Wireless Pers Commun (2014) 76:231–244
DOI 10.1007/s11277-014-1687-1

CloudThinking as an Intelligent Infrastructure for Mobile Robotics

Rui L. Aguiar · Diogo Gomes · João Paulo Barraca · Nuno Lau

Published online: 12 March 2014

Abstract Mobile robotics is a transforming field that presents a varying set of challenges. The discussion on the autonomy of (self-powered) robots is not settled, and as the communication infrastructure evolves, centralized concepts become more attractive over distributed concepts. This paper presents the CloudThinking architecture applied to intelligent cloud-based robotic operation. CloudThinking offloads most of complex robotic tasks to a central cloud, which retrieves inputs from the environment as a whole in order to instruct the robots to perform its actions. CloudThinking is a natural approach to the orchestration of multiple specialized robotic systems, defining the best mechanisms for reaching a goal. Furthermore, this architecture provides a set of automatic features which can be useful for application developers. These features can fully exploit novel cloud tools development as it becomes available, providing a time-resilient infrastructure of easy upgrade. The resulting approach has the potential to create a different set of market for robotic application developers.

Keywords Mobile robotics · Cloud-based approaches · Computer Thinking

1 Introduction

Robotics is an area with increasing economical impact. Mobile robotics is now on the center of this revolution, with high-profile initiatives such as Google Self-Driving Car [1] or the

R. L. Aguiar (✉) · D. Gomes · J. P. Barraca
Instituto de Telecomunicações, Universidade de Aveiro, Aveiro, Portugal
e-mail: ruilaa@ua.pt

D. Gomes
e-mail: dgomes@av.it.pt

J. P. Barraca
e-mail: jpbarraca@av.it.pt

N. Lau
IEETA, Universidade de Aveiro, Aveiro, Portugal
e-mail: nunolau@ua.pt

Fig. 1 Robotic soccer

Mars Rover reaching the headlines. The reality is that mobile robot design present challenges which are cumbersome to overcome: mechanical actuators, self-power limitations, size and weight, and furthermore total cost, are all aspects that strictly bound robot development [2]. A "full-function" robot, able to perform all the tasks which could be desired inside an organization/plant would be (at least) very expensive and most often unfeasible in current technologies. Thus it comes as no-surprise that besides these high-visibility initiatives, mobile robotics is also undergoing a revolution, with efforts such as [3] or swarm [4], where different types of robots cooperate for performing specific tasks, which are unfeasible for a single one. In fact, plant production lines are designed with dozens of robots operating in a coordinated fashion in order to produce the final product (e.g., a car). In fact, the cost of mobility is so large that in many of these production lines, robots are non-mobile, and it is the product under construction that is being moved from one place to the other (effectively a robot with the single task of moving the item from one place to the other).

Robotic soccer, an applied exercise for the development of robotics, is another excellent example of the challenges posed for coordination in robotics. Notwithstanding its strict rules of autonomous robots, robotic soccer (see Fig. 1) is an excellent paradigm of the potential advantages of having a centralized coordination to the actions of the overall team—mimicking a fabled "perfect football team", where each play could be remotely orchestrated exactly the way the coach desires. Each robot would be perfectly placed and would have the perfect movement required to realize a defending or attacking move, given the superior knowledge of all the actions in the field that the "outside coach" has.

Current developments on communication technologies and cloud computing raise the question of why not to design such a perfect team, where the "coach" could be almost all-powerful and all-knowledgeable, coordinating the team movements to a level unattainable by any fully autonomous system.

This paper, albeit not focused on robotic soccer (which has its own strict rules regarding robot autonomy), tries to address this challenge. We present a cloud-based system that can be used to orchestrate arbitrary mobile robotic systems in real time, and at the end summarily show its application to robotic soccer. In Sect. 2 we overview concepts of autonomic robotic systems and the advantages of using the cloud for their overall coordination/control. Section 3 briefly highlights some other approaches developed under these lines, and shows some of their shortcoming. Section 4 shows our approach, CloudThinking for Mobile Robotics, and discuss its primary blocks. Section 5 goes then to the discussion of the novel ecosystems which are inherent from our approach. Finally we conclude the paper in Sect. 6.

2 Autonomic Robotic Systems and the Cloud

Robotic systems have long been used for several ranging and remote sensing applications. While they are able to sense the environment and trigger actuators (like traditional static sensing platforms), mobile robotic systems are given some more computational capabilities and the possibility of (at least partially) moving in the environment they reside. Robots can be self-mobile, or their mobility capabilities may need to be assisted by other specialized devices, or even human operators [5]. A frequent scenario consists in operator-assisted mobility until near the area of actuation, followed be self-mobility in a geographically more reduced area. Independent of their mobility characteristics (self vs assisted), mobile robotics are computation platforms which are deployed in order to fulfil a set of tasks, frequently in cooperation, as in the robotic soccer example above (and which we will retain as example in our discussion).

