Suitable task allocation in intelligent systems for assistive environments

Manuel Vinagre, Joan Aranda and Alicia Casals

Universitat Politècnica de Catalunya. BarcelonaTECH
{manuel.vinagre,joan.aranda,alicia.casals}@upc.edu

Abstract. The growing need of technological assistance to provide support to people with special needs demands for systems more and more efficient and with better performances. With this aim, this work tries to advance in a multirobot platform that allows the coordinated control of different agents and other elements in the environment to achieve an autonomous behavior based on the user's needs or will. Therefore, this environment is structured according to the potentiality of each agent and elements of this environment and of the dynamic context, to generate the adequate actuation plans and the coordination of their execution.

Keywords: System architectures, Cognitive Systems, Tasks manipulation, semantic nets.

1 Introduction

The increasing demand of assistive services due to the continuous aging population carries with it a deficit on the availability of human resources able to cope with these needs in the future. Assistive technologies are thus the potential solution to these growing requirements of human welfare in an ageing population. By facilitating the different assistive services to attend daily life tasks, this technology is expected to increase the user's autonomy and their own self-esteem. Looking at what technology offers today, one can observe that technology in this area is quite diverse, ranging from simple devices that help in communicating and in mobility, those which are currently available, to more intelligent devices, as robotic systems capable of perceiving certain environment conditions and dynamically adapt to their changes.

These more advanced systems differ in the way of managing the perceived information. While reactive systems respond to stimulus or orders in an immediate manner, deliberative systems perform an abstraction of the perceived information and do some reasoning to provide a behavior adapted to every situation. Being those latter systems able to achieve capacities closer to those of humans, they become of interest in assistive systems since they allow increasing usability and efficiency.

When dealing with daily live activities, gesture and activity recognition aiming to interpret the user's will, or their needs, allows the generation of robot actuation strategies in a proactive way [1] [2]. These proactive strategies may even require the interaction of multiple robots, when more than one arm is needed to accomplish a given task

[3]. At present, many applications require the generation of intelligent environments from the combination and coordination of different technologies. Industry 4.0 is an example of intelligent environment, which goal is the increase of automation, flexibility and scalability of industrial settings. The advantages of this technology that provides companies with the ability to quickly adapt to production changes are identified in [4], emerging from here the concept of smart retrofitting. Other environments that are experimenting significant changes are those related to surgery, which lead to the concept of surgery 4.0. These new advances in surgical innovation, constitute a step forward in the progressive implantation of Minimally Invasive Surgery (MIS). As explained in [5], the key factor is the intelligent collaboration between assistant personnel, surgeons, and assistance and autonomous systems in the Operating Room. The aim is to support decision making in surgery, contextual assistance and surgical training. When referring to daily life, the concept of Healthcare 4.0 appears too. In this context the efforts focus on the implantation of intelligent services to support daily life therapies. In [6], an open code system is presented aimed at generating an intelligent environment able to monitor the patient from diverse sensors, analyze the obtained data and manage the actions to perform in an intelligent manner through a complex events processor.

The increasing availability of intelligent systems and robots at home will make them part of our daily life. Then, it will be necessary an adequate interconnexion and coordination of all these systems to generate an adaptable and useful ecosystem. Thus, the aim of the work is the development of methodologies that allow this coordination through the design of a platform composed of intelligent agents, in a domestic environment.

2 Intelligent environments

The way to generate intelligent environments in any of the above ambits can follow different methodologies. Those more frequently used are the implementation of cyber-physic systems, multirobot environments and heterogeneous environments, which combine the previous ones.

2.1 Cyber-physic systems

Most intelligent environments are composed of cybernetic means that process information in the digital space by means of physical devices, which by means of sensors and actuators acquire information and act on the real world. The so-called Cyber-Physic Systems (CPS) integrate three fundamental parts: computation, communication and physical control. They allow building environments that can exhibit multiple actuation modes in function of the context through digital or virtual information and the real world.

Internet of Things (IoT) is a kind of CPS where a connection is created between objects and humans, processing information in real time. An example of an intelligent assistive environment for elder is that presented in [7], which provides a fluid interaction between the user and the environment and the care givers.

