Mechatronics 000 (2016) 1–17

Contents lists available at ScienceDirect

Mechatronics

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

An object-unified approach to develop controllers for autonomous underwater vehicles

T. Soriano [a,*], N.V. Hien [b,**], K.M. Tuan [b], T.V. Anh [b]

[a] LISMMA, Laboratoire d'Ingénierie des Systèmes Mécaniques et des Matériaux, SUPMECA, Université de Toulon 83957 La Garde, France
[b] Department of Ship Engineering and Fluid Mechanics, School of Transportation Engineering, HUST, Hanoi University of Science and Technology, Vietnam

ARTICLE INFO

Article history:
Received 16 June 2014
Accepted 28 December 2015
Available online xxx

Keywords:
AUV control
Underwater robot
Mechatronic object-oriented design
Hybrid automata
Real-time UML/SysML
MDA

ABSTRACT

This paper presents a novel approach, which is based on the integration of hybrid automata, real-time Unified Modeling Language (UML)/System Modeling Language (SysML) and Model-Driven Architecture (MDA) in order to systematically develop controllers for Autonomous Underwater Vehicles (AUVs). This approach entirely covers the analysis, design, implementation and deployment phases focused on controllers for AUVs and also allows the designed control elements to be customizable and re-usable in the development of new applications of various AUV types. The paper brings out step-by-step the AUV dynamics together with the control structure, specializations of MDA's features such as the Computation Independent Model (CIM) with use-cases and hybrid automata, the Platform Independent Model (PIM) carried out by using the real-time UML, as well as its Platform Specific Model (PSM) implemented by sub-system paradigms and object-oriented mechanisms to completely perform the development lifecycle of AUV controllers. The transformation rules are then introduced and applied to convert the detailed design model of PIM into the implementation model of PSM by using open-source platforms in order to quickly simulate and realize AUV controllers. Based on this approach, a planar trajectory-tracking controller of a miniature autonomous submersible was completely developed and successfully taken on trial trip. In this application, the controller has been implemented with the simulation model in OpenModelica. The obtained simulation results then can permit us to mainly define the designed control elements and their properties, as well as building the implementation libraries for performing quickly the realization model in Arduino Mega2560 microcontroller.

1. Introduction

Autonomous underwater vehicles (AUVs) have seen unprecedented levels of growth over the last decade. Even though AUVs have been mainly used for military applications, there is a considerable and increasing interest for civilian applications combined with more complex and dangerous missions. This is due to the underlying characteristics of safety and cost effectiveness when compared to manned vehicles. AUVs require no human operator be subject to the conditions and dangers inherent in the underwater environment that the vehicle is exposed to, and therefore the risk to human life is greatly minimized or even removed. It is postulated that AUVs will be used in the future extensively for environmental monitoring, the biology of ocean resources, disaster and tsunami warnings, etc.

Control systems have a significant impact on the performance of AUVs and marine structures allowing them to perform tasks in severe sea states and during long periods of time. AUVs are designed to operate with adequate reliability and economy, and in order to achieve this, it is essential to control the motion. The problem of designing motion controllers for AUV is equally challenging because these controllers are tightly connected with the dynamic models. In addition, AUV controllers and actuators take into account models with discrete events and continuous behaviors that can be called hybrid dynamic systems (HDS) [1,2].

These behaviors are distributed on different operating modes, which are associated with processes related to the interactivity with users such as the designer, supervisor, maintainer etc. Furthermore, controlled systems of AUVs do not always have the same behavior because they are associated with validity hypotheses to check at any moment; the security requirement forces to envisage events, and behaviors different from nominal behaviors. Hence, the whole behaviors of these systems are thus complex that can be modeled by hybrid automata (HA) [1,2] for modeling entirely requirements of control analysis in the development lifecycle

---

[*] Corresponding author. Tel.: +33 0 483 166 619; fax: +33 0 483 166 601.
[**] Corresponding author. Tel.: +84 0 438 692 501; fax: +84 0 436 230 592.
E-mail addresses: thierry.soriano@supmeca.fr (T. Soriano), hien.ngovan@hust.vn (N.V. Hien).

http://dx.doi.org/10.1016/j.mechatronics.2015.12.011
0957-4158/© 2016 Elsevier Ltd. All rights reserved.

Please cite this article as: T. Soriano et al., An object-unified approach to develop controllers for autonomous underwater vehicles, Mechatronics (2016), http://dx.doi.org/10.1016/j.mechatronics.2015.12.011

Nomenclature

AUVs	Autonomous Underwater Vehicles
BDD	Block Definition Diagram
CIM	Computation Independent Model
DoF	Degrees of Freedom
EKF	Extended *Kalman* Filters
GPS	Global Positioning System
HA	Hybrid Automata
HDS	Hybrid Dynamic Systems
IB	Integral Backstepping
IBD	Internal Block Diagram
IDE	Implementation Development Environment
IGCB	Instantaneous Global Continuous Behavior
IMU	Inertial Measurement Unit
LOS	Line-Of-Sight
MAS	Miniature Autonomous Submersible
MAS	Miniature Autonomous Submersible
MDA	Model-Driven Architecture
MDSE	Model-Driven Systems Engineering
MES	Marine Environment System
OMG	Object Management Group
OO	Object-Oriented
PID	Proportional-Integral-Derivative
PSM	Platform Specific Model
PIM	Platform Independent Model
SDLC	System Development Life Cycle

of systems. Furthermore, the immersion in an industrial control context makes that designers and programmers must take into account costs and existing standards for analyzing, designing and implementing effectively these systems. The customization and reutilization are factors to be associated with the production of a new application in order to reduce its costs, resources and time development.

