# A Top Down Approach to Ensure the Continuity of the Different Design Levels of Swarm Robots

Khalil Aloui, Amir Guizani, Moncef Hammadi, Mohamed Haddar, Thierry Soriano

# A Top Down Approach to Ensure the Continuity of the Different Design Levels of Swarm Robots

Khalil Aloui
QUARTZ Lab EA7393 SUPMECA
3 rue Fernand Hainaut 93400 Saint Ouen France
alouika95@gmail.com

Amir Guizani
*University of Sfax LA2MP Ecole Natioanle d'Ingénieurs de Sfax*
km 4 route de la Soukra Sfax 3038 Tunisia
amir.guizani@live.fr

Moncef Hammadi
QUARTZ Lab EA7393 SUPMECA
3 rue Fernand Hainaut 93400 Saint Ouen France
moncef.hammadi@supmeca.fr

Mohamed Haddar
*University of Sfax LA2MP, Ecole Natioanle d'Ingénieurs de Sfax*
km 4 route de la Soukra Sfax 3038 Tunisia
mohamed.haddar@enis.rnu.tn

Thierry Soriano
QUARTZ Lab EA7393 SUPMECA
3 rue Fernand Hainaut 93400 Saint Ouen France
thierry.soriano@univ-tln.fr

***Abstract*—Swarm engineering is a systematic application of scientific and technical knowledge to specify requirements, model, design, realize, verify, validate, operate and maintain a swarm intelligence system. In swarm robotics, there is not a well-structured methodology until today for developing robotic swarm systems. Several researchers have developed steps to design swarm robots but these steps are still incomplete.**

**In this paper, we focus on the functional architecture of the swarm robots where we propose a top-down approach to ensure consistency and continuity from requirement level to behavioral level up to the functional and structural levels. This approach is based on the Model-Based Systems Engineering method (MBSE) using the Systems Modeling Language (SysML) where we present the allocations between the functions of each swarm member and the overall swarm behaviors. Then, we will be interested in the architecture of Robot Operating System (ROS) of a swarm behavior where we identify the allocations between the component and the functions of a robot.**

***Keywords— Swarm robotics, MBSE method, Robot Operating System, collective behaviors, swarm functions***

## I. INTRODUCTION

The engineering of swarms is complicated. it is difficult to understand, analyze and design swarm robot systems due to the difficulty of separating swarms into simpler parts[1]. In recent years, developing a structured design method for designing swarm robot systems is an open challenge but the effort on this topic has been very limited [2]. Traditionally, designers have generally used the code-and-fix approach to design and develop swarm robot systems [3]. During this approach, the developer designs, tests and modifies the individual behaviors of the robots until a desired collective behavior is achieved. This approach relies on the ingenuity and expertise of the designer and takes a lot of time.

Kazadi et al. [4] presented a design approach based on Hamiltonian vector fields called the Hamiltonian method. The method uses a mathematical description of collective behavior to derive microscopic rules and minimize or maximize a selected numerical value. The limitation of this method is that it only deals with spatial organization behaviors such as pattern formation.

Brambilla et al. [5] developed a top-down design method for developing collective behaviors of swarm robot systems called Property Driven Design. This approach describes the swarm robot system through a set of desired properties. The method consists of four phases: In the first phase, the developer specifies the requirements of the swarm by identifying the desired properties. In the second phase, the developer creates a normative model of the swarm and verifies that this model satisfies the desired properties. In the third phase, the developer simulates and validates the prescriptive model developed in the previous steps. In the last phase, the developer implements the simulated version of the swarm on real robots.

Aloui et al. [6] proposed an approach to improve the Property Driven Design method. They used the SysML language to specify requirements and model collective behaviors: SysML state machines are used to describe robot behaviors. The robot behavior models described

with SysML are then implemented with a multi-agent technique [7], [8].

The majority of the design approaches developed above require the continuity and the consistency between the different design aspects. Indeed, a model of swarm robotic system shall represent, in a consistent way, different aspects of the system that can be summarized by: requirements, collective behavior, local and global structures and parameters. These aspects must be interconnected to ensure the model continuity and consistency. The model of swarm robotic system must also include links to model interdependency between the different views. Systems engineering tackles the system as a whole and considering all the facets [9]. Using MBSE method, we can represent: structural composition, interconnection, and classification, function-based, message-based, and state-based behavior, constraints on the physical and performance properties, requirements and their relationship to other requirements, design elements, and test cases, allocations between behavior, structure, and constraints (e.g., functions allocated to components), and finally the traceability of requirements [10].