The overall architecture needs to consider then two different sets of challenges:

(1) The component level aspects, where the overall robot subsystems have to be managed and controlled in order to perform its function (e.g., moving, kicking the ball, scanning the environment, define orientation, etc… Security and safety safeguards also need to be present at this level).
(2) The coordination aspects, which implements system-wide behaviour using these essential robotic features as building blocks, and follows global restrictions (e.g., when to move, where to move, when to kick the ball, check that you remain inside the field, point to the goal).

It is no surprise that mobile robotics architectures follow this two level hierarchy, presenting a high level where policies and inter-unit communication are paramount, and a lower level with focus on sensors and actuators processing. These two levels frequently use different computing resources. For instance, CAMBADA [6], a software architecture for soccer robotics, winner of several prizes, is depicted in Fig. 2. One can identify the low level sub-system management and the higher level functions.

It is particularly important to highlight the RTDB (Real Time DataBase) at the core of the architecture. The needs for real time operation is one of the major reasons that impair the remote location of the "sensorial interpretation, intelligence and coordination".[1] It is challenging to develop a communication (and processing) loop that fulfils the real time requirements of mobile robotics, so traditionally all these functions are located in the robots themselves.

The basic structure running on these autonomic systems implements a form of the Monitor, Analysis, Planning and Execution architecture (MAPE) [7], or, in more recent versions, improved versions to consider knowledge (MAPE-K) [8] (see Figure). MAPE-K involves both software elements (such as vision processing software or services of the communication stack) as well as hardware (such as sensors and actuators, e.g., temperature sensors or step

[1] Strictly speaking, the rules of robotic soccer require such autonomous architectures, regardless of technical feasibility.

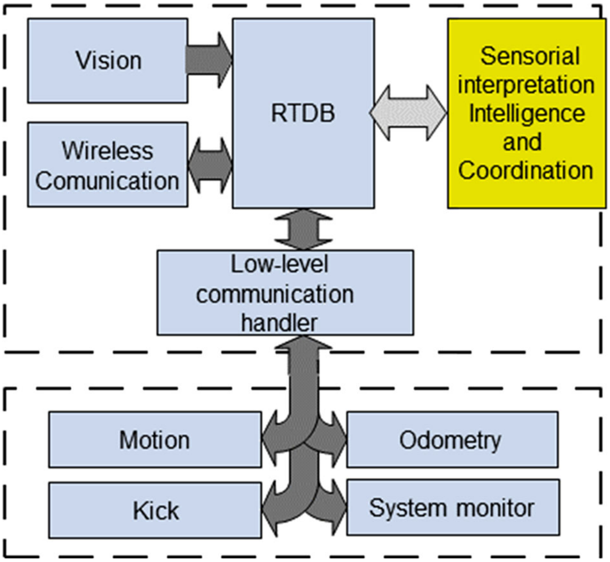


Fig. 2 CAMBADA software architecture

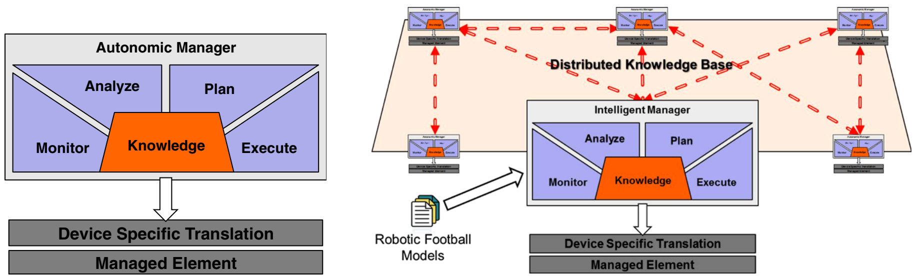


Fig. 3 Autonomic feedback loop with knowledge (*left*) and distributed knowledge (*right*)

motors). Knowledge is sometimes distributed among system nodes, but not necessarily across all nodes, and not necessarily uniformly over all nodes.

The manager deployed in such a distributed system then may benefit of a centralized approach, instead of being replicated in each node (robots) (Fig. 3).

Technologies for deploying this knowledge plane are known. Service-oriented architecture (SOA) solutions such as SOAP [9], REST [10], or CORBA [11] provide standardized interfaces that enable efficient distributed system development, and arbitrary complex control loops. Agents [12], actors [13], or bio inspired cooperating systems [14, 15] are some of the techniques that can be used for the control loop, often in a distributed manner.

