Distributed Intelligent Navigation Architecture for Robots

Eduardo Javier Pérez Rodríguez
Department of Tecnología Electrónica, University of Málaga, Spain
E-mail: edu@dte.uma.es

This thesis presents a basic architecture named **DLA** (**D**istributed and **L**ayered **A**rchitecture) to support navigation in unstructured dynamic environments for any autonomous mobile robot. DLA supports intuitive adaptation to physically different agents and simple expansion of their capacities via addition of new modules. DLA works by combining the responses of different deliberative and reactive algorithms through the interaction of freely distributed processes in an asynchronous way. This architecture provides transparency to the user through a high simplicity and portability.

Keywords: Control Architecture, Autonomous Navigation, Case-Based Reasoning, Metric-Topologic Integration

1. Introduction

Robots are currently helping humans by performing complex, repetitive, unpleasant and/or dangerous tasks in all kinds of environments. In order to progressively develop more complex tasks, it is usually necessary that these robots are able to navigate in dynamic and potentially unstructured environments. Although many navigation schemes for autonomous mobile agents have been proposed to solve this complex problem, lack of standardization makes it enormously difficult to adapt existing schemes to systems different from those for which they were conceived.

This thesis [3] proposes a new control architecture to develop a basic infrastructure for navigation in unstructured dynamic environments. It supports easy adaptation to new robots and simple expansion by adding new behaviours in a modular way. This architecture is named **DLA** (**D**istributed and **L**ayered **A**rchitecture) [1], and it allows coordination of modules distributed through different machines in an asynchronous way.

An architecture can be divided into two parts [1]: **structure** and **style**. Structure refers to system decomposition into modules and to how they are implemented, whereas style refers to software mechanisms supplied by the architecture to allow interaction among modules. Next sections describe the style and structure for the DLA architecture.

2. Architecture style

The DLA architecture has been conceived to work in a transparent way to the developer. This is achieved through high simplicity and portability in its use [5]. DLA follows a distributed shared memory scheme based on Dulimarta's proposal [2], where a central server manages the interaction among modules as shown in Figure 1.

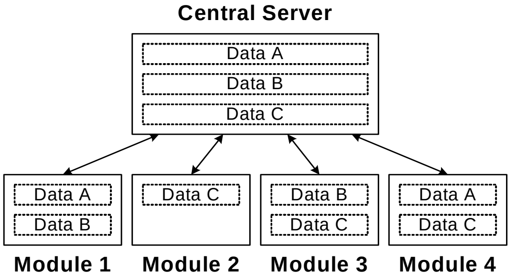

Fig. 1. DLA style.

The main drawback in Dulimarta's proposal is that only one server manages all interactions between modules, so the system might collapse if traffic between modules is high. DLA automatically splits the central server into a set of temporal children servers to manage interactions among modules while they exist. Thus, concurrency is considerably improved, and the response time of the central server is reduced.

This indirect connection scheme among modules provides very important properties:

---

[1] Available at (*GPL*): http://campusvirtual.uma.es/eperez

ISSN 0921-7126, IOS Press.

- **Simplicity:** to use DLA only an API library is needed.
- **Scalabitity:** processing capabilities can be increased by incorporating new machines to the system.
- **Extensibility:** new equipment and behaviours can be progressively added to the system.
- **Portability:** modules can be developed in several languages and platforms (C, JAVA, Matlab, Linux, Windows, ...), and navigation infrastructure can be directly used by new agents.
- **Debugging:** modules can be independently implemented and debugged.

3. Architecture structure

The basic navigation infrastructure developed relies on a hybrid structure, which successfully connects fast reactive behaviours and deliberate planning processes in an asynchronous way. This strategy allows at the same time fast reaction to environment stimuli and also planning for the development of complex tasks.

3.1. Reactive system

Reactive behaviours have been achieved by means of a learning scheme based on the **Case-Based Reasoning** paradigm, that supports acquisition of different navigation behaviours through a supervised training stage [6]. Case definition and system operation are shown in Figure 2.

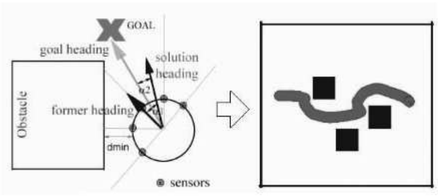

Fig. 2. CBR case features and emerging reactive trajectory.

Learning is achieved by supervisedly guiding the robot through an obstacle field. A human driver easily adapts to any situation and provides flexibility to act one way or another in similar situations, coming close to obstacles if necessary or performing sharp direction changes if required. The human driver implicitly takes into account the kinematics and dynamics of the problem, as well as the nature of the environment, so such factors are learnt through the supervised training process in an intuitive way. Thus, adaptation of reactive behaviours to new environments or new agents with different dynamic characteristics can be achieved without any explicit analytical study. Learning by own experience is also supported to incorporate new situations to the system when the robot is moving on its own and to adapt to different configurations and environments on-line.

