4 Architecture Design

4.1 Requirements Definition

This Subsection presents how the user and system requirements are defined and specified by analysing different representation models.

4.1.1 Overview of the System Context

This first part of Section 4 summaries the basic ideas and aims for the AMR system in order to extract key requirements from the user and the AMR system for the intelligent control system architecture. In addition, it also addresses the background, framework conditions, and other relevant information from the project environment.

The aim for the AMR system is to perform the missions given by the stakeholder. These missions are described in Subsection 5.3 of this book. To achieve the mission goals, the AMR platform is required to have certain capabilities (generally, multipurpose dexterous manipulation capabilities for intervention operations in unknown, unstructured and underwater environments).

The design and development of the embedded knowledge representation framework and the high-level reasoning agents is required in order to enabling autonomy and on-board decision making of the marine vehicles. To empower the agent technology, the agents are required to be service-oriented entities so that they have the operational flexibility given by SOA, i.e. it provides plug & play facilities that make it possible to integrate agents' capabilities and facilities the diagnosis of available capabilities.

4.1.2 Stakeholder Requirements

The primary stakeholder of the AMR is the end user. The end-user requirements (what the system should be able to do) are the following:

The platform components should be able to advertise their capabilities.
The system should discover all the capabilities of the platform.

C.C. Insaurralde, *Intelligent Autonomy for Unmanned Marine Vehicles,*
Studies in Systems, Decision and Control 29, DOI: 10.1007/978-3-319-18778-5_4

The user and marine vehicles should collaborate with each other in order to achieve goals.

The marine vehicles should be autonomous. At least a high degree of autonomy.

The system should be automatically reconfigurable in order to deal with changes when planning missions.

The marine vehicles should be able to communicate with each other and the operator.

The vehicles of the system should provide internal knowledge representation about the context where it is working.

4.1.3 Systems Requirements

Figure 4.1 shows the AMR system context. The AMR system inhabits an environment where it interacts with the end user and probably with other systems. As above mentioned, the AMR system has one or more use cases where it fulfils missions. Every system has its architecture. In particular, the intelligent control architecture of the AMR is developed in the current document.

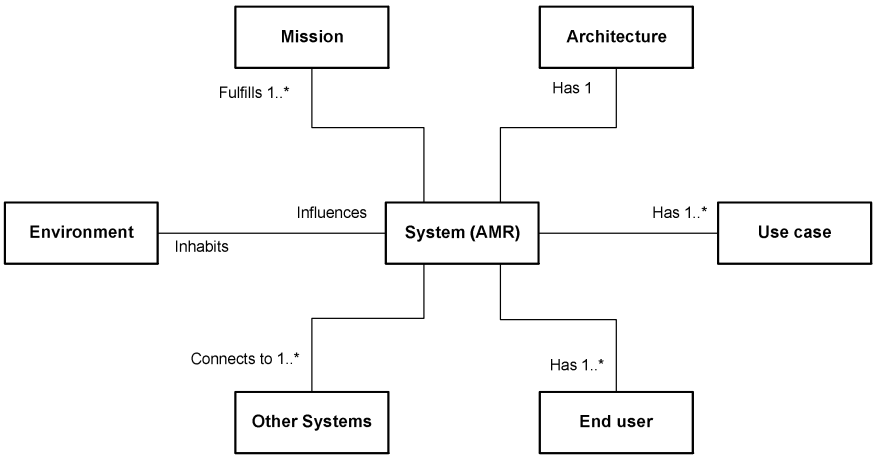

Fig. 4.1 System context

Table 4.1 presents the External AMR system interfaces with the environment. It basically shows the perceptions, and actuations by means of which the AMR system interacts with the near environment, the end user, and other systems.

As mentioned in Chapter 1, the mission proposed by TRIDENT [1] for the AMR system is a multipurpose generic intervention. It is divided into two phases: survey and manipulation. Following the proposal, as system requirement for the intelligent control architecture, the above missions are implemented as three different sub-missions: seabed survey, target selection, and object manipulation.

Table 4.1 AMR system interfaces

	Perception	Actuation	Interaction with
OCU	End user input	Scene and object display	End user
ASC	Local pose	Wrench effort command*	
	Velocity state		Environment
IAUV	Local pose	Wrench effort command*	
	Velocity state		
	Scanned image		
	Visual image		
Manipulator	Manipulator joint position	Join effort setting	
	Manipulator end-effector pose	End-effector pose setting	

* A wrench effort command is to guide the AMV to waypoints [22].

Figure 4.2 shows a network of goal and sub-goals for the above three sub-missions. The main mission has a goal called *Underwater Intervention* and it can be decomposed in three sub-goals called *Vehicles Positioning for Survey, Path/Terrain Following, and IAUV Docking* respectively. The former can be in turn decomposed in two sub-goals called *ASC Positioning*, and *IAUV Positioning*. The target selection goal can be achieved if three sub-goals are reached. They are called *Seabed Image Mosaicing*, *View and Object Characterization*, and *Grasp Specification*. The latter main mission goal can be decomposed in four sub-goals called *Vehicles Positioning for Manipulation (*decomposed in *ASC Positioning*, and *IAUV Positioning)*, *Object Search*, *Object Intervention (*decomposed in *Floating Manipulation/Station Keeping)*, and *Object Grasping)*, and *IAUV Docking*.

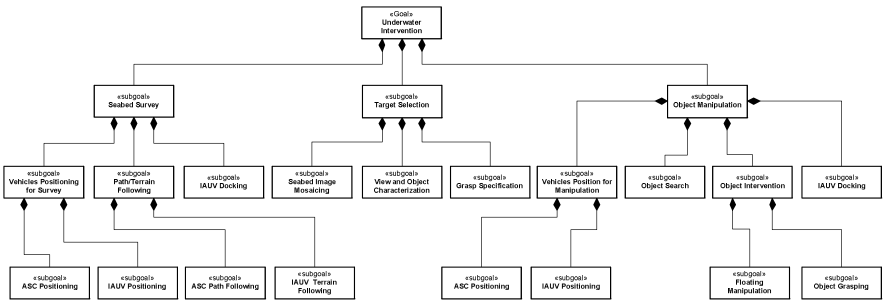

Fig. 4.2 Goals and sub-goals for the selected mission

The high-level functionalities are defined based on the goal and sub-goals shown in Figure 4.2. They can be seen as capabilities at the mission and operation levels. To achieve the main mission goal (top of Figure 4.2) the AMR system is required to have functionality at the mission level in order to achieve the *Underwater Intervention* goal. At the operation level, the AMR system is required to have functionality to achieve the *Vehicles Positioning for Survey, Path/Terrain Following, and IAUV Docking*. Thus, there are two high capability levels are defined: mission and operation capabilities.

The following use cases are identified for the AMR system based on the above mission goals (Figure 4.3). There are three use cases: seeded survey, target selection, and object manipulation. They are the AMR system sub-missions defined above.

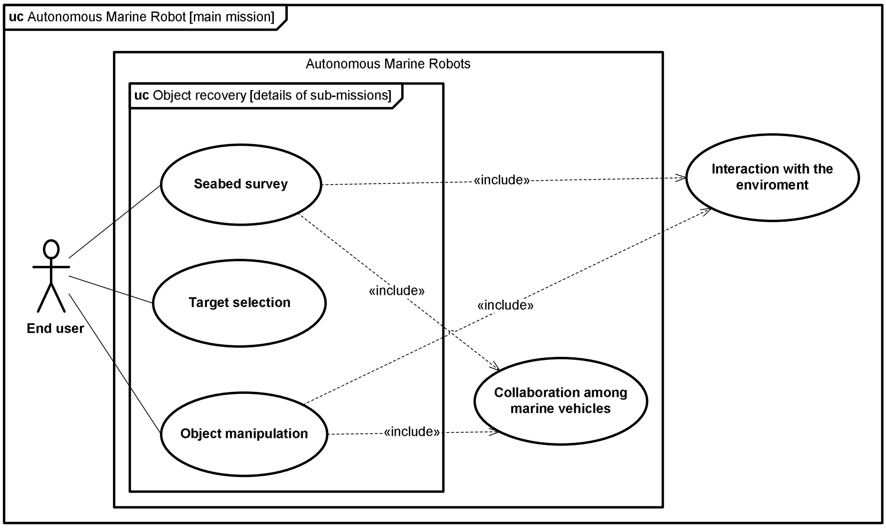

Fig. 4.3 System use cases

The AMR system actors (SysML actors) are OCU (end user), ASC, and IAUV. Figure 4.4 shows the scenario and interaction among AMR system actors for the seabed survey sub-mission.

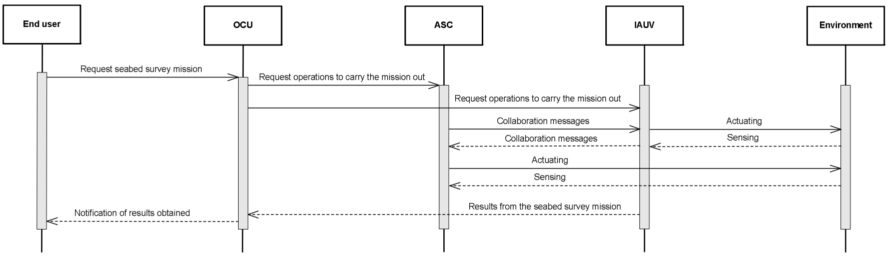

Fig. 4.4 Scenario for the seabed survey sub-mission

Figure 4.5 shows the scenario and interaction among AMR system actors for the target selection sub-mission.

Figure 4.6 shows the scenario and interaction among AMR system actors for the object manipulation sub-mission.

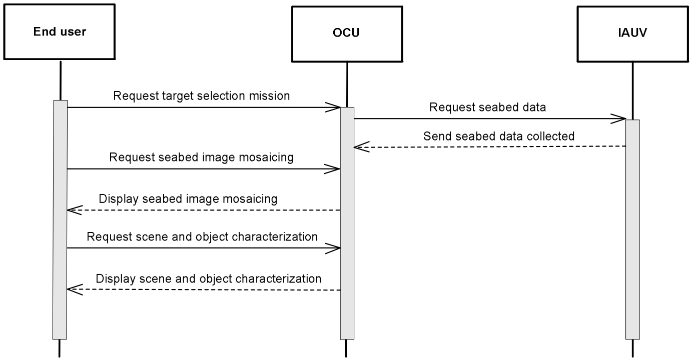

Fig. 4.5 Scenario for the target selection sub-mission

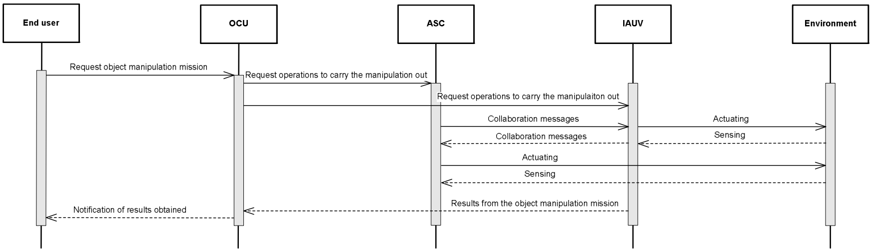

Fig. 4.6 Scenario for the object manipulation sub-mission

Alternative scenarios are defined in the case something goes wrong. Figure 4.7 shows alternative scenarios for the seabed survey mission when a seabed obstacle is found in the path of the IAUV. There is a need for re-planning of the mission in such a case. The interaction among actor to deal with it is shown inside the diagram box called *Re-planning requirement*.

