Vietnam Journal of Mechanics, VAST, Vol. 34, No. 1 (2012), pp. 55 – 65

A REALIZATION MODEL TO DEVELOP THE AUTOPILOT SYSTEM OF SHIPS BY SPECIALIZING MDA

Ngo Van Hien, Vu Duy Quang
Hanoi University of Science and Technology, HUST

Abstract. This paper presents a method which is based on the Model-Driven Architecture (MDA) and functional blocks to realize effectively the autopilot systems of ships. It brings out an executable MDA process to cover completely the requirement analysis, design and deployment phases of these systems. This process also allows the determined design elements to be customizable and re-usable in the new applications of controlled ship steering systems. The paper indicates straightforwardly the ship dynamic model-to-be used, the Computation Independent Model (CIM) of a ship autopilot system, the Platform Independent Model (PIM) of this system by using the Real-Time Unified Modeling Language (UML), and its Platform Specific Model (PSM) implemented by the functional blocks. Furthermore, the important transformation rules are also brought out and applied to convert the identified PIM into PSM for implementing quickly this system with different industrial frameworks such as the IEC61499 in a programmable controller. Then, its deployment model completely is tested on a model ship with the predetermined program and control performance.

Key words: Control of ships, MDA, Real-Time UML, functional blocks.

1. INTRODUCTION

The control systems are increasingly more intelligent and easier to operate for improving the control performance. The immersion in an industrial control context makes that the designers and programmers must take into account costs and existing standards for analyzing, designing and implementing effectively these systems. The customization and re-utilization are factors to be associated with the production of a new application in order to reduce its costs, resources and time development.

The Model-Driven Architecture (MDA) [11] starts 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.

Moreover, one of the key industries in Vietnam, the shipbuilding technology, is being developed very rapidly. It has imported much equipment for building large and modern

ships especially control devices, so that the cost of a ship could be increased in comparison with the living standards.

Starting from these considerations, we have developed a model-based realization process to effectively carry out the control parts of Autopilot Systems of Ships (ASS), which have the dynamic behaviours modelled by using hybrid automata [1]. This system allows the trajectory of a ship to be stabilized with predetermined programs.

In our process, we adapt the dynamic model of ships [3], MDA's features, Functional Blocks (FB) of IEC61499 standard [14] and S7-200 programmable controller [15] to perform completely an executable process of the analysis, design and realization phases of ASS.

This paper is described by the following sections:

- Section 2 presents the overview of ship dynamic models, which permit us to gather the requirement analysis of an ASS;

- Section 3 indicates the MDA specialization for an ASS to obtain an executable MDA process model to develop this system;

- Section 4 shows out the detailed executable MDA process to realize effectively an ASS with the standard three degrees of freedom dynamic model of ships.

2. OVERVIEW OF SHIP MODELING AND CONTROL GUIDANCE

2.1. Main motion tasks of a ship

To obtain the most suitable feedback controller design for each operation mode of a ship, it is convenient to classify its main motion tasks in the following types [3]:

- *Point to point*: the ship must reach a desired goal configuration starting from a given initial configuration.

- *Path following*: the ship must reach and follow a geometric reference path in the Cartesian space starting from a given initial configuration (on or off the path).

- *Trajectory tracking and Path tracking*: the ship must reach and follow a reference trajectory / path in the Cartesian space (i.e., a geometric path with an associated timing law) starting from a given initial configuration.

The control realization of these motion tasks can be carried out by using either feed-forward commands, or feedback control, or a combination of the two.

2.2. Dynamic model of ships

According to SNAME (*Society of Naval Architects and Marine Engineers*) [17], the six different motion components are defined as *surge*, *sway*, *heave*, *roll*, *pitch*, and *yaw* (Table 1).

From the large field of guidance, navigation and control of ships, we repeat here the 6D (six degrees of freedom) dynamic model in body frame [3], [10], which can be written in the following form:

$$\begin{aligned} &\dot{\eta} = J(\eta)v \\ &M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau + g_0 + \omega, \end{aligned} \tag{1}$$

where: $\eta = (x, y, z, \phi, \theta, \psi)^T$ is the position (NED: *North, East and Down*) and orientation (Euler: RPY - *Roll*, *Pitch* and *Yaw* angles); $\nu = (u, v, w, p, q, r)^T$ is the velocity and angular velocity; $M = M_{RB} + M_A$ is the mass matrix; $C(\nu) = C_{RB}(\nu) + C_A(\nu)$ is the

