Experiences Using a Component-Oriented Architectural Framework for Robots and its Improvement with a MDE Approach*

Francisco J. Ortiz, Juan A. Pastor, Diego Alonso, Bárbara Álvarez, Pedro Sánchez
Division of Electronics Engineering & Systems (DSIE)
Universidad Politécnica de Cartagena, Campus Muralla del Mar s/n
30202 Cartagena, Murcia (SPAIN)
francisco.ortiz@upct.es

Abstract. This paper describes the experience of the DSIE research group in the developing of the EFTCoR family of robots using an abstract architectural framework ACRoSeT, following the component-based paradigm. Using abstract components allow us to define very different architectures in a platform independent way. The translation of the abstract components to platform specific code is a hard and difficult task that can be partially automated with the help of the model transformation tools provided by the MDE approach.

Keywords: MDE, component-based software architecture, teleoperated robot.

1. Introduction

This paper describes the authors' experiences using software architectures in the development of teleoperated cranes and vehicles for ship hull cleaning in the context of the EFTCoR project. This development was specially challenging due to:

- The use of different execution platforms and different programming languages.
- Different functional requirements makes impossible to use a single architecture.

We needed a way to define different architectures sharing common components. With these ideas in mind, we defined *ACRoSeT* [1], an abstract architectural framework for the domain of teleoperated robots.

Teleoperated robotic systems cover a broad range of mechanisms that usually perform a small number of highly specialized tasks. Such specialization implies high variability that makes very difficult to design a single architecture flexible enough to deal with such heterogeneity. For this reason it is required a flexible and extensible architectural framework that (1) does not impose a concrete architecture, but allow defining different architectures, (2) allows reusing components in systems with different architectures, (3) allows the integration of components may be software or

* This work was partially supported by the Spanish CICYT project MEDWSA, ref. TIC2006-15175-C05-02 and the Regional Government of Murcia Seneca Program, ref. 02998-PI-05.

hardware, and (4) makes possible to integrate “intelligence,” or to interoperate with “intelligent systems”.

There have been numerous efforts to provide developers of software for robots with component frameworks to ease the development of robotic systems. Among these frameworks it is possible to highlight the following: OROCOS [3], CLARAty [9], MCA [6], ORCA [2], CARMEN [4] and PLAYER [8]. All of them make very valuable contributions that simplify the systems development.

2. Software Architecture for the Teleoperated Devices of the ETFCoR Family.

The EFTCoR system comprises a family of teleoperated systems which mission is to retrieve and confine paint, oxide and marine adherences from ship hulls. The working environments are not fixed, there is a great variety of ship types, hull areas and shipyards characteristics, the systems consider different degrees of autonomy and different systems may have to work cooperatively at the same time.

ACRoSeT provides a common framework of abstract components to design software for teleoperated robots with very diverse behaviours. The subsystems defined by ACRoSeT are the following (see Fig. 1):

- *Coordination, Control and Abstraction Subsystem (CCAS): a*bstracts and encapsulates the functionality of the system physical devices.
- *Intelligence Subsystem* (*IS*): comprises the subsystems that provide intelligence to the global system. These systems are considered users of the *CCAS* functionality.
- *User Interaction Subsystem (UIS):* interprets, combines and arbitrates between orders that may come simultaneously from different users of the *CCAS*.
- *Safety, Management and Configuration Subsystem (SMCS):* Initializes, configures and manages the application.

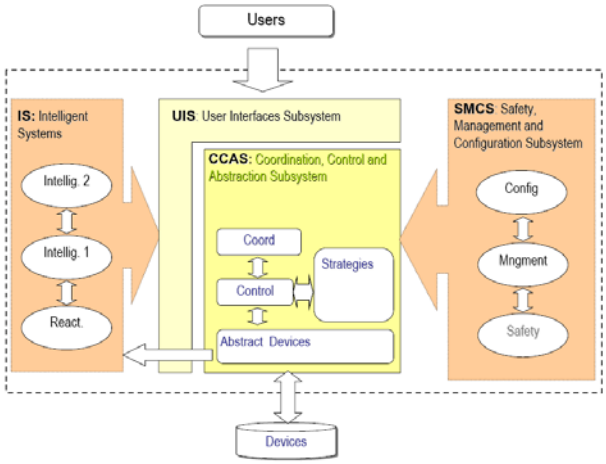

Fig. 1. An overview of the subsystems of ACRoSeT

The CCAS comprises components that are defined in four levels of granularity: (1) atomic components: abstract the characteristics of sensors and actuators, (2), Simple Controllers, (3) Mechanisms Controllers, and (3) Robot Controllers.