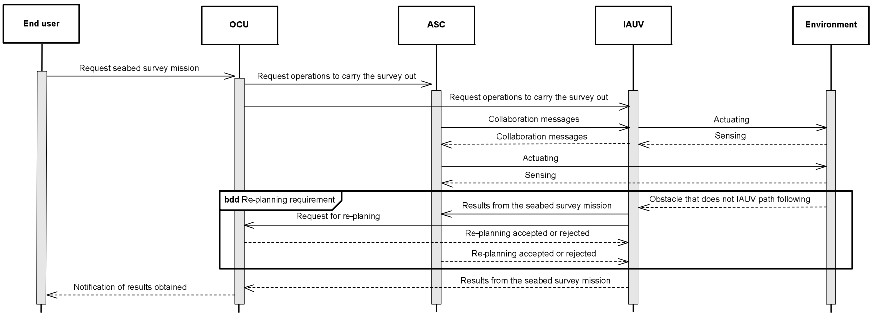

Fig. 4.7 Alternative scenario for re-planning seabed survey mission (seabed obstacle found in the path)

Figure 4.8 shows alternative scenarios for the object manipulation mission when a grasp strategy fails. There is a need for re-planning of the mission in order to choose and try another grasp strategy which is, in this case, successful. The interaction among actor to deal with it is shown inside the diagram box called *Re-planning requirement*.

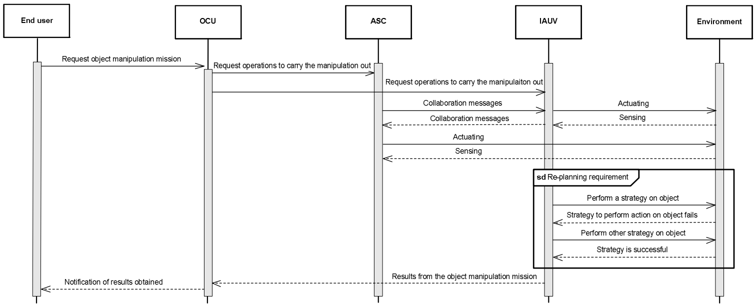

Fig. 4.8 Alternative scenario for re-planning object manipulation mission (grasp strategy fails)

4.2 System Architecture Specification

This Section presents the specification for the intelligent control architecture. The outcomes expected are the full specification of the AMR system components, its low-level functionalities and data coupling. The AMR system specification is done according to the JAUS reference architecture specification [22].

4.2.1 AMR System Components

Figure 4.9 shows a hierarchical representation of the different AMR system components and how they are organized according to the JAUS standard.

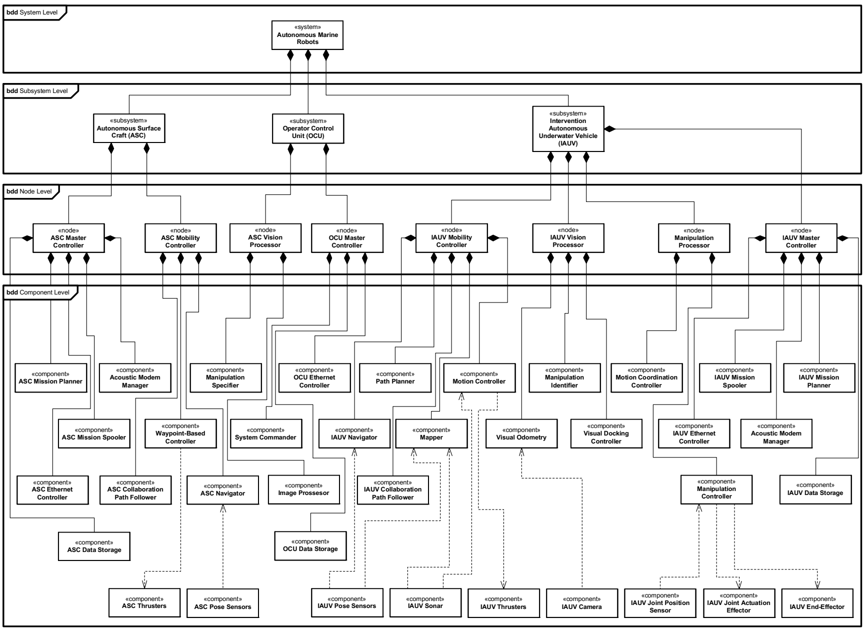

Fig. 4.9 Composite view of the JAUS-compliant system structure

The JAUS reference architecture specification defines a composite topology where a system is built of subsystem(s). A subsystem is built of node(s), and a node is built of component(s). The components are grouped according to their functionalities in order to build nodes, e.g. node *Vision Processor* is built of the following components: *Visual Odometry*, *Manipulation Identifier*, and *Visual Docking Controller*.

Figure 4.10 shows how the AMR system components are connected with each other. The small arrows placed in the component boxes show the direction of the data flow.

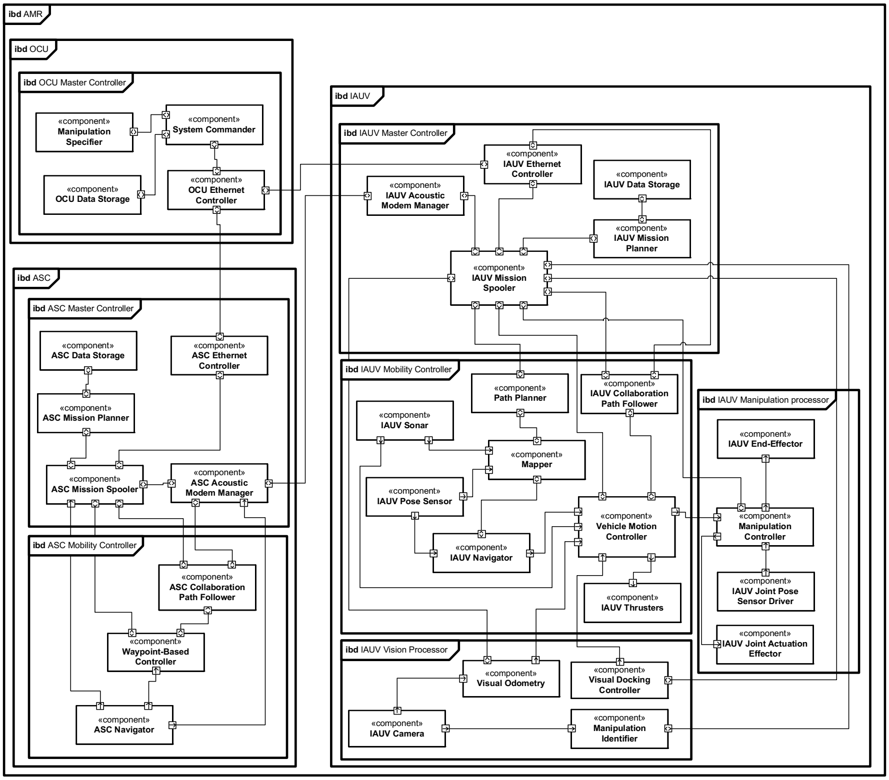

Fig. 4.10 Functional view of the JAUS-compliant system structure

4.2.2 Low-Level Functionalities and Data Coupling

Functionalities are the atomic elements of a functional structure of the system. The key point to identify basic services in components (Figure 4.10) is to group or un-group low-level component functionalities. It is done by following the next criterion:

to group functionalities:

- o Component functionalities are related
- o Component functionalities require similar information

to separate functionalities:

- Component functionalities are not related
- Component functionalities exist on different hardware platform
- Different numbers of component functionalities are required at runtime.

A good practice to identify component functionalities that then are wrapped in services is to make a list with all the components available in the platform, and their functions. In addition, the inputs and outputs of the function are provided in order to specify the data coupling among components. The functionalities of the AMR system (or platform) components are listed in Appendix C.

The functionalities (high level functionalities defined in Section 4.1, and low level functionalities above specified) of the platform components allow carrying out activities in the system. The activities can be classified at different levels of interaction. Thus, activities are classified as follows:

Missions are divided into Operations.
Operations are divided into Tasks.
Tasks are divided into Actions.
Actions are the atomic part of the hierarchy.

4.3 System Architecture Design

According to the JAUS reference architecture specification, services are provided by AMR system components. Services encapsulate functionalities of AMR system components, and can be seen as a block with one or more functions.

4.3.1 Description of Services

The intelligent control architecture supports two types of services:

Basic services: They are indivisible. Therefore, they are the atomic elements of the SOA in which the intelligent control architecture is based on.
Composite services: they are composed by in other services (basic or composite services). This composition of service is called orchestration of services in SOA.

Figure 4.11 shows the different parts of the anatomy of the basic and composite services. The former (a) is built of primitive functions linked to actions (one by one) that have pre-conditions and post-conditions. The latter (b) is built of other services that can be basic or composite services.

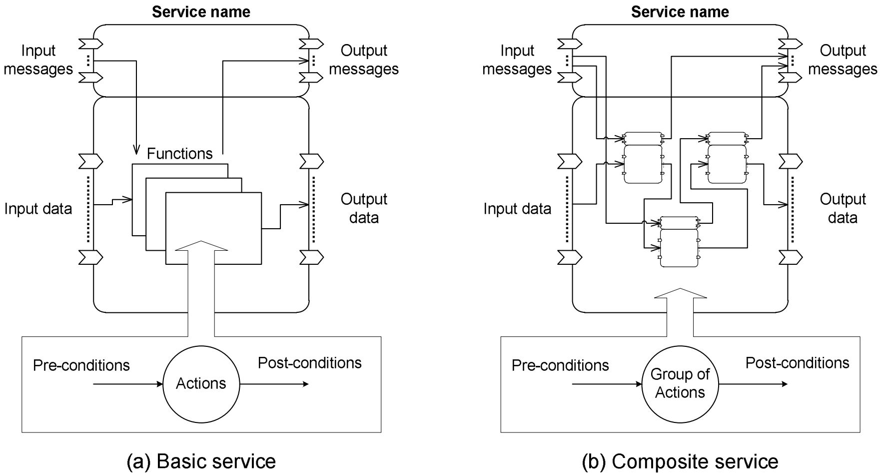

Fig. 4.11 Types of services. (a) Basic service. (b) Composite service.

Since services encapsulate functionalities, they are also classified following the functionality categorization given in Section 0. Therefore, the service classification is as follows:

A mission service is composed of Operation services
An operation service is composed of Task services
A task service is composed of Action services
An action service are the atomic part of the service hierarchy

The first three service classifications are composite services, and the last one is a basic service. The mission and operation services are designed by wrapping high-level functionalities as defined in Section 0. The task and action services are designed by wrapping low-level functionalities that provide the platform components shown in Figure 4.10.

A definition and description of all the services from the ICA is in the Appendix D.

4.3.2 Protocols of Services

The protocols of services are designed based on the use cases and scenarios defined in Subsection 5.4. The events and data exchange among services are identified from the following interaction diagrams.

Figure 4.12 shows the interaction among AMR system actors (i.e. as defined in Section) OCU, ASC, and IAUV) for mission and operation services.

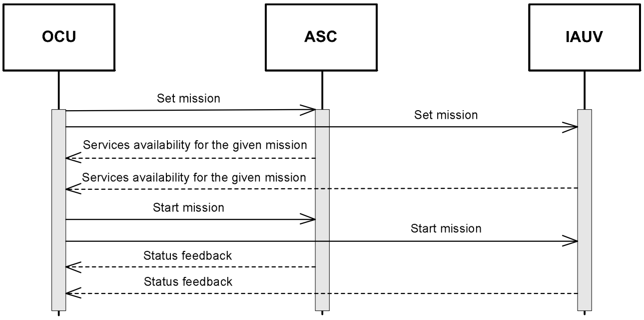

Fig. 4.12 External interaction among OCU, ASC, and IAUV

Figure 4.13 shows the interaction among the OCU components for task and action services.

