Abstract—This paper presents a hybrid control model, which is based on the Hybrid Automata (HA) and real-time Unified Modeling Language (UML) to systematically and homogeneously implement cooperative controllers for an Autonomous Surface Vehicle (ASV) combined with Multiple Autonomous Underwater Vehicles (MAUVs) operating in the team. The paper brings out the main points as follows: the dynamics of an individual underwater vehicle are adapted for control; the coordinated operation scenarios, hierarchical control architecture and the Hybrid Cooperative Control Model (HCCM) are proposed to capture the control requirements for an ASV-MAUVs team. The HA's features are specialized for modeling the global control algorithms of ASV-MAUVs cooperated in the team; the specialized HA of HCCM is then implemented for the team-based cooperative operations of the ASV-MAUVs team by using the real-time UML. Finally, a cooperative controller permits a team of a small-scale ASV combined with 03 AUVs to perform pre-determined search scenarios with the coordination mechanisms for ocean exploration, was designed and simulated. In this application, the implementation model was converted into the simulation environment by using an open-source platform of OpenModelica to quickly perform the simulation model for the controller. The obtained simulation results could allow us to verify and evaluate the proposed control model with good reliability and feasibility.

Index Term—Autonomous Surface Vehicles/Autonomous Underwater Vehicles (ASVs/AUVs), Cooperative Control, Model-Based Mechatronic Systems Design, Marine Engineering, Hybrid Automata, Real-Time UML.

Nomenclature

<table>
<thead>
<tr>
<th>ASVs</th>
<th>Autonomous Surface Vehicles</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUVs</td>
<td>Autonomous Underwater Vehicles</td>
</tr>
<tr>
<td>CSS</td>
<td>Command and Control Station</td>
</tr>
<tr>
<td>DGPS</td>
<td>Differential Global Positioning System</td>
</tr>
<tr>
<td>DIS</td>
<td>DGPS Intelligent Sonobuoys</td>
</tr>
<tr>
<td>DoF</td>
<td>Degrees of Freedom</td>
</tr>
<tr>
<td>HA</td>
<td>Hybrid Automata</td>
</tr>
<tr>
<td>HCCM</td>
<td>Hybrid Cooperative Control Model</td>
</tr>
<tr>
<td>HDS</td>
<td>Hybrid Dynamic Systems</td>
</tr>
<tr>
<td>IB</td>
<td>Integral Backstepping</td>
</tr>
<tr>
<td>IDE</td>
<td>Implementation Development Environment</td>
</tr>
<tr>
<td>IGCB</td>
<td>Instantaneous Global Continuous Behavior</td>
</tr>
<tr>
<td>LOS</td>
<td>Line-Of-Sight</td>
</tr>
<tr>
<td>OMG</td>
<td>Object Management Group</td>
</tr>
<tr>
<td>SNAME</td>
<td>Society of Naval Architects and Marine Engineers</td>
</tr>
<tr>
<td>SysML</td>
<td>Systems Modeling Language</td>
</tr>
<tr>
<td>UKF</td>
<td>Unscented Kalman Filter</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
</tbody>
</table>
of complex systems. However, both UML and SysML lack the constructs for modeling time and duration constraints of the industrial control system. For these reasons, the real-time UML version [19-23], which includes the ‘capsules, ports, protocols, connectors’ concepts, can be specialized in order to implement in detail the design artifacts for real-time and embedded control systems, e.g. the cooperative controller of an ASV-MAUVs team.

This study is focused on the development of a control model integrated HA with the real-time UML features, which can permit us to conveniently implement the cooperative controller of an ASV-MAUVs team. This cooperative controller, which permits an ASV combined with the MAUVs group to be deployed for performing quickly missions in the wide range of actions, in order to improve the efficiency of ocean exploration and survey. In our model, coordination operation scenarios, hierarchical control architecture and hybrid cooperative control model are specified to gather the requirements of control system; HA features [10-16] are specialized to globally model the behaviors of ASV-MAUVs coordination, as well as the real-time capsule collaboration performed by using the real-time UML [19-23] in order to depict the detailed implementation components. Finally, a cooperative controller of a team consisting of a small-scale ASV combined with 03 AUVs was completely designed and simulated to perform predetermined cooperative scenarios for ocean exploration and search. In this application, the implementation model are converted into the simulation model by using an open-source platform of OpenModelica [24] based on Modelica language [25] to quickly perform the simulation model for the controller.