The new paradigm of Cloud Computing brings a new opportunity for these robotic systems. Cloud Computing offers virtually unlimited resources in terms processing power, storage space, scalability, and communication bandwidth between core components. This means that robots can explore virtually an arbitrary amount of complexity on their functions of "intelligence and coordination". Robots can benefit from cloud computing by transferring algorithms that demand high computational power to the cloud, by having access to a much larger (shared) knowledge base that can be stored in the cloud, or to have a place where they can retrieve and store new information. Primary functions can still be located in the robot, but all aspects related with inter-robot coordination or information sharing can be handled by the cloud. This allows for focusing Robot development in real time sensing and actuation (e.g., movement), and outsources computationally expensive tasks to systems where the implementation of such algorithms has less cost. We believe that aiming for the development of an all-autonomous robotic system, although a great research challenge, is not the best approach. Decoupling the several robotic subsystems, and considering its integration with a highly scalable computational cloud is a more feasible and cost effective approach, if network communications are reliable.

3 Cloud Robotics Frameworks

The term "Cloud Robotics" [16] is generically refer to the robotics field that explore the cloud as enabling technology. This model relocates part of the robot intelligence into the cloud, retaining only essential security, movement and sensorial functions in the robot. Cloud Robotics is expected to develop Robotics in five different topics [17]:

- Big Data—Robots can generate vast amounts of information either by using high throughput sensors, like RGD-D or high frame rate cameras, or by operating throughout large amounts of time. Cloud technologies enable robots to use Big Data techniques, saving computer power, storage and robot weight. Other services like data backup, reuse, sharing and availability can be improved through the use of the cloud.
- Cloud Computing—Computational power is one of the most important requirements of modern robot algorithms. When all processing is performed locally, there is a compromise on the amount of computing power that will be available on the robot and the algorithms that will be used: more processing power means increasing power consumption, less mobility, increased weight and increased costs. Cloud computing provides a solution where the most computational intensive algorithms may be executed in the cloud, using virtually unlimited parallel resources [18]. A robot using the cloud may offload its learning and image processing, or other high demanding tasks, to the cloud, using the best algorithms for these tasks.
- Open Source/open access—The cloud may also be very effective as an open repository of code, data, algorithms, benchmarks and hardware designs. The developers will have at their disposal state of the art modules that can be used to build new robots. This way developments from one research or development group can easily be made available to other developers, increasing productivity and fostering new development that can themselves be stored in the repository. (e.g. ROS—Robot Operating System [19]).
- Collective Robot Learning—Robot learning needs not only good computational resources but also a large amount of training data. The cloud may be used to store, share and reuse data, collected from robots performing in different environments all around the world, which is made available for collective learning tasks. By having a larger training dataset that is continuously updated by different robots, the cloud provides a very interesting learning environment [20]. Moreover, a collective of robots can be easily upgraded by exploring service composition and aggregation of external datasources.
- Human learning—The cloud may be used not only as a repository of robotic solutions but also as a way to actively share knowledge/implementation, in crowdsourcing and call center approaches. Cloud based services allow developers to share their problems and contribute to each other developments, or even to develop a robotic system in a

crowdsourcing paradigm where all contribute with their knowledge and work to the final robotic system.

Cloud Robotics is an early area, with several of its aspects being tackled in multiple papers, as seen before. Global approaches are also being proposed. For instance, GostaiNet [21] enables to run complex modules on remote servers, such as speech recognition and face detection, which could not run inside cheap robots. Nevertheless, there are two initiatives that should specially be referred by its ambition: the DAvinCi [22] and the RoboEarth [23].

The A*STAR Data Storage Institute of Singapore proposed a software framework, DAvinCi, that uses the cloud computing paradigm for simultaneous localization and mapping in large environments. The Distributed Agents with Collective Intelligence (DAvinCi) Project aims to provide software architecture to enable heterogeneous agents to share sensor data and also to upload data to processing nodes for computationally intense algorithms. It combines the distributed ROS architecture, the Hadoop Distributed File System (HDFS), and the Hadoop Map/Reduce Framework. The DAvinCi server allows external entities (e.g., robots, human interfaces) to access the cloud services, and binds the ROS middleware to the backend Hadoop computational cluster, used to provide a reliable and scalable computing platform. The ROS framework provides a standard communication layer across robots, and between the DAvinCi server and the robots. A set of algorithms (versions of widely used Simultaneous Localization And Mapping used for path planning) are exposed as cloud services that can be accessed over the local network, as if in a private cloud, or by the Internet through ROS messages encapsulated in HTTP requests/responses. End results show that by delegating the computation to the Cloud, any robot with commodity hardware can use the algorithm with acceptable times.

Another project is RoboEarth. Based on standard Internet protocols and an open-source cloud architecture, this project attempts to develop a World Wide Web for robots, emphasizing its role as database repository for information sharing and for robots to learn from each other. Rapyuta [24] is the open-source PaaS Cloud Engine used in RoboEarth. It is designed to help robots to offload their heavy computations by providing secure and customizable computation environments in the Cloud. Before Rapyuta, robots would only query the RoboEarth repository, and all the processing and reasoning over the data happened locally. With Rapyuta, the robots can perform these tasks in the cloud. Additionally, Rapyuta provides a computing environment compatible with ROS, making all ROS open-source software packages available to be used in the Cloud without any modification.