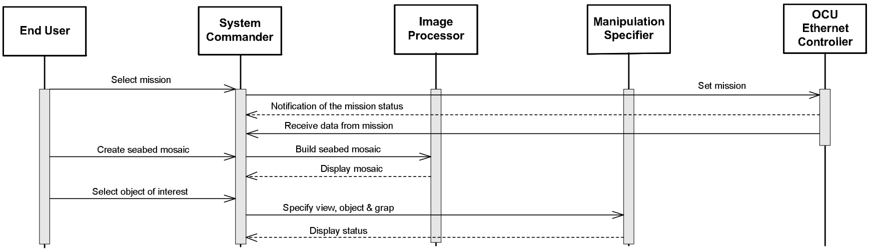

Fig. 4.13 Internal interaction for the OCU

Figure 4.14 shows the interaction among the ASC components for task and action services.

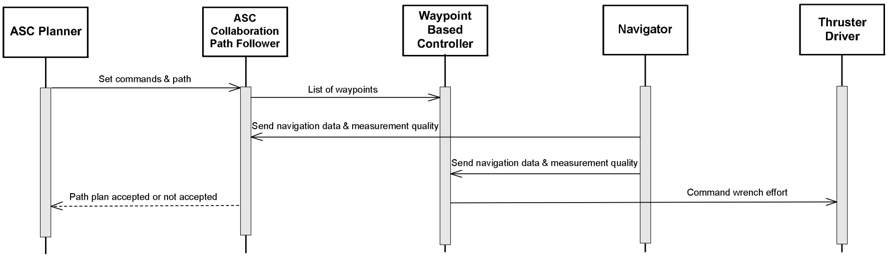

Fig. 4.14 Internal interaction for the ASC

Figure 4.15 shows the interaction among the IAUV components for task and action services.

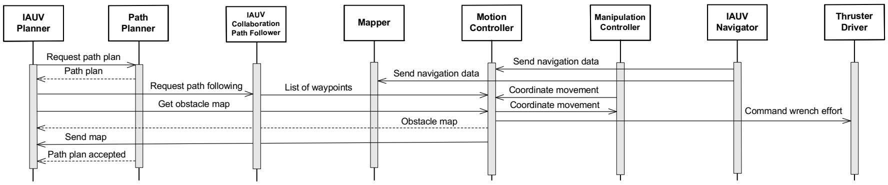

Fig. 4.15 Internal interaction for the IAUV

Table 4.2 presents the characteristics of the interaction among services. From left to right, providers are AMR system components that provide services. Consumers are the components that demand the services from the component providers.

The communication model is a representation of the data-linking mechanisms to transmit and receive information between communicating parts. This model follows three basic way to get communicated; one to one (Peer-to-Peer), one to many (Publisher/Subscriber), many to one (Client/Server).

The interaction model is a representation of the data-exchanging mechanisms to transmit and receive information between communicating parts. This model involves four way of services interaction; request-response, only request, solicitation-response, only notification. The blocking mechanism takes into account two communication ways; synchronous (blocking), asynchronous (non-blocking).

Table 4.2 Interaction among services

<table>
<tr><th colspan="2">Connection between services</th><th rowspan="2">Communication model</th><th rowspan="2">Interaction Model</th><th rowspan="2">Blocking mechanism</th></tr>
<tr><th>Provider</th><th>Consumer</th></tr>
<tr><td>Image Processor (Seabed Image Mosaicing)</td><td rowspan="4">OCU System Commander</td><td>Peer-to-peer</td><td>Request-Response</td><td>Synchronous</td></tr>
<tr><td>Manipulation Specifier (View Characterization)</td><td>Peer-to-peer</td><td>Request-Response</td><td>Synchronous</td></tr>
<tr><td>Manipulation Specifier (Object Characterization)</td><td>Peer-to-peer</td><td>Request-Response</td><td>Synchronous</td></tr>
<tr><td>Manipulation Specifier (Grasp Specification)</td><td>Peer-to-peer</td><td>Request-Response</td><td>Synchronous</td></tr>
<tr><td>ASC Collaboration Path Follower (Behaviour Management)</td><td rowspan="3">ASC Planner</td><td>Peer-to-peer</td><td>Request-Response</td><td>Synchronous</td></tr>
<tr><td>Waypoint-Based Controller (Waypoint List Setting)</td><td>Peer-to-peer</td><td>Request-Response</td><td>Synchronous</td></tr>
<tr><td>IAUV Data Storage (IAUV Operation Area Setting)</td><td>Peer-to-peer</td><td>Request-Response</td><td>Synchronous</td></tr>
</table>

Table 4.2 (*continued*)

Connection between services		Communication model	Interaction Model	Blocking mechanism
Provider	Consumer			
ASC Navigator (ASC Navigation Data Sending)	Waypoint-Based Controller	Publisher/Subscriber	Notification	Asynchronous
Path Planner (Path Plan Setting)	IAUV Planner	Peer-to-peer	Request-Response	Synchronous
Visual Odometry (Seabed Data Collection)		Peer-to-peer	Request-Response	Synchronous
Visual Docking Controller (Vehicle Docking)		Peer-to-peer	Request-Response	Synchronous
ASC Data Storage (ASC Operation Area Setting)		Peer-to-peer	Request-Response	Synchronous
Manipulation Identifier (Scene Identification)		Peer-to-peer	Request-Response	Synchronous
Manipulation Identifier (Object Identification)		Peer-to-peer	Request-Response	Synchronous
Manipulation Controller (Intervention Configuration Setting)		Peer-to-peer	Request-Response	Synchronous
Manipulation Controller (Object Intervention Manoeuvre)		Peer-to-peer	Request-Response	Synchronous
Mapper (Obstacle Map Generation)	Path Planner	Peer-to-peer	Request-Response	Synchronous
Mapper (Map Sending)		Publisher/Subscriber	Notification	Asynchronous

Table 4.2 (*continued*)

Connection between services		Communication model	Interaction Model	Blocking mechanism
Provider	Consumer			
IAUV Navigator (IAUV Navigation Data Sending)	Waypoint-Based Controller	Publisher/Subscriber	Notification	Asynchronous
	Motion Controller	Publisher/Subscriber	Notification	Asynchronous
	Mapper	Publisher/Subscriber	Notification	Asynchronous
	Manipulation Controller	Publisher/Subscriber	Notification	Asynchronous
	IAUV Planner	Publisher/Subscriber	Notification	Asynchronous
Visual Odometry (Visual Navigation Data)	Manipulation Controller	Publisher/Subscriber	Notification	Asynchronous
Manipulation Controller (Body Force Control)	Motion Controller	Publisher/Subscriber	Notification	Asynchronous

Communication model = {peer to peer, publisher/subscriber, client/server}
Interaction mode = {request-response, request, solicitation-response, notification}
Blocking mechanism = {Synchronous (blocking), Asynchronous (non-blocking)}

4.3.3 Service-Oriented Architecture Interoperability

The component services have a higher degree of autonomy than conventional ones since Component Profile for Ocean System Services (CPOSS) is supported by the SOA approach proposed. CPOSS defines a minimal set of implementation constraints to enable secure interoperation among services on resource-constrained components.

Form the control engineering viewpoint, AMR components are categorized as either controlling components or controlled components. However, a given component may play both roles. The interoperation patterns of a component-level SOA (or CPOSS) can be categorized according to the following basic interoperation mechanism for services (set of networking protocols, i.e. Universal Plug and Play). Protocols adapted from [37].

Addressing. This is the foundation for component networking. The way to address services from components is through a Uniform Resource Identifier (URI). The addressing is provided by the IP protocol.

Discovery. Once addressing is established, components need to discover each other. When a controlled component is added to the network, a discovery protocol enables it to advertise its services on the network. When a controlling component enters the network it sends out a search request, and then the components that match the request send a corresponding reply.

Description. Once a controlling component has discovered a controlled component, to learn more about the latter and its capabilities, the controlling component must retrieve the controlled component description. For each service provided by a component, the component description defines the command messages that the service responds to, as well as the associated message formats.

Control. A controlling component can exert control over a controlled component. A controlling component sends a control message to the network endpoint for that service to invoke a component service. The service may or may not return a response message providing any command specific information.

Eventing. Components may communicate through asynchronous eventing. It is usually implemented by a "publisher-subscriber" mechanism through which a service exposes events corresponding to internal state changes. Controlling components can subscribe in order to receive event notifications whenever the corresponding internal state change occurs.

4.4 Agent Design

This Subsection presents all the capabilities existing in the AMR system. They are built based in the definition of platform services specified above, and the activities the agents have to cope with.

4.4.1 Summary of AMR System Capabilities

Table 4.8 presents a summary of the capabilities provided by the AMR system in TRIDENT. The capability nature classification is as follows: vision, manipulation, motion, and communication.

Table 4.3 Summary of the AMR system capabilities

Capability Entry		
Capability	**Nature Classification**	**Specification**
Seabed Survey	[vision] [manipulation] [motion] [communication]	Appendix B Section B.1
Target Selection	[vision] [manipulation] [motion] [communication]	
Object Manipulation	[vision] [manipulation] [motion] [communication]	
ASC Positioning	[motion]	Appendix B Section B.2
IAUV Positioning	[motion]	
Leader Following	[motion] [communication]	
Path/Terrain Following	[vision] [motion]	
IAUV Homing	[vision] [motion] [communication]	
Pattern Search	[vision] [motion]	
OCU Ethernet Data Transfer	[communication]	
ASC Ethernet Data Transfer	[communication]	
IAUV Ethernet Data Transfer	[communication]	
Seabed Data Processing	[vision]	
Intervention Configuration	[manipulation]	
Dynamic Positioning	[motion] [communication]	
Object Intervention Manoeuvre	[vision] [manipulation] [motion]	
Floating Manipulation	[vision] [manipulation] [motion]	
Object Search	[vision] [motion]	

Table 4.3 (*continued*)

Capability Entry		
Capability	**Nature Classification**	**Specification**
Seabed Image Mosaicing	[vision]	Appendix B Section B.3
View Characterization	[vision]	
Object Characterization	[vision]	
Grasp Specification	[vision] [manipulation]	
Scene Identification	[vision]	
Object Identification	[vision]	
Vehicle Docking	[vision] [motion] [communication]	
Map Sending	[vision]	
Seabed Data Collection	[vision] [motion]	
Motion Estimation	[vision] [motion] [communication]	
Intervention Configuration Setting	[manipulation]	
Object Manipulation Control	[manipulation] [motion]	
Behaviour Management	[motion] [communication]	
Waypoint List Setting	[motion]	
ASC Navigation Data Sending	[motion]	
Path Plan Setting	[motion]	
Motion Control	[motion]	
Station Keeping	[motion] [communication]	
IAUV Navigation Data Sending	[motion]	
OCU Ethernet Data Sending	[communication]	
OCU Ethernet Data Receiving	[communication]	

Table 4.3 (*continued*)

Capability Entry		
Capability	**Nature Classification**	**Specification**
ASC Ethernet Data Sending	[communication]	
ASC Ethernet Data Receiving	[communication]	
IAUV Ethernet Data Sending	[communication]	
IAUV Ethernet Data Receiving	[communication]	
OCU Modem Data Sending	[communication]	
OCU Modem Data Receiving	[communication]	
ASC Modem Data Sending	[communication]	
ASC Modem Data Receiving	[communication]	
IAUV Modem Data Sending	[communication]	
IAUV Modem Data Receiving	[communication]	
End Effector Driving	[manipulation]	Appendix B Section B.4
Joint Effector Driving	[manipulation]	
Vehicle Motion Driving	[motion]	

4.4.2 Specification of Agents

This Subsection presents the specification of all the agents of the AMR system. The agents in the TRIDENT context are a sub-system form the user interface unit, and the two marine vehicle involved, i.e. the OCU, and the ASC and the IAUV.