Although CPS allow a wide range of services, these services are mainly oriented to the connection and communication through internet. In most IoT platforms, the coordination of devices to generate services is pre-programmed, and the links between inputs and outputs are stablished to generate some given behaviors.

2.2 Multirobot systems

A Multirobot system (MRS) can be characterized as a set of robots operating in the same environment, having been widely applied in different domains [8]. An MRS can improve the effectivity of a robotic system, providing better performance in the execution of some tasks, more robustness and reliability, thanks to its modularity. This happens not only when the robots have different functions, but also when they all have the same capacities. However, most frequently robots are heterogeneous to exploit their respective characteristics according to the tasks and thus, obtain more efficient solutions, although this implies complex coordination strategies.

In this work, much attention is paid to MRS developments that operate in dynamic environments, where uncertainness and unforeseen changes can occur due to the presence of other external agents (for instance the presence of humans). Even considering uniquely the subset of MRS in assistive robotic applications, it is not easy to identify a common frame for comparing the technical solutions known up to now [9]. This is a common phenomenon that occurs in other application domains, even easier to specify, as shown for instance with the teams of Robot-Soccer in the RoboCup competition.

An MRS cannot be considered simply as the generalization of a single robot system, and the proposed approaches must be characterized in terms of how suppositions over the environment are established and how the system is internally organized [10].

Among the most common problems of multi-robot systems, the focus of this work is the design of new methods of tasks assignment, selecting the minimum number of robots most suitable to perform a certain task [11].

2.3 Heterogeneous systems

The combination of cyber-physical and robotic systems in an environment generates a great added value. On the one hand, multi-robot systems can benefit from accessing the resources provided by a CPS system, such as information from the environment and the user. This allows robots to have more information to elaborate their decision processes and more capacity of acting on the environment. For instance, if the CPS contains a set of cameras in the scene and a software component that detects objects in it, then the robot can use this information to locate a certain object that cannot be easily found through its perception system. In another situation, if a given actuation requires that the robot must access inside a refrigerator with automatic opening, the robot can take advantage of this resource to access its content in a simpler way. On the other hand, the CPS system benefits from the resources provided by the robot. For instance, the CPS system can acquire not available information from the robot scanning actions to areas inaccessible by the CPS.

In the interrelation process between CPS and robotic systems different aspects must be considered: communications protocol, interoperability method, information sharing, modeling and capacity coordination, among others. One of the fields that stands out in the development aid of this type of systems is robotics in the cloud. In this field cloud-based technologies are investigated in order to give the robots access to digital resources. The most relevant advantages of this type of systems is the access to computing resources as servers for data analysis or big data and flexible data storage resources. Now, a new paradigm pursues the convergence of cyber-physical and robotic systems, the Internet of Robotic Things (IoRT). This paradigm is based on three fundamentals fields: IoT, robotics and cloud computing.

3 Architecture design for intelligent environments

To assist people efficiently and user-friendly, an intelligent environment architecture has been designed based on different devices, which can be sensors, actuators, applications or robots (Fig. 1). This platform has been implemented and tested within the project AURORA, with the aim to create services in a dynamic way in healthcare settings.

This platform offers different features. The first is the connectivity and management of devices in the environment. The second is the modeling of devices from a general model that contains their relevant information as their capabilities. Third, a modeling of the environment describes its state which is used as context information. There is also a dynamic service generator that depends on the capabilities of available devices and on the context. Then, the assistive services available at a certain time are presented to the user through an interactive interface. Once the user demands for a specific service, or a pro-active service is offered and accepted, the platform controls and monitors its correct execution. The platform is composed of three main modules: the module of knowledge management, the module of devices' abstraction and the module of services.

The knowledge management module is responsible for maintaining all the information about the environment status and the relationships and rules between the different elements. This information is stored in classes and instances in ontological models using OWL (*ontology web language*). For this module, the knowledge processing system developed in the KnowRob project is used [12]. It provides mechanisms of storage and recovery of actions, objects, processes, events, properties and relations. The knowledge of agents' capacity is established as a property owned by actors, which represents the possibility to perform a certain task. Capabilities are structured hierarchically from two basic abilities: information and operation. At functional level, this module is responsible for the knowledge of the 3D model of the different objects and the dynamic model of the environment.

