Towards a Unified Platform for Agent-based Cloud Robotics

Francisca Rosique, Pedro Sánchez, Diego Alonso and Juan Antonio López
Systems and Electronic Engineering Division (DSIE), Universidad Politécnica de Cartagena, Cartagena, Spain

Keywords: Teleo-reactive, Agents, Cloud, Robotics, Drone.

Abstract: This paper describes a platform that aims to design, build, and validate a new generation of cloud robotic platforms that enable agent-based intelligent control of robots deployed in unknown and dynamic environments. The platform will consider: (1) novel techniques for programming reactive plans and robotic behaviours through missions and novel mechanisms for building new behaviours from existing ones, both for experienced and non-expert users; (2) novel multi-layered cloud platform as the infrastructure to maintain a continuous link between the robots acting on a physical environment and their agent counterparts, to provide sensor data from robots to agents, and to provide high-level autonomous decisions from agents to robots.

1 INTRODUCTION

Intelligent agents are autonomous and heterogeneous entities that can perceive their environment and are able to achieve tasks on their own or collaborating/competing with other agents (Russell and Norvig, 2003). Agent technology advances can be helpful in the robotic areas providing capabilities such as planning, learning and decision-making (Thrun et al., 2005), (Kaminka, 2012). Distributed robotic applications can be more easily obtained considering robotic devices as avatars of virtual agents (Dipsis, 2010). These agents can use Teleo-Reactive (TR) programs to support flexible agent control, high-level goal achievement, and collaboration between robots (Benson and Nilsson, 1995), (Clark and Robinson, 2014).

Traditionally, robotic developers have to set up multiple computing layers to begin building an application. From identifying a server, installing the Operating System, to deciding on programming languages and running preliminary testing. It is anything but simple. It needs a team of experienced developers to make the solution work. In this context, we propose a unified platform for agent-based cloud robotics (Robotic Agents in the Cloud - RAC) that will leverage the knowledge and capabilities needed for developing such systems.

RAC will abstract away from almost all of this through the development of a cloud-based execution platform that augments low-level cognitive robots with high-level cognitive behaviours using theTR approach suitably extended with reasoning under uncertainty. The TR formalism will make the specification of the expected behaviour easier than traditional programming languages, since programs are written in terms of a sequence of goals that the system has to reach (i.e., missions). Because TR programs are much easier to read than to write, non-technical users could easily understand TR programs written by experienced developers, deciding which missions fit best to the problem they try to solve. So, the programming of robots can reach a wider set of end users. At the same time, the cloud infrastructure will abstract almost all the resources needed to perform simulation and computation tasks.

In this context, the proposal will provide technical and non-technical users with intuitive application development tools to easily create custom assemblies of autonomous robots to solve application specific tasks, as if they were involved in a mission. Applications will be hosted in the cloud without the need of programming code, using simple, and easy-to-use visual development tools based on a drag and drop development model (see Scratch tool (Resnick, 2009) from MIT as an example).

The proposal will be validated through a cloud-based platform with scenarios from the civil area including: critical infrastructure surveillance and search and rescue in emergencies. By using Unmanned Aerial Vehicles (UAVs), commonly known as drones, we expect our approach to be

Rosique F., Sánchez P., Alonso D. and López J..
Towards a Unified Platform for Agent-based Cloud Robotics.
DOI: 10.5220/0005553700590064
In *Proceedings of the 10th International Conference on Software Engineering and Applications* (ICSOFT-EA-2015), pages 59-64
ISBN: 978-989-758-114-4

transferrable to other potential drone applications areas like agriculture, commercial and logistics and transport.

Figure 1 shows an overview of the platform from the point of view of the end-user. Though drones have been the robots selected for the demonstrators, it is meant that other type of robots could be consider and integrated in the solution. So, a typical non-technical user could build a mission by: (1) accessing the TR behaviour repository with missions previously tested by other, experienced users, so that the sharing of knowledge and experiences between users is promoted; (2) configure, modify or extend the reused missions; (3) simulate in a 3D environment the expected behaviour of the TR program associated to the mission; (4) deploy in real the validated TR program to execute mission; and (5) visualize and analyse the data (video, sensor data, etc.) received from the drone during the execution.