Table 4.4 shows the record for the OCU agent. Basically, the following characteristic an agent are shown: a brief description about him, its perception capacities, messages exchanged with other agents, its external capabilities, its goals, and its involvement in plans.

Table 4.4 Record of the OCU agent

OCU Agent	
Description	This agent is the user workstation. It has capabilities for seabed data processing, and communication. Following the hierarchical agent organization, it is able to coordinate to the other system agents under the figure based on supervision capacities. It allows the user to deal with AMR system, i.e. selection of the missions to be carried out, storage of data acquisition, display of outcome visualization, etc.
Role	Supervisor
Perceptions	User input ASC messages IAUV messages Environment information
Messages	Seabed data processing request/response Ethernet data transfer request/response Modem data transfer request/response
Capabilities	Operation capabilities: • Seabed data processing • OCU Ethernet Data Transfer • OCU Modem Data Transfer Task capabilities: • Seabed Image Mosaicing • Object Characterization • View Characterization • Grasp Specification
Goals	Collective goals: Underwater intervention carried out (mission plan Figure 4.18) • Seabed Surveyed (mission sub-plan Figure 4.19) • Target Selected (mission sub-plan Figure 4.20) • Object Intervened (mission sub-plan Figure 4.21) Individual goals: Seabed Data Processed (within Target Selection) • Seabed Image Mosaic done • View Characterized • Object Characterized • Grasp Specified Ethernet Data Transferred (within Target Selection)

Table 4.5 shows the record for the ASC agent.

Table 4.5 Record of the ASC agent

ASC Agent	
Description	This agent is one of the resources available in the AMR system to perform marine operations on the surface. It has capabilities for static and dynamic positioning, leader following, and communication. Following the hierarchical agent organization, it is able to be coordinated by the other system agents under the figure based on work capacities.
Role	Worker
Perceptions	OCU messages IAUV messages Environment information
Messages	ASC Positioning request/response Leader Following request/response Dynamic Following request/response ASC Ethernet Data Transfer request/response ASC Modem Data Transfer request/response
Capabilities	Operation capabilities: • ASC Positioning • Leader Following • Dynamic Following • ASC Ethernet Data Transfer • ASC Modem Data Transfer Task capabilities: • Behaviour Management • Waypoint List Setting • ASC Navigation Data Sending • ASC Operation Area Setting • ASC Modem Data Receiving • ASC Modem Data Sending • ASC Ethernet Data Receiving • ASC Ethernet Data Sending • ASC Operation Area Setting
Goals	Collective goals: Underwater intervention carried out (mission plan Figure 4.18) • Seabed Surveyed (mission sub-plan Figure 4.19) • Target Selected (mission sub-plan Figure 4.20) • Object Intervened (mission sub-plan Figure 4.21) Individual goals: Ethernet Data Transferred ASC Positioned ASC Followed the Leader ASC Dynamically Positioned Ethernet Data Transferred Modem Data Transferred

Table 4.6 shows the record for the IAUV agent.

Table 4.6 Record of the IAUV agent

IAUV Agent	
Description	This agent is one of the resources available in the AMR system to perform marine operations underwater. It has capabilities for positioning, path and terrain following, pattern search, manipulation, and communication. Following the hierarchical agent organization, it is able to be coordinated by the other system agents under the figure based on work capacities.
Role	Worker
Perceptions	OCU messages ASC messages Environment information
Messages	IAUV Positioning request/response Path/Terrain Following request/response Pattern Search request/response IAUV Homing request/response Intervention Configuration request/response Object Intervention Manoeuvre request/response Floating Manipulation request/response IAUV Ethernet Data Transfer request/response IAUV Modem Data Transfer request/response
Capabilities	Operation capabilities: • IAUV Positioning • Path/Terrain Following • Pattern Search • IAUV Homing • Intervention Configuration • Object Intervention Manoeuvre • Floating Manipulation • IAUV Ethernet Data Transfer • IAUV Modem Data Transfer Task capabilities: • IAUV Navigation Data Sending • Path Plan Setting • Obstacle Map Generation • Map Sending • Motion Control • Station Keeping • IAUV Modem Data Receiving • IAUV Modem Data Sending

Table 4.6 (*continued*)

IAUV Agent	
	• IAUV Ethernet Data Receiving • IAUV Ethernet Data Sending • IAUV Operation Area Setting • Seabed Data Collection • Motion Estimation • Vehicle Docking • Object Identification • Scene Identification • Intervention Configuration Setting • Object Manipulation Control
Goals	Collective goals: Underwater intervention carried out (mission plan Figure 4.18) • Seabed Surveyed (mission sub-plan Figure 4.19) • Target Selected (mission sub-plan Figure 4.20) • Object Intervened (mission sub-plan Figure 4.21) Individual goals: IAUV positioned IAUV followed the path and terrain ASC positioned Map Sent Motion Controlled Seabed Data Collected Motion Estimated Vehicle Docked Object Identified Scene Identified Ethernet data transferred Modem data transferred

4.4.3 Global Planning for Local Plans

This Section presents the agent plans for the missions required by the TRIDENT project. They come from an automated planning carried out by the agents at two hierarchical planning levels according to the agents' roles, i.e. coordinator (as well supervisor since it can monitor and optimise the mission plans), and cooperator/collaborator (worker): supervision (global and centralized planning), and fieldwork (local and distributed plans). The former is concerned with the external activities and results coming from the execution of the local agent plans (worker). The latter is concerned with the internal activities and results coming from the execution of the local agent plans (worker). This approach matches the concept of centralized planning for distributed plans found in the literature. The plan elements are states, transitions (with guards), and activities (with preconditions as guards).

4.4.3.1 Global Agent Planning

This Subsection presents the global agent planning in terms of high-level capabilities and plans at the supervision level for the missions required by the TRIDENT project. They come from an automated planning carried out by the AMR system in the OCU agent. The planning nature comes from a simplified version of Hierarchical Task Network (HTN) called Single Task Network (STN) planning. Initially, the planning process is based on STN but the HTN approach could be implemented in the future in order to support any decomposition procedures of activities for solving problems.

<u>Intervention Mission</u>
The following planning representations correspond to the global planning in the coordinator agent (OCU). Figure 4.16 shows the global OCU plan for generic underwater interventions. The all plan is depict as a big state machine where the states are simplified, and summarized at the very high level in term of the sub-missions the AMR system has to go through to complete the main mission. Thus, it has the initial state which is the "Underwater intervention not carried out", and the final state (mission goal) which is the "Underwater intervention not carried out". The latter is composed of three sub-mission states (Seabed surveyed, Target selected, and Object intervened). The main activity to be performed in order to go from the initial state to the final one is described by the OCU plan.

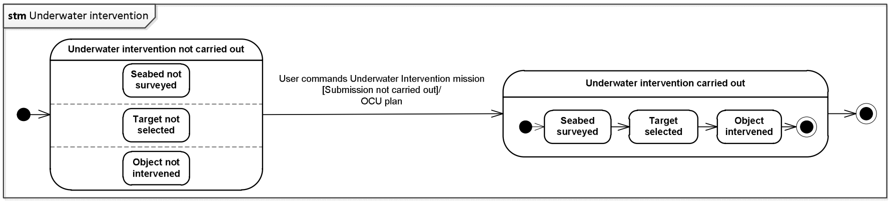

Fig. 4.16 (Global) OCU plan for the Underwater Intervention mission

Table 4.7 presents a summary of the characteristics and elements of the global and centralized OCU plan for generic underwater interventions.

Table 4.7 Global and centralized OCU plan specification

OCU Plan Record	
Name	Underwater Intervention (Figure 4.16)
Description	It performs a generic underwater intervention according to the parameters given by the end user.
Goal	Underwater Intervention carried out
Trigger	User commands Underwater Intervention mission to the AMR system
Guard	Sub-mission not carried out
Context	User interface running on the OCU ASC & IAUV deployed and ready to operate
Worker agent(s)	ASC, IAUV
Mission plan	Underwater Intervention plan built of mission sub-plans (below)
Mission sub-plans	Seabed Survey (Figure 4.17) Target Selection (Figure 4.18) Object Intervention (Figure 4.19)
Capabilities	Table B.2 (Appendix B) Table B.3 (Appendix B) Table B.4 (Appendix B)
Considered failures	Faults cases: • Lack of capabilities • OPC stops working • ASC stops working • IAUV stops working
Recovery mechanism	See Table 4.20 for recovery mechanisms for the above fault cases.

Seabed Survey Mission (Supervision)
Figure 4.17 shows the internal state-based composition for the Seabed Surveyed sub-goal (or sub-state) at the supervision level (missions and operations). It basically has four goals or states: AMVs at origin position (1), AMVs at beginning position (2), AMVs at origin position (3), and AMVs at origin position (4). To go from (1) to (2) the operation activities to be performed are ASC and IAUV Positioning. To go from (2) to (3) the operation activities to be performed are ASC Leader Following and IAUV Path/Terrain Following. To go from (3) to (4) operation activities to be performed are ASC Positioning and IAUV Homing.

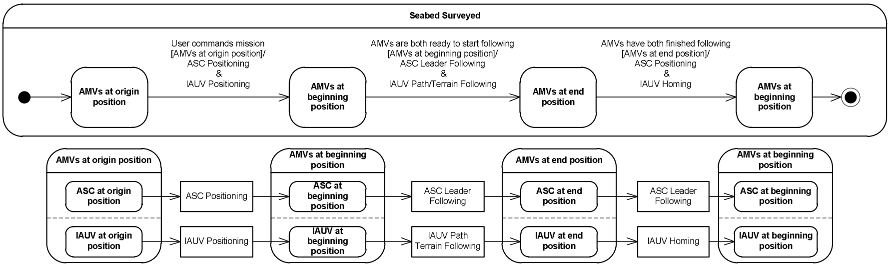

Fig. 4.17 (Global) OCU sub-plan for Seabed Survey

Table 4.8 presents a summary of the characteristics and elements of the OCU sub-plan for seabed surveys.

Table 4.8 OCU sub-plan record for Seabed Survey

OCU Sub-plan Record	
Name	Seabed Survey
Description	The initial state is that where Autonomous Marine Vehicles (AMVs) are in the origin location. The first intermediate state is that when the IAUV is in the beginning position, ready to start the seabed data collection and the ASC is positioned to support the above IAUV operation. The second intermediate state is that the IAUV has completely followed the path planned for the exploration area, and it is ready to go back to the surface to dock the ASC. The ASC has followed the leader (IAUV), and it is stopped and ready to meet the IAUV in the origin position when docking. The end state (mission goal) is that when AMVs are both back in the origin location after being carried out the seabed survey.
Goal	Seabed surveyed
Trigger	User commands Underwater Intervention mission to the AMR system
Guard	AMVs at origin position
Context	User interface running on the OCU ASC & IAUV deployed and ready to operate

Table 4.8 (*continued*)

OCU Sub-plan Record	
Worker agent(s)	ASC, IAUV
Sub-mission plan	Seabed Survey Plan (Figure 4.17)
Capabilities	ASC Positioning & IAUV Positioning ASC Leader Following & IAUV Path/Terrain Following ASC Positioning & IAUV Homing
Considered failures	Faults cases: • Lack of capabilities • OPC stops working
Recovery mechanism	See Table 4.20 for recovery mechanisms for the above fault cases.

Target Selection Mission (Supervision)
Figure 4.18 shows the internal state-based composition for the Target Selection sub-goal (or sub-state) at the supervision level (missions and operations). It basically has three goals or states: OCU connected with the IAUV (1), Data transferred from IAUV to OCU (2), and Data processed (3). To go from (1) to (2) the operation activities to be performed are Transfer data from IAUV to OCU. To go from (2) to (3) the operation activity to be performed are Seabed Data Processing.