The device abstraction module manages all existing devices in two layers: an interconnection and a modeling layer. The first layer is responsible for handling communications between all devices using ROS middleware (Robotic Operating System [13]. To build a heterogeneous system, a communication from ROS to the Open-Hab platform is used. Open-Hab is a platform for the interconnection of smart devices (IoT).

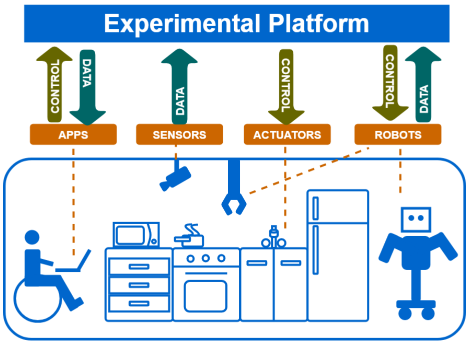


Fig. 1. Experimental platform

Above this layer, the device modeling layer contains a definition of the types of devices and their capabilities. These definitions are managed considering the information provided by the knowledge management module. At functional level, this module allows the interconnection of the actors: control agents, interface agents and recognition and location agents.

The service module handles the definition, updating, progress and achievement of services. Contextual information is managed in a similar way to [14], with the adaptation of the concept of skill as a quantifiable unit of a service. The services are then the result of the planning through the composition of different possible skills depending on the state of the environment and the capabilities of the actors. These skills have as objective to fulfill a service and are evaluated by a factor of competence of the skill. This factor is calculated by the evaluation of each skill and the available capability. These evaluations are calculated under the current context conditions.

We distinguish between three different types of evaluation functions. First, those related to actions that modify the environment. For example, there is an evaluation on the manipulation of objects depending on its shape, performed action and environment status. Second, functions for evaluating actions addressed to the user, mainly skills and abilities of person-robot interaction. And third, those that assess the acquisition of the information about the state of the environment and the user, as well as its interpretation.

4 Implementation

4.1 Hardware

An experimental platform has been implemented to validate the architecture. This platform emulates a kitchen living environment with the automation of different devices. A table surface with an induction cooker, an automated water tap of a sink, a cabinet and a refrigerator equipped with a motorized system for opening/closing doors. Two 3D cameras cover the whole environment and a user interface allows the interaction with the environment from an interactive graphical interface to select elements of the environment and trigger the actions necessary to achieve the proposed objectives.

To support actions over the environment, the system counts on three robots with very different and complementary features: a Cartesian robot hanging on the ceiling (CAPDI), a second double-handed robot for complex manipulation operations (BAXTER) and a single robot arm over the table surface (MICO).

The implementation of the proposed control strategy is summarized in Fig. 2. This figure shows the agents currently incorporated and in attenuated mode the possibility to add new agents.

4.2 Multi-robot task allocation system

In the setup considered in this work, the team of robots have different architectures. They coordinately assist a disabled user performing daily tasks in a kitchen. Several task levels can be considered. The first level is constituted by the high-level tasks, or services, such as "making a coffee" or "pouring a glass of water". These types of actions are those that will most likely be triggered by the user through a user-machine interface. These services or goals are compound or even complex tasks which require the execution of various sub-tasks. For instance, "pouring a glass of water" will require (1) to fetch an empty glass, (2) to grasp a bottle of water, (3) to pour water from the bottle into the glass. These lower level tasks can themselves be decomposed into sub-subtasks such as "picking the glass from its location", "placing the glass at a given location"...