The structure of this paper is organized as follows: Section II introduces the overview of dynamics an individual underwater vehicle for control. Section III proposes the coordinated operation scenarios and industrial hybrid control structure for an ASV-MAUVs team. In Section IV, the hierarchical control architecture and hybrid cooperative control model are defined to carry out the coordinated operations of an ASV-MAUVs team. Section V presents the specializations of HA and real-time UML to completely implement the HCCM for an ASV-MAUVs team. Following this approach, the simulation model, which is implemented for a cooperative controller of a team consisting of a small-scale ASV combined with 03 AUVs for performing the concrete coordination scenarios, is presented in Section VI. Finally, the conclusions and future works are made in Section VII.

II. OVERVIEW OF DYNAMIC MODEL AND CONTROL STRUCTURE FOR AN INDIVIDUAL UNDERWATER VEHICLE

A. Dynamic model for an individual underwater vehicle

According to SNAME [26] and the large field of guidance, navigation and control of vehicles, the 6 DoF dynamic model of an underwater vehicle in body frame [15, 16, 27-33] can be written in equation (1).

\[
\begin{align*}
\dot{\eta} &= J(\eta)v \\
(Mv + C(v)v + D(v)v + g(\eta)) &= \tau(v, u)
\end{align*}
\]

(1)

Where: \(\eta=[x, y, z, \phi, \theta, \psi]^T\) includes the position (NED: North, East and Down) and the orientation (Euler RPY: Roll, Pitch and Yaw angles); \(v = [u, v, w, \rho, \alpha, \beta]^T\) is composed the linear and the angular velocities; \(M = M_{RB} + M_{A}\) is a mass matrix, which denotes the 6x6 system inertia matrix containing \(M_{RB}\) - the generalized constant inertia matrix, and \(M_{A}\) - the added mass inertia matrix; \(C(v) = C_{RB}(v) + C_{A}(v)\) is the 6x6 Coriolis and centripetal forces matrix including added mass; Linear and nonlinear hydrodynamic damping are contained within the 6x6 matrix \(D(v) = D_{h} + D_{n}(v)\), \(D\) contains the linear damping terms, and \(D_{n}(v)\) contains the nonlinear damping terms; \(g(\eta)\) is the 6x1 vector of gravitational and buoyancy effects; \(\tau(v, u)\) is the vector of resultant force and moment acting on the underwater vehicle, and \(u\) is the control inputs, e.g. the rotational speed of the motors related to the generated thrusts, the driving angles sent to the needed servo-motor for sail planes and rudder.

A discrete state-space representation is required for modeling the individual AUV controller in order to use a recursive digital motion estimation filter, e.g. the Unscented Kalman Filter (UKF); the developed system can be then described by a set of equations (2).

\[
\begin{align*}
\dot{x}_{k} &= f_{k-1}(x_{k-1}, u_{k-1}) + w_{k-1} \\
y_{k} &= h_{k}(x_{k}) + v_{k}
\end{align*}
\]

(2)

Here, \(x = \begin{bmatrix} \eta \end{bmatrix}\) is a 12-dimensional state vector for describing the motion of AUV, while \(x_k\) is the vector of state variables at the kth instant of \(x\); \(u_k\) and \(v_k\) respectively are the inputs and outputs of the system; \(h_k\), \(w_k\) and \(v_k\) are the respectively the measurement function, additive process and measurement noise; the first equation in (2) is called the system’s evolution equation, while the second one is called the measurement equation.

The details of the 6 DoF dynamic model and state-space model of underwater vehicles for control can be found in [27, 34].

