Safety Management in Smart Ships

Massimo Cossentino, Luca Sabatucci, and Flavia Zaffora

National Research Council
Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR-CNR), Palermo, Italy
{name.surname}@icar.cnr.it

Abstract. Smart Ships represent the next-generation of ships and they use ICT to connect all the devices on board to support integrated monitoring and safe management. In such cyber-physical systems, software has the responsibility of bridging the physical components and creating smart functions. Safety is a critical concern in such kind of systems whose malfunctioning may result in damage to equipment and injury to people. In this paper, we deal with this aspect, by identifying two interconnected sub-systems: shipboard power system and emergency management. The proposed architecture is developed through the H-entity multi-paradigm approach, in which heterogeneous technologies are interconnected. We propose to extend the MOISE+ organisational model to deal with systems of H-entities.

Keywords: Safety, Multi-Paradigm, System of Systems

1 Introduction

Smart Ships are new generation of vessels using ICT to connect all the systems and devices on board, aiming at supporting integrated monitoring and control but also safety, energy-efficient operations and management. They can be defined as a system of systems, their main requirements include smartness and safety. Safety-critical systems are those in which a system failure could harm human life, other living things, physical structures, or the environment. Safety in a smart ship may be defined as the avoidance of hazards to people and ship components due to the operation of a device under normal or single fault condition (including mechanical, electrical and software failures). Several European Projects (e.g. Decision Support System for Ships in Degraded Condition, DSS-DC[1],) have been developed in the direction of shipboard safety, deepening the field of Integrated safety and emergency management systems (ISEMS) [16]. Our approach, in line with Cyber-Physical Systems [19], is to integrate heterogeneous components, namely in this case: the Shipboard Power System (SPS) and the emergency procedure management system. The challenge is to combine different parts, both physical and software, to achieve a smart ship solution pursuing the overall goal of safety. We propose a multi-paradigm approach, that combines multi-agent systems and actor models, and adopts existing organizational

[1] https://trimis.ec.europa.eu/project/decision-support-system-ships-degraded-condition

models (MOISE+ [10]) to create a system of systems. The paper is structured as follows: a first part introducing the concept of safety in smart ships and the emergency procedures especially related to the SPS, a second part presenting the proposed architecture and a third one describing the multi-paradigm approach, with a special focus on the organizational side.

2 Safety in the Smart Ship

In the last decade, we are witnessing an advancement of the marine sector, due to the integration of smart technology and automation, aimed at making ships more and more autonomous. In this context, crew and passengers safety is a paramount civil responsibility. In practice, onboard safety depends on: 1) assuring the correct functioning of the equipment necessary for addressing the current mission, and 2) adopting specific emergency procedures to contain or recover failure conditions.

The **Shipboard Power System (SPS)** is the component responsible for granting energy to navigation, communication, and operational systems [2]. It consists of various electric and electronic equipments, such as generators, cables, switchboards, circuit breakers, fuses, buses, and many kinds of loads. In order to grant continuity of services, the electrical circuit is designed for being robust to failures. Loads often are distributed in zones and fed power from the main electric buses. It is usual to classify loads according to their importance into vital and non-vital categories [15], where vital loads are non-sheddable loads that directly affect the survivability of the ship, while the non-vital ones may be shed in order to prevent a total loss of ship's electrical power, or for protection purposes. Moreover, the circuit is instrumented with sensors and actuators for enabling reconfiguration procedures and maintaining safety during operations. The classification of loads into vital and non-vital (sometimes the semi-vital category is considered as well) is not static but it rather depends on the current operational profile and status of the ship. The strategy that enables restoration of the electrical power system is called *SPS Reconfiguration*. Recently, reconfiguration techniques are based on close integration between hardware and software, enabling smart and timely reconfiguration of the electrical layer due to a fault (typically multiple/cascade faults).