Up to date, the Object Management Group (OMG) has standardized the model-driven architecture (MDA) [3] together with UML/SysML [4,5], which have been started with the well-known and long established idea to separate the specification of system operations from the details of the way that system uses the capabilities of its platform. MDA provides an approach for and enables tools to be provided for: specifying a system independently of the platform that supports it; specifying platforms; choosing a particular platform for the system; and transforming the system specification into one for a particular platform.

Starting from the above considerations, we have developed an object-unified model to systematically analyze, design, implement and deploy the AUV controller of which dynamic behaviors can be modeled by HA. This control system permits an AUV to reach and follow a reference trajectory in the *Cartesian* space. In this model, we specify the AUV dynamic model combined with the generic control structure, MDA's features such as the CIM performed by specifying the use-case model of UML/SysML [4,5] related to the HA specialization, the PIM carried out by using the real-time UML [6,7,8] as well as its PSM implemented by the object-oriented mechanisms of open-source platforms in order to make up an executable process for analyzing, designing, implementing and realizing systematically the AUV controller.

The structure of this paper is organized as follows: Section 2 is depicted the works that have led us to develop the object-unified approach focused on AUV controllers. Section 3 introduces the general control structure and MDA-based process for the AUV dynamic model. In Section 4, the analysis, design and realization models of AUV controllers are developed by implementing the defined MDA-based process combined with the detailed components of CIM, PIM and PSM. Following this approach, a planar trajectory-tracking controller of a miniature autonomous submersible was completely developed and successfully taken on trial trip that is presented in Section 5. Finally, the discussion and conclusions are made in Section 6.

2. Related work

2.1. Structural computing techniques for AUV controllers

There are actually many AUV control applications that have used soft computing techniques to optimally solve the control of AUV dynamics. For example, Titan and Collins [9] have proposed a novel trajectory planning method for a robot manipulator whose workspace includes several obstacles. In this method, a polynomial based on *Hermite* cubic interpolation was applied to approximate the time histories of the trajectory in task space; and a genetic algorithm was introduced to search for valid and optimal solutions to the trajectory. Li and Lee [10] have introduced an adaptive nonlinear controller for diving dynamics of an AUV without any restricting condition on the vehicle's pitch angle in the diving plane by using a traditional back stepping method. A control system using sliding mode control was applied to implement the controller of underwater robot [12] that has shown the controller's effective capabilities in plant nonlinearity and parameters uncertainties. Son and Kim [13] have published a study on manoeuvring control of underwater vehicle from the perspective of a combined discrete-event and discrete-time system simulation [11]; the simulation model is established on the basis of discrete-event system specification formalism, which is a representative modeling formalism of a discrete-event system simulation. This hybrid approach makes possible to build a simulation-based expert system which supports the decision making for the acquisition of an underwater vehicle. Jouffroy and Opderbecke [14] have presented an approach for the underwater vehicle navigation, which consists of diffusion-based observers processing a whole trajectory segment at a time, allowing the consideration of important practical problems such as different information update rates, outages, and outliers in a simple framework. Perera and Soares [15] have developed a calculation model for the estimation and prediction of trajectory-tracking controller that can be grouped into the continuous-time and discrete-time categories including the dynamic system model as well as the measurement model. To combine the continuous-time model with the discrete-time model, the nonlinear kinematic system model and linear model of measurement system are respectively associated with a white *Gaussian* state noise and a white *Gaussian* measurement noise. The Extended *Kalman* Filters (EKF) algorithm in this model is an adaptive filter that estimates the position, depth, attitude and velocity of the AUV.

However, we find that the above guidance and control models are based on the structural procedures for the implementation. Thus, they could be difficult to customize and re-use the designed control elements for implementing controllers of different AUVs into various software and hardware platforms in order to suitably realize them. To achieve this goal, the model-driven approach can be specialized for the whole development lifecycle of AUV controllers.

2.2. Model-driven development for system engineering

2.2.1. Using the real-time UML/SysML

The real-time UML [6,7,8] has its own the graphical notation set to model structures and behaviors of real-time systems. A capsule is represented as a class, stereotyped «*capsule*». Capsules have much of the same properties as classes, e.g. the operations,

Please cite this article as: T. Soriano et al., An object-unified approach to develop controllers for autonomous underwater vehicles, Mechatronics (2016), http://dx.doi.org/10.1016/j.mechatronics.2015.12.011