

Model of Integration and Management for Robotic Functional Components Inspired by the Human Neuroregulatory System

José Vicente Berná Martínez
Computer Science Department
University of Alicante
jvberna@dtic.ua.es

Francisco Maciá Pérez
Computer Science Department
University of Alicante
pmacia@dtic.ua.es

Abstract

This paper presents a model of integration and management for robotic functional components that make up the robotic control system. To that end, we use the human neuroregulatory system as the basis for the decomposition of tasks and actions behavior, and we rely on the SOA paradigm for the design of a distributed architecture that allows the viability of the system. This proposal will ensure a total decoupling between modules by promoting the reusability and features such as pattern-based design, while the system is fully distributed ensuring high flexibility, scalability, robustness and fault tolerance.

1. Introduction

Robotics has currently become one of the most active fields for researchers and enterprise and has wide repercussions for all society, changing our way of thinking about our daily lives. Examples of where the application of robots has captured the world's attention are nuclear accidents, finding shipwrecks, surgical interventions, exploring volcanoes and space travel. They have changed the way in which we build, how we remain safe, and how we produce and distribute energy and food throughout the world. Their high degree of involvement in society has made the diversity and variety of robots as great as the tasks to which they are put [1].

The design of a robot is inherently multidisciplinary and the design and implementation of its control software is one of the most important aspects [2]. There are a huge number of questions to be taken into account and in the majority of cases these are finally resolved by ad hoc solutions with different approaches. Nowadays, the use of biological systems as a source of inspiration has become one of the trends used for the simulation of behavior and to extract cooperation mechanisms and subsystems management [3]. These solutions are very efficient and robust, but the high degree of coupling between software and machine implies a great complexity in its maintenance or

evolution. Thus there are groups of researchers specializing in the creation of generic management architectures [4][5][6][7], trying to minimize the effort required in the design and development of the frameworks that support the control elements, so that more effort can be focussed on the implementation of the functionalities necessary for a robot to undertake the tasks for which it was conceived [8][9].

Therefore, this paper presents a proposal for the integration and management of functional elements that make up a robot. This proposal is based on the functionality and organization of human neuroregulatory system as a basis for designing a robot as a set of concurrent elements that attempt to influence the world around them. We propose to implement the use of techniques, tools, technologies and paradigms of ICT related to distributed systems and specifically the SOA paradigm (Service-Oriented Architecture) due to the nature of decoupling, flexibility, reusability and interoperability of heterogeneous components. We also propose to use Web Services to encapsulate the elements of control, because it is a widely published and mature technology that implements a natural way to SOA.

2. Modeling the Functional Components

Studies on the biological system can create realistic models capable of simulating and act as the neuroregulatory system, even in situations not covered [10][11][12]. These models are based on considering the system as a group of control centres that exert influence in an environment of action and reaction. The sum of all influences produce an emergent behavior similar to that expected. Each control center is fully autonomous, receiving data, processing them and sending their influence over other control centres. These models consider the biological system as a fully distributed system composed of various control centres, each responsible for carrying out specific tasks. One of the main advantages of this design is the decoupling between each control center, allowing you to modify, delete or add control centres changing interconnections. This makes the system highly

scalable and flexible, able to develop or incorporate new features, like the human neuroregulatory system. These models have proven useful not only in biological control problems but also in ICT environments [13]. Because the models based on the biological system have demonstrated its effectiveness in complex control problems, we propose to use them to model the system management and integration of functional components of a robot, because we consider that both types of systems are equivalent.

Table 1. Equivalences between the neuroregulatory system and de robot system.