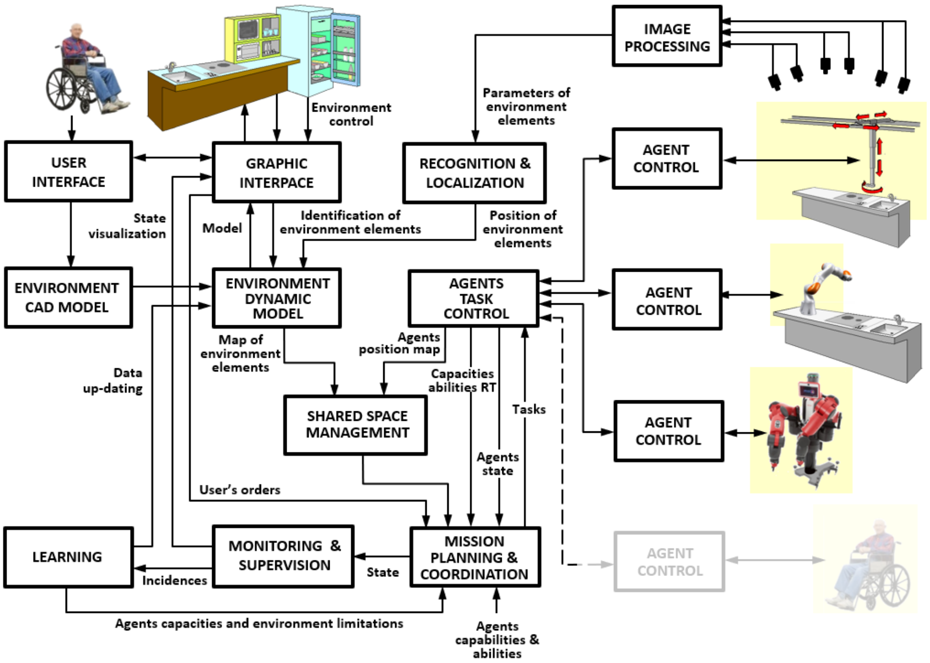


Fig. 2. System architecture showing the main physical components, the agents, and the modules

The services are decomposed into a *precedence graph* of simple or elementary tasks by means of preprogramed receipts considering some ordering constraints. These tasks must be allocated to the convenient agent at any time, permitting the services to overlap in time, by starting one service even before others finish. Every task will be assigned considering three constraints: capability, capacity and precedence.

The Multi-Robot task allocation problem can be stated as:

Finding an optimal allocation A of a set of tasks T to a subset of robots R, that will be in charge of carrying it out: A: T → R

The implemented solution is based on an iterated multi-robot auction process inspired in the work by [15] conveniently modified for our precedence-constrained task scheduling. Despite certain drawbacks, auction-based methods are indeed the most appropriate to work with heterogeneous teams, since they are the most suited to consider the different capabilities [16]. The optimization objective will be to minimize the make span, i.e. the total schedule duration.

In the proposed solution, the tasks are scheduled in batches. It is assumed that in each batch, the tasks are pairwise independent. This enables to schedule every task of a batch without considering the scheduling of the other tasks in the same batch. Batch selection is performed by the auctioneer, which takes as input the *precedence graph* defined by receipts. At each iteration, it selects the tasks that have either no predecessor or which predecessors have all been scheduled.

Once the batch of tasks to auction has been selected, it is broadcasted to all bidders. To take into account heterogeneity, McIntire's solution is modified by adding the following step: before computing the bid for a task, the robot evaluates if it can be done with respect to its ***capabilities***. Each robot then computes a bid for each feasible task in the batch and for each possible position in its current schedule. The best result of each robot is sent to the auctioneer, which selects the best "offer" and communicates the winner to the robots. The winner updates its schedule and all participants remove the assigned task from the tasks to be auctioned. The process is repeated until all the tasks from the auctioned batch have been scheduled.

The task schedules generated in this way are represented using Simple Temporal Networks (STN), introduced in [17]. STNs are data structures similar to graphs, in which nodes symbolize events and edges symbolize timing constraints on the events.

Notice that in this stage the system follows the *decompose-then-allocate* paradigm, in which the services are decomposed into tasks that are then allocated optimizing duration under capability, capacity and precedence criteria. However, before task execution, the *allocate-then-decompose* approach will be used, in which the tasks assigned to robots can be decomposed locally using a spatial 3D suitability function.