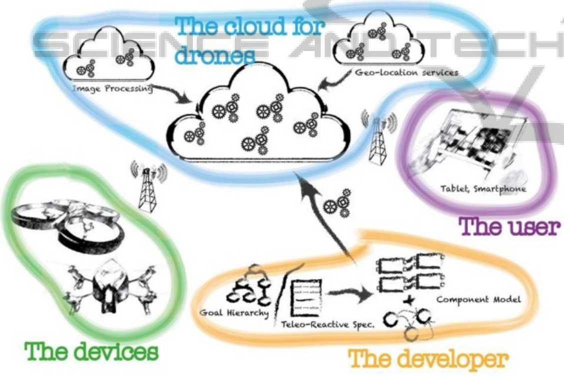

Figure 1: Overview of the approach with the involved actors.

Moreover, RAC is aligned with the Robotics 2020 Strategic Research Agenda (SRA) of Europe (produced by Euroboticsaisbl) (EUROP, 2010), as it allows exploiting technology in key areas such as agriculture, commercial and society security. It is then needed more disruptive technologies ahead of the wave, more markets to exploit emergent robotics, and more awareness in Society of the potential for robotic systems. Robot's ability to monitor large areas from the air and to work in hazardous environments will provide new and cost effective ways to gather valuable data in civil security (border protection, patrolling of plants, rescue, surveillance, etc.). This may represent an additional opportunity for the creation of jobs and the stimulation of the European economy given the potential set of applications and SME that can benefit from the results. Besides, making robot programming easier and more affordable will greatly increase their demand, boosting the creation of new manufacturers of drones.

This paper is structured as follows: Section 2 describes a general overview of the proposal. Next, Section 3 presents several works related to TR and Cloud systems development. This is followed, in Section 4, by the description of the scenarios of implementation in which the proposal is used for demonstrating the application of the approach. Finally, in Section 5, we discuss our future work and present some concluding remark.

2 GENERAL OVERVIEW

The main idea behind RAC is the development of a cloud-based execution platform that augments low-level cognitive robots with high-level cognitive behaviours using the TR approach extended with reasoning under uncertainty. The work will be undertaken from the next four main perspectives: intelligent robotic agent modelling, tools and infrastructure (TR program implementation, 3D simulation, support in the cloud, etc.), drone integration, and real life demonstrators in domain of high interest as stated in the SRA for Robotics in Europe.

For this proposal, we will provide the TR specification language and the implementations for executing missions (with possible collaboration of drones, making use of probabilistic reasoning under uncertainty). These outcomes will be needed in a 3D simulation environment for simulating the TR programs executed on drones. Moreover, we will provide a platform in the cloud that supports and brings together the software artefacts developed, giving to users a front-end to develop and deploy TR missions. This cloud platform will cover the full life-cycle of TR applications enabling users to: (i) develop TR programs, (ii) test them with the 3D simulation environment, and (iii) deploy them in real drones, monitor their operation in the real environment and store sensors readings for future needs. A reusable repository of TR specifications will enhance the reuse of open missions between users and developers. Users will be able to learn about the TR paradigm through educational material specifically conceived for the proposal. The front-end will provide users a novel visual language to build their own TR missions. The demonstrators will need first to be modelled and simulated using the Virtual Simulator. Once the results of the simulation are deemed adequate the missions can be executed.

Analysis of the simulations will also be used to cross-validate with the results of the real-world missions. To address this idea, the functionality provided for RAC is organized in a three layer architecture: (1) Infrastructure Layer; (2) Platform Layer; and (3) Application/Service Layer. Figure 2 shows the main components of the RAC platform where the stakeholders make use of the services provided at the Application Service Layer (TR program simulation and execution, TR program edition, educational resources for learning the paradigm, repositories of TR missions, etc.). The platform layer includes those modules to be developed for the proposal.