Biological Control System	Robot Control System
Neuroregulatory Centre	Control Centre - cc
Biological-Mechanical system	Robot-Mechanical system
Neuronal connections	Connections between cc
Nerve impulses	Signals and variables

In general, a robot can be seen as a physical system that performs several tasks to achieve a global goal. We can consider the overall task as a composition of different tasks along with their dependencies. Using the following heuristic can do top-down analysis and break down a global behavior in the different tasks and their dependencies:

- 1.- *Specifying the desired behaviour in qualitative terms.* Describing the global objective of the system. From this description we extract considerations, constraints and action plans and we determine the control centres that compose the system. The emergent behavior of the system will respond to that description.
- 2.- *Specifying the behaviour in terms of actions.* It is the process of decomposing the behaviour into independent actions serving as sub-objectives of the global objective. Each of these actions is a system control center.
- 3.- *Description of actions based on the following parameters:*

- Action: what is the function performed.
- Memory: contextual information stored to provide the function.
- Input: what data needs to receive to perform its function.
- Data source: sources of the data required (other control center, a device, etc).
- Outputs: what information it produces. This information will be responsible for modifying other centres or actuators.
- Data receiver: who to send the output.

Using the model proposed in [10] and the information obtained in the above analysis, the functional elements of a robot can be viewed as the interaction between different control centres that perform non-composite tasks (figure 1). There will be

reactive control centres responsible for communicating with the physical elements, others who perform highly deliberative tasks such as trajectory planning, And between these two types exist a wide range of hybrid control centres. Each center carries out individual tasks and delivers its results to other centres or to physical components. The model allows to characterize each of these elements separately but integrated into a single system according to the relationships defined in the analysis.

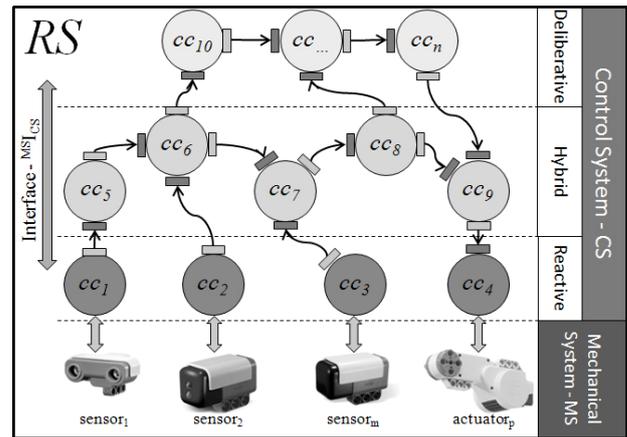


Figure 1. Proposed model.

Each of the control centres will encapsulate the application logic that develops the specific control task such as PID control for wheels, positioning, calculating trajectories, security control, etc.

Since the behavior are very complex, the decomposition process can be performed iteratively, breaking down each behavior into subtasks until we get the basic actions. This process is repeated as many times as necessary.

At the same time, this process of decomposition can be done in parallel, We just need to define the interfaces and respect throughout the process of decomposition. Finally, each group can integrate the subsystem into the global model.

3. Software Architecture

Due to the distributed nature of the resulting system, SOA paradigm is selected to provide infrastructure for integration and management model. This paradigm provides features like architectural composability, capable of improving QoS, based on open standards, vendor diversity, intrinsic interoperability, discoverability, federation, inherent reusability, extensibility, layers of abstraction, loose coupling and organizational agility [14]. The intrinsic characteristics of SOA allows us to divide the control centres in different layers, placing control centres responsible for communication with sensors and actuators in the layer closest to the physical components. In this way we can

integrate or replace heterogeneous devices within the system. Moreover, the ability of federation allows the integration of legacy systems or other external systems that can help us carry out tasks.

Each of the control centres will be encapsulated in a service able to offer its function to any other service that requires. This ensures reusability because services can be fully public and accessible. To retain their independence, services encapsulate logic within a distinct context. This context can be specific to a task or some other logical grouping. The concern addressed by a service can be small or large. Therefore, the size and scope of the logic represented by the service can vary. Further, service logic can encompass the logic provided by other services. In this case, one or more services are composed into a collective.

The dynamic nature of SOA systems also allows us great agility and flexibility of the system, since we can include, replace, modify or duplicate those services as needed, allowing for example to alter partial behavior or even full system behavior. This gives us the flexibility to evolve the control system by adapting or implementing increasingly complex functions without altering the existing elements.