B. Control structure of an individual underwater vehicle

The general autonomy architecture of an underwater vehicle basically consists of three main sub-systems as follows: the guidance sub-system, navigation sub-system and control sub-system. Here, the guidance sub-system is responsible for producing the desired trajectory for the vehicle to follow. This task can be completed by implementing a common guidance approach based on the generations of way-points that take the desired way-points defined pre-mission and, with the possible inclusion of external environmental disturbances, generates a path for the vehicle to follow in order to reach each successive way-point. The navigation sub-system addresses the task of determining the current state of the vehicle. The control sub-system is responsible for providing the corrective signals and events to enable the vehicle to follow a desired path. This is achieved by receiving the desired state of vehicle from the guidance sub-system, and the current state of vehicle from the navigation sub-system. It then calculates and applies correcting forces and moments, through use of the various actuators on the vehicle, to minimize the difference between desired and current states.
This allows the vehicle to track a desired trajectory even in the presence of unknown disturbances, which can be ocean currents, winds, waves, etc. All three of these sub-systems have their own individual tasks to complete, yet must also work cooperatively in order to reliably allow the vehicle to complete its objectives.

In addition, control systems and their actuators take into account models with discrete events and continuous behaviors that can be called HDS. The behaviors of HDS 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 do not always have the same behavior because they are associated with validity hypotheses to check at any moment; the security forces to envisage events, and behaviors different from nominal behaviors [10-16].

From the above described dynamic model (1) and general control structure of an underwater vehicle together with defined characteristics of HDS, we find that controllers of this single vehicle are HDS whose dynamic behaviors can be modeled by HA. These controllers have the continuous/discrete parts and their interactions such as the motions in surge, sway, heave, roll, pitch, and yaw, and external interacting events from the guidance and navigation system and environmental disturbances.

III. PROPOSAL OF COORDINATED OPERATION SCENARIOS AND INDUSTRIAL HYBRID CONTROL STRUCTURE FOR AN ASV-MAUVS TEAM

A. Coordinated operation scenarios of an ASV-MAUVs team

Fig. 1 shows out a coordinated operation structure for presenting the cooperative model to implement the controller of an ASV-MAUVs team.

![Fig. 1. Coordinated structure of an ASV-MAUVs team.](image)

Here, the Command and Control Station (CCS) installed on the mother ship periodically requires the gathered information from the ASV, and also commands the ASV-MAUVs to survey some particular regions of interest. The AUVs carry out the exploration mission and periodically provide the information to the ASV. Once the information is transferred from the ASV to the CSS, the CSS may provide a new path to the AUVs through the ASV for exploration. The ASV periodically meets the MAUVs, collects the information, and returns to the CCS for providing the acquired information.

The communication links between the CCS and ASV can be carried out by RF XTend combined with the Differential Global Positioning System (DGPS). Furthermore, the ASV is also considered as an acoustic navigation vehicle combined with one higher cost central AUV (i.e. master AUV) based on DGPS Intelligent Sonobuoys (DIS) to provide several different types low-cost AUVs (i.e. slave AUVs) with navigation information. Using the underwater DGPS concept together with a set of intelligent surface sonobuoys, the precise position of the master AUV carrying an acoustic pinger, could be estimated by the measured time of arrival of acoustic signals and the DGPS positions of sonobuoys. Hence, the ASV always conveniently moves above to the master AUV that permits the master AUV to remain inside the projected area of communication of the ASV, and to get the precise position from the ASV. With this coordinated structure and scenarios, the master AUV could get accurate position from the CCS and ASV, without coming up to the surface. The above coordinated operation scenarios also permit the master AUV to calibrate its positions (e.g. the trajectory-tracking), which would severely disturb or even deteriorate the whole strategy of the team coordination and formation, besides the unwanted energy consumed to emerge to the surface.

B. ASV-MAUVs team with industrial hybrid dynamic systems

As the previously stated, control systems and their actuators can be considered as models combined with discrete events and continuous behaviors that can be named as a HDS. In general, a HDS has a continuous evolution and occasional jumps; the jumps correspond to the change of state in an automaton that transits in response to external events or to the continuous evolution. A continuous evolution is associated to each state of the automaton by means of ordinary differential equations. The structure of the equations and the initial condition may be different for each state. In this paper, we are also interested in designing a HDS in the industrial context. This HDS can be called as an Industrial HDS (IHDS), and contains two parts, which are the HDS controller and controlled HDS. These parts mutually exchange periodic signals and episodic events. The episodic event is either external or internal. Fig. 2 shows out the block diagram of an IHDS.