Table 1. SNAME notations for ships

Degree of freedom	Motions	Force and moment	Linear and Angular velocity	Position and Euler angles
1	*Surge*	X	u	x
2	*Sway*	Y	v	y
3	*Heave*	Z	w	z
4	*Roll*	K	p	φ
5	*Pitch*	M	q	θ
6	*Yaw*	N	r	ψ

skew symmetric matrix of centripetal and *Coriolis* effects: M_{RB} - the generalized constant inertia matrix and C_{RB} together with the linear and angular velocities, M_A - the added mass inertia matrix; $D(\nu) = D + D_n(\nu)$ is the symmetric and positive definite damping matrix consisting of linear and nonlinear parts; $g(\eta)$ is the vector of gravity and buoyancy effects; τ is the control force and torque; g_0 is the ballast force and torque; and finally ω is the force and torque of environmental effects caused by wind, waves and ocean current.

2.3. Using three degrees of freedom (3D) horizontal model

The horizontal motion of a surface ship is often described by the motion in *surge*, *sway* and *yaw*. Therefore, we choose $\eta = (x, y, \psi)^T$, $\nu = (u, v, r)^T$ and the *Jacobian* matrix typed 3×3 for obtaining the dynamic model of this case from the general model (1). The detailed dynamics of this 3D model can be found in [3, 10].

In this paper, we are interested in the course keeping of ships, so we use the 3D dynamic model to find out the control algorithms with a concrete guidance such as the "Line-of-Sight" (LOS) [10].

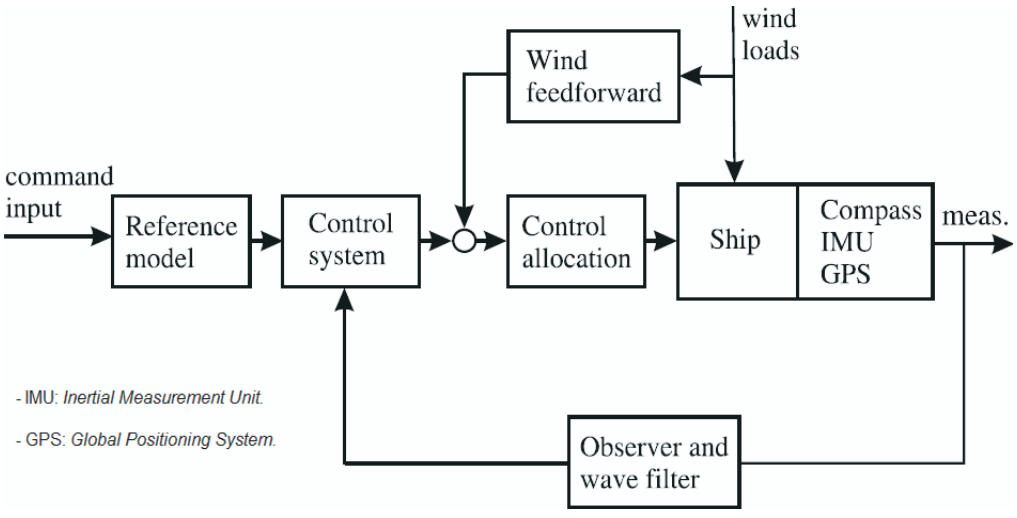

Fig. 1. Typical control structure of ships

From the definition of Hybrid Dynamic Systems (HDS) described in [1, 4] and the ship dynamic model-to-be used, we find that Autopilot Systems of Ships (ASS) are HDS whose dynamic behaviors can be modeled by Hybrid Automata (HA) [1, 5]. Because an ASS has the continuous/discrete parts and their interactions such as the motions in *surge*, *sway* and *yaw*, and external interacting events from the wind, waves and ocean currents. The typical control structure of an ASS is shown in Fig. 1.

The detailed dynamic models of ships in body and NED frames with marine control systems can be found in the excellent book of Fossen [3].

3. SPECIALIZING THE MODEL-DRIVEN ARCHITECTURE FOR A CONTROL SYSTEM OF SHIPS

3.1. Overview of Model-Driven Architecture

The Model-Driven Architecture (MDA) [11] is an approach to system development, which increases the power of models in that work. MDA contains three models to separate the specification of the operation of a system from the details of the way that system uses the capabilities of its platform:

- *Computation-Independent Model (CIM)*: A CIM is referred to as a domain or business model, the CIM presents the system at the highest level of abstraction.