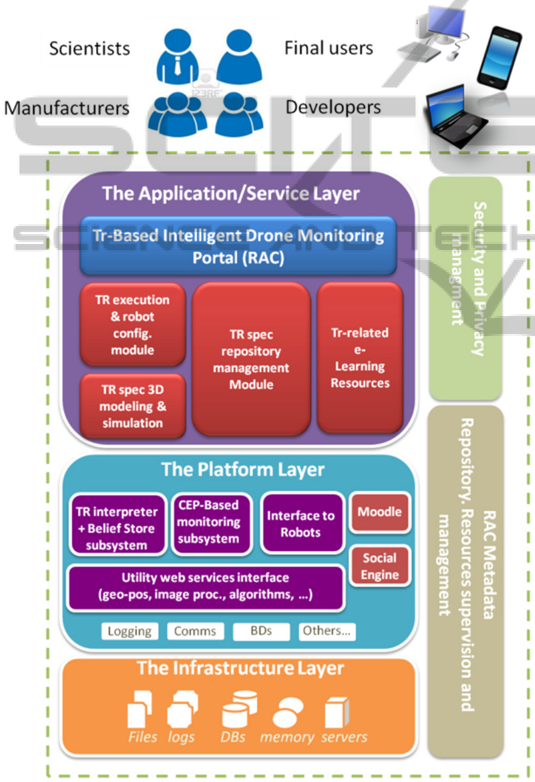

Figure 2: Architecture of RAC.

3 RELATED WORK

The use of TR programs for intelligent agents as well as robotics has been extensively studied in the literature. In (Benson and Nilsson, 1995), a quite elaborate agent architecture is described that makes use of TR procedures represented as trees, which was introduced in (Nilsson, 1994). Nilsson's agent tower architecture paper (Nilsson, 2001) extended his robust goal-directed TR robotic agent programming language with an inference capability using the tower of rule defined predicates sitting on top of a set of rapidly changing percept facts. The predicate definitions allow TR procedures towards the top of the tower of TR procedures to have Guard -> Action rules with guards that query and interpret current percepts using application specific concepts and a declarative model of the agent's environment. TR programs have also been experimented in the RoboCup competitions (Gubsich et al., 2008). In RAC we will interpret TR-procedures with TeleoR (Clark and Robinson, 2014), a typed higher-order major extension of Nilsson's Teleo-Reactive robotic agent language. We will also combine TeleoR with a flexibly-typed moded logic and functional language QuLog (Clark and Robinson, 2015) for representing the agent's belief store and making inferences.

On the other hand, several research groups are exploring the idea of robots that rely on cloud-computing infrastructures to access vast amounts of processing power and data. Some of the existing cloud robotics projects and initiatives are listed below:

- RoboEarth (Roboearth, 2013) is a European project led by the Eindhoven University of Technology, in the Netherlands, to develop a "World Wide Web for robots", a giant database where robots can share information about objects, environments and tasks;
- Researchers at Singapore's ASORO (A-Star Social Robotics Laboratory) have built a cloud-computing infrastructure that allows robots to generate 3D maps of their environments much faster than they could with their on board computers. It allows robots to perform simultaneous localization and mapping much efficiently than relying on their on-board computers;
- Google engineers developed Android-powered robot software that allows a smartphone to control robots based on platforms like LegoMindStorms, iRobot Created, and Vex Pro;
- Researchers at Laboratory of Analysis and Architecture of Systems in Toulouse, France, have created a user manual repository for everyday objects to help robots with manipulation tasks like opening a door. The idea is to develop a software framework where objects come with a "user-manual" for the robot to manipulate them.
- Yinong Chen et al., (2010), presented a research done on service oriented robotics computing and their design, implementation and evaluation of Robot as a Service (RaaS) unit. Their work was

sponsored by Microsoft, which demonstrates the interest that big companies are putting in these topics.

From the above list, RoboEarth is the initiative that has more in common with RAC. The platform obtained with RoboEarth allows robots connected to the Internet to directly access the powerful computational, storage, and communications infrastructure of modern data centres (Google, Facebook, Amazon, etc.) for robotics tasks and robot learning. To create a system for robots to communicate with each other, RoboEarth's researchers use a language that organizes information into environments, objects, and actions. The developed PaaS (Platform as a Service) solution allows robots to perform complex functions like mapping, navigation, or processing of human voice commands in the cloud, at a fraction of the time required using on-board computers.