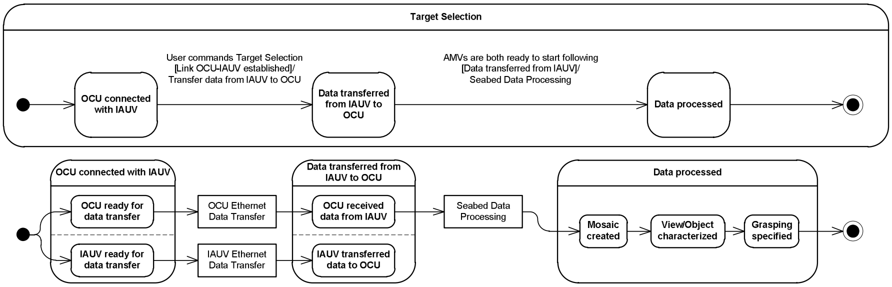

Fig. 4.18 (Global) OCU sub-plan for Target Selection

Table 4.9 presents a summary of the characteristics and elements of the OCU sub-plan for target selections.

Table 4.9 OCU sub-plan record for Target Selection

OCU Sub-plan Record	
Name	Target Selection
Description	The initial state is that where OCU is connected to the IAUV via Ethernet. The first intermediate state is that when the data have already been transferred from the IAUV to the OCU. The final state (sub-mission goal) is that when the data have already been processed, and a potential target has been selected.
Goal	Data processed in order to select potential object of interest (target)
Trigger	User commands Target Selection mission to the AMR system
Guard	Seabed surveyed
Context	User interface running on the OCU
Worker agent(s)	OCU, IAUV
Sub-mission plan	Target Selection Plan (Figure 4.18)
Capabilities	OCU Ethernet Data Transfer IAUV Ethernet Data Transfer Seabed Data Processing
Considered failures	Faults cases: • Lack of capabilities • OCU stops working
Recovery mechanism	See Table 4.20 for recovery mechanisms for the above fault cases.

Object Intervention Mission (Supervision)
Figure 4.19 shows the internal state-based composition for the Object Intervention sub-goal (or sub-state) at the supervision level (missions and operations). It basically has six goals or states: IAUV is connected with OCU (1), Data is transferred from OCU to IAUV (2), Intervention is configured and AMVs are at origin position (3), AMVs are at beginning position (4), AMVs are at end position (5), and AMVs are at origin position (6). To go from (1) to (2) the operation activity to be performed are Transfer data from OCU to IAUV. To go from (2) to (3) the operation activity to be performed are Intervention Configuration. To go from (3) to (4) operation activities to be performed are ASC and IAUV Positioning. To go from

(4) to (5) operation activities to be performed are ASC Dynamic Positioning and Pattern Search. To go from (5) to (6) operation activities to be performed are ASC Positioning and IAUV Homing.

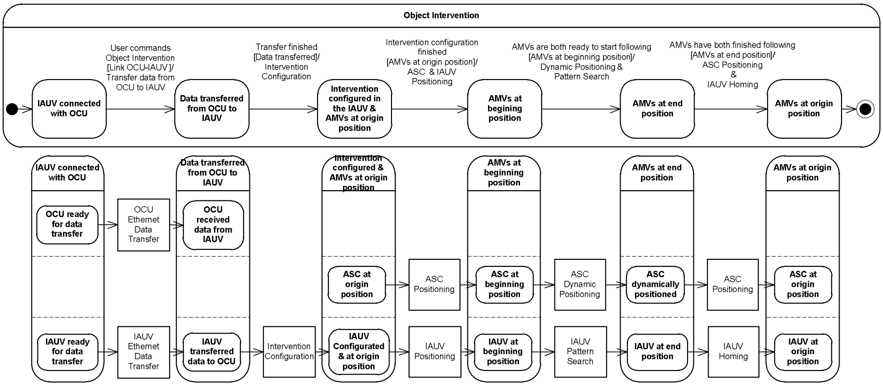

Fig. 4.19 (Global) OCU sub-plan for Object Intervention

Table 4.10 presents a summary of the characteristics and elements of the OCU sub-plan for object intervention.

Table 4.10 OCU sub-plan record for Object Intervention

OCU Sub-plan Record	
Name	Object Intervention
Description	The initial state is that where OCU is connected to the IAUV via Ethernet. The first intermediate state is that when the data have already been transferred from the OCU to the IAUV. The second intermediate state is that where intervention configuration was set in the system. The third intermediate state is that where Autonomous Marine Vehicles (AMVs) are in the origin location. The fourth intermediate state is that when the IAUV is in the beginning position, ready to start the pattern search and the ASC is dynamically positioned to support the above IAUV operation (covering a cone which vertex is in the ASC dynamic position). The fifth intermediate state is that the IAUV has intervened completely the object selected within the given sub-area, and it is ready to go back to the surface to dock the ASC. The ASC has supported IAUV, and it is stopped and ready to meet the IAUV in the origin position when docking. The final state (mission goal) is that when AMVs are both back in the origin location after being carried out the object intervention.

Table 4.10 (*continued*)

OCU Sub-plan Record	
Goal	Object intervened
Trigger	User commands Object Intervention mission to the AMR system
Guard	Seabed surveyed & Target selected
Context	User interface running on the OCU ASC & IAUV deployed and ready to operate
Worker agent(s)	ASC, IAUV
Sub-mission plan	Object Intervention Plan (Figure 4.19)
Capabilities	OCU Ethernet Data Transfer IAUV Ethernet Data Transfer Intervention Configuration ASC Positioning & IAUV Positioning Dynamic Positioning & Pattern Search ASC Positioning & IAUV Homing
Considered failures	Faults cases: • Lack of capabilities • OCU stops working
Recovery mechanism	See Table 4.20 for recovery mechanisms for the above fault cases.

4.4.3.2 Local Agent Plans

This Subsection presents the local agent planning in terms of low-level capabilities and plans at the fieldworker level for the missions required by the TRIDENT project. They come from an automated planning carried out by the AMR system in the ASC and IAUV agents. The planning nature comes from the classical planning [31].

The following planning representations correspond to the local planning in the work agents, i.e. ASC and IAUV.

<u>Seabed Survey Plan at Fieldwork</u> (vehicle level)
Figure 4.20 shows the internal state-based composition for the Seabed Surveyed sub-goal (or sub-state) at the fieldwork level (tasks and actions) for the ASC. Matching its definition for the supervision level (Subsection 0), it basically has four goals or states:

(1) ASC at origin position which means that it is ready for positioning at the vehicle level.

(2) ASC at beginning position which means that it is ready for leader following at the vehicle level.

(3) ASC at origin position which means that it is ready for homing at the vehicle level.

(4) ASC at origin position which means that it has finished supporting the IAUV at the vehicle level.

To go from (1) to (2) the task activity to be performed as building blocks of the ASC Positioning are Waypoint List Setting. To go from (2) to (3) the task activities to be performed as building blocks of the ASC Leader Following are the Behaviour Management and the Waypoint List Setting. To go from (3) to (4) task activity to be performed as building blocks of the ASC Positioning is the Waypoint List Setting.

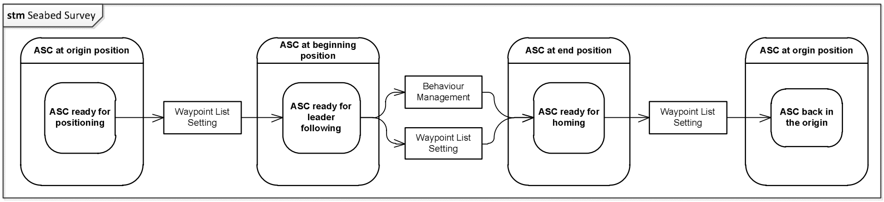

Fig. 4.20 (Local) ASC plan for Seabed Survey

Table 4.11 presents a summary of the characteristics and elements of the ASC plan for seabed surveys.

Table 4.11 ASC plan record for Seabed Survey

ASC Plan Record	
Name	Seabed Survey
Description	The initial state is that where ASC is ready for positioning (at origin location). The first intermediate state is that when the ASC is ready to start the leader following (at beginning position). The second intermediate state is that when the ASC is ready for homing (at end position). The final state (submission goal) is that when ASC is back in the origin position after being supporting the IAUV to carry out the seabed survey submission.
Goal	Support the IAUV to survey the seabed

Table 4.11 (*continued*)

ASC Plan Record	
Trigger	IAUV has just started its own operation to survey the seabed
Guard	ASC is ready for positioning
Context	ASC deployed and ready to operate
Module/Component	ASC Mobility Controller • Waypoint Based Controller • ASC Collaboration Path Follower • ASC Navigator
Submission plan	Local ASC Seabed Survey Plan (Figure 4.20)
Capabilities	ASC Positioning ASC Leader Following ASC Positioning
Considered failures	Faults cases: • ASC stops working • DVL does not work
Recovery mecha-nism	See Table 4.20 for recovery mechanisms for the above fault cases.

Figure 4.21 shows the internal state-based composition for the Seabed Surveyed sub-goal (or sub-state) at the fieldwork level (tasks and actions) for the IAUV. Matching its definition for the supervision level (Subsection 0), it basically has four goals or states:

(1) IAUV at origin position which means that it is ready for positioning at the vehicle level.

(2) IAUV at beginning position which means that it is ready for path/terrain following at the vehicle level.

(3) IAUV at end position which means that it is ready for homing at the vehicle level.

(4) IAUV at origin position which means that it is docked to the ASC at the vehicle level.

To go from (1) to (2) the task activity to be performed as building blocks of the IAUV Positioning is Path Planning Setting. To go from (2) to (3) the task activities to be performed as building blocks of the Path/Terrain Following are the

Behaviour Management and the Waypoint List Setting. To go from (3) to (4) task activity to be performed as building blocks of the IAUV Homing is the Waypoint List Setting.

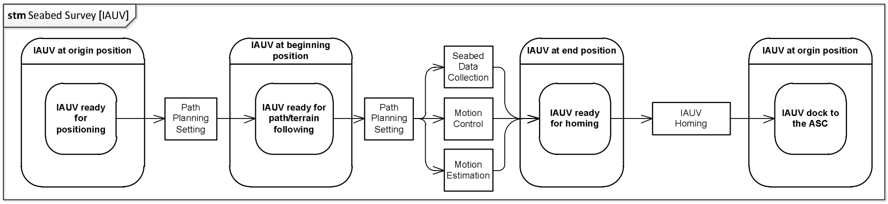

Fig. 4.21 (Local) IAUV plan for Seabed Survey

Table 4.11 presents a summary of the characteristics and elements of the IAUV plan for seabed surveys.

Table 4.12 IAUV plan record for Seabed Survey

IAUV Plan Record	
Name	Seabed Survey
Description	The initial state is that where IAUV is ready for positioning (at origin location). The first intermediate state is that when the IAUV is ready to start the path/terrain following (at beginning position). The second intermediate state is that when the IAUV is ready for homing (at end position). The final state (submission goal) is that when IAUV is back in the origin position, and docked to the ASC after being carrying out the seabed survey submission.
Goal	Data collection from the seafloor as part of the seabed survey
Trigger	ASC & IAUV are both deployed and ready for operation
Guard	ASC & IAUV are ready for positioning
Context	IAUV deployed and ready to operate
Module/Component	IAUV Mobility Controller • Path Planner • Mapper • Motion Controller • IAUV Navigator

Table 4.12 (*continued*)

IAUV Plan Record	
Submission plan	Local IAUV Seabed Survey Plan (Figure 4.21)
Capabilities	Path Planning Setting Seedbed Data Collection Motion Control Motion Estimation IAUV Homing
Considered failures	Faults cases: • IAUV stops working • Obstacle in the IAUV path • Target has moved away from initial position • DVL does not work
Recovery mechanism	See Table 4.20 for recovery mechanisms for the above fault cases.