![Fig. 2. Block diagram of industrial HDS (IHDS).](image)

Here, $E_0$ and $E_1$ are respectively output and input events; $S_0$ and $S_1$ are respectively output and input signals; $\Delta T$ is a sampling period of the evolution model for control; and $\text{Actor}_1$, $\text{Actor}_2$, ..., $\text{Actor}_n$ are descriptions of a coherent set of
roles that users (i.e. persons or involved external systems) play when they interact with the developed IHDS. An IHDS and its actors asynchronously exchange messages that can be carried out by a state machine. The controlled HDS may evolve along with several models from the industrial control perspective; interactions between these models can be presented by using one of formalisms such as HA, Hybrid Grafcet, Hybrid Petri Nets, etc. [35-37].

From the above coordinated operation scenarios of an ASV-MAUVs team, the dynamic models for control of each individual underwater vehicle (1) together with the characteristics of IHDS, we also find that cooperative controllers of an ASV-MAUVs team are IHDS whose dynamic behaviors can be modeled by HA. These controllers have the continuous/discrete parts and their interactions such as the motional components of each individual vehicle in the team, the external interacting events from the CCS and ASV, the internal events exchanged between the master AUV and the slave AUV, etc.

IV. HYBRID ANALYSIS MODEL FOR THE COOPERATIVE CONTROLLER OF AN ASV-MAUVs TEAM

A. Hierarchical control architecture for an ASV-MAUVs team

As previously discussed, the heterogeneous ASV-MAUVs team is coordinated in such a strategy, that one central AUV as a master, leads some low-cost AUVs as slaves, with the help of one ASV and some sonobuoys providing accurate acoustic navigation. Hence, an object-oriented hierarchical control structure could be proposed for the ASV-MAUVs team, and depicted in architectural layers, as shown in Fig. 3, by using the class diagram in the real-time UML. In the real-time UML models, a stereotype is a model element that is an extensibility mechanism, which can be used to identify the purpose of the model element to which we apply it. For example, the <<master>> stereotype can be applied to the abstract AUV class in order to indicate that it is an instruction function for the AUV as a master AUV.

![Class diagram](image)

Fig. 3. A class diagram of hierarchical control model for an ASV-MAUVs team.

Here, the AUV class (the master AUV) handles the event-based coordination (e.g. the “Handle()” function), by assigning new way-points for the AUV_i, AUV_j, ..., AUV_n classes (the slave AUVs in a team). The dynamic model described by (1) of the i-th slave AUV may not be the same as that of the j-th slave AUV, i ≠ j = 1..n, n>1. The master AUV determines when the mission is completed and when the whole system needs to reconfigure. The controller of each vehicle (master or slave AUVs) takes care of the cooperation with the other AUVs and takes decision on the event-based coordination of the whole system. The controller also gets the specified computational tasks, e.g. the “Operation()” function, by the master AUV issued from the CSS and ASV classes. It computes low-level control commands and executes them on each AUV, the AUV could then arrive at a desired position with optimal velocities. In addition, the Integral Backstepping (IB) techniques implemented in [29, 38] and the standard navigation filters based on the UKF algorithm [3, 39-41] are hierarchically applied to the individual dynamic model (1) and its state-space model (2) in each individual AUV controller for estimating and controlling the depth, position, attitude and velocity of each corresponding AUV (master and slave AUVs). As previously stated in Sub-section III.A, it is supposed that the central AUV which gets the accurate DGPS positions issued from the ASV and CSS (i.e. the CSS and ASV classes) acting as the master AUV. The master AUV provides inputs to the slave AUVs, and vice versa. The interaction between master and slave AUVs can be performed by the low-cost underwater communication network.