- *Platform-Independent Model (PIM)*: A PIM is used by control system architects and designers to describe the control solution at a high level, independent of the solution's deployment platform.

- *Platform-Specific Model (PSM)*: A PSM specifies a combination between the details found in the PIM with the details representing how a solution can be implemented on a platform.

Furthermore, MDA supports also for model transformation [11]. The model transformation is the process of converting one model to another model of the same system. Transformations can use different mixtures of manual and automatic transformation.

3.2. General MDA process for an ASS

In fact, MDA applications are largely spread and appreciated in the control software industry; and the manufacturers have achieved great success in different domains [11, 18] such as the "*Infrastructure*", "*Business Applications*" and "*Devices and Embedded Systems*".

Starting from MDA specifications and characteristics of dynamic model of ships, we define here an executable process (Fig. 2), which permits us to carry out the development of an industrial ASS modeled by hybrid automata model and to re-use it in different ASS applications. This process includes the following main points:

- Object collaborations with UML (Unified Modeling Language) and Hybrid Automata (HA) present the CIM, which allow analyzing structure and behaviors of an industrial ASS.

- To describe a control system such as ASS with the HA's formalism [1] and carry out its evolution, we have introduced constraints and rules, which can be found in [4]. In CIM, HA are used to describe mathematical behaviors (i.e., the dynamic model of ships: *Situations, State Variables, Transition, etc.* of its HA) of this system.

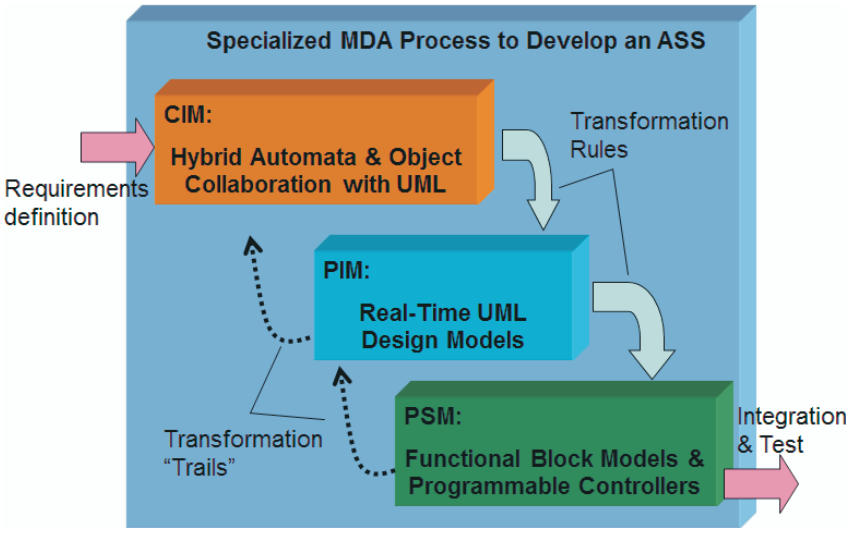

Fig. 2. Executable MDA process for developing an ASS

- Object collaborations with UML [11] permit the identified HA model to be exactly converted into business objects, which present control elements of the ASS being developed.

- Real-Time UML models indicate the PIM, which permit us to cover the design phase of the developed system. These models are described by using the "*capsules, ports, protocols*" concept [2, 8] that we adapted by specializing a set of capsules in precise behaviours.

- Functional block models are used to introduce the PSM of this system in order to carry out its implementation phase with concrete industrial platforms such as IEC61499 [14], and in programmable controller such as S7-200 [15].

- There are transformations rules, which allow the identified CIM to convert into a PIM, and to transform the PIM into a PSM. It also contains transformation trails, which permit the models can track their transitions.

This process will be gone into detail in next sections for describing the development phases of ASS control parts.

4. DETAILED EXECUTABLE MDA PROCESS TO DEVELOP AN ASS

4.1. CIM of an ASS

Main stages to build the CIM of an ASS are the followings:

- Identifying complex behaviors of the ASS being developed by using the use case model [11]. In this step, it is necessary to provide industrial constraint conditions (for example: maximum swing angles of rudder - *ThetaMax*) of ASS in order to make sure the precise operation of this system. Fig. 3 shows the main use case model of an ASS (MDS - *Measurement & Display System*; MES - *Marine Environment System* including disturbances such as the wind, waves, ocean currents etc...).