The **Emergency Management** is a wide field of studies involving many different disciplines and stakeholders. Its definition is connected to Preparedness and, consequently, to Safety, respectively referring to the ability to respond to an emergency and the need to avoid damages and disasters in a broad sense. Its applications spread from plans at regional/citizen scale to solutions for confined spaces, such as ships and vessels. The issue concerning confined spaces has a precise literature because it leads to very peculiar risks and dangerous situations [6]. The management of emergency has a direct impact on the safety of ship crew and passengers. The onboard procedures established with the emergency plans should handle at least the following emergencies [12] or combinations of them: fire; damage to ship; pollution; unlawful acts threatening the safety of the ship and the security of people; personnel accidents; cargo-related accidents; emergency assistance to other ships. The hardware basis of the emergency management provides all the information collected via devices arranged in different parts of the ship. Sensing is used for the acquisition of physical events and data, also including various types of physical quantities (for example, fire source, temperature, humidity etc). The software basis is the enactment of emergency procedures. It should be in charge to provide all the possible plans related to the current emergency (or concurrent emergencies) and be the smart and safe support for human decisions.

3 The Proposed Architecture

When smartness comes from the union between cyber and physical components, it is the case of framing the product as a Cyber-Physical System (CPS). A CPS integrates the dynamics of the physical processes with those of the digital systems [19] intending to extend the capabilities of the physical layer. Adding the word "smart" to CPS means to define it with a specific quality of *intelligence* that the CPS does not necessarily require. Nevertheless, the smartness of a CPS lies in the property of the reasoning, and in the ability of communicating and sharing knowledge among dissimilar components to take run-time decisions. Developing a smart cyber-physical system is an open challenge due to 1) the inherent complexity of integrating the physical layer with the software layer and 2) the fact that the cyber part must encompass adaptivity, environment programming, open, distributed and scalable functionalities, with an intensive participation of humans. By surveying the state of the art, two main paths emerge for facing this complexity: actors and agents. The choice of a suitable paradigm is not trivial.

The problem of *SPS reconfiguration* has been recently faced off in [17]. The proposed solution has been obtained by adopting MUSA, a Middleware for User-driven Service Adaptation. This middleware, in its last version[2], is implemented as a hierarchy of Akka actors. The advantage of adopting a middleware for self-adaptation in this context consists in achieving the capability to adapt the repair strategy to the operative context, including the external environment and the current mission the ship is undertaking. In practice, different goals derive from different missions, and therefore, the various components of the ship (loads) assume a different priority. Reconfiguration plans are a verticalization in this domain of the approach proposed by [18]. The actor model enables implementing an autonomous monitor-analyse-plan-execute model. In the literature, *Emergency Management* is often faced off by integrating Multi-Agent Systems and Internet of Things [13,20]. The IoT is supposed to provide functionalities for the overall perception of information, reliable transmission of information and intelligent processing of information which can achieve the object of intelligent control and management, especially useful for emergency cases [21]. An integrated safety and emergency management system applied to ships can be defined as computer-based support to maintain all safety functions onboard [16].

[2] MUSA is available online at https://github.com/icar-aose/musa_2_scala

It should coordinate all the involved entities, by integrating the captain's decision, crew's operations, passengers' movements and navigation main functions to pursue safety during an emergency.

The proposed architecture [3] is obtained as the integration of the Multi-Agent System for the enactment of emergency procedures, as discussed in [13], with the actor model for the SPS reconfiguration [17]. However, the problem of integrating these two subsystems is not trivial because of the mutual dependencies that may exist. The objective of the following sections is providing a framework for multi-paradigm development.

4 The H-entity Multi-Paradigm Approach

The proposed aim here is to develop a smart cyber-physical system respecting some requirements especially involving the information exchange among heterogeneous systems, an ability we can refer to as "interoperability". Therefore, interoperability could determine a sort of "translation" among elements, languages, frameworks originally independent, acting on the communication-side of a CPS. A "smart" CPS uses specific frameworks involving some reasoning, and the need for communication has to include a social aspect along with an aptitude to adaptation. These elements explicitly belong to agent-oriented languages, while the necessity of enacting feedback loops and scaling system's functionalities suggest to adopt an actor-based language. Dealing with the above-mentioned Smart Ship architecture, under the constraint of using different available technologies (the CPS challenge), lead to the adoption of a multi-paradigm approach. Unlike Multi-Paradigm Modeling [11], that refers to a domain dealing with a complex heterogeneity of models, the problem here exposed is to use different programming paradigms with different underlying programming languages: multi-agent systems and actor models. In the specific, JADE [3], Jason [5] and Akka [9] have been taken into account, highlighting their common points and their peculiarities [7]. Although they are all Java-based, their differences are several, and they descend from the concepts behind their main entity, the *agent* for JADE and Jason, the *actor* for Akka. Moreover, Jason's agents are based on a BDI model, whereas the JADE platform is compliant with FIPA standards [1]. In the following, their strengths will be summarized.