B. Hybrid cooperative control model (HCCM) for an ASV-MAUVs team

The global continuous model of an ASV-AUV team based on the individual dynamic model described by (1) can be generally built by considering a set \( F = [F_1, F_2, ..., F_i, ..., F_{n'}] \) of n>1 Autonomous Vehicles (AVs). The dynamic property of the i-th AV may not be the same as that of the j-th AV, i ≠ j = 1..n, n>1, i.e. these n AVs also set up a heterogeneous system in the ASV-MAUVs team. The dynamic model for control of each AV can be modeled as the following nonlinear system (3) [6].

\[
F_i(t) = f_i(F_i(t), u_i(t))
\] (3)

Here, \( F_i(t), u_i(t) \) and \( f_i \) are respectively the continuous state, the admissible control value or state feedback and vector field, which define the dynamic model of the i-th individual AV described by (1). With the soft computing technique combined with various control laws (e.g. the IB techniques and UKF algorithm), the AVs could arrive at the desired position from one waypoint to another.

The global discrete model of an ASV-MAUVs team can be realized by an event-based controller, and generates a set \( W = [W_1, W_2, ..., W_n] \) of way-points. The team coordination is
then defined and updated by the following law:

\[ W_i(t+1) = \Psi(W_i, t, e) \]  (4)

Where: \( e \) is an event that is triggered when all AVs arrive at the desired position; \( t \) is the time step; \( W_i(t+1) \) indicates the next value of \( W_i \); finally, \( \Psi \) is the team coordination strategy, e.g., the coordinated scenarios. The control \( u_i \) is derived for the \( i^{th} \) AV based on \( W_i(t) \) and \( W_i(t+1) \).

An interaction between the global discrete and continuous models can be carried out by the control \( u_i \) because it depends on both the continuous behaviors and the state of the discrete model; the interaction is determined by event \( e \) as well as providing a set of coordination commands (5) corresponding to waypoints \( W_i \).

\[ u_i = \varphi_i(W_i, e) \]  (5)

Here, \( \varphi_i \) is the interaction function in the team coordination strategy. It should be noted that all AVs observe the same enabling event \( e \), which is triggered when all AVs have reached their previously computed way-points.

V. DESIGN MODEL FOR THE COOPERATIVE CONTROLLER OF

AN ASV-MAUVs TEAM

A. Specializations of HA for the HCCM of an ASV-MAUVs team

The evolution of the above defined Hybrid Cooperative Control Model (HCCM) for an ASV-MAUVs team can be carried out by using the HA’s formalism because HA has only one global continuous behavior at time given, contains the invariant notation to verify hypotheses on the continuous state, is derived from an automaton modeling also the dynamic behavior of interactive software systems, and can be verified with proof tools such as HyTech, CheckMate [11] and OpenModelica [24].

A HA of HCCM is defined by the following data:

\[ H_{HCCM} = (Q, X, \sum, A, Inv, \Phi, Q_0, X_0) \]  (6)

Where:

- \( Q \) is a set of states describing operational modes of \( H_{HCCM} \), e.g., the system Coordination, system Reconfiguration, system Motion, system Stop and system Idle, which are combined with an applicable state machine issued from the coordinated scenarios (i.e. the team coordination strategy \( \Psi \)).
- \( Q \) can be called situations of the applicable controller of ASV-MAUVs team; \( Q_0 \) is the initial situation.
- \( X \) presents the continuous state space of \( H_{HCCM} \).
- \( \sum \) is a finite set of events, e.g., the external interacting events from the ASV linked to the CSS and the internal event \( e \) triggered for \( W_i \) in the HCCM.
- \( A \) is a set of transitions between \( q \in Q \) and \( q' \in Q \). It is defined by \( (q, Guard, a, Jump, q') \) and represented by an arc between situations, here: Guard is a subset of the state space in which the continuous state must be, so that the transition can be crossed; Jump represents the continuous state transformation during the change of situation; it is expressed by a state value function, whose result is affected like initial value of the continuous state in the new situation.
- \( Inv \) is an application for the interaction function \( \varphi_i \) of the HCCM which associates a subset of the state space to each situation; it is called the invariant of the situation, in which the continuous state must remain, when the situation is \( q \), the continuous state must verify \( X_i \in inv(q_i) \).
- \( \Phi \) is defined by using the global continuous model \( F \) of the HCCM for each situation; the evolution of continuous state is occurred when the situation is activated.