Cloud robotics can be applied to any kind of robots, large or small, humanoid or not. Eventually, some of these robots could become more standardized and sharing applications would be easier. The "app paradigm" is one of the crucial factors behind the success of Apple and Google. Applications (apps) that are easy to develop, install, and use. This is the main characteristic that distinguishes RAC from existing approaches (like RoboEarth): the idea of providing an intuitive and easy to use notation for representing the desired behaviour of robots (the TR formalism) combined with a cloud computing infrastructure that allows to deploy complex missions into cheap devices (the micro-drones). This significantly will reduce the learning curve, requiring less training and effort to develop small-size missions (which perhaps represent the largest portion of cases). At the same time, the cloud serves not only as the common infrastructure where the missions are executed, but also provides a repository of missions where developers can take the benefit of sharing and reusing programs to control their own robots, obtaining in doing so the benefits of the mentioned app store philosophy.

4 SCENARIOS

As a demonstrator of the proposal, three scenarios have been defined. These scenarios will allow us to implement the final architecture shown in Figure 2 in a staggered and progressive way. The evolution of architecture in different scenarios is as follows:

- Scenario 1: this scenario implements the minimum part of the architecture required in order to run a first demonstration. The Infrastructure Layer features the local cloud infrastructure. The Platform Layer includes the TR interpreter and a basic interface to the selected drones. Finally, the Application Layer implements the TR repository and the facilities needed for loading, saving and executing concrete TR programs and configure drones. It should be highlighted that in this first scenario we only consider reusing previously defined TR programs that use a single agent.
- Scenario 2: Continuing with the architecture developed in Scenario 1, the Platform Layer will provide the possibility of using external Web services (useful for example for image processing). In the Application Layer we will add a graphical TR editor like Scratch in order to create and modify TR programs, and a module for 3D simulation. In this scenario several agents are incorporated, obtaining a more complex and mature case study.
- Scenario 3: Finally, in this scenario the proposed architecture is fully implemented.

Below the scenario 1 that has already been implemented is described.

4.1 Scenario 1 Description

For the scenario demonstration missions, a real environment replica has been deployed in a medium size area located in Cartagena (Spain). The mission considered the detection and tracking of persons in an emergency situation. Appendix A contains an excerpt of the TR rules developed in order to accomplish the mission requirements.

For this task, we have selected micro-drones. Micro-drones are a particular type of UAVs of great interest to test new technological approaches given their low cost, the large number of application fields, and the possibility to use the results obtained with them in other fields. We have selected the Ar. Drone 2.0 from Parrot, which has a good quality-price ration and includes two cameras (HD 720p frontal and QVGA vertical) and other sensors that assure a high stability during the flight and provides an API that allows us reading any sensor of the quadcopter as well as sending commands to the propellers, among other specifications. In addition, the well-known Robot Operating System (ROS) (Quigley et al., 2009) is compatible with this aerial platform thanks to the ardrone_autonomy ROS driver. ROS runs in a Raspberry Pi B+ installed on the drone.

This Single Board Computer (SBC) is connected to the Internet by using a 4G USB dongle. Therefore, data can be stored in the cloud and evaluated by the TR approach.

As figure 3 shows, the user carries out a passive visual beacon that could be detected by the HD frontal camera. This images as other sensor data are sent to the cloud by using the C++ application running in the SBC. This application provides several high level commands (UP, DOWN, FORWARD, among others) which are executed by the implemented TR algorithm.

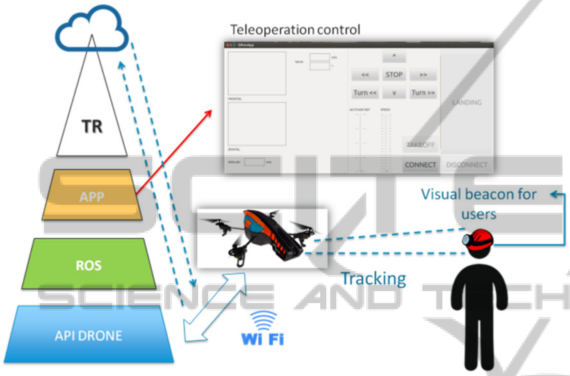

Figure 3: Scenario 1.