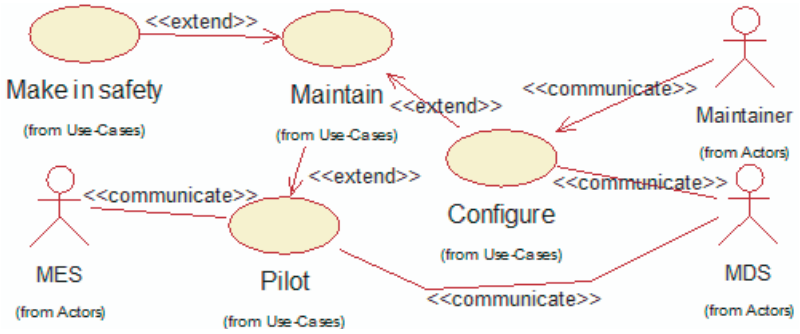

Fig. 3. Main use case model of an ASS

We find that the "*Pilot*" use case is oriented towards control modes; the complex behaviors of this case must be specified by using sequence diagrams and local state machines. In our model, we use 3D dynamic model of ships and LOS guidance because we are only interested in the course keeping [3, 10]. The detailed scenarios of the "*Pilot*" use case its corresponding local state machine can be found in [7].

- Defining the extended functional diagram [4], which permits us to model transformational activities of this system with events coming from outside. Starting from the considered dynamic model of ships, its industrial constraint conditions, and the identified use case model with LOS guidance, we bring out here an extended functional diagram of an ASS (Fig. 4).

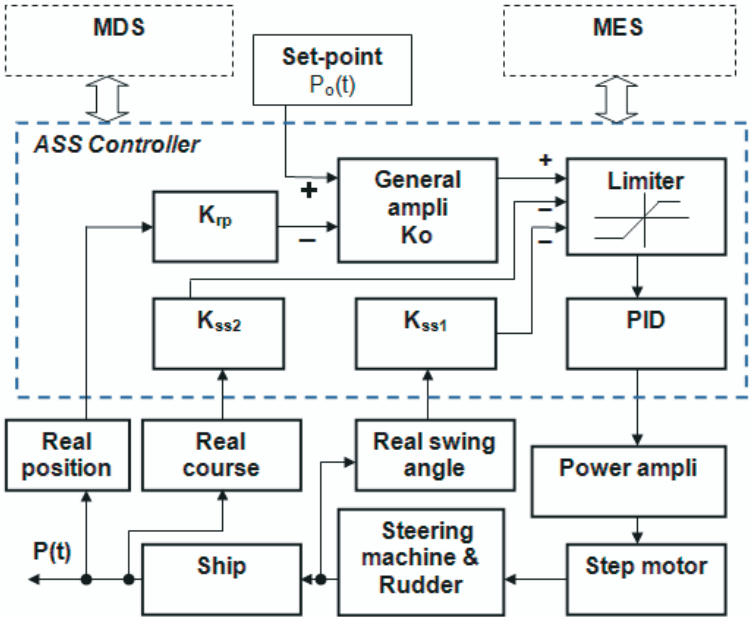

Fig. 4. Extended functional diagram of the ASS being developed

Here, K_{rp}, K_{ss1} and K_{ss2} are respectively the amplification coefficients of feedback signals including the real position, course and swing angle; $P_o(t)$ and $P(t)$ are the desired and real course or position of the piloted ship.

Then we build a global state machine in order to bring out the global dynamic behavior of the ASS being developed system from all local identified state machines. The specific rules which permit us to discover this global state machine can be found in [4]. The detailed global state machine, which corresponds with the above use case model and extended functional diagram, is shown in Fig. 5.