4.3 Supervising and optimizing robot capabilities

Once all tasks have been assigned to each agent, an assessment of the best execution by means of the degree of adequacy of the capabilities of each agent is calculated over the workspace. For that, there is a common data structure of the workspace for general coordination and management. This data is a 3D map represented by the voxelization of the workspace volume and a given value for each voxel (workspace map). One is the occupancy map that provides the occupancy of the workspace that includes static elements loaded from the pre-known CAD of the environment and dynamic elements that are updated in real-time from the processing of the images provided by the different

After the task allocation has been stablished, the implemented evaluation of capacities and abilities aims to be spatially evaluated on each action. To achieve this objective, a spatial three-dimensional suitability function is used, calculating for each free agent a suitability value in each voxel of the workspace map using Eq. 1.

$$\Phi_a(x,y,z) = \frac{1}{n}\sum_{i=1}^{n} p_i(x,y,z) \tag{1}$$

where a is the evaluated agent, and p_i a function based on the capabilities of each agent given by quantitative and qualitative parameters. Those quantitative that have been considered initially for the validation of the proposed strategy are shown in table 1.

$p_1 = \frac{(P_{max} - P_r)}{P_{max}}$	**Load margin around workspace**. P_{max} is the load capacity of robot and P_r is the load of the object to be manipulated.
$\mathrm{p}_2 = 1 - \frac{(\mathrm{T_i} - \mathrm{T_{min}})}{\mathrm{T_i}}$	**Execution speed factor.** Measured from the origin of the trajectory to each intermediate point of the trajectory.
$p_3 = \left(1 - \frac{1}{d_{\mathrm{min}} + 1}\right)$	**Accessibility to the operating point**. Where $\mathrm{d_{min}}$ is the distance to a singularity of the articulation, which represents a minimum among all of them.
$\mathrm{p}_4 = \frac{(\mathrm{h_{max}} - \mathrm{h_r})}{\mathrm{P_{max}}}$	**Grasp capacity**, where $\mathrm{h_{max}}$ is the opening value of the end effector, and $\mathrm{h_r}$ the opening resulting from prehension

Table 1. Quantitative parameters

The qualitative parameters introduced as indicators of the quality of the task performed by each agent, in this case robotic arms, are shown in table 2.

$\mathrm{p}_5 = 1 - \frac{\mathrm{Vpp_{max}}}{\mathrm{Vpp}}$	**Smoothness of task execution.** The value Vpp, is the mean value of the peak values of the frequency signal generated by an accelerometer located at the end effector of each agent, in each path. $\mathrm{Vpp_{max}}$ is the maximum value from the signal along the entire trajectory.
$\mathrm{p}_6 = 1 - \frac{\Gamma_{\mathrm{max}}}{\Gamma_{\mathrm{r}}}$	**Ease of manipulation** throughout the completion of each task, in which $\Gamma_{\{r\}}$ is the pair made by the last articulation of the kinematic chain of each robotic arm, and $\Gamma_{\{max\}}$ the maximum admissible torque.

Table 2. Qualitative parameters

The action will switch to the free agent with the maximum suitability in the workspace map from a predefined trajectory of the actual action. Since performing this process can lead to excessive segmentation and transfer of tasks between agents, a hysteresis factor is introduced in the allocation change algorithm.

5 Performance evaluation

The performance of the system architecture has been tested with a given problem in two simulated scenarios, similar to the real one. These simulations involve modifications and physical constraints with the objective of testing the system performance.

The conditions of the first simulation are slightly different from the implemented setup. We consider the case where the kitchen is equipped with one CAPDI robot and three, instead of one, MICO arms. This enables to test more demanding scenarios, as well as to evaluate the possibility of parallel task execution, which is rather limited in the real Aurora setup. In the rest of this subsection, the three MICOs will be identified as "M0", "M1" and "M2". This arrangement, which can be seen in Fig. 3 enables to reach both cupboards to store objects and to create parallelism.

In order to evaluate the interest of having two MICOs on the left side of the table, the second simulation is carried out on the same scenario but without M1.