- *Jason* is based on a BDI (belief-desire-intention) model, thus the agent shows a decision-making ability [5]. Human-like, this ability is realised by a sequence of tasks composing a plan that is enacted to pursue a goal.
- *JADE* [3] exploits the object-oriented paradigm and a cooperative task scheduling to implement autonomous agents. The main features are protocols-oriented communications, strong support to ontologies and full support for FIPA specifications.

[3] Further details about the proposed Shipboard Power System reconfiguration system and its architecture are available at http://ecos.pa.icar.cnr.it/research-topic/.

– *Akka* is the Scala implementation of the actor model whose peculiar structure lies in the father-children hierarchy [9]. It is greatly useful when developing a reactive system in terms of parallel, asynchronously communicating processing.

For the above-mentioned aim, none of them can be substituted without losing something, so instead of settle with just one, the adopted strategy was creating an *H-entity* in order to choose all of them, basically adopting each entity strength points. A briefly introduction of the H-entity is provided in the following section.

4.1 The H-entity Metamodel

The so-defined H-entity, where 'H' stands for *heterogeneous*, holds the entire philosophy here proposed. It can be considered as a polyhedral organism characterized by autonomous entities and explicable only by reference to the interconnection of these parts. We distinguish an internal and an external view of an H-entity.

The **Internal view**: an H-entity is a (closed world) organization of heterogeneous but collaborating autonomous entities (typically agents and actors). The whole H-entity is characterized by its own design goal, and by a complex behavior. Each member is devoted to one or more tasks for addressing the collective goal. Members have different responsibilities. The *Manager* is responsible for reasoning and decision making (it is typically a Jason agent). The *Worker* is responsible for providing core functionalities and services. It is typically an Akka actor (more than one are usually employed in the solution). The *Diplomat* maintains external relations with other H-entities. It is typically a JADE agent. In the context of a CPS, an H-entity is a cohesive team aimed at solving a specific macro-functionality. However, the overall system is generally constituted by several H-entities. Designing the smart ship, we identify an H-entity for the SPS reconfiguration subsystem and 1..n H-entities for each emergency procedure (fire, flooding, sanitary, evacuation and so on).

The **External view**: the interaction of many H-entities leads to the concept of system of systems, where the open world hypothesis holds. H-entities are not pre-defined and may appear/disappear at run-time; the overall behaviour emerges contextually. For instance, the SPS H-entity must consider active emergency H-entities, when it modifies the configuration of powered components. To allow unknown entities to profitably interact, communications must be done on formal basis, establishing a common language, semantics and protocols. This paper focuses on the external view (only a few details will refer to the internal one). Figure 1 details the external view of the H-entity meta-model.

4.2 The External Organization

A system of systems is defined as a large-scale concurrent and distributed system, the components of which are complex systems themselves. This paradigm grants the operational independence of the individual systems [14]. H-entities are