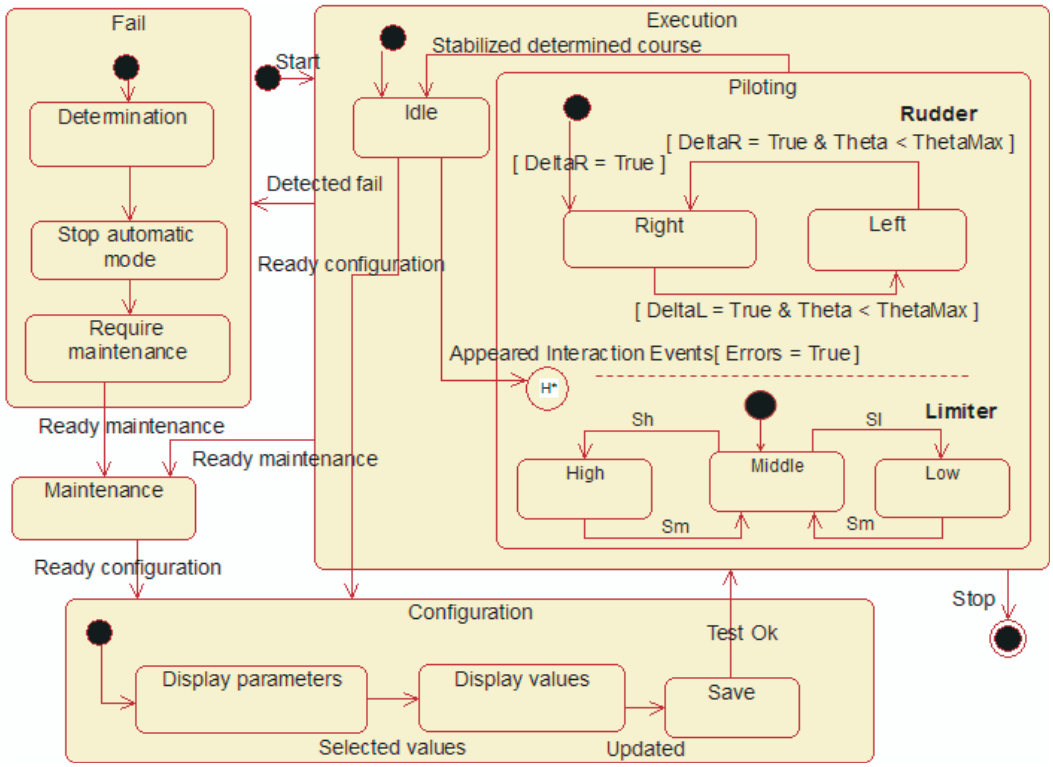

Fig. 5. Global state machine of the ASS being developed

Where: "*Left*", "*Right*" and "*Idle*" are the main states of the rudder (or step motor); "*High*", "*Low*" and "*Middle*" are the operational modes of the *Limiter* component; *Sm*, *Sh* and *Sl* are the internal generated events causing by the operational modes of *Limiter*. These events are handled in HA of this ASS.

- Specifying the hybrid automaton of an ASS; this includes situations, invariants, continuous state spaces, events, initial situation, initial continuous state, and continuous fluids. We have defined the steps and realization hypotheses, which can be found in [5, 7] to determine HA of the ASS being developed.

Within limit of the paper, we do not show here the detailed CIM results of this system, so all of these results can be seen in [7].

4.2. PIM of an ASS

4.2.1. *Global PIM's structure*

We find that the direct transformation of CIM to the implementation environment must be supplemented to carry out an industrial ASS and its re-use in the new application development phase. For example, the above identified CIM are not well adapted to visualize, model interconnection types between control objects or sub-systems. In the detailed design phase of an industrial ASS, we transform the identified CIM into PIM, which is based on transformation rules described in [6, 13] and uses the Real-Time UML notation [2, 8]. This PIM closes to implementation but not tied to a platform; because the Real-Time UML version includes the "*capsules, ports, protocols*" concepts that we adapted by specializing a set of capsules in precise behaviors to carry out completely this ASS and to re-use generic capsules in different ASS applications. From the approach described in [4], we proposed 5 main control capsules of PIM, which take part in the HA realization of the ASS being developed: the continuous part's capsule, discrete part's capsule, internal interface's capsule, external interface's capsule and Instantaneous Global Continuous Behavior (IGCB's capsule). The general communication structure of these control capsules is described by using the Real-Time UML component diagram (Fig. 6).