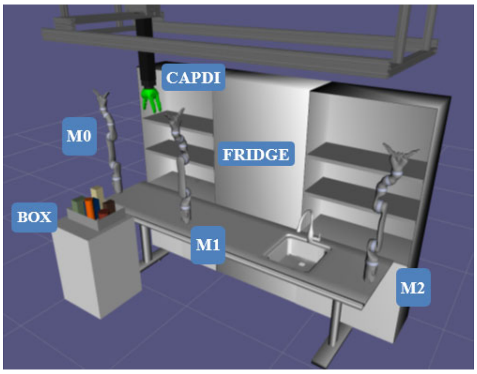


Fig. 3. Evaluation scenario

Given both simulations, the following problem is presented to the system: food items from a box must be unpacked, and each item must be stored to its place. The robots must store six items: Sugar (green box) and coffee (brown box), to be stored on the shelves of the left cupboard. A tomato sauce can (in red) and pasta (in blue), to be placed on the shelves of the right cupboard. A carton of milk (in pale yellow) and a bottle of juice (in orange), that must be stored in the fridge (see fig. 4).

Some of the tasks can be performed by several agents, while others can only be achieved by one of the team members, depending on the agent's capabilities and reachable workspace. The precedence graph is represented in Fig. 4. Each task is represented by a circle divided into two: in the upper half is the action type, and in the lower the associated object. The global precedence graph is composed of six disconnected branches which can operate in parallel. The bold number at the left of each circle indicates the task index. For every task its duration is known (sec.), the object, initial and final point (if needed), and other internal parameters needed for task execution (table 3).

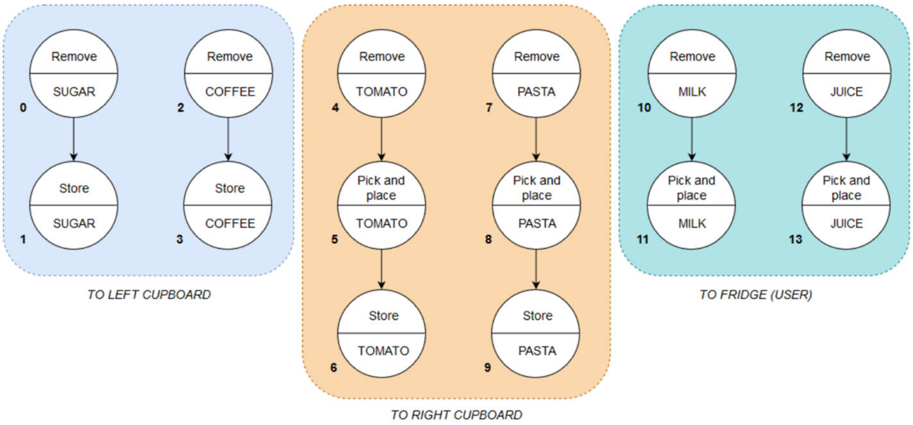


Fig. 4. Task precedence graph

Id	Type	duration(s)	object	Surface	
0	8	30	sugar	Table	
1	0	30	sugar	lower left cupboard	
2	8	30	coffee	table	
3	0	30	coffee	middle left cupboard	
4	8	30	tomato	Table	
5	0	45	tomato	Table	
6	0	30	tomato	middle right cupboard	
7	8	30	pasta	Table	
8	0	45	pasta	Table	
9	0	30	pasta	middle right cupboard	
10	8	30	milk	fridge	
11	0	30	milk	fridge	
12	8	30	juice	fridge	
13	0	30	juice	fridge	

