraith, M. Paolucci, K. Sycara, D. L. Mcguinness, E. Sirin, and N. Srinivasan, "Bringing semantics to web services with owl-s," *World Wide Web*, vol. 10, no. 3, pp. 243–277, 2007.

[29] Robot operating system. [Online]. Available: http://www.ros.org/

[30] Web services glossary. [Online]. Available: http://www.w3.org/TR/ws-gloss/

[31] W3c semantic sensor network incubator group. [Online]. Available: http://www.w3.org/2005/Incubator/ssn

[32] L. Zhang, H. Zhang, Z. Fang, R. Zapata, and M. Huchard, "A domain specific architecture description language for autonomous mobile robots," in *Information and Automation (ICIA), 2012 International Conference on*. IEEE, 2012, pp. 283–288.