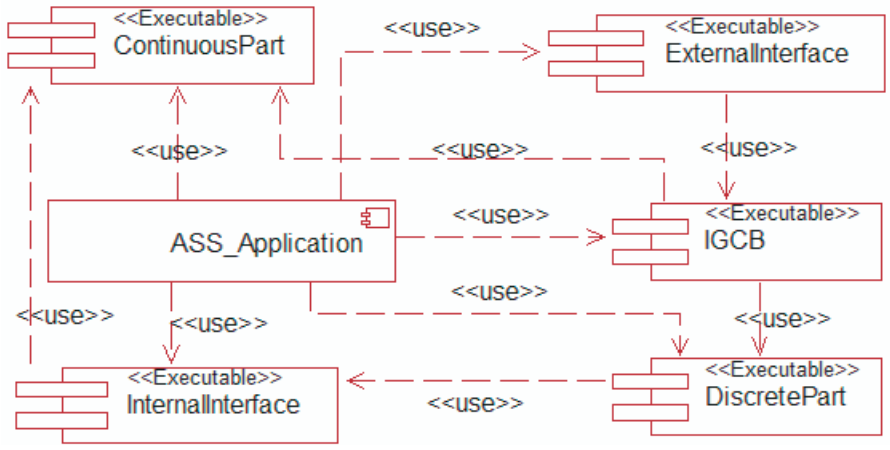

Fig. 6. General communication structure of main control capsules

4.2.2. *Detailed PIM*

- *The discrete part's capsule* contains a set of situations and transitions in HA of the ASS being developed (i.e., the *macro*-motion in *surge, sway* and *yaw*). This capsule also contains a state machine to make its own evolution with other capsules such as the internal interface's capsule and the IGCB's capsule and to treat the default internal event.

- *The continuous part's capsule* is related to transformational activities in the identified extended functional diagram such as *PID controller, Limiter, elements of feedback etc.* according to our case study. The continuous part's capsule also has a state machine to

make its own evolution with other capsules such as the IGCB's capsule and the internal interface's capsule.

- *The IGCB's capsule contains* concrete continuous fluids of the ASS being developed at time given in its HA. Each continuous fluid corresponds with a situation in this HA. The IGCB's capsule has a state machine to make its own evolution with other capsules such as the discrete part's capsule and the external interface's capsule. In this evolution, the IGCB's capsule exchanges periodic signals.

- *The internal interface's capsule* generates internal events of the ASS being developed, so that the discrete part's capsule can make its own evolution by these events. It has a state machine to make its own evolution with other capsules such as the continuous part's capsule and the discrete part's capsule.

- *The external interface's capsule* is an intermediary, which receives or sends episodic events and periodic signals between the developed ASS and their interacted systems such as MES and MDS in our case study. The external interface's capsule has a state machine to make its own evolution with other capsules such as the discrete part's capsule and the IGCB's capsule.

In addition, the re-use is very important for developing the general industrial control system; because it makes it possible to reduce the time, resource and development cost. We find different re-usable view in the development of industrial ASS such as:

- The re-use view based on the virtual mechanism of objects, classes, or class hierarch.

- The re-use view based on the capsule collaboration. For example, the generic state machine of identified main capsules, industrial constraints can be specialized to develop different industrial applications.

The specialization description, which makes it possible to re-use the capsule collaboration of a developed ASS in the new application for different types of ships, can be found in [4, 7].

The detailed PIM of this ASS has been shown in [7]. The validation and verification of this model have been corrected by using the "*IBM Rational Rose Real-Time & Architect Real-Time*" tools [8].

4.3. PSM of an ASS

To carry out an ASS, we can convert the above identified PIM into PSMs by using different specific technology or platforms such as Java, .NET, IEC61131, IEC64199 etc. in order to implement its PIM. But we are interested in the realization of an industrial ASS. We have decided to choose industrial platforms such as IEC61499 [14] to realize the implementation model of this ASS in a programmable logic controller such as S7-200 [15].

From the characteristics of capsules [2, 8], Functional Block (FB) instances of IEC61499 and specifications of S7-200 programmable controller, the PSM of an ASS is accomplished by applying the following transformation rules:

- Each capsule is implemented by an FB.

- Each sub-capsule is executed by a component FB, the super-capsule corresponds with a composite FB.

- Protocols are carried out by the input/output event and signal flow of FB instances.

- Ports are performed by a set of inputs and outputs, which can be associated to corresponding event/signal flow.

- Entity classes such as continuous elements or Instantaneous Global Continuous Behaviors (IGCB) are mapped to resource capacities, which are notified to the appropriate algorithms for realizing their evolution.

- State machines of the main control capsules are carried out by execution control functions, which notify the resource capacities to schedule an algorithm for the evolution.

All of PSM's elements have been produced to completely realize the above considered ASS. The detailed results can be are found in [7]. We have also tested the realization model of this application with the installation of a concrete model ship. This model ship had been built by the other research project [12]. The scenarios of tests are based on the above identified use case model and global state machine. All of tests are satisfied with the predetermined control performance.