We use a classic architectural pattern to implement de SOA system, the marchmaker style. This architecture is the most basic. The intermediate element or broker is only responsible for the registration of services. Thus the services can be located and consumed by other services. All subsequent communication is between consumer and supplier. The matchmaker style define the role of stakeholders, the service requestor, the service provider and the service registry.

We selected web services technology for the implementation of the integration and management system because Web services have collaborated in some of the recent advances SOA [14]. Contemporary SOA is intrinsically reliant on Web services so much so that Web services concepts and technology used to actualize service-orientation have influenced and contributed to a number of the common SOA characteristics. In addition, web services now have a large number of tools and a large support community. In our case, Web services allow us to distribute the system over any network including the internet This will allow us to give notice to each service on the appropriate remedy.

Using Web Services, the matchmaker patter have these technologies:

- WSDL described the service
- SOAP provided the messaging format used by the service and its requestor
- UDDI provided the standardized service registry format.

4. Case Study

We used a Lego robot as a base for our proposal, the robot was composed of several motors and sensors. The robot moves in an unstructured environment. That is, there are no marker or localisation elements specially designed for the robot's movement. The behaviour we wished to implement was the movement of the robot through the environment, from the point where it started to another point located at a distance and a certain direction avoiding obstacles.

In order to define the systems, we used the heuristic proposed in Section 2 as a base, defining 3 levels of detail to arrive at the specific actions:

Level 1: given a specific point where the robot currently is located, initially (0,0), it must move to a destination point (A,B) avoiding possible obstacles that it might find on its route.

Level 2: (1) be capable of communicating with the hardware (to receive sensor values or to communicate values to motors), (2) know the current position as a function of the movements performed, (3) calculate the distance and the direction necessary to reach the objective, (4) perform the movement of the robot, (5) detect obstacles at the front, (6) modify the route of (4) to avoid obstacles detected in (5).

Level 3: the table 2 describes in each module; what communication takes place, module of origin, module of destination and signals transmitted.

The Table 2 resumes the different centres, signals and functions of each of them.

Table 2. Parameters and dependences.

cc	S	R	Signal
1	4	1	(V1,V2) motor speed
	1	2	(a1, a2, ..., an) motor and sensor values
	1	5	(a1, a2, ..., an) motor and sensor data
2	1	2	(a1, a2, ..., an) motor and sensor data
	2	3	(x,y) current position
	2	4	(α) current direction
3	2	3	(x,y) current position
	3	6	(D,δ) distance to the objective and direction to follow
4	2	4	(α) current direction
	6	4	(D,δ) target distance and direction to follow
5	1	5	(a1, a2, ..., an) . motor and sensor data
	5	6	(β,γ) angles between which there are obstacles
6	3	6	(D, δ) distance to the objective and direction to follow
	5	6	(β,γ) angles between which there are obstacles
	6	4	(D, δ) distance to the objective and direction to follow

cc -> Control Center S -> Data Source R -> Data Receiver

Graphically, the system and connections is show en figure 2.

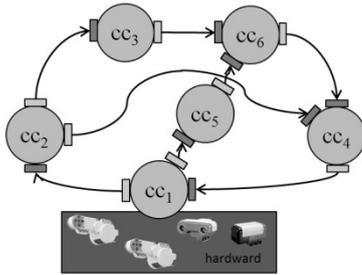


Figure 2. Logical architecture of control centres.

5. Test and Validation

We have implemented a web service that encapsulates the logic of each control center. The specific algorithm used for each task is a secondary objective in this paper. All services have been connected as shown in Table 2. The result is a system that integrates several elements of control. Each separate element does nothing but all the elements acting simultaneously achieve the global goal, as shown in Figure 3. All components run concurrently and can be deployed in any resource that supports Web services.