4 CloudThinking Architecture for Mobile Robotics

The cloud robotic architecture proposed in the CloudThinking concept encompasses both hardware and software components, that enable computationally intensive tasks to be remotely executed, in a similar approach to DaVinci and RoboEarth. This allows the construction of robots with limited computational resources, thus reducing power consumption, weight, size and overall cost. The cloud can, not only optimize current robot functionalities, but also expand them beyond the current state of the art thanks to virtually unlimited computational and storage resources. Our architecture supports Real-Time aspects, so that the performance of the robots is not affected by variations in the quality of cloud services. In comparison with existing approaches, CloudThinking was designed in order to present several advantages:

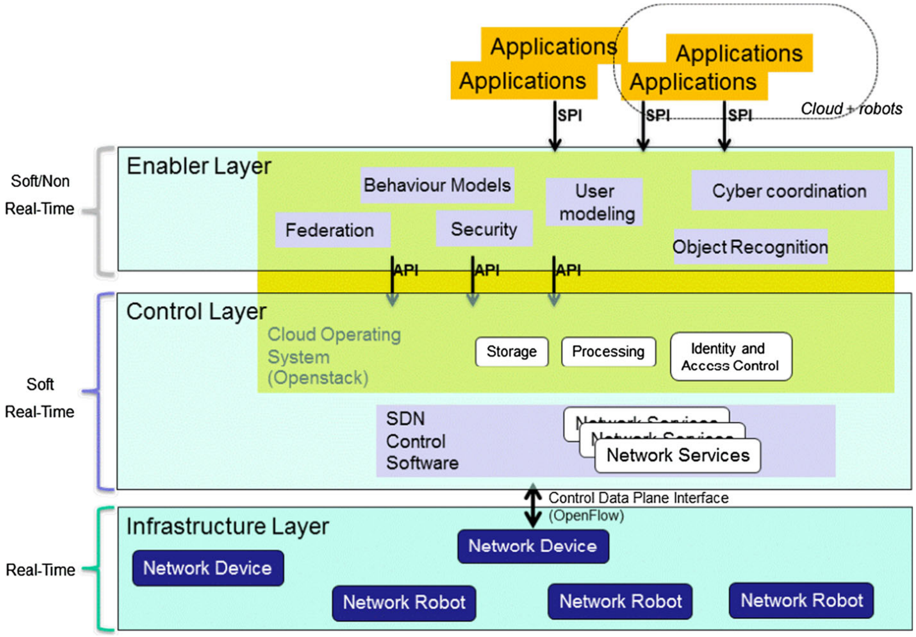


Fig. 4 Overall CloudThinking architecture

- Provides integrated control of the network layer, imposing protected and reliable communications with robots according with the existing network infrastructure.
- Integrates multiple data sources into the same overall data processing environment, including sensors and databases which are not necessarily associated with robotics (e.g. weather, geolocation).
- Supports data and ontology models designed for optimized robotic tasks, such as behavior learning and object recognition.
- Inherently provides continuous monitoring of all system elements and provides performance metrics near-real time (or even alarms).
- Presents an advanced set of functions for robotic application developers to easily integrate in their design.
- Follows clear standardized interfaces for cloud computing and network control, enabling federation of multiple databases.
- Enables multiple business models, and multiple business actors, creating a novel ecosystem.

For realizing these advantages, CloudThinking has an architecture structure along three layers (Fig. 4). A first layer, reflecting the physical infrastructure. A second layer, related with the overall control of the communications and cloud infrastructure, is divided in two different sub-layers: a first one dedicated to the control and adaptation of the communication with the robots; and a second one, dedicated to the support of the cloud operating system. The top enabler layer, running over the cloud operating system, supports the multiple enablers required for the mobile robotics. Finally, over this infrastructure, different robotic applications can be developed, with components necessarily in the robots, and components resorting to the enabler layer. Typically these applications will be run in the same cloud environment (and of course, part will be run in robot, which will be running a modified version of ROS), but