Starting from the above discussed points: the coordinated operation scenarios, hierarchical control architecture, HCCM and HA specialization, the ASV-MAUVs team coordinately has an evolution, which is based on hybrid automata as shown in Fig. 4.

![Fig. 4. HA evolution of the HCCM for an ASV-MAUVs team.](image)

Here, the evolution constraints are applied as follows: the interaction function \( \varphi_i \) and \( \sum \) are considered in term of inputs/outputs and internal/externality; \( X \) contains input/output signals applied to globally perform the HA evolution. In addition, the realization hypotheses of the HA evolution, which permit the invariant Inv and guard control Guard can generate internal events for the HA of the HCCM, are summarized in Table 1.

<table>
<thead>
<tr>
<th>Inv and Guard</th>
<th>System evolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>If ( X_i \in Inv(q) ) and Guard(a)=True, then there is a generated internal event, and the system changes to the situation ( q' ) described in a set of situations of the AUV with Jump identified by the initial value of the continuous fluid ( \Phi_q ).</td>
<td></td>
</tr>
<tr>
<td>If ( X_i \in Inv(q) ) and Guard(a)=True, then the system remains its actual situation ( q ).</td>
<td></td>
</tr>
<tr>
<td>If ( X_i \in Inv(q) ) and Guard(a)=False, then the system remains its actual situation ( q ).</td>
<td></td>
</tr>
<tr>
<td>If ( X_i \in Inv(q) ) and Guard(a)=False, then there is a generated internal event; the system changes into the situation ( q'' ), which is called the irreversible default situation.</td>
<td></td>
</tr>
</tbody>
</table>

B. Implementation model of the specialized HA of HCCM for

an ASV-MAUVs team

We find that the direct transformation of the specialized HA for a HCCM to the implementation environment must be supplemented to carry out an ASV-MAUVs team and its reusability in the new application development phase. For example, the above identified HA are not well adapted to
visualize, model interconnection types between control objects or sub-systems. In the detailed implementation model of this system, we transform the identified HA into the real-time object paradigm, which is based on the real-time UML profile and IBM Rational Rose RealTime or IBM Rational Software Architect RealTime tools [42]. The goal of the implementation model is to closely build up the real-time control capsules, ports, protocols and their timing concurrency of evolutions, which allow designing the precise behaviors and structures of the HCCM for an ASV-MAUVs team. We have specialized here the 5 main control capsules, which take part in the HA realization of the HCCM: 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). Fig. 5, Fig. 6 and Fig. 7 indicate respectively the real-time communication pattern, its structure and behaviors of main control capsules by using the real-time UML’s collaboration, class and sequence diagrams.

- **The discrete part’s capsule** contains a set of situations \( Q \) and transitions \( A \) in the HA of the HCCM (i.e. the macro-system operations in Coordination, Reconfiguration, Motion, Stop and Idle, which are combined with an applicable state machine issued from the coordinated scenarios, e.g. the team coordination strategy \( \mathcal{P} \)).
- **The continuous part’s capsule** contains entity classes to store and process continuous model of each AUV, which combine with the continuous state-space \( X \), e.g. the continuous components \( F_i \) of the \( i \)th individual AUV described by (1).
- **The IGCB’s capsule** contains entity classes, which present the instantaneous global continuous behaviors (continuous fluids \( \Phi \) in the HA of the HCCM); each continuous fluid \( F \in \Phi \) is connected to a concrete situation \( q \in Q \). There is only one concrete global continuous behavior activated at time given.
- **The internal interface’s capsule** verifies the Inv in the HA of the HCCM, and generates internal events such as the internal events exchanged between the master AUV and the slave AUV; so that the discrete part’s capsule can make its own evolution by these events.
- **The external interface’s capsule** is an intermediary, which receives or sends episodic events and periodic signals between the developed ASV-MAUVs team and their interacted systems, e.g. the external interacting events and signal from the CCS and ASV.

