Software architecture of the PACOM project

David Filliat

ENSTA ParisTech - INRIA Flowers team
Unité Informatique et Ingénierie des Systèmes
828 boulevard des Maréchaux
91762 Palaiseau Cedex
`david.filliat@ensta-paristech.fr`

Introduction

We present the architecture of a mobile robot whose goal is to perform autonomous semantic mapping of indoor environments. This robot has been developed under the PACOM[1] project in order to participate in the "CAROTTE" challenge organised by the french Armament Procurement Agency (DGA) and Research Funding Agency (ANR). This challenge takes place in an arena of approximately 100 m^2 wherein objects and obstacles are placed, which contains several rooms, with variable ground types and difficulties (fitted carpet, tiling, grid, gravel, ...). Several types of objects are present in multiple instances which must be detected, located, and identified by the robot. The complete description of the environment can be found on the challenge website[2].

The architecture described in this paper was used during the third CAROTTE competition and is an evolution of our previous work for the first two editions [Jebari et al., 2011, Baillie et al., 2011, Filliat et al., 2012].

System Architecture

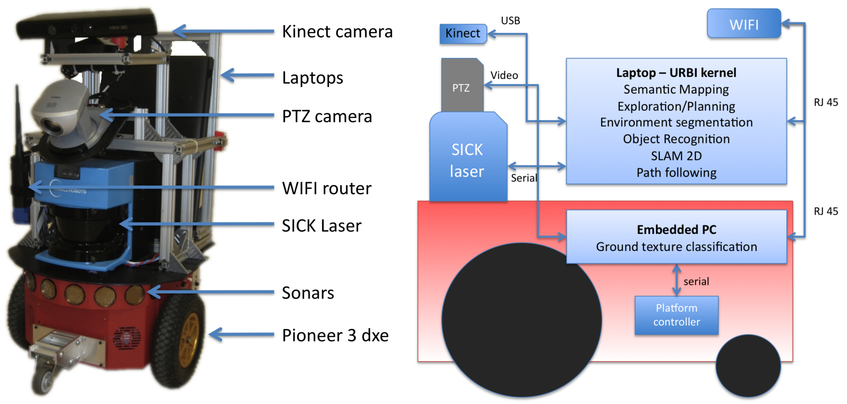

Figure 1: The robot developed for the PACOM project and the software architecture of our robot showing the repartition of the software modules on the 2 onboard computers.

Our robot (Figure 1, left) is a pioneer 3 dx from Mobile Robots Inc. The robot carries a SICK laser range finder, a ring of sonar sensors, a Pan-Tilt-Zoom color camera and a Microsoft Kinect camera as RGBD sensor. Two on-board computers linked through an Ethernet router run all the software modules involved in semantic mapping and exploration.

The software architecture uses the Urbi framework[3]; an open-source middleware for programming complex robotic systems developed and supported by Gostai. Urbi is composed of a distributed component architecture (UObject), and an innovative orchestrator language (urbiScript) to coordinate all components. This language incorporates high-level features that facilitate the development of parallel and event-based applications. For the project, we thus developed a set of UObjects in C++ carrying out the various necessary functionalities (Figure 1, right). The whole mission of the robot is implemented in urbiScript which uses these UObjects' functionalities and coordinates their activation.

[1]http://cogrob.ensta-paristech.fr/pacom/
[2]http://www.defi-carotte.fr
[3]http://www.urbiforge.org/

For the third competition, besides hardware interface modules, the main software components were : **2D SLAM**, **Exploration and planning** (chooses next exploration goal and computes trajectory), **Path following**, **Environment segmentation** (detects walls and objects using kinect), **Object recognition** (recognizes segmented objects), **Ground texture recognition** (detects gravel and ground type) and **Semantic mapping** (detects rooms and integrates all objects and ground type detections into the map). Most of these modules are described in [Filliat et al., 2012]. Each software component is executed in a thread inside a separate UObject and communicates through message passing or shared memory. This repartition facilitates independent component development through well defined interfaces and their testing in simulation, but makes integration more difficult. In particular, the presence of many different threads and their orchestration through both the Urbi and the OS schedulers could lead to difficult to understand behaviours which need thorough testing to be tuned.

The dataflow between components as well as a hierarchical finite state machine that control the robot behaviour during the mission are both coded in urbiScript. This feature make it very easy to test various components configuration and to very rapidly adapt the robot behaviour to the cost of having a complex script that could be difficult to maintain.

Navigation subsystem

A particularly important part of the software architecture is devoted to safe navigation, taking into account account glass walls, gravel holes in the ground and large 3D objects such as tables under which our robot cannot pass. Navigation therefore takes into account 4 different sensors : laser for 2D SLAM and avoidance of most obstacles, sonars for glass detection, a color camera for gravel detection on the ground and the kinect for 3D object detection.

In our previous approach [Filliat et al., 2012], we integrated all these detections projected on the ground inside a 2D obstacle map which was constructed as the robot moved. However, this approach proved to be difficult to tune. In particular, it was very sensitive to any time shift between sensor acquisitions, for which synchronisation was difficult to enforce given the variety of sensors and the variable processing time for each sensor. Including obstacles detected by sonar sensors was also difficult because of the poor localization of these obstacles, which resulted in closing possible passages beside detecting glass walls.

We therefore resorted to a more conservative strategy. Planning is first performed using the 2D laser map, and during robot navigation, the three other sensors monitor only the robot path in order to check for the presence of obstacles. Upon obstacle detection, the robot is stopped through the event based mechanism of urbiScript and the presence of a real static obstacle is verified before inclusion in the map. Planning is then performed again to avoid the newly discovered obstacle. This approach strongly reduces noise in the obstacles added to the map and avoid erroneously blocking free passages.

Conclusion

Urbi and urbiscript are powerful tools that make event-based programming and data-flow control very easy to use when developing a robot. They make it possible to gain a lot of time for development and to orchestrate all the work that have been done by many different people. The limitation is the complexity of the script necessary for an exploration robot such as ours that makes error understanding difficult and requires extensive tuning.

Acknowledgment

The PACOM project was carried out by ENSTA ParisTech, ISIR and Gostai, supported by DGA in the frame of the CAROTTE competition and funded by ANR under the subvention 2009 CORD 102.

References

[Baillie et al., 2011] Baillie, J.-C., Demaille, A., Duceux, G., Filliat, D., Hocquet, Q., and Nottale, M. (2011). Software architecture for an exploration robot based on urbi. In *Proceedings of the 6th National Conference on Control Architectures of Robots*.

[Filliat et al., 2012] Filliat, D., Battesti, E., Bazeille, S., Duceux, G., Gepperth, A., Harrath, L., Jebari, I., Pereira, R., Tapus, A., Meyer, C., Ieng, S., Benosman, R., Cizeron, E., Mamanna, J.-C., and Pothier, B. (2012). Rgbd object recognition and visual texture classification for indoor semantic mapping. In *Proceedings of the 4th International Conference on Technologies for Practical Robot Applications (TePRA)*.

[Jebari et al., 2011] Jebari, I., Bazeille, S., Battesti, E., Tekaya, H., Klein, M., Tapus, A., Filliat, D., Meyer, C., Ieng, S., Benosman, R., Cizeron, E., Mamanna, J.-C., and Pothier, B. (2011). Multi-sensor semantic mapping and exploration of indoor environments. In *Proceedings of the 3rd International Conference on Technologies for Practical Robot Applications (TePRA)*.