5. CONCLUSION

We have presented a realization model to develop the Autopilot System of Ships (ASS). This model is based on the specialization of Model-Driven Architecture (MDA) in order to effectively implement control parts of this system. We indicated the following important points:

- Adapting the dynamic model and control structure of ships to gather the requirement analysis of an ASS.

- Specializing MDA's features to obtain an executable MDA process model to entirely develop this system.

- Specifying the Computation Independent Model (CIM) of an ASS to carry out its object-oriented analysis phase.

- Introducing the Platform Independent Model (PIM) for obtaining the object-oriented design model. This model can be customized and re-used in the new applications of ASS.

- Defining the Platform Specific Model (PSM) which is transformed from the identified PIM by applying the specific transformation rules, 'Functional Block' instances of IEC61499 standard and S7-200 programmable controller in order to carry out entirely the evolution of an ASS and to test it with the predetermined control performance.

In the future, we will develop completely our model with the connection to reality of real actuators, in order to deploy perfectly it in the ship engineering of Vietnam.

ACKNOWLEDGEMENT

This work has been completed by financial supports of scientific research projects at the Department of Ship Engineering and Fluid Mechanics, Hanoi University of Science and Technology.

REFERENCES

[1] Carloni L. P., Passerone R., Pinto A. and Sangiovanni-Vincentelli A. L., Languages and Tools for Hybrid Systems Design, *now Publishers Inc.*, (2006).

[2] Douglass B. P., Real Time UML: Advances in the UML for Real-Time Systems, *Third Edition, Addison-Wesley*, (2004).
[3] Fossen T. I., Guidance and Control for Ocean Vehicles, *John Wiley & Sons*, (1994).
[4] Hien N. V., Une Méthode Industrielle de Conception de Commande par Automate Hybride Développée en Objets, *Thèse de Doctorat, SUPMECA et Univertsité Marseille III, France*, (2001).
[5] Hien N. V., Soriano T., Implementing Hybrid Automata for Developing Industrial Control Systems, 8^{th} *IEEE - ETFA, Nice, France*, (2001).
[6] Hien N. V., Quang V. D., Vinh H. T., Soriano T., A General Implementation Model of Industrial Control Systems Using Real-Time UML and Functional Block, *VICA6, Vietnam*, (2005).
[7] Hien N. V. et al., A Method of Model-Driven Architecture to Develop Industrial Hybrid Dynamic Systems: Application to the Automatic Ship Steering Systems, *Final report of research project, code: B2010-01-354, HUST*, (2011).
[8] IBM Rational Software, IBM Rational online documentation, Redbooks and training kit, https://www.ibm.com/developerworks/university/, (2009).
[9] Karl-Heinz J., Tiegelkamp M., IEC 61131-3: Programming Industrial Systems, *Second Edition, Springer*, (2010).
[10] Lantos B., Marton L., Nonlinear Control of Vehicles and Robots, *Springer*, (2011).
[11] OMG, MDA & UML Specifications, *http : //www.omg.org/* (2008).
[12] Quang L. et al., Calculate, Design and Manufacture the Model Ship of Salvage, *Report of research project, code: B2009-01-291, HUST*, (2010).
[13] Quang L., Hien N. V., Soriano T., Specializing model-driven architecture to develop industrial Control systems, *5th SEATUC, Vietnam* (2011).
[14] Lewis R., Modeling Control Systems Using IEC 61499: Appling function blocks to distributed systems, *the institution of Engineering and Technology*, (2008).
[15] Siemens AG; S7-200 Programmable Controller System Manual, *Siemens AG*, (2002).
[16] Soriano T., Hien N. V., Using Objects Collaboration to Model the Control of an Industrial System, 7^{th} *IEEE - ETFA, Barcelona, Spain*, (1999).
[17] SNAME, Nomenclature for Treating the Motion of a Submerged Body Through a Fluid, *SNAME Technical and Research Bulletin 1-5, Jersey City, New Jersey, U.S.A.*, (1952).
[18] TURKI S., Ingénierie système guidée par les modèles: Application du standard IEEE 15288, de l'architecture MDA et du langage SysML à la conception des systèmes mécatroniquesrs, *Thèse de Doctorat, Université du Sud Toulon-Var, France*, (2008).

Received April 04, 2011