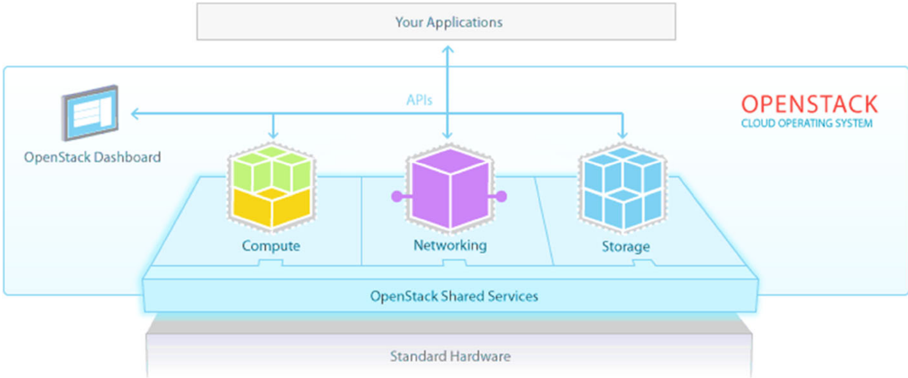


Fig. 5 OpenStack reference architecture

this is not necessarily so. The enabler layer will use standardize cloud API interfaces, and specific service API are made available for application developers.

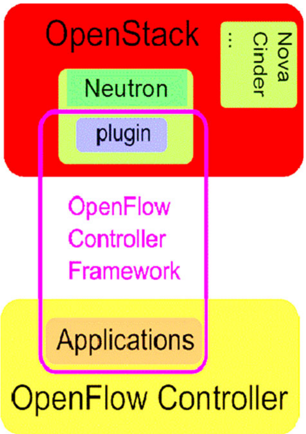
Network and device control and communications are a key concern of our architecture. To accomplish mobile robotic, the entire infrastructure has to be tightly integrated, including data centers, networks, applications and services. The network plays a critical and fundamental role on this, being the sole component that interacts with all the elements in a cloud service. It is also the element that connects the robots remotely to the cloud, ultimately delivering the Quality of Experience to the robots in terms of availability, security, and latency. Management APIs that allow automating provisioning and network services management are essential.

The most flexible approach to network and device management is to resort to Virtualization [25]. Here we define the abstraction of physical resources as logical units, such that a single physical resource can appear as multiple logical units (one-to-many) or vice-versa, multiple physical resources appear as a single logical unit (many-to-one). This separation between the connections and the physical network hardware makes the way to a fully abstracted view of the resource allocation management. This approach, while hiding the irrelevant details at the physical layer, allows a separate entity to provide network communications as a service. This separation of concerns, between network control (which needs to address network specifics, such as how to impose QoS or real-time control) and the Cloud Robotics computational nodes, allows cloud robotics application developers to ignore the intricacies of communication issues, and concentrate on their application.

In our prototype environment, we use OpenFlow [26] to control the network, which can then be reconfigured upon need. Other management protocols are used for non-OpenFlow routers and access points, but from the point of view of the cloud robotics applications, overall control is placed at the OpenFlow controller. Non-OpenFlow networking elements have their own interface defined to translate OpenFlow commands into adequate control messages.

The software reference for the core of CloudThinking is the OpenStack cloud operating system. CloudThinking aims to provide an environment where multiple efforts can converge. Such a multi-tenant requirement imposes the adoption of popular and easy to federate cloud operating systems. OpenStack is specially well-structured to CloudThinking needs, as it separates computational, networking and storage aspects in comportamentized entities (see Fig. 5), above a common set of cloud services, allowing easy development of enhancements. The Service-Oriented-Architecture of OpenStack is an adequate fit to the underlying service logic from CloudThinking, and a key enabler for the development of a mobile robotics ecosystem.

Fig. 6 Network integration inside OpenStack



Network virtualization integration in the OpenStack environment is possible by the usage of the Neutron package. This allows the integration of a OpenFlow controller in the overall OpenStack environment (see Fig. 6). The so-called Southbound interface for the CloudThinking infrastructure assumes an OpenFlow enabled network, following the description above. This allows the extension of the virtualization concepts at the data center into the network.

Essential services running in the cloud are quite usual, and fit to the virtualization approach mentioned above. Each robotic application is able to access its own virtualized resources, providing efficient data security, and providing inherent reliability. Although it is desirable to provide controlled sharing of information, uncontrolled software feature influence across multiple robots is undesirable. For an application developer, the whole infrastructure should appear as its own environment, trustable and reliable, where his set of robots "reside". At the limit, each robot image can be safely stored in its virtual space, and downloaded upon startup, providing an easy management system to ROS (e.g., supporting updates). As computational requirements increase, more resources can be provisioned in the cloud for that robotic application (although for our robotic soccer prototype, a single virtual machine is enough), providing horizontal scalability to robotic applications.

Note that the virtualization approach allows all data objects to be stored securely, but in reality they will be lying in an unified system. This makes data interchange across robotic applications a matter of opening the adequate access permissions in the cloud. This common knowledge layer consists then of a repository of information that is available to authorized members (that is, either in the same virtualized environment, or federated), and to which sensed data can be published and queried for further correlation. This common knowledge layer also provides for a continuous aggregate of information about common tasks and reactions to common events, such as structural damage, tampering (in urban scenarios), or simply hardware malfunction. Robot operations and features are modeled according with policy concepts [3], including roles, obligations and actions.