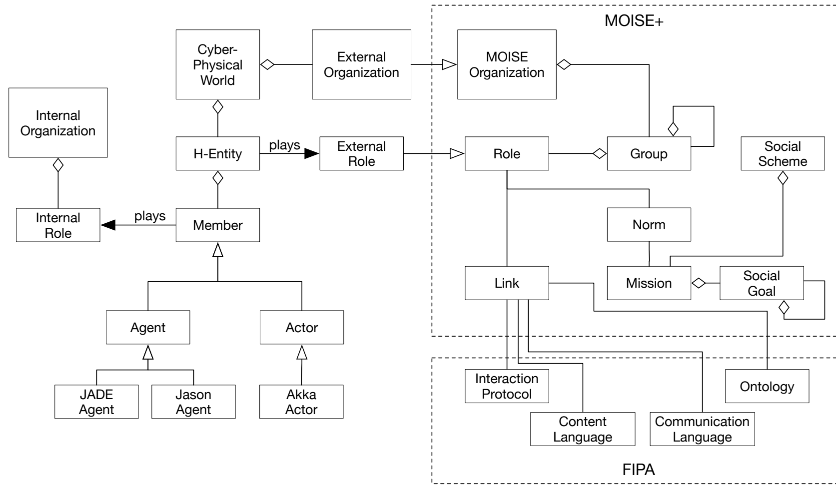

Fig. 1. A portion of the meta-model of the H-entity highlighting the External View.

complex systems because they are made of many heterogeneous entities. They are designed to address some functional requirement, but also to live in an open world. Therefore, the idea is that an overall behaviour emerges by the interaction among H-entities. It is necessary to adopt an instrument to formalize the formation of organizations of H-entities to realize a "system of systems". An example of such instrument is MOISE+ [10], a framework for specifying the organization of a multi-agent system by defining the structure as well as functional and deontic aspects. In MOISE+, the functional aspect describes organization goals, the structural aspect defines groups and roles and the deontic aspect describes the relation between goals and roles via permissions and obligations. Roles are an excellent instrument to decompose and distribute goals without specifying agent responsibilities. Indeed agents may dynamically play roles. This dynamicity is necessary when designing open systems whose participants are not assigned apriori. The choice of MOISE+ is supported by the fact that it has already integrated with Jason agents (in the Jacamo framework [4]). We propose to extend the current version of MOISE+ in order to explicitly deal with some of the problems of the open-world systems. Indeed, the ability of agents to interoperate in open and dynamic environments would be facilitated by the use of public and standard specifications. To this aim, the Foundation for Intelligent Physical Agents [1] (FIPA) provides the specifications for open distributed computing environment integrating unknown agents through the use of technologies such as interaction protocols, agent communication languages, and ontology.

4.3 MOISE+ For Systems of Systems

Integrating MOISE+ with the grounding principles of FIPA would create a language to specify organizations operating as open and interoperable systems of systems. In MOISE+, the Organisational Specification (OS) is defined by its three dimensions: structural, functional, and normative [10]. We focus on the Structural Specification (SS) that is defined by: 1. a set of roles, 2. a set of inheritance relations among roles, 3. the root group.

Groups represent the shared context for agents playing roles in it. A Group is composed by Roles, Links and Role Compatibility Relations. The original definition of Link is "the relation between roles that directly constrains the agents in their interaction with the other agents playing the corresponding roles". A Link is defined by:

- source and target roles of the link;
- the type of the link (e.g. acquaintance, communication, or authority);
- the scope of the link (e.g. inter-group or intra-group).

In order to support the open-world hypothesis, we add four constraints into the definition of Link:

- the Interaction Protocol, i.e., the specification of the pre-agreed sequence of messages to be exchanged to communicate effectively;
- the Communication Language (typically grounding on the speech-act theory) that defines the set of performatives and their meaning; examples are FIPA-ACL [1] and KQML [8] (Knowledge Query and Manipulation Language);
- the Ontology, i.e., the concepts, predicates and actions to be used to formalize the semantics of messages content.
- the Content Language, i.e., the language used to serialize and deserialize the content of a message; examples are SL, KIF and RDF.