Table 3. Task list definition

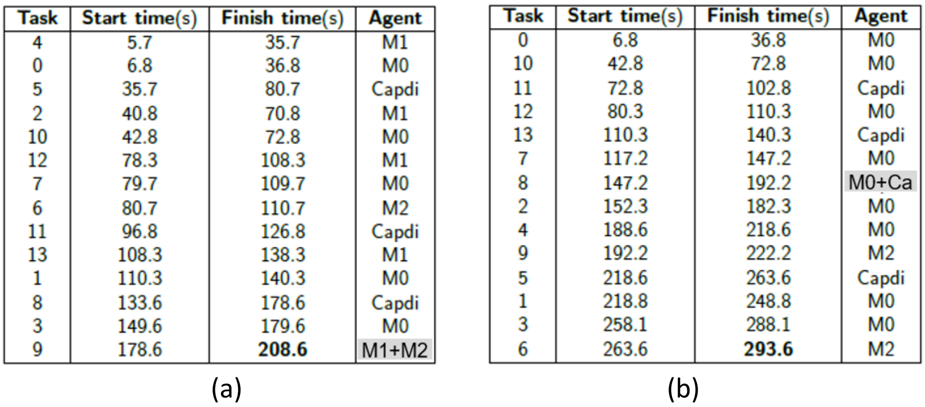
The output schedules obtained for the proposed problem and for each simulation are shown in table 4. The first column of each table presents the task ids in ascending order with respect to their start times. For each task, its start and finish times are specified along with the agent to which it has been assigned. The make span is indicated in bold; for all cases, it corresponds to the finishing time of the last task to be started.

This result obviously presents a different task distribution and task sequence. In the first simulation, CAPDI has only three tasks to complete, those of bringing the pasta and tomato sauce to the other side of the table and that of bringing the milk to the fridge. In the second case, CAPDI is also in charge of bringing the juice to the fridge, while M1 is not present so all tasks must be assigned to other robots. M0 oversees the rest of tasks previously assigned to M1, since M2 performs the same tasks in both cases.

Comparing the two variants, results are as expected: the make span is 41% longer with one robot less. This is due to the fact, visible in the task distribution of the first variant, that M0 and M1 work in parallel most of the time.

The improvement of the quality has been applied in both simulations. From the variant of four robots, task 9 was assigned only to robot M2, but it has been replaced by a shared task between robot M1 (free) and M2. The position of the pasta placed by CAPDI in task 8 gives a suitability of $\Phi_{M1} = 0.8$ and $\Phi_{M2} = 0.6$ mainly due to a better accessibility and easiest manipulation in the workspace. When M1 carries pasta to right direction at the middle between both robots, M1 places pasta and it is picked by M2 because it has a major suitability mainly due to zero accessibility of M1 to reach the right cupboard.

In the second simulation, task 8 firstly allocated to CAPDI it's finally shared between robot M0 (free) and CAPDI. This is due to the less suitability of CAPDI to reach the pasta in the position placed after it has been removed from the box. M0 has a better execution speed and smoothness for the task, so, it translates pasta towards the center of the kitchen until $\Phi_{CAPDI} > \Phi_{M0}$.



Task	Start time(s)	Finish time(s)	Agent
4	5.7	35.7	M1
0	6.8	36.8	M0
5	35.7	80.7	Capdi
2	40.8	70.8	M1
10	42.8	72.8	M0
12	78.3	108.3	M1
7	79.7	109.7	M0
6	80.7	110.7	M2
11	96.8	126.8	Capdi
13	108.3	138.3	M1
1	110.3	140.3	M0
8	133.6	178.6	Capdi
3	149.6	179.6	M0
9	178.6	**208.6**	M1+M2

(a)

Task	Start time(s)	Finish time(s)	Agent
0	6.8	36.8	M0
10	42.8	72.8	M0
11	72.8	102.8	Capdi
12	80.3	110.3	M0
13	110.3	140.3	Capdi
7	117.2	147.2	M0
8	147.2	192.2	M0+Ca
2	152.3	182.3	M0
4	188.6	218.6	M0
9	192.2	222.2	M2
5	218.6	263.6	Capdi
1	218.8	248.8	M0
3	258.1	288.1	M0
6	263.6	**293.6**	M2

(b)

Table 4 a) Output schedule for simulation 1 and b) Output schedule for simulation 2

Conclusion

When dealing with different kind of agents that collaboratively should perform a task, an adequate management of all the resources considering the capabilities and abilities of the agents, the state of the elements in the scene and the sequence of actions should be considered. This work has been addressed to formalize a methodology that allows considering all the involved elements to stablish the sequence of actions and the allocation of the robot assigned to each action. Then, the methodology has been evaluated on slightly modified scenarios of a robotized kitchen.