3.2. Deliberate system

Deliberate planning processes increase the efficiency and navigation capabilities of the system. Two different levels of planning have been developed: i) a path planner based on an occupancy grid to compute a trajectory free of obstacles to reach a target, and ii) a route planner to provide a succession of regions to finish a task [4]. These planning processes are supported by a metric probabilistic representation built from environment stimuli, and a topological representation extracted from the metric one by means of a **hierarchical pyramidal structure** [4]. The generation of this structure is shown in Figure 3.

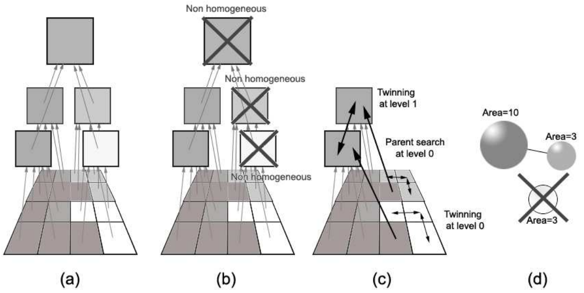

Fig. 3. Hierarchical pyramidal structure: a) generation; b) prunning; c) merging; d) resulting topology.

The pyramid is derived from the *Split and Merge* paradigm and a linked pyramid. First, the whole pyramid is constructed. Then, non homogeneous nodes are removed from the structure. Finally, orphaned nodes (nodes not linked to their upper level) search for a parent (node in the upper level) or sibling (neighbour node in their own level) presenting a probability of occupation similar to theirs and link to them. Remaining orphan nodes provide a decomposition of the environment into a set of connected homogeneous regions, including explored and unexplored areas. Occupied ones are removed from the structure to achieve a topological map.

Both metric and topological representations are constantly related through a set of links, so plans can be quickly propagated from one representation to the other. Planning is performed at highest possible level and the system does the best it can with the available plans at each time instant.

4. Results

The DLA architecture has been exhaustively tested over simulated and real environments with different robots. Its style performance has been evaluated and compared to the original Dulimarta's proposal. The structure of DLA architecture has also proven to be very robust, and works correctly in unstructured dynamic environments with different robots after a short training of the reactive layer. It has been checked that the longer the robot operates, the better it behaves because the reactive layer is more and more adapted to its environment. Furthermore, if the platform or environment suffer changes, the reactive algorithm adapts itself again to the new situation through unsupervised learning. It has also been tested that a deliberative layer improves the efficiency and functionality of the system. When a topological representation is available, a route is computed at topological level and then a path between each pair of consecutive nodes is calculated at metric level by using a Potential Fields based approach. The proposed method is fast enough to recalculate the route without stopping the robot when unexpected obstacles appear in unexplored or changed areas. While a new route is calculated, the reactive layer keeps the robot moving safely.

5. Conclusions

The developed DLA architecture has proven to be a powerful and efficient tool to develop autonomous navigation applications in unstructured dynamic environments for different robots. The DLA architecture style easily supports cooperative modular systems with a high flexibility, where modules can be independently implemented using several different languages and platforms. The DLA architecture structure combines reactive behaviours with deliberate planning. The main advantage of the proposed reactive layer is that the system easily adapts to changes. In fact, unlike most popular reactive approaches, no kinematics or dynamics need to be explicitly taken into account, so it can be easily adapted to other robots. Deliberate processes rely on a metric-topological representation of the environment, and allow explicit representation of unexplored areas and their relationships at topological level. The main advantage of the proposed scheme is that relationship between topological and metric levels is explicitly kept by means of a link set, so that plans at both levels can easily be related.

Acknowledgements

This work has been partially supported by the Spanish Ministerio de Ciencia y Tecnologia (MCYT) and FEDER funds, project No. TIN2004-07741. The author is also very grateful to Cristina Urdiales, the advisor of this thesis.

References

[1] E. Coste-Manière and R. Simmons, Architecture: The Backbone of Robotic Systems, in: *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, California, 2000, pp. 67-72.

[2] H. Dulimarta and A.K. Jain, Client/Server Control Architecture for Robot Navigation, *Pattern Recognition* **29**(8) (1996), 1259-1284.

[3] E.J. Pérez, Arquitectura de Navegación Distribuida para Agentes Robóticos, PhD Thesis, Department of Tecnología Electrónica, University of Málaga, Málaga, Spain, 2006. (available at: http://campusvirtual.uma.es/eperez)

[4] A. Poncela, E.J. Pérez, A. Bandera, C. Urdiales and F. Sandoval, Efficient Integration of Metric and Topological Maps for Directed Exploration of Unknown Environments, *Robotics and Autonomous Systems* **41**(1) (2002), 21-39.

[5] C. Urdiales, A. Bandera, E.J. Pérez, A. Poncela and F. Sandoval, Hierarchical Planning in a Mobile Robot for Map Learning and Navigation, in: *Autonomous Robotic Systems: Soft Computing and Hard Computing Methodologies and Applications*, Physica-Verlag, 2003, pp. 165-188.

[6] C. Urdiales, E.J. Pérez, J. Vázquez-Salceda, M. Sànchez-Marrè and Francisco Sandoval, A Purely Reactive Navigation Scheme for Dynamic Environments Using Case-Based Reasoning, *Autonomous Robots* **21**(1) (2006), 65-78.