5 CONCLUSIONS

RAC aims to design, build, and validate a new generation of cloud robotic platforms that enable agent-based intelligent control of robots deployed in unknown and dynamic environments. The system will consider: (1) state-of-the-art resources to control physical robots; (2) state-of-the-art simulation techniques to define a 3D simulation environment which can be used to simulate TR mission execution; (3) state-of-the-art Artificial Intelligence and Knowledge Representation techniques to realize high-level robotic cognitive abilities; (4) novel techniques for programming reactive plans and robotic behaviours through missions and novel mechanisms for building new behaviours from existing ones, both for experienced and non-expert users; (5) novel multi-layered cloud platform as the infrastructure to maintain a continuous link between the robots acting on a physical environment and their agent counterparts, to provide sensor data from robots to agents, and to provide high-level autonomous decisions from agents to robots.

RAC seeks to develop robots with the capability to interact with humans and operate dynamically in a constantly changing robotic-mission environment. As the notion of robotics often refers to a very broad term, encompassing such diverse capabilities as low-level perception and action, network connectivity and decision making, the focus of RAC's effort is to close the innovation gap between by extending these different technologies in the civil robotics market and its associated stakeholders.

Among the expected benefits of the proposal, we want to highlight that we expect RAC to significantly improve the level of development of service robots; to exploit the results in civil markets; to provide a holistic robotic system development process, facilitating validation and deployment; to share knowledge and to harmonise the design of software for robots; to get a novel action planning technique to increase both flexibility and programming easiness of robotic missions; to improve upon the cognitive tasks of robots using goal-oriented reactive approaches, their autonomy and ability to react to changes in the environment; to leverage the use of knowledge representation and reasoning methods; and to get the resources for enabling a culture of robot programming for everyone, among others.

ACKNOWLEDGEMENTS

This work has been partially funded by the Spanish Ministry of Economy and Competitiveness under the CICYT Projects cDrone (ref.~TIN2013-45920-R) and ViSelTR (ref.~TIN2012-39279).

REFERENCES

Russell, S. J., Norvig, P., 2003. Artificial Intelligence: A Modern Approach, *Pearson Education*.

Thrun, S., Burgard, W., Fox, D., 2005. Probabilistic Robotics, MIT Press.

Murphy, R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., Erkmen, A. M., 2008. Search and Rescue Robotics, *Springer Handbook of Robotics*, pp. 1151-1173.

Kaminka, G. A., 2012. Autonomous Agents Research in Robotics: A Report from the Trenches, *2012 AAAI Spring Symposium: Designing Intelligent Robots*.

Dipsis, N., Stathis, v, 2010. EVATAR - A Prototyping Middleware Embodying Virtual Agents to Autonomous Robots. *Ambient Intelligence and Future Trends-International Symposium on Ambient Intelligence (ISAmI 201.)*. pp. 167-175.

Benson, S., Nilsson, N., 1995. Reacting, Planning and Learning in an Autonomous Agent, Machine Intelligence. *Machine Intelligence 14*, pp. 29-64.

Nilsson N. J., 2001. Teleo-reactive programs and the triple-tower architecture. *Electronic Transactions on Artificial Intelligence*, vol. 5, pp. 99-11.

Clark, K. L., Robinson, P. J., 2014. Programming Robotic Agents: A Multi-tasking Teleo-Reactive Approach, in preparation for publication, *Springer*.

Resnick, M., 2009. Scratch: Programming for All. *Communications of the ACM*, vol. 52, pp. 60-67.

EUROP, 2010. Robotic Visions to 2020 and beyond, The strategic research agenda for robotics in Europe. *IEEE*, vol. 17, pp. 15-16.

Nilsson, N. J., 1994. Teleo-reactive programs for agent control, *Journal of Artificial Intelligence Research*, vol. 1, pp. 139-158.

Gubisch, G., Steinbauer, G., Weiglhofer, M., Wotawa, F., 2008. A Teleo-Reactive architecture for fast, reactive and robust control of mobile robots. *New Frontiers in Applied Artificial Intelligence*, pp. 541-550.

Clark, K. L., Robinson, P. J., 2015. QuLog: Engineered for Agent Applications, in preparation for publication, *Springer*.

Roboearth, 2013. www.roboearth.org, funded by the European Union Seventh Framework Programme FP7/2007-2013.

Chen, Y., Du. Z., Garcia-Acosta, M., 2010. Robot as a Service in Cloud Computing. *2010 Fifth IEEE International Symposium on Service Oriented System Engineering*.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., & Ng, A. Y., 2009. ROS: an open-source Robot Operating System. *ICRA workshop on open source software*, vol. 3, pp. 5.