<u>Target Selection Mission at Fieldwork</u> (vehicle level)
Figure 4.22 shows the internal state-based composition for the Target Selection sub-goal (or sub-state) at the fieldwork level (tasks and actions) for the OCU. Matching its definition for the supervision level (Subsection 0), it basically has three goals or states:

(1) OCU is connected with the IAUV which means that it is ready for data transfer at the agent/node level.

(2) Data have been transferred from the IAUV to the OCU. This means that the OCU has received data form the IAUV at the agent/node level.

(3) Data have been processed in the OCU in order to generate the seabed mosaic. This means that the mosaic is ready for use in the system at the agent/node level.

(4) Data have been processed in the OCU in order to characterize the view and the object. This means that the view has been characterized first, and the the object have been characterized at the agent/node level.

(5) Data have been processed in the OCU in order to generate the grasping techniques. This which means that the grasping specification is ready for use at the agent/node level.

To go from (1) to (2) the task activity to be performed as building block of the OCU Ethernet Data Transfer is OCU Ethernet Data Receiving. To go from (2) to (3) the task activity to be performed is Seabed Image Mosaicing. To go from (3) to (4) the task activities to be performed are View Characterization and Object Characterization. To go from (4) to (5) the task activity to be performed is Grasp Specification.

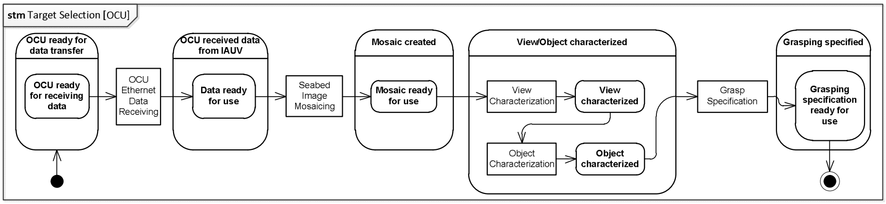

Fig. 4.22 (Local) OCU sub-plan for Target Selection

Table 4.13 presents a summary of the characteristics and elements of the OCU sub-plan for target selections.

Table 4.13 OCU sub-plan record for Target Selection

OCU Plan Record	
Name	Target Selection
Description	The initial state is that where the OCU is ready for receiving data. The first intermediate state is that when the data are ready for use. The second intermediate state is that when the mosaic is ready for use. The third intermediate state is divided into two states, and it happens when the view/object have been characterized. The final state (submission goal) is that when, after having been through the above states, the grasping is specified.
Goal	Seabed data processing as part of the seabed survey
Trigger	OCU is commanded to begin the data transfer
Guard	OCU is ready for data transfer
Context	OCU & IAUV are connected with each other via Ethernet
Module/Component	OCU Master Controller • OCU Ethernet Controller OCU Vision Processor • Manipulation Specifier • Image Processor

Table 4.13 (*continued*)

OCU Plan Record	
Submission plan	Local IAUV Target Selection Plan ()
Capabilities	OCU Ethernet Data Receiving Seabed Image Mosaicing View Characterization Object Characterization Grasp Specification
Considered failures	Faults cases: • OCU stops working
Recovery mechanism	See Table 4.20 for recovery mechanisms for the above fault cases.

Figure 4.22 shows the internal state-based composition for the Target Selection sub-goal (or sub-state) at the fieldwork level (tasks and actions) for the IAUV. Matching its definition for the supervision level (Subsection 0), it basically has three goals or states:

(1) OCU is connected with the IAUV which means that it is ready for data transfer at the agent/node level.

(2) Data have been transferred from the IAUV to the OCU. This means that the OCU has sent data form the IAUV at the agent/node level.

To go from (1) to (2) the task activity to be performed as building block of the IAUV Ethernet Data Transfer is OCU Ethernet Data Sending.

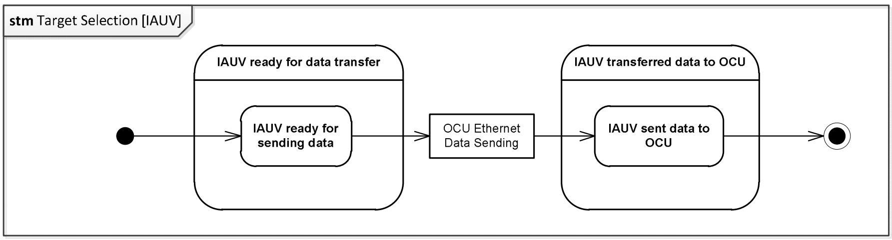

Fig. 4.23 (Local) IAUV sub-plan for Target Selection

Table 4.14 presents a summary of the characteristics and elements of the IAUV sub-plan for target selections.

Table 4.14 IAUV sub-plan record for Target Selection

IAUV Plan Record	
Name	Target Selection
Description	The initial state is that where the OCU is ready for receiving data. The final state is that when the data have been completely sent to the OCU.
Goal	Data transfer from the IAUV to the OCU
Trigger	OCU is able to receive data
Guard	IAUV is ready for data transfer
Context	OCU & IAUV are connected with each other via Ethernet
Module/Component	IAUV Master Controller • IAUV Ethernet Controller
Submission plan	Local IAUV Target Selection Plan (Figure 4.23)
Capabilities	IAUV Ethernet Data Sending
Considered failures	Faults cases: • IAUV stops working
Recovery mechanism	See Table 4.20 for recovery mechanisms for the above fault cases.

<u>Object Intervention Mission at Fieldwork</u> (vehicle level)
Figure 4.24 shows the internal state-based composition for the Object Manipulation sub-goal (or sub-state) at the fieldwork level (tasks and actions) for the OCU. Matching its definition for the supervision level (Subsection 0), it basically has two goals or states:

(1) OCU is connected with the IAUV which means that it is ready for data transfer at the agent/node level.

(2) Data have been transferred from the OCU to the IAUV. This means that the OCU has sent data to the IAUV at the agent/node level.

To go from (1) to (2) the task activity to be performed as building block of the OCU Ethernet Data Transfer is OCU Ethernet Data Sending.

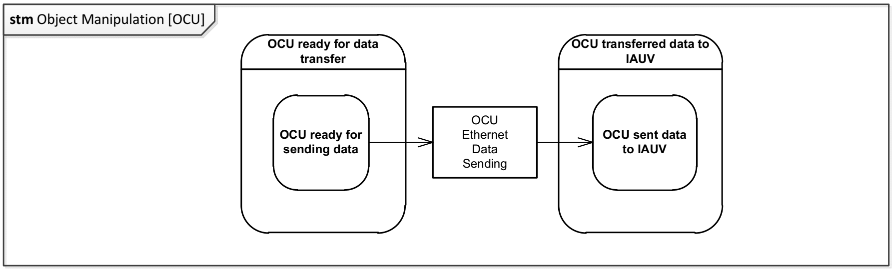

Fig. 4.24 (Local) OCU plan for Object Intervention

Table 4.15 presents a summary of the characteristics and elements of the ASC plan for Object Interventions.

Table 4.15 OCU plan record for Object Intervention

OCU Plan Record	
Name	Object Intervention
Description	The initial state is that where the OCU is ready for sending data. The final state is that when the data have been completely sent to the IAUV.
Goal	Data transfer from the OCU to the IAUV
Trigger	OCU is able to send data
Guard	OCU is ready for data transfer
Context	OCU & IAUV are connected with each other via Ethernet
Module/Component	OCU Master Controller • OCU Ethernet Controller
Submission plan	Local OCU Object Intervention Plan (Figure 4.24)
Capabilities	OCU Ethernet Data Sending
Considered failures	Faults cases: • OCU stops working • Obstacle in the IAUV path • Target has moved away from initial position
Recovery mechanism	See Table 4.20 for recovery mechanisms for the above fault cases.

Figure 4.25 shows the internal state-based composition for the Object Intervened sub-goal (or sub-state) at the fieldwork level (tasks and actions) for the ASC. Matching its definition for the supervision level (Subsection 0), it basically has four goals or states:

(1) ASC at origin position which means that it is ready for positioning at the vehicle level.

(2) ASC at beginning position which means that it is ready for leader following at the vehicle level.

(3) ASC at origin position which means that it is ready for homing at the vehicle level.

(4) ASC at origin position which means that it has finished supporting the IAUV at the vehicle level.

To go from (1) to (2) the task activity to be performed as building blocks of the ASC Positioning are Waypoint List Setting. To go from (2) to (3) the task activities to be performed as building blocks of the ASC Leader Following are the Behaviour Management and the Waypoint List Setting. To go from (3) to (4) task activity to be performed as building blocks of the ASC Positioning is the Waypoint List Setting.

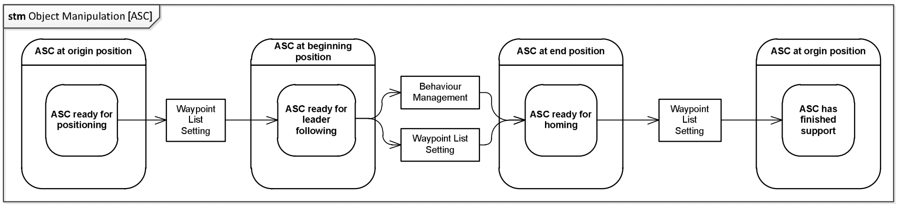

Fig. 4.25 (Local) ASC plan for Object Intervention

Table 4.16 presents a summary of the characteristics and elements of the ASC plan for Object Interventions.

Table 4.16 ASC plan record for Object Intervention

ASC Plan Record	
Name	Object Intervention
Description	The initial state is that where ASC is ready for positioning (at origin location). The first intermediate state is that when the ASC is ready to start the dynamic positioning. The second intermediate state is that when the ASC is ready for homing (at end position). The final state (submission goal) is that when ASC is back in the origin position after being supporting the IAUV to carry out the Object Intervention submission.
Goal	Support the IAUV to manipulate the object selected
Trigger	IAUV has just started its own operation to survey the seabed
Guard	ASC is ready for positioning
Context	ASC deployed and ready to operate
Module/Component	ASC Mobility Controller • Waypoint Based Controller • ASC Collaboration Position Supporter • ASC Navigator
Submission plan	Local ASC Object Intervention Plan (Figure 4.20)
Capabilities	ASC Positioning ASC Collaboration Position Supporter ASC Positioning
Considered failures	Faults cases: • ASC stops working • DVL does not work
Recovery mechanism	See Table 4.20 for recovery mechanisms for the above fault cases.

Figure 4.26 shows the internal state-based composition for the Object Intervened sub-goal (or sub-state) at the fieldwork level (tasks and actions) for the IAUV. Matching its definition for the supervision level (Subsection 0), it basically has four goals or states:

(1) IAUV at origin position which means that it is ready for positioning at the vehicle level.

(2) IAUV at beginning position which means that it is ready for object manipulation at the vehicle level.

(3) IAUV at end position which means that it is ready for homing at the vehicle level.

(4) IAUV at origin position which means that it is docked to the ASC at the vehicle level.

To go from (1) to (2) the task activity to be performed as building blocks of the IAUV Positioning is Path Planning Setting. To go from (2) to (3) the task activities to be performed as building blocks of the Path/Terrain Following are the Behaviour Management and the Waypoint List Setting. To go from (3) to (4) task activity to be performed as building blocks of the IAUV Homing is the Waypoint List Setting.

