Simulation Modelling Practice and Theory 49 (2014) 305–319

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

# Multi-robot data mapping simulation by using microsoft robotics developer studio

Antonio Matta-Gómez *, Jaime Del Cerro, Antonio Barrientos

*Centre for Automation and Robotics, CAR-UPM-CSIC, Madrid, Spain*

ARTICLE INFO

*Article history:*
Received 26 May 2014
Received in revised form 10 September 2014
Accepted 3 October 2014
Available online 19 November 2014

*Keywords:*
Service Oriented Computing (SOC)
Multi-robot systems
Data mapping
Robotics frameworks
Microsoft robotics developer studio

ABSTRACT

This document summarizes the goals achieved in the development of a data mapping application, for a multi-robot system, implemented as a service with the guidelines found in the Service Oriented Computing paradigm (SOC). The obtained service generates both local and global maps in the reconstruction of a virtual scenario: the local maps represent the surrounding area around each one of the mobile robots, and the global one the totality of the scenario where the robots move. The information of the global map is continuously updated by merging the data coming from the local maps by using a novel approach: each one of the maps manages a confidence level value that defines which of the data coming from the maps is worthy of being updated into the global one. This technique is not present in related work. The *Microsoft Robotics Developer Studio* framework was chosen for its implementation because of the advantages that this tool offers in the management of concurrent and distributed processes, typically found in both a robotics platform and in a multi-robot system.



## 1. Introduction

It is advisable to count with a simulation software that allows the study of the behavior of a multi-robot system in a virtual scenario previous to their execution in a real one [1]. This simulation software allows the supervision and analysis of the tasks that the multi-robot system can perform and offers the capability, to human personnel, to train and control unmanned robots in collaborative environments [2].

These simulators must be consistent with the type of concurrent and distributed processes that are typically found in a robot or in a multi-robot system; processes that allow both the execution of multiple tasks concurrently and the communication among them in a distributed approach [3].

This is why the *Microsoft Robotics Developer Studio* (MRDS) framework was chosen as the robotics simulator of this project. It not only offers two powerful dynamics and graphics engines [4], it also allows the implementation of concurrent and distributed processes in a novel approach that it can only be found in this simulator [5].

This approach is possible because the MRDS is built upon two unique libraries: the Concurrency and Coordination Runtime (CCR) and the Distributed Software Services (DSS). They both run as.NET DLLs and offer the possibility to build asynchronous, concurrent, and service-oriented applications with different programming languages [4]. These are key elements found both in robotics applications and in a software programming paradigm called Service Oriented Computing (SOC), which sees a software system as a set of services that are hierarchically coordinated and in continuous communication.

* Corresponding author.
*E-mail address:* antonio.matta@upm.es (A. Matta-Gómez).

http://dx.doi.org/10.1016/j.simpat.2014.10.003
1569-190X/

By following this paradigm a data mapping service was obtained. This service was in charge of coordinating and interrelating all the components that were part of the data mapping service, which were also modeled as MRDS services. With the obtained collaborative behavior, it was possible to reproduce the main components of a virtual scenario through the generation of accurate maps using a multi-robot system.

The generated maps take into consideration information such as obstacles, terrain gradients, traction estimates, quality of the communications, and GPS coverage. These maps are later used in the virtual system for navigation, path planning, and obstacle detection purposes.

Prior to the development of the mapping service, several mapping techniques were studied [6–8]. It was noticed, that most of these techniques divide the information needed for the recreation of the scenario in two separated parts. The first part is a global map where the static elements of the scenario, such as walls or roads, are represented [6]. This map basically allows the robot to perform path planning tasks for long distances or long periods of time. The second part introduces the concept of instant representation, which basically takes the last sensor readings and creates a map local to the robot that allows the robot to react if dynamic obstacles are detected [7].

Moreover, it was found that two different models can be derived from the different existing representations: the geometrical elements model and the occupancy cells one [8]. The first one is made of a set of geometrical primitives (points, corners, walls, shapes and so forth) whose positions are constantly estimated from the information provided by the sensors data. The second one represents the environment as a set of cells where each one of them holds information related to the scenario.

The occupancy cell model, also known as occupancy grids [9], was used over the geometrical one because it makes easier the fusion of data coming from different sensors readings [10], something that it is constantly performed by the data mapping service in order to obtain better surfaces and correct faulty readings with more recent and accurate ones.

Taking all this into account, a multi-layer grid structure was created to represent the maps generated by the robotics mapping service. This structure holds a representation of the scenario in a set of layers, where each layer is divided into a grid of cells that represents an element of the scenario.

The maps are characterized by both the resolution of their information, high or low, and the size of the area of the scenario that they can represent: local to the robot or global if it contains the entire scenario. To generate these maps, the data mapping service has the following data as input parameters: an initial low resolution 2.5-dimensional global map that represents the static data present in the virtual scenario, the position and attitude of each one of the robots, and the 3D laser readings coming from every robot.

With these inputs, the data mapping service generates a global map that contains both the information of the initial global map and the one coming from the local maps generated by each one of the robots. This is achieved through a fusion of information process based on the level of confidence present in the cells of the maps that are being fused, something that has not been found in previous related work, and that was possible because of the features present in Service Oriented Computing paradigm.

This document is organized as follows: Section 2 gives a brief description of the Service Oriented Computing programming paradigm and the benefits that MRDS services can offer to multi-robots applications. In Section 3 the obtained data mapping service is carefully described, as well as the two types of maps that this service generates. Section 4 describes the software architecture of the obtained service. Later, in Section 5 the simulation results obtained through the use of the data mapping service are shown and Section 6 presents the conclusions of this article and the future work related to the mapping techniques described in this document so as to improve it. The advantages of using the *Microsoft Robotics Developer Studio* framework in the implementation of the services are also described.

## 2. SOC computing and multi-robot systems

Service oriented computing, also known as SOC computing, has become a standard of the software industry in the area of the creation and distribution of services across networks. It is based on object oriented computing (OOC) and the coordination and synchronization of services. Basically, a service is an object that holds a proxy server used to be connected, in a distributed approach, with other services that can be located either in the same machine, in different ones, or in other networks [11].

By this way, developers can easily share services for their apps, which can be programmed by using different languages and different developing teams; allowing in this manner modularity, reliability and easy code reuse [4]. All the main software companies have adopted and supported this new paradigm [5].

This technology could not have been ignored by the robotics software industry. On the contrary, as early as 2005, SOC computing concepts started to be applied in robotics applications [12] and in 2006 Microsoft launched its SOC-based robotics framework called Microsoft Robotics Developer Studio (MRDS) [4].

This framework is a Microsoft Windows® based system focused on the creation of robotics applications that can either be simulated through the use of a three-dimensional virtual simulator called Visual Simulated Environment (VSE) or connected to real robots through a programming interface known as Visual Programming Language Environment (VPL).

The Visual Simulated Environment is composed by the XNA graphics engine, an engine that gives such a realistic rendered images that it is used by most of the state of the art video games that run inside the Microsoft XBOX video game machine, and a dynamics engine called AGEIA, widely used for realistic simulations. These both tools turns this framework into an attractive and advantageous one for the three-dimensional simulation of multi-robot systems [13].