Acknowledgments. The work has been developed in the frame of Project RTC-2015-3926-1, from MINECO and with Feder funds.

References

1. Vinagre, M., Aranda, J., Casals, A., Jan 2014. An interactive robotic system for human assistance in domestic environments. Lecture notes in computer science 8548, 152–155.
2. Vinagre, M., Aranda, J., Casals, A., Jan 2015. A new relational geometricfeature for human action recognition. En: Informatics in Control, Au-tomation and Robotics: 10th International Conference, ICINCO 2013:Reykjav´ık, Iceland, July 29-31, 2013. Springer, pp. 263–278.
3. Aranda, J., Vinagre, M., Aug 2016. Anticipating human activities from object interaction cues. En: IEEE International Symposium on Robot and Human Interactive Communication. Institute of Electrical and Electro-nics Engineers (IEEE), pp. 58–63.
4. Guerreiro, B. V., Lins, R. G., Sun, J., Schmitt, R., 2018. Definition of smartretrofitting: First steps for a company to deploy aspects of industry 4.0,161–170.
5. Maier-Hein, L. et al, 2017. Surgical data science: Enabling next-generation surgery. CoRR abs/1701.06482.
6. Gonzalez-Usach, R., Collado, V., Esteve, M., Palau, C. E., May 2017. Aalopen source system for the monitoring and intelligent control of nursinghomes. En: 2017 IEEE 14th International Conference on Networking,Sensing and Control (ICNSC). pp. 84–89.
7. Dohr, A., Modre-Opsrian, R., Drobics, M., Hayn, D., Schreier, G., April2010. The internet of things for ambient assisted living. En: 2010 Seventh International Conference on Information Technology: New Gene-rations. pp. 804–809.
8. Yasuda, T., 2011. Multi-Robot Systems: Trends and Development. InTe-chOpen.Zezulka, F., Marcon, P., Vesely, I., Sajdl, O., 2016. Industry 4.0 – an intro-duction in the phenomenon. IFAC-PapersOnLine 49 (25), 8 – 12, 14thIFAC Conference on Programmable Devices and Embedded SystemsPDES 2016.
9. Das, G. P., McGinnity, T. M., Coleman, S. A., Behera, L., Oct 2015. A dis-tributed task allocation algorithm for a multi-robot system in healthcare facilities. Journal of Intelligent & Robotic Systems 80 (1), 33–58.
10. Farinelli, A., Iocchi, L., Nardi, D., October 2004. Multi robot systems: A classification focused on coordination. IEEE Transactions on System Man and Cybernetics, part B 34 (5), 2015–2028, new York, (USA).
11. Hernansanz, A., Casals, A., Amat, J., Jun. 2015. A multi-robot cooperationstrategy for dexterous task oriented teleoperation. Robot. Auton. Syst.68 (C), 156–172.
12. Tenorth, M., Beetz, M., April 2013. KnowRob – A Knowledge ProcessingInfrastructure for Cognition-enabled Robots. Part 1: The KnowRob Sys-tem. International Journal of Robotics Research (IJRR) 32 (5), 566 –590.
13. Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Whee-ler, R., Ng, A. Y., 2009. Ros: an open-source robot operating system. En:ICRA Workshop on Open Source Software
14. Santofimia, M. J., Villa, D., Villanueva, F. J., Escolar, S., Lopez, J. C., Oct2016. A semantic middleware architecture for supporting real smartness.En: IECON 2016 - 42nd Annual Conference of the IEEE Industrial Elec-tronics Society. pp. 6925–6930.
15. M. McIntire, E. Nunes, and M. Gini. Iterated multi-robot auctions for precedence-constrained task scheduling. In AAMAS 2016 - Proceedings of the 2016 International Conference on Autonomous Agents and Multi-agent Systems, pages 1078–1086. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), Jan. 2016.
16. A. Khamis, A. Hussein, and A. Elmogy. Multi-robot Task Allocation: A Review of the State-of-the-Art, pages 31–51. Springer International Publishing, Cham, 2015.
17. R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. Artificial Intelligence, 49(1):61–95, May 1991.