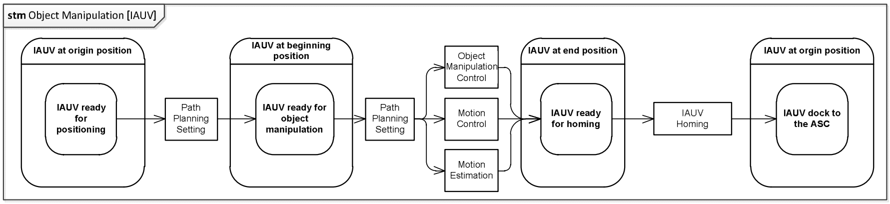

Fig. 4.26 (Local) IAUV plan for Object Intervention

Table 4.17 presents a summary of the characteristics and elements of the IAUV plan for Object Interventions.

Table 4.17 IAUV plan record for Object Intervention

IAUV Plan Record	
Name	Object Intervention
Description	The initial state is that where IAUV is ready for positioning (at origin location). The first intermediate state is that when the IAUV is ready to start the pattern search (from the beginning position), and then object manipulation once the object is found. The second intermediate state is that when the IAUV is ready for homing (at end position). The final state (submission goal) is that when IAUV is back in the origin position, and docked to the ASC after being carrying out the Object Intervention submission.
Goal	Object search from the seafloor as part of the Object Intervention

Table 4.17 (*continued*)

IAUV Plan Record	
Trigger	ASC & IAUV are both deployed and ready for operation
Guard	ASC & IAUV are ready for positioning
Context	IAUV deployed and ready to operate
Module/Component	IAUV Mobility Controller • Path Planner • Motion Controller • Manipulation Controller • IAUV Navigator
Submission plan	Local IAUV Object Intervention Plan (Figure 4.21)
Capabilities	Path Planning Setting Object Manipulation Control Motion Control Motion Estimation IAUV Homing
Considered failures	Faults cases: • IAUV stops working • Obstacle in the IAUV path • Target has moved away from initial position • Inadequate manipulation technique • DVL does not work
Recovery mechanism	See Table 4.20 for recovery mechanisms for the above fault cases.

4.4.3.3 Fault-Tolerant Agent Planning

The following faults are initially considered in the context of TRIDENT but few of them are implemented. It is expected to expand the implementation of faults taken into account in the diagnostic process in the future.

Fault Diagnosis
Table 4.18 presents the faults cases initially considered in TRIDENT.

Table 4.18 Faults cases initially considered in TRIDENT

Faults Cases	**Classification**					
	Nature	**Perception**	**Boundary**	**Origin**	**Occurrence**	**Hierarchy**
Lack of capabilities	Intentional/ Unintentional	Logical/ Physical	External/ Internal	Sustainment	Non-periodic	Any
OPC stops working	Unintentional	Physical	External	Sustainment	Non-periodic	Mission
ASC stops working	Unintentional	Physical	External	Sustainment	Non-periodic	Mission
IAUV stops working	Unintentional	Physical	External	Sustainment	Non-periodic	Mission
Obstacle in the IAUV path	Intentional/ Unintentional	Physical	External	Sustainment	Non-periodic	Operation
Target has moved away from initial position	Intentional/ Unintentional	Physical	External	Sustainment	Non-periodic	Operation
Inadequate manipulation technique	Unintentional	Physical	External	Sustainment	Non-periodic	Task
DVL does not work	Unintentional	Physical	Internal	Sustainment	Sporadic	Action

Table 4.19 presents the failures generated by states errors in the system due to faults found.

Table 4.19 Failures derived from the faults cases

Faults (event)	**Error (states)**	**Failure (behaviour)**
Lack of capabilities	Missing and unreachable sub-goals (sub-states).	The mission plan is incomplete.
OCU stops working	The external OCU state is that “no response”.	There is not user interface support for the mission.
ASC stops working	The external ASC state is that “no response”.	There is not surface support for underwater localization

Table 4.19 (*continued*)

Faults (event)	Error (states)	Failure (behaviour)
IAUV stops working	The external IAUV state is that "no response".	There is no way to carry underwater missions out.
Target has moved away from initial position	The IAUV may come back to a sub-area where the object is not in.	The object may not be found in the seafloor.
Obstacle in the IAUV path	The IAUV is temporally in wrong places	The IAUV cannot follow the path given
Inadequate manipulation technique	Object not manipulated.	The manipulation technique is not successful in dealing with the object.
DVL does not work	Wrong data samples from the DVL.	Not-trusted velocity measured.

Fault Recovery

Table 4.20 presents the diagnosis and recovery mechanisms to tackle the above faults cases.

Table 4.20 Diagnosis and recovery mechanisms for the faults cases

Faults Case	Diagnosis Mechanism	Recovery Mechanism	
		Mitigation Plan	**Technique**
Lack of capabilities	Consistency analysis	It depends on in which mission stage the system is. Thus, it may: 1. Abort mission, 2. Perform partially mission, or 3. Perform fully mission with some capability replaced by similar ones	1. No planning 2. Re-planning 3. Plan repair
OCU stops working	Watchdog timing	Before starting mission • Abort mission After starting mission • Let the mission reach one of the following sub-mission goals: seabed surveyed, or object intervened. If it happens during the target selection mission, there is nothing to be done.	Re-planning

Table 4.20 (*continued*)

Faults Case	**Diagnosis Mechanism**	**Recovery Mechanism**	
		Mitigation Plan	**Technique**
ASC stops working	Watchdog timing	Mission re-planning without the ASC	Re-planning
IAUV stops working	Watchdog timing	Before starting mission Abort mission After starting mission Stop Mission	No planning
Target has moved away from initial position	Effectiveness detection	The object is in the area specified by the OCU. • Nothing to do if the manipulation technique is working. If not, do what is described for inadequate manipulation below. The object is away from the area specified by the OCU. • Perform new object search	Re-planning / Plan repair
Obstacle in the IAUV path	Effectiveness detection	1. Collision avoidance 2. Path re-planning	Plan repair
Inadequate manipulation technique	Effectiveness detection	Recursive try using different manipulation techniques until running out of alternative options.	Plan repair
DVL does not work	Malfunction detection	Use of visual motion estimation instead.	Plan repair

The faults cases presented are just the starting point to deal with the AMR system dependability. There is still a lot work to do in terms of listing faults cases and analysis at the deliberation level, e.g. the failure conditions should be categorized by their effects on the marine vehicles, human operators, and environment in order to make decision more accurately.

Fault-Tolerant System Performance

The fault cases are considered in the scenarios presented above. In addition, the test procedures are also shows in Chapter 6 in order to evaluate the fault-free system performance. The outcomes of these evaluation processes are reported in Chapter 5.

4.4.4 Executive Processes in Agents

This Section describes the internal agent modules from the execution unit shown in figure 3.7, Chapter 3. It basically focuses on the execution of the services as activities planned in the Mission Planner. Before executing the services, the Mission Spooler schedules them following the initial order given in the plan but containing the services execution according to the conditions already established.

4.4.4.1 Spooling of Activities

The spooling of activities is carried out in the Mission Spooler. The mission plan is a XML-like file which has all the information about the activities to be performed as well as the instructions about how to execute the services. Once the Mission Planner has created the mission plan, it invokes the service "Mission Management" from the Mission Spooler in order to set the mission plan. Thus, the mission plan can be managed from the Mission planner, i.e. run, aborted, paused, and resumed. The scheduling of service is a process carried out before spooling the activities planned.

Agents Plan Description

Following the idea of facilitating the description of the services execution used in SOAs in business process (WS-BPEL [42]), and robotics planning based in the agent technology, a language to describe the execution of agent activities is proposed. It is called Service-Oriented Agent Plan Execution Language (SOAPEL).

The SOAPEL structure is oriented to automated planning processes, i.e. an automatic state machine which transitions are conditioned by the previous state, and some external condition in the environment. The transition can be

Like other execution languages, SOAPEL has key words. They basically are categorized as follows: Instructions and attributes.

<u>Control instructions</u>

- **invoke.** This SOAPEL instruction allows the mission plan to invoke a one-way or request-response service. It also has two control sub-instructions as follows.
    - **input.** Data are passed to services invoked by using this key word.
    - **output.** Data are retrieved from services invoked by using this key word.

- **if-else.** This SOAPEL instruction allows the mission plan to select one branch of service or other from a set of two choices.
- **switch-case.** This SOAPEL instruction allows the mission plan to select exactly one branch of service from a set of choices.
- **while.** This SOAPEL instruction allows the mission plan to indicate that a service is to be repeated until a certain success criteria has been met.

<u>Data instructions</u>

- **assign.** This SOAPEL instruction can be used to update the values of variables with new data. It can contain any number of elementary assignments.
- **variable.** This SOAPEL instruction can be used to define variables of the mission plan.
- **type.** This SOAPEL instruction can be used to define new data types available in the mission plan. It also has one control sub-instruction as follows.
    - **enumerator.** Ordered list of data type.
    - **structure.** Data structure data type.

<u>Process instructions</u>

- **plan.** This SOAPEL instruction is to specify a container where declarations of relationships to external definitions of XML schema, declarations for process data, handlers for various purposes and, as key elements of the mission plan, the list of services to be invoked according to some conditions for execution.
- **function.** This SOAPEL instruction can be used to define simple functions in the mission plan. Functions should typically be defined to deal with execution conditionals (control instructions). It also has two control sub-instructions as follows.
    - **argument.** Parameter(s) passed to the functions.
    - **body.** List of operators to be performed when the function is called.

Table 4.21 shows what attributes are supported by the instructions of the mission plan in SOAPEL. Where there is a note (m) means it is mandatory. Where there is a note (o) means it is optional.

Table 4.21 Instruction attributes

Instruction & Sub-instruction	Attribute						
	name	**type**	**value**	**field**	**re-turn**	**ser-vice**	**con-di-tion**
plan	Yes (o)	No (m)	No (m)	No (m)	No (m)	No (m)	No (m)
type	Yes (m)	Yes (m)	No (m)	No (m)	No (m)	No (m)	No (m)
enumerator	Yes (m)	No (m)	Yes (m)	No (m)	No (m)	No (m)	No (m)
function	Yes (m)	Yes (m)	No (m)	No (m)	Yes (o)	No (m)	No (m)
variable	Yes (m)	Yes (m)	Yes (m)	Yes (o)	No (m)	No (m)	No (m)
assign	No (m)	No (m)	No (m)	No (m)	No (m)	No (m)	No (m)
while	No (m)	No (m)	No (m)	No (m)	No (m)	No (m)	Yes (m)
switch	Yes (m)	No (m)	No (m)	No (m)	No (m)	No (m)	No (m)
case	No (m)	No (m)	No (m)	No (m)	No (m)	No (m)	Yes (m)
if	No (m)	No (m)	No (m)	No (m)	No (m)	No (m)	Yes (m)
invoke	No (m)	No (m)	No (m)	No (m)	No (m)	Yes (m)	No (m)
input	Yes (m)	No (m)	Yes (m)	No (m)	No (m)	No (m)	No (m)
output	Yes (m)	No (m)	Yes (m)	No (m)	No (m)	No (m)	No (m)

Agents Plan Scheduling

The following terms, i.e. “activities” or “services execution” are used with the same meaning in this Subsection, and along this book. The former refers to missions, operations, tasks, and actions required at the planning level. The latter makes refers to (and matching one by one) how the above activities are being implemented in terms of services available in the system.

Once the mission plan is set in the Mission Spooler, it can be executed just invocating the corresponding Mission Spooler services from the Mission Planner. Figure 4.27 shows a structural view of the functional blocks of the Mission Spooler.