---

**Fig. 5. Real-time communication pattern for the main control capsules for the HA of HCCM.**

**Fig. 6. Structure of main control capsules, ports and protocols for the HA of HCCM.**
In Figure 7, the messages exchanging between the main control capsules are synchronous, and the interval between two adjacent timeout messages indicates the sampling period (AT) of the IGCB’s capsule. The external interface’s capsule receives period signals coming from external continuous components. It then gives the BuildingGlobalContinuousModel message to the IGCB’s capsule so that the IGCB’s capsule can call all of the continuous models of participated AUV’s in the team corresponding to the concrete ‘IGCB: IGCB1’. During the call of the IGCB’s capsule, the external interface’s capsule can receive an event named ‘3: InputEvent’ issued from the master AUV issued from ASV and CSS, and gives this event named ‘ee1: DetectedEvent’ to the discrete part’s capsule. The discrete part’s capsule then memorizes and later processes this event. If the IGCB’s capsule receives the LastSlaveAUV message coming from the continuous part’s capsule, then it gives the All_AUV_Evolution message to the continuous part’s capsule so that the internal interface’s capsule can receive all updated variables. The internal interface’s capsule then verifies the invariant \((X_q \in Inv(q))\) of the situation \(q\); in this case, there is a generated internal event. The internal interface’s capsule gives this event to the discrete part’s capsule that permits the IGCB’s capsule to identify the concrete ‘IGCB: IGCB2’ and give output signals to the external interface’s capsule. At the end of this sampling period, the external interface’s capsule gives the output event and control signals to the external environment of the ASV_MAUV team operating with its concrete ‘IGCB: IGCB2’.

![Diagram](image-url)/Fig. 7. Behaviors in one sampling period (AT) of main control capsules for the HA of HCCM.

In addition, the reusability is very important to implement controllers for different applications of ASV-MAUVs teams because it permits the development time and cost to be reduced. The various reusable views in the development phase are considered as follows: The reusable view is based on the virtual mechanism of objects, classes or class hierarchies; the other re-use view can be based on design components, e.g. the continuous model and main control capsules that can be specified to develop various control applications of ASV-MAUVs teams using the same technique. The specializations, which permit the capsule collaboration of the developed ASV-MAUVs team to customizable and reusable in the new control application for various ASV-MAUVs teams, could be found in the author’s reports [43, 44].

The validation and verification of this implementation model and its traceability have been corrected by using IBM Rational Rose RealTime or IBM Rational Software Architect RealTime tools [42]. IBM Rational’s leading role in defining the real-time UML is widely acknowledged, as is the preeminence of the IBM Rational Rose RealTime product in implementing UML to support the architecting of large-scale real-time and embedded software systems. It combines a rich modeling environment with a code-oriented tool set to create a comprehensive practitioner desktop for creating solutions in a variety of architectural styles, and targeted at specific runtime infrastructures. Many other important lifecycle artifacts also
benefit from this tool (e.g. requirements lists, test cases and build scripts) to entirely cover development phases for the cooperative controller of ASV-MAUVs team.

VI. APPLICATION

Based on above proposed approach, the simulation model was implemented for a cooperative controller of a small-scale ASV combined with 03 AUVs for performing the coordination scenarios described in Sub-section III.A. The desired coordinated control behavior in this application is MAUV flocking like birds flying in loose formations, which is useful for underwater collaborative operation. We have considered a simplified model of each AUV constraining to move in a horizontal plane. There are three basic elements to maintain MAUV flocking: i) Cohesion: attraction to distant neighbors up to a reachable distance. ii) Separation: repulsion from neighbors within minimal distance. iii) Alignment: velocity and average heading matching with neighbors. This is a small part of our long-term research project (code: 107.03-2019.302) funded by National Foundation for Science and Technology Development (NAFOSTED) of the Ministry of Science and Technology of Vietnam, which is led by the authors at Hanoi University of Science and Technology (HUST), Vietnam [45].