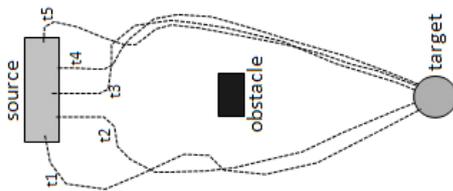


Figure 3. Test of trajectories.

6. Conclusions

The proposal presented in this paper allow us create a integration and management system that preserve the best software characteristics like modularity, decentralization, standards, security, decoupling modules, reusability, federation, etc. In this system we can put any trajectory algorithm, any vision pattern match or any other function because every function or control element is encapsulate inside a web services. The web service hidden the application logic and allow us to put them in the system easily. The proposal allow us create a management and integration system that fully decoupled each elements and underlying physical components.

Currently we are working on two aspects. Firstly, on the construction of control centres that implement more functionalities and so enable more complex emergent behaviours (controlling steps and limits, constructing maps and modifying trajectories in order not to repeat routes in search of the objective). Secondly, on the designing a set of robots and scenarios that allow us to

compare the proposed system to other systems such as CoRoBa [2], ORCA [7] o SC-Agent [6].

References

- [1] G. Bekey, R. Ambrose, V. Kumar, D. Lavery, A. Sanderson, B. Wilcox, J. Yuh, Y. Zheng. *Robotics. State of the art and future challenges*. Imperial College Press. 2008.
- [2] Colon, E., Sahli, H., Baudoin, Y. "CoRoBa, A multi mobile robot control and simulation framework". *International Journal of Advanced Robotic Systems* 3 (1), 73–78. 2006.
- [3] R.A. Brooks. *Flesh and Machines*. Vintage . 2003.
- [4] A. Sayouti, F. Qrichi Aniba, H. Medromi, A. Lebbat. "Remote control over Internet of robotic systems based on multi agents systems". *RCIS 2008*, Marrakech, Morocco, June 3-6, 2008.
- [5] C. Novales, G. Mourioux, G. Poisson. "A multi-level architecture controlling robots from autonomy to teleoperation". *Proc. In First National Workshop on Control Architectures of Robotics*. Montpellier. 2006.
- [6] J.L. Posadas, J.L. Poza, J.E. Simó, G. Benet, F. Blanes. "Agent-based distributed architecture for mobile robot control". *Engineering Applications of Artificial Intelligence*. Vol. 21, pp. 805-823. 2008.
- [7] A. Brooks, T. Kaupp, A. Makarenko, S.B. Williams, A. Oreback. "Orca: a component model and repository". *Software Engineering for Experimental Robotics*. Springer, Berlin, pp. 231–251. 2007.
- [8] J.S. Albus. "A reference model architecture for intelligent unmanned ground vehicles". *16th Annual International Symposium on Aerospace/Defense Sensing*. 2002.
- [9] R. Volpe, I. Nesnas, T. Estlin, D. Mutz, R. Petras, H. Das. "CLARAty: coupled layer architecture for robotic autonomy". *Jet Propulsion Laboratory*, Technical Report. 2000.
- [10] A. Soriano, J.M. García, F. Maciá. "An Agent Based Framework for Modelling Neuronal Regulators of the Biological Systems". *Lecture Notes in Computer Science*. Vol: 2972. 2004.
- [11] F. Maciá, J.M. García, A. Soriano, D. Ruiz, "A robust model of the neuronal regulator of the lower urinary tract based on artificial neural networks". *Elsevier*. Vol.: 71. Num.: Nos 4-6. Pps: 743-754. 2008.
- [12] D. Ruiz, J.M. García, F. Maciá, A. Soriano. "Modeling the Distributed Control of the Lower Urinary Tract Using a Multiagent System". *Lecture Notes in Artificial Intelligence*. 2004. Pp. 104-114.
- [13] F. Maciá, J.M. García. "Mobile Agent System Framework Suitable for Scalable Networks". *Kybernetes. The International Journal of Systems and Cybernetics*. Vol. 35-5, 2006.
- [14] T. Erl. *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*. Prentice Hall. 2005.