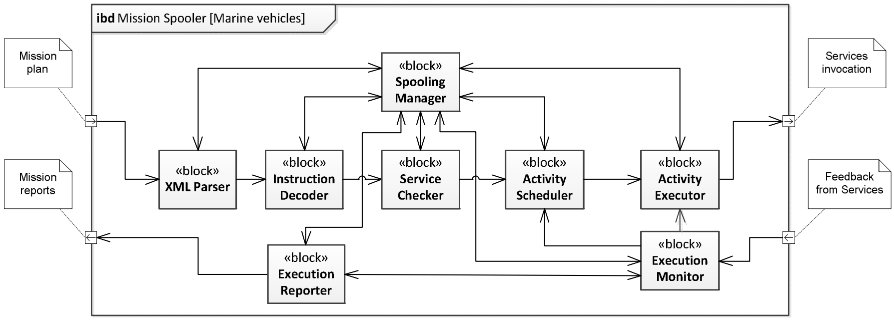

Fig. 4.27 Functional structure of the Mission Spooler (Static representation)

The mission plan is the input data for the function "Mission Setting" of the service "Mission Spooling". Once this service is invocated, the mission plan is parsed through a XML parser, decoded by means of the Instruction Decoder and analysed by the Service Checker which looks for more details about the services required by querying the Service Matchmaker. The following functions (i.e. "Mission Run", "Mission Abort", "Mission Pause", and "Mission Resume") of the service "Mission Spooling" are to control the plan execution carried out by the Activity Scheduler and the Activity Executor. The execution process is followed by the Execution Monitor that monitors the execution of the services, and notifies it to the Activity Executor, the Activity Scheduler, and the Spooling Reporter (this block generates reports on the mission status to the Mission Planner). All the Mission Spooler processes are controlled by the Spooling Manager.

Figure 4.28 shows a behavioural view of the functional blocks of the Mission Spooler. The mission plan is the input data for the service "run mission". Initially, it is considered that the Mission Spooler is in the state "Idle", when it receives a mission plan it goes to the state "Pending" which means that there is mission plan waiting to be executed. This transition is triggered automatically, and performs the XML parsing, instruction decoding, and services checking. From the state "Pending", the possible following states are "Running", and "aborted". The former is reached when the Mission Spooler is commanded by the Mission Planner to run the mission plan. The latter is reached when the Mission Spooler is commanded by the Mission Planner to abort the mission for any reason. While running, the mission plan can enter in the state "blocked" if any activity requires a blocking mechanism when executing. The, it can return to "running" state when finished. From the state "running", the mission plan can be paused, reaching so the state "paused" where it can come back to the state "running" if mission plan is resumed, or goes to the state "aborted" if it is aborted.

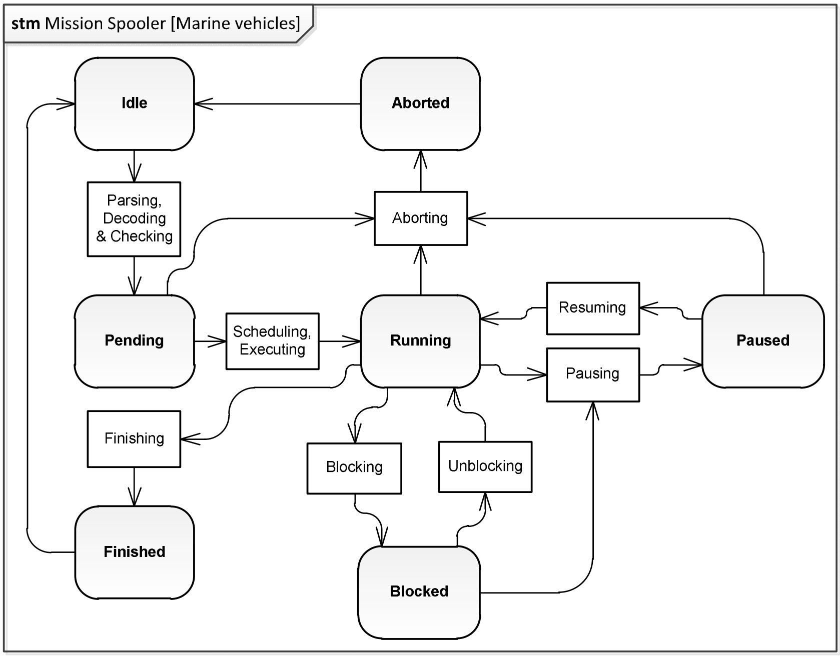

Fig. 4.28 Functional behaviour of the Mission Spooler (Dynamic representation)

Based on the above architecture of the Mission Spooler, three execution models for the mission plan are investigated.

1. **Sequential list of activities.** This model implements the mission plan based on a natural list of activities (as they should be executed) with or without taking into account the conditional defined by the Mission Planner for the execution of each activity. The Mission Spooler just goes through the activities listed, scheduling as a sequence of activities, and executing them one by one. The advantage of this approach is that the mission plan and its execution are simple. The disadvantage is that there is not any chance to perform recursively activities (other than list as many as needed in the mission plan). In addition, no real parallelism can be carried out whilst performing activities.
2. **State machine of activities embedded in the Mission Spooler.** This model takes the activities following the order (sequence) given in the mission plan, schedules them in such an order, and then performs them as scheduled. The advantage of this approach is that the execution block of the Mission Spooler remains relatively simple in comparison with the previous approach. The disadvantage is that the mission plan increases its

complexity a degree. A real parallelism can be implemented in this approach but Mission Spooler should include a multi-threading mechanism to process the scheduled activities.

3. **State machine of activities created whilst scheduling in the Mission Spooler.** This model takes the activities as they are specified in the mission plan. This means the execution block in the Mission Spooler does not have a built-in state machine as in the second approach, and the mission plan fully describes the control and data flow used by the Mission Spooler to schedule the activities. Then, the activities are executed as scheduled. The advantage of this approach is that it is fully flexible to support any mission plan specification (including future instructions). The disadvantage is that the execution block of the Mission Spooler increases its complexity in order to dynamically implement an on-the-fly state machine for the execution of activities. As in the second approach, it supports a real parallel execution of activities if the Mission Spooler includes a multi-threading mechanism to process the scheduled activities.

From the above discussion for the three approaches for the execution model of activities, the last approach (3.) is expected to be implemented in the final version of the Mission Planner and Mission Spooler. The remaining approaches (1. and 2.) are expected to be previous steps to test the architecture implementation in the early stages of development.

4.4.4.2 Matchmaking of Services

Within this architecture, matchmaking of services is enabled by two components:

- matchmaker server
- matchmaker client library

The matchmaker server itself contains a service registry, storing the current state of services in the system. The matchmaker client library allows service providers and consumers to easily perform service advertisement, discovery, and monitoring via the server. Figure 4.29 shows the relationship between service provider, consumer, and matchmaker server. Note that the actual service data connection is direct between the service provider and consumer; the matchmaker server links carry service metadata only.

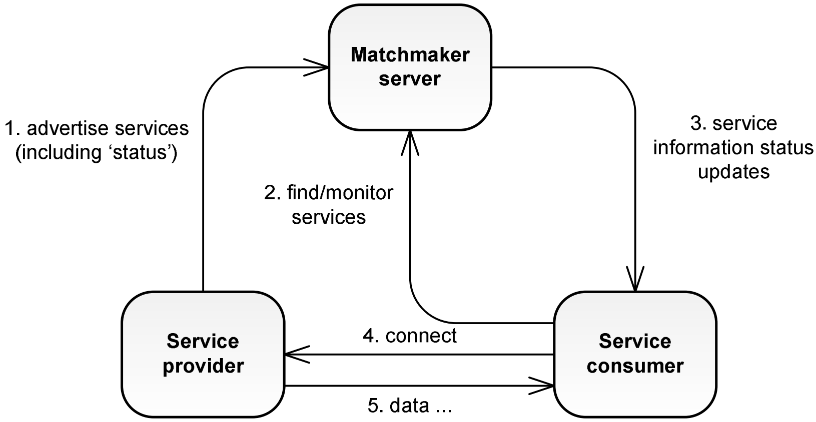

Fig. 4.29 High level operation of matchmaker server with provider and consumer

Table 4.22 describes the states that a service may take; service providers are free to set any of these states. In addition to specifying a status value, service providers may supply an optional text string. This may be used to elaborate on the status value, for example when an error has occurred. Status requests for services that are no longer provided or never existed will result in status REMOVED.

Table 4.22 Possible service states

State	Description
ERROR	Service cannot be provided due to an error
REMOVED	Service advertisement has/is ending
UNMET_DEPENDENCIES	Dependencies needed to provide this service are not available
AVAILABLE	Service is available and currently unused
ACTIVE	Service is in use by one or more consumers, but still available
ACTIVE_BUSY	Service is in use and not available to other consumers

The matchmaker is designed to be used to match service providers to direct consumers of those services. It is the responsibility of each service consumer to use the matchmaker to resolve its own direct service dependencies. For example, the mission spooler uses the matchmaker to find the services it will directly connect to, but not dependencies of these services. If we consider a tree of services as in Figure 4.30, the status of services at the low level will be reflected in the status of services at the higher level; that is, if a low level dependency becomes unavailable, those services that depend on it will become unavailable in turn. Note that this is a logical view of information flow; in reality, service status is communicated via the matchmaker server, as shown in Figure 4.29.

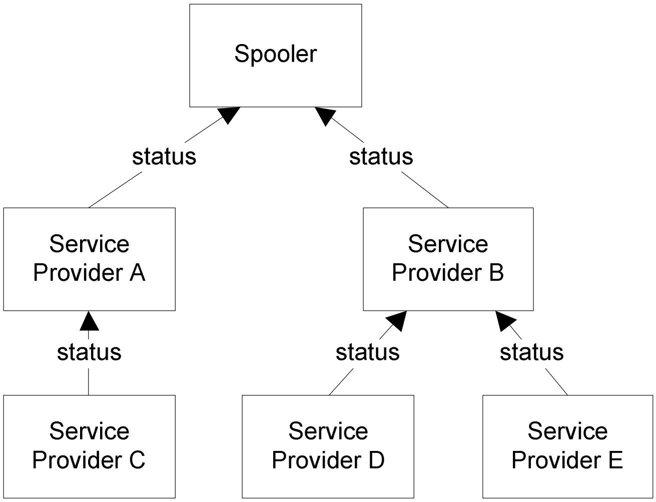

Fig. 4.30 Logical view of hierarchical dependencies; effect of dependents' status flows upwards

A structural view of the matchmaker server is given in Figure 4.31. The service registry is filled from the adverts received from service providers. One-shot queries may be performed on the registry via the query handler. Registries for status monitors and persistent service queries are populated from client update messages. These registries, together with the service registry, are used when sending server update messages. A timeout checker scans all of the registries for expired entries, and removes them when necessary.

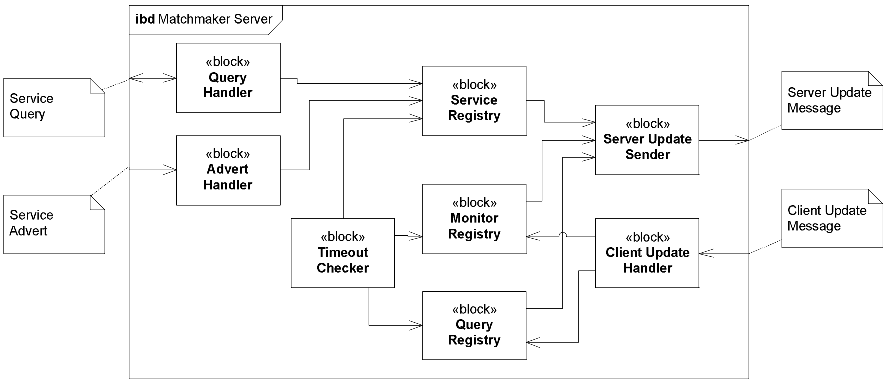

Fig. 4.31 Structural view of matchmaker server