The following source code shows an example of structural specification with the new definition of Link. At lines 19-22, protocol, language and ontology are specified. These additional elements represent the formal aspects that are expected to be used for implementing a communication among H-entities. A Link prescribes interaction protocol, communication language and an ontology for the interaction be effective.

```
1  <structural-specification>
2    <role-definitions>
3      <role id="SPS"></role>
4      <role id="emergency"></role>
5      <role id="fire"><extends role="emergency" /></role>
6      <role id="evacuation"><extends role="emergency" /></
          role>
7      ...
8    </role-definitions>
9
10   <group-specification id="smart_ship_safety">
11     <roles>
12       <role id="SPS" min="1" max="1" />
13       <role id="emergency" min="1" max="10" />
```

```
    </roles>
    <links>
      <link
        from="SPS_diplomat" to="emergency_diplomat"
        type="communication"
        protocol="IteratedValidation"
        acl="FIPAACL"
        language="SL0"
        ontology="onboard_components"
        scope="intra-group" bi-dir="false"/>
    </links>
  </group-specification>
</structural-specification>
```

4.4 The Diplomat Role

To let the internal view and the external view be integrated, one (or more) specific member(s) of the H-entity will play the role of *Diplomat*, being responsible for the communications with other H-entities.

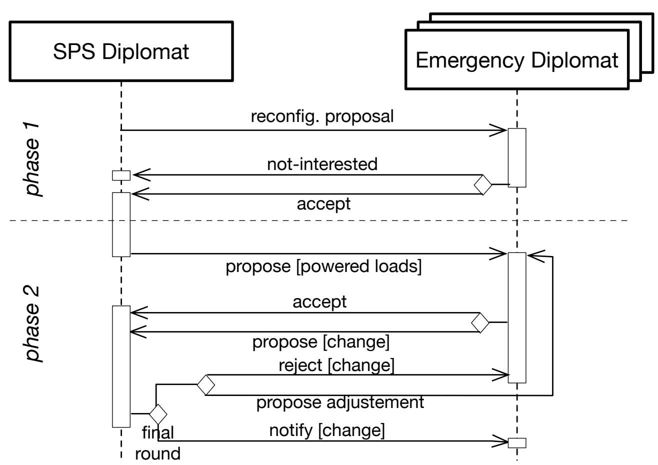

Fig. 2. The IteratedValidation, i.e. an Iterated Contract Net adapted for the SPS/Emergency interaction.

The *Diplomat* is an expert in using standard languages and protocols, also referring to a formalized conceptualization of the domain (ontologies). The presence of the Diplomat ensures the ability of the H-entity to act in an open world where entities developed by different providers could live, interact, and cooperate. We suggest JADE agents act as diplomats because they can exploit FIPA-compliant semantic communications organized through interaction protocols.

In Figure 2, we provide an example of the IteratedValidation protocol used to allow an orchestrated validation of the SPS reconfiguration plan with the help of the interested emergency H-entities. It is a variant of the Iterated Contract Net [1], modified to allow the SPS manager to refine the electrical reconfiguration taking into account conflicts with the active emergency management H-entities. Indeed, when the generators do not produce enough energy for powering the whole ship components, the reconfiguration strategy often provides power to only a subset of the loads, and therefore it shades some physical components, according to the mission priorities. However, it may happen an emergency plan requires a specific load to be switched on (for instance a corridor for evacuation purposes). The protocol has a first phase in which emergency diplomats who are involved in enacting emergency procedures respond to the call. The second part is multi-step interaction in which the SPS diplomat and the active Emergency diplomats negotiate about which components of the ship may be switched off. The use of this protocol allows applying market strategies to let a better allocation of the physical resources.

5 Conclusion

Safety is a paramount requirement in a smart ship, in which the automation of most of the functions is critical for the crew and the passengers. A possible solution for integrating many independent safety sub-systems is to adopt a multi-paradigm approach: agents and actors live together in abstract structures called H-entities. This paper proposes to use agents' organization to glue together H-entities, but the open-world hypothesis mandates the use of an additional layer based on interaction protocols, languages and ontologies that is realised by extending MOISE with link constraints. An example of a specific interaction protocol (Iterated Validation) is presented as a solution for coordinating the actions of the SPS and Emergency H-entities.