This commonality of file system storage and common information layer is one of the features enabling some of the enablers provided to the application developers. These enablers are available in the Northbound interface of the Cloud infrastructure, and can be divided in five different classes:

- Object learning and recognition—Enablers here offer specific functionalities for object learning and recognition. In the context of object recognition, large sets of training images can be used to run heavy learning/classification algorithms.

- Behavior learning—The process of behavior learning is very time consuming and must, in general, be accomplished, at least partially, manually by using computational resources external to the robots. Several behavior learning methods can be used as enablers, namely reinforcement learning (Q-learning, Neural Fitted Q iteration) and optimization (hill climbing, genetic algorithms, tabu search) based methods.
- User and agent modeling—This category of enablers provide agent modeling, including user modeling and opponent modeling (used in competitive tasks, such as robotics football).
- Cyber coordination—Multi-robot teams typically need to share environment information as well as configurations and control/coordination decisions. Enablers allow robots to share their knowledge, and access others knowledge, keeping delays and network contention as low as possible.
- Generic infrastructure—Requirements for access control, device and system management, security, and federation capabilities are transversal to all functions, and related adequate enablers can be explicitly called when required.

Note that the usage of the cloud allows for multiple data sources to be integrated in e.g. fusion algorithms. For instance, the network itself holds a huge amount of information regarding topology, end-to-end performance status and even user location. This is per se a valuable data source that can be made available in the cloud, and which can provide added information for robotic control algorithms (such as aided repeaters [3]).

5 CloudThinking Cloud Robotics Ecosystem

The CloudThinking framework was developed taking in consideration the need for a global mobile robotics ecosystem (see Fig. 7), although our example case was a simple robotic soccer trial.

In our vision, robots will be widespread in society. Some of these robots will be locally controlled (e.g. the Romba vacuum cleaner [27] but the trend will be for increasingly intelligent robots to be deployed, relying in a backend for its operation. This will lead to the appearance of robotic solutions providers. These will cater for their specific markets, and develop solutions optimized (both in performance and in cost) to these end-costumers. This trend can be seen today, with the development of robotic solutions for home [28] and professional applications [29].

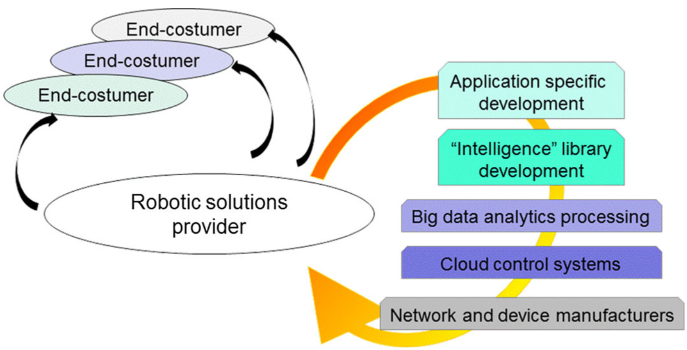


Fig. 7 Cloud robotics eco-system

Nevertheless, this industry has not reached an expansion level comparable with other emerging ICT applications. One of the reasons is associated with the difficulties in building new products based in previous industry experience: power requirements keep moving, technology evolves, actuators change, robot purpose changes, algorithms need to be adapted, networks are hard to control, are all issues that contribute to the difficulties of reusing knowledge and technologies.

Cloud robotics has the potential to change this situation, and CloudThinking was designed to enable such a change, by adherence to existing standardize interfaces at all levels. The ability to remotely deploy most of the intelligence of the robot (or even completely manage the software image of the robot), has a transformation potential that should not be neglected.

For a widespread development of the cloud robotics industry, multiple actors will need to be in place, in order for synergistic developments to arise.

On a very first level, *device and network manufacturers* need to settle on an agreed set of standardized interfaces. Device standardization is mostly related with hardware and firmware issues (such as those implied by ROS [19]), while network standardization addresses technology and communication processes between the device and the cloud. On an associate level, network control interfaces need to be standardized (our usage of OpenFlow is not currently an industry practice, and several types of interfaces exist depending on the network operator). Without a simple, reliable and trusted way to control communications with a mobile robot, the amount of intelligence that can be relocated into the cloud will always be limited, and complex delegation architectures, managing interoperating different levels of complexity, will need to be deployed in the robots.

While this level is related with the processing, flow and control of information for the robots, *cloud control systems* are related with the infrastructure required for the storage, processing and running operation of cloud robotics. This is currently a topic undergoing large developments, and industry is expected to consolidate along a specific set of interfaces in a medium term. Our choice of OpenStack is a representative of one of the most promising standards.