3. Instantiations of ACROSET for the EFTCoR family

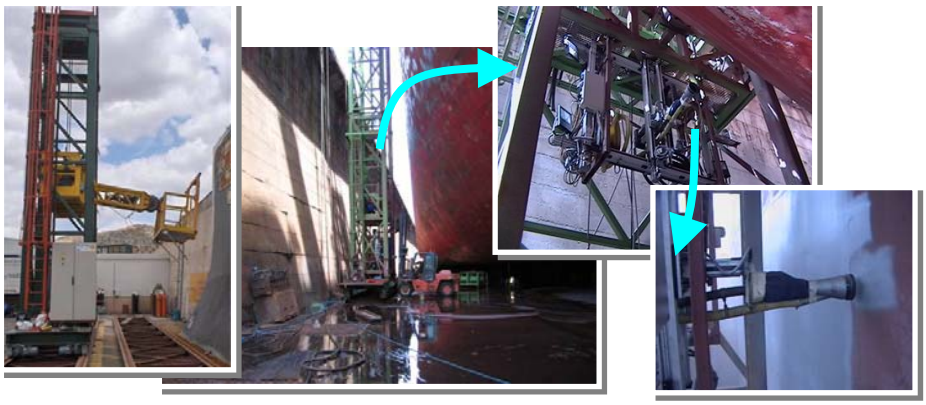

Fig. 2. XYZ table mounted on a crane. Tests in NAVANTIA shipyards

In response to the special industrial requirements of the EFTCoR project, the cranes (see Fig. 2) has been implemented using a PLC SIMATIC S7-300 and a Field-Bus (PROFIBUS-DP). The second instantiation is a caterpillar vehicle capable of scaling a hull thanks to permanent magnets (Fig. 3), carrying a manipulator that holds a cleaning tool. The execution platform is an on-board embedded PC with RTLinux Operating System.

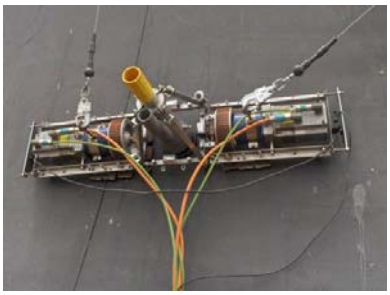

Fig. 3. Lazaro climbing vehicle

4. MDE

Model-Driven Engineering (MDE) [5] is an approach to software development in which *models* are first-class entities that guide each and every step of the design process. The other key concept in which rests MDE is *model transformation* [7].

We have adopted a MDE approach to develop the software architecture of robotic systems based on the abstract components proposed by *ACRoSeT*, using the *Eclipse* development environment and plug-ins. Different transformations make possible to map the *ACRoSeT* components to different platforms.

5. Conclusions

It is not possible to define a software architecture generic enough to be adapted to the entire domain, but usually there is no need to develop such architecture. The aim is to reuse components in different architectures and this is just what CBD and component frameworks propose.
Current component frameworks for robotic applications generally impose a concrete programming language and execution platform. As it is desirable to be able to define components that are independent of both system architecture and execution platform, ACROSET defines abstract components. However, the translation of the ACROSET abstract components into concrete, platform specific components is a difficult and error prone task. So, the ACROSET approach will only show its full potential if we are able to find a way to automatically translate abstract components into concrete components. The adoption of the MDE approach is a key step to achieve this goal.

References

1. Álvarez B, Sánchez P, Pastor JA, Ortiz F,: An Architectural Framework for Modeling Teleoperated Service Robots, ROBOTICA. ISSN 0263-5747, Cambridge University Press. Volume 24, Issue 04, pp 411-418.
2. Brooks, A.; Kaupp, T.; Makarenko, A.; Williams, S.; Oreback, A.: Towards component-based robotics. 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems., Vol., Iss., 2-6 Aug. 2005, pp 163- 168
3. Bruyninckx, H., Koninckx, B. & Soetens, P.,2002. A Software Framework for Advanced Motion Control, Dpt. of Mechanical Engineering, K.U. Leuven. OROCOS project inside EURON. Belgium.
4. Montemerlo, M.., Roy, N., and. Thrun S. Perspectives on standardization in mobile robot programming: The Carnegie Mellon Navigation (CARMEN) Toolkit. In IEEE/RSJ Intl. Workshop on Intelligent Robots and Systems, 2003.
5. Schmidt, D.: Model-Driven Engineering. IEEE Computer, 2006, 39(2), IEEE Computer Society. ISSN 0018-9162. doi: 10.1109/MC.2006.58.
6. Scholl, K.U. Albiez, J. & Gassmann, B. (2001) MCA: An Expandable Modular Controller Architecture, Karlsruhe University, 3rd Real-Time Linux Workshop, Milano, Italy
7. Sendall, S. and Kozaczynski, W.: Model Transformation: The Heart and Soul of Model-Driven Software Development. IEEE Software, 2003, pp. 42-45, 20(5), IEEE Computer Society. ISSN 0740-7459. doi: 10.1109/MS.2003.1231150.
8. Vaughan R,. Gerkey B, and Howard A.. On device abstractions for portable, reusable robot code. Proc. of the IEEE/RSJ Intl. Conf. On Intelligent Robots and Systems (IROS), 2003.
9. Volpe, R.; Nesnas, I.; Estlin, T.; Mutz, D.; Petras, R.; and Das, H.,2001. The CLARAty architecture for robotic autonomy. In IEEE Proceedings., ed., Aerospace Conference, vol. 1, pp 121-132, 2001 Montana, USA.