The simulation model was performed by using OpenModelica [24] software in this application. OpenModelica is an open-source modelling and simulation environment intended for industrial and academic usage. It is an object-oriented declarative multi-domain modelling language for complex dynamic systems. The OpenModelica environment allows most of the expression, algorithm, and function parts of Modelica to be executed interactively, as well as equation models and Modelica functions [25] to be compiled into efficient C++ codes. The generated C++ codes are combined with a library of utility functions, a run-time library and a numerical Differential Algebraic Equation (DAE) solver. The obtained simulation results in OpenModelica permits us to theoretically evaluate the control performance and functionalities, and to easily optimize the control design elements before they are implemented and deployed.

The transformations rules, which are used to convert the implementation model described in Sub-section V.B into the simulation environment of OpenModelica, and vice versa, through the round-trip engineering of the intermediate C++ codes as follows:

- Each capsule is realized by a class or a block model;
- Each sub-capsule is carried out by a component class or block model; the super-capsule corresponds to the composite class or block model;
- Messages are implemented by the “functions” of classes or block models;
- Interfaces are realized by the set of inputs and outputs of a block model;
- Passive classes such as continuous elements or Instantaneous Global Continuous Behaviors (IGCB) are mapped to the “expressions” terms;
- State machines of the main capsules can be performed by state graphs.

Furthermore, the HA evolution of HCCM (Fig. 4) for this application can be automatically implemented in the object-oriented convention by using the common State Pattern described in. This pattern allows an object to alter its behavior when its internal state changes; the object will appear to change its class.

Fig. 8 illustrates the velocity transients in a MAUVs flock due to the velocities of two slave AUVs in convergence corresponding to the velocity of the master AUV at 1.0m/s received from the ASV linking to the CSS.

![Image](image.png)

Fig. 8. Velocity convergence in the “3 AUVs flock” case.

All of obtained simulation results permit us to theoretically evaluate the control performance of this system within the control criteria such as the admissible timing response, transition, static errors and run-time concurrency in the team, and to evidence a good reliability of this approach. From that point, we can decide to choose the designed control elements and their properties in order to accurately deploy the realization model of the above application [44]. The test bed...
and detailed experimental scenarios are currently built up to realize the control performances and features of this application in the laboratory of fluid dynamics and automation engineering.

VII. CONCLUSIONS AND FUTURE WORK
The paper has introduced a hybrid control model to develop cooperative controllers for an ASV-MAUVs team that can be used for performing quickly missions in the wide range of actions in order to improve the efficiency of ocean exploration and survey. This model is based on the specialization of and Hybrid Automata (HA) and real-time UML to entirely cover the analysis, design and implementation phases for the cooperative controller of an ASV-AUAVs team. The paper contains the following main points:

- Adapting the dynamic model and control structure for an individual underwater vehicle;
- Specifying the coordinated operation scenarios for an ASV-MAUVs team to gather control requirements, and to link them to the characterististics of Industrial Hybrid Dynamic Systems (IHDS);
- Defining the Hybrid Cooperative Control Model (HCCM) for an ASV-MAUVs team;
- Specializing the HA to design in detail the HCCM that permits the ASV-AUAVs cooperatively to operate in a team;
- Constructing the implementation model that is closely build up by the real-time control capsules, ports, protocols and their relationships, which allow modeling the precise behaviors and structures of the HA of HCCM for an ASV-MAUVs team.

Finally, a cooperative controller of a team consisting of a small-scale ASV combined with 03 AUAVs was designed and simulated to illustrate a good reliability of the proposed hybrid cooperative control model.

Furthermore, using the approach described in this paper, development engineers will be more capable of managing the system complexity through the visual modeling of artifacts and their transformations in the development lifecycle of system.

In the next time, we will describe the steps taken to transition from theory to practice and the results of actual tests performed with 03 physical AUAVs having different dynamic properties, in the Summer of 2020.

ACKNOWLEDGMENT
The authors would like to thank the financial supports of 107.03-2019.302 project funded by National Foundation for Science and Technology Development (NAFOSTED), from the Ministry of Science and Technology of Vietnam.

REFERENCES
[26] SNAME, Nomenclature for Treating the Motion of a Submerged Body through a Fluid, Technical and Research Bulletin No. 1-5, SNAME (the
Society of Naval Architects and Marine Engineers), New York 18, N. Y., USA, 1950.