However, from this level upwards, there is a lack of common practices. *Big data analytics tools* are essential for large scale learning and complex environment operation, and while this is a promising industry in development, there is not yet a common standard that can be used for cloud robotics applications. In fact, there is not yet an established vision on large data fusion approaches for robotic operation. Incipient efforts exist nevertheless in the development of *intelligent libraries for cloud robotics*: the efforts of DaVinci and RoboEarth are driven in this direction, but no easy access or reusability exist.

These four areas seem to be able to present its specific economic value, and to lead to the creation of a value chain, that ultimately will be explored by *robotic applications developers* in order to provide solutions in an economic way.

The CloudThinking approach defines adequate interfaces at these multiple levels, providing a path for the development of such a cloud robotics ecosystem.

6 Conclusions

The CloudThinking framework presents a flexible architecture for Cloud Robotics. This concept aims to provide the advantages of a global system with the security and control of a dedicated solution. The infrastructure resorts to an approach where network communications are controlled (mostly using OpenFlow) by a cloud operating system (based on

OpenStack), providing a reliable and controllable environment with inherent virtualization. This environment supports a large set of tools for mobile robotics, including object recognition, training functions, environment assessment, and multi-user coordination. A centralized data storage allows for the easy development of novel robotic systems, and for the incorporation of increasingly complex environment assessment sensors into the robots decision process. Furthermore, in this environment, a virtuous ecosystem can arise, with strict separation of concerns between multiple actors enabling reusage of intelligence and experience from previous robots.

The system is being trialed into a modified robotics football environment, by adapting the basic CAMBADA robotic system.

Acknowledgments This work was supported by project Cloud Thinking (CENTRO-07-ST24-FEDER-002031), co-funded by QREN, "Mais Centro" program.

References

1. Thrun, S. (2010). What we're driving at. Official Blog, Google, October 9, 2010. http://googleblog.blogspot.com/2010/10/what-were-driving-at.html. Accessed Apr 18, 2013.
2. Arumugam, S., Kalle, R. K., & Prasad, A. R. (2013). Wireless robotics: Opportunities and challenges. *Wireless Personal Communications*, *70*(3), 1033–1058.
3. Sadeghi, R., Barraca, J. P., & Aguiar, R. L. (2013). Collaborative relaying strategies in autonomic management of mobile robotics. *Wireless Personal Communications*, *70*(3), 1077–1096. doi:10.1007/s11277-013-1104-1.
4. Şahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application. *Swarm Robotics, Lecture Notes in Computer Science*, *3342*, 10–20.
5. Nehmzow, U. (1993). *Mobile robotics: A practical introduction*. Berlin: Springer.
6. Almeida, L., Santos, F., Facchinetti, T., Pedreiras, P., Silva, V., & Lopes, L. S. (2004). Coordinating distributed autonomous agents with a real-time database: The CAMBADA project. In *Computer and Information Sciences-ISCIS 2004* (pp. 876–886), Berlin: Springer.
7. Horn, P. (2001). Autonomic computing: IBM's perspective on the state of information technology. IBM technical report.
8. IBM. (2003). An architectural blueprint for autonomic computing. IBM technical report.
9. Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J.-J., Nielsen, H. F., Karmarkar, A., & Lafon, Y., (Eds.). (2007). SOAP version 1.2 part 1: Messaging framework (2nd ed.). World Wide Web Consortium, 27 April.
10. Fielding, R. T. (2000). Architectural styles and the design of network-based software architectures. Ph.D. Dissertation, chapter 5, University of California, Irvine, AAI9980887.
11. Object Management Group. Common object request broker architecture. Object Management Group Standard. http://www.omg.org/spec/
12. Shin, S.-O., Lee, J.-O., & Baik, D.-K. (2007). A mobile agent-based multi-robot design method for high-assurance. In *High assurance systems engineering symposium, 2007. HASE '07. 10th IEEE* (pp. 389–390).
13. Darche, P., Raverdy, P.-G., & Commelin, E. (1995). ActNet: The actor model applied to mobile robotic environments. In *OBPDC 1995* (pp. 273–289).
14. Mohan, Y., & Ponnambalam, S.G. (2009). An extensive review of research in swarm robotics. In *World Congress on nature and biologically inspired computing. NaBIC 2009* (pp. 140–145).
15. Han, Q., Wang, Q., Zhu, X., & Xu, J. (2011). Path planning of mobile robot based on improved ant colony algorithm. In *2011 international conference on consumer electronics, communications and networks (CECNet)* (pp. 531–533).
16. Kuffner, J. J. (2010). Cloud-enabled robots. In *IEEE-RAS international conference on humanoid robotics*. Nashville, TN.
17. Goldberg, K. (2013). Cloud robotics. Retrieved July 2013. http://goldberg.berkeley.edu/cloud-robotics/.
18. Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, *51*, 107–113.
19. Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Rob, W., & Ng, A. Y. (2009). ROS: An open-source Robot Operating System. *ICRA Workshop on Open Source Software, 3*(3.2)