References

1. Foundation for Intelligent Physical Agents (FIPA): FIPA Specification Repository (2005), http://www.fipa.org/repository/index.html
2. Agnello, L., Cossentino, M., De Simone, G., Sabatucci, L.: Shipboard power systems reconfiguration: a compared analysis of state-of-the-art approaches. In: Smart Ships Technology 2017, Royal Institution of Naval Architects (RINA). pp. 1–9 (2017)
3. Bellifemine, F., Poggi, A., Rimassa, G.: Jade: a fipa2000 compliant agent development environment. In: Proceedings of the fifth international conference on Autonomous agents. pp. 216–217. ACM (2001)
4. Boissier, O., Bordini, R.H., Hübner, J.F., Ricci, A., Santi, A.: Multi-agent oriented programming with jacamo. Science of Computer Programming **78**(6), 747–761 (2013)
5. Bordini, R., Hübner, J., Wooldridge, M.: Programming multi-agent systems in AgentSpeak using Jason, vol. 8. Wiley-Interscience (2007)

6. Botti, L., Duraccio, V., Gnoni, M.G., Mora, C.: A framework for preventing and managing risks in confined spaces through iot technologies. In: Safety and Reliability of Complex Engineered Systems-Proceedings of the 25th European Safety and Reliability Conference, ESREL. pp. 3209–3217 (2015)
7. Cossentino, M., Lopes, S., Nuzzo, A., Renda, G., Sabatucci, L.: A Comparison of the Basic Principles and Behavioural Aspects of Akka, JaCaMo and Jade Development Frameworks. 19th Workshop From Objects to Agents (WOA 2018), Palermo, 28-29 June 2018
8. Finin, T., Fritzson, R., McKay, D., McEntire, R.: Kqml as an agent communication language. In: Proceedings of the third international conference on Information and knowledge management. pp. 456–463. ACM (1994)
9. Gupta, M.: Akka Essentials. Packt Publishing, Birmingham (2012)
10. Hannoun, M., Boissier, O., Sichman, J.S., Sayettat, C.: Moise: An organizational model for multi-agent systems. In: Advances in Artificial Intelligence, pp. 156–165. Springer (2000)
11. Hardebolle, C., Boulanger, F.: Exploring multi-paradigm modeling techniques. Simulation **85**(11-12), 688–708 (2009)
12. House, D.J.: Seamanship techniques: shipboard and marine operations. Routledge (2013)
13. Katayama, K., Takahashi, H., Yokoyama, S., Gäfvert, K., Kinoshita, T.: Evacuation guidance support using cooperative agent-based iot devices. In: 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE). pp. 1–2. IEEE (2017)
14. Olivier, J.P., Balestrini-Robinson, S., Briceño, S.: Approach to capability-based system-of-systems framework in support of naval ship design. In: 2014 IEEE International Systems Conference Proceedings. pp. 388–395. IEEE (2014)
15. Padamati, K.R., Schulz, N.N., Srivastava, A.K.: Application of genetic algorithm for reconfiguration of shipboard power system. In: 2007 39th North American Power Symposium. pp. 159–163. IEEE (2007)
16. Rødseth, Ø.J., Joung, T., Lie, H., Rialland, A., Mo, B., Hermundstad, O.A., Lindstad, H.E., Haase, M., Halvorsen-Weare, E., Berg, T.: Passenger ship safety and emergency management control. In: Lloyds register and Fairplay conference Cruise and Ferry (2005)
17. Sabatucci, L., Cossentino, M., Simone, G.D., Lopes, S.: Self-reconfiguration of shipboard power systems. In: In Proceedings of the 3rd eCAS Workshop on Engineering Collective Adaptive Systems (2018)
18. Sabatucci L., Cossentino M.: Supporting dynamic workflows with automatic extraction of goals from bpmn. ACM Transactions on Autonomous and Adaptive Systems (TAAS). (In printing)
19. Shi, J., Wan, J., Yan, H., Suo, H.: A survey of cyber-physical systems. In: Wireless Communications and Signal Processing (WCSP), 2011 International Conference on. pp. 1–6. IEEE (2011)
20. Turoff, M., Hiltz, S.R., Bañuls, V.A., Van Den Eede, G.: Multiple perspectives on planning for emergencies: An introduction to the special issue on planning and foresight for emergency preparedness and management (2013)
21. Yang, L., Yang, S.H., Plotnick, L.: How the internet of things technology enhances emergency response operations. Technological Forecasting and Social Change **80**(9), 1854–1867 (2013)