20. Dang, H., & Allen, P. (2012). Learning grasp stability. In *2011 IEEE international conference on robotics and automation (ICRA)* (pp. 2392–2397). IEEE.
21. Gostai (2013). GostaiNet. www.gostai.com/activities/consumer.
22. Arumugam, R., Enti, V., Bingbing, L., Xiaojun, W., Baskaran, K., Kong, A. F. F., et al. (2011). DAvinCi: A cloud computing framework for service robots. In *IEEE international conference on robotics and automation (ICRA)* (pp. 3084–3089).
23. Waibel, M., Beetz, M., Civera, J., d'Andrea, J., Elfring, J., Galvez-Lopez, D., et al. (2011). RoboEarth—a World Wide Web for robots. *IEEE Robotics and Automation Magazine*, *18*, 69–82.
24. Hunziker, D., Gajamohan, M., Waibel, M., & D'Andrea, R. (2013). Rapyuta: The RoboEarth cloud engine. In *Proceedings IEEE international conference on robotics and automation (ICRA)* (pp. 438–444). Karlsruhe, Germany.
25. Anderson, T., Peterson, L., Shenker, S., & Turner, J. (2005). Overcoming the internet impasse through virtualization. *Computer*, *38*(4), 34–41.
26. McKeown, N., et al. (2008). OpenFlow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, *38*(2), 69–74.
27. Jones, J. L. (2006). Robots at the tipping point: The road to iRobot Roomba. *IEEE on Robotics and Automation Magazine*, *13*(1), 76–78.
28. Malehorn, K., Liu, W., Im, H., Bzura, C., Padir, T., & Tulu, B. (2012). The emerging role of robotics in home health care. *AMCIS 2012 Proceedings*, Paper 62.
29. Moradi, H., Kawamura, K., Prassler, E., Muscato, G., Fiorini, P., et al. (2013). Service robotics (the rise and bloom of service robots). *IEEE on Robotics and Automation Magazine*, *20*(3), 22–24.

Rui L. Aguiar received a Ph.D. degree in electrical engineering in 2001 from the University of Aveiro, Portugal. He is currently a professor at the University of Aveiro, being responsible for networking aspects. He is leading a research team at the Institute of Telecommunications, Aveiro, on future generation network architectures and systems. His participation in European cooperative research is extensive, and he is currently involved in the 5G-PPP steering board. His current research interests are centered on the implementation of advanced networks and systems with special emphasis on future Internet and mobile architectures. He is a member of ACM and a senior member of IEEE. He has more than 300 published papers in those areas. He has served as technical and general chair of several conferences, such as ICNS'05, ICT'06, ISCC'07, Monami'2012, ISCC'2014, NTMS'2014.

Diogo Gomes graduated in Computers and Telematics Engineering from the University of Aveiro in 2003 with first class honors, and concluded his Ph.D. by the same University on Resource Optimization for Broadcast Networks in 2010. He's currently an Invited Auxiliar Professor at the University of Aveiro. Since his graduation year, he has participated in several IST Projects such as IST-Mobydick, IST-Daidalos, IST-Akogrimo, IST-C-MOBILE and ICT-C-Cast where besides conducting research on QoS, IP Mobility and Multicast/Broadcast has always been deeply involved in the deployment prototypes demonstrations. He was one of the founders of the University Linux User Group and has contributed to several OSS projects. Recently he participates in ab Industrial Partnership with Portugal Telecom Inovação on Machine-2-Machine Communication and Information retrieval from sensor networks.

João Paulo Barraca received a Ph.D. degree in informatics engineering from the University of Aveiro, Portugal in 2012. He joined Instituto de Telecomunicações (IT) in 2003, were he develops his research activities, and in 2008 he became an Invited Lecturer at the University of Aveiro. His research activities include management mechanisms for community-oriented autonomic networks, development of experimentation facilities for wireless environments. He has coauthored more than 30 publications, published in journals and conference proceedings.

Nuno Lau is an assistant Professor at Aveiro University, Portugal. He got is Electrical Engineering Degree from Oporto University in 1993, a DEA degree in Biomedical Engineering from Claude Bernard University, Lyon, France in 1994 and the PhD from Aveiro University in 2003. Nuno Lau is one of the team leaders of FC Portugal team that has been 3 times World champion of RoboCup (simulation leagues). He is also a member of the CAMBADA Middle-Size League team which has also been RoboCup World Champion. His research interests include Intelligent Robotics, Artificial Intelligence, Multi-Agent Systems and Simulation. He has lectured courses at Phd and MSc levels on Distributed Artificial Intelligence, Intelligent Robotics, Computer Architecture, Programming, etc. Nuno Lau is the author of more than one hundred publications in international conferences and journals.