To develop a model of swarm robots with the MBSE method, we use the SysML language [11], [12]. This language consists of a set of diagrams such as:

- Requirements diagram represents the textual requirements.
- Activity diagram represents the behaviors in terms of the relationship between inputs and outputs and represents how actions transform inputs into outputs.
- Sequence diagram represents the behavior in terms of the sequence of the messages exchanged between the parties.
- State machine diagram represents the behavior in terms of its transitions between states triggered by events.
- The use case diagram represents the functionality to achieve a set of goals.
- The block definition diagram represents the structural elements (blocks) and their composition and classification.
- The internal block diagram represents the interfaces and the interconnection between the parts of a block.

In this paper, we will focus, in the first section, on the functional architecture of the swarm robots where we will propose a top-down method to ensure continuity from requirement level to behavior level up to the function and structural level. This method is based on the MBSE method using the different SysML diagrams. In the second section, we will be interested in the Robot Operating System (ROS) architecture of a swarm behavior where we will apply our method to an aggregation case of swarm robots to identify the structural architecture of our system.

## II. PROPOSED METHODOLOGY

According to Fig.1, a group of autonomous robots constitutes collective swarm behaviors that meet given swarm requirements. It is a top-down method starting with the identification of global swarm requirements. these requirements are divided into sub-requirements to be expressed by collective robot behaviors. These global behaviors are constituted of the functions of each swarm member. Each robot must perform certain functions to ensure the proper functioning of the swarm. And finally, it is necessary to identify the structural architecture of the swarm through the different functions and behaviors.

Throughout this approach, different SysML diagrams are used such as requirements diagrams to specify swarm requirements. State machine diagrams and activity diagrams are used to identify collective behaviors and individual functions of swarm members. Besides, block definition diagrams and internal block definition diagrams are used to describe the structural architecture of the system.

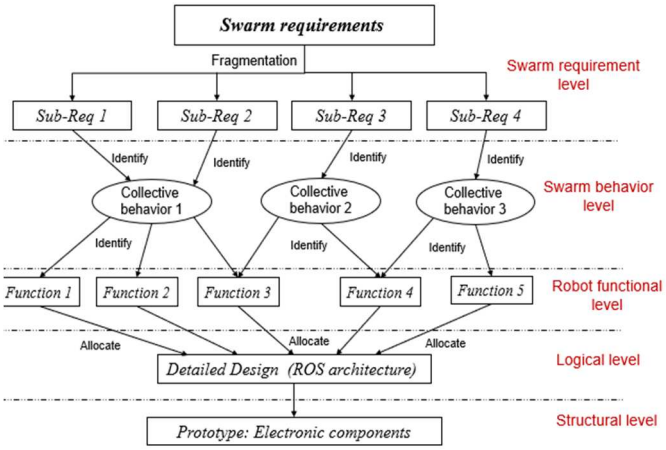


Fig. 1. A top-down method for developing swarm robot systems

### A. Requirement Level

In this step, SysML requirements diagrams are used to define the requirements that the swarm has to meet, such as flexibility which means the ability to cope with a wide range of different environments and tasks, the autonomy which means that individuals must physically interact with the world and be autonomous. Also, scalability means the ability to perform well with different group sizes, and robustness means the ability to cope with the loss of individuals. And finally, maintenance plays a key role in achieving a desired

global behavior among a swarm of robots, plus problems related to energy consumption. See Fig. 2.

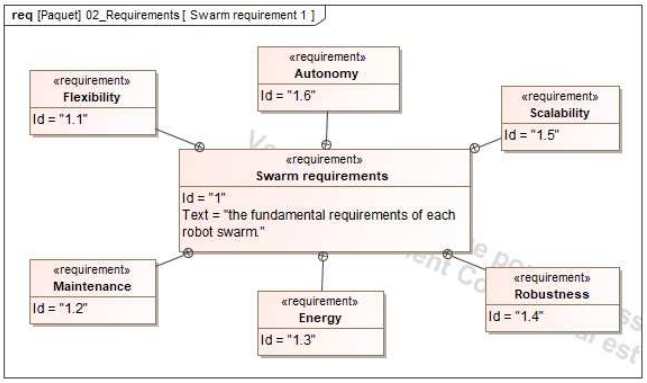


Fig. 2. Basic swarm requirements

## B. Behavioral level

Swarm robots have many collective behaviors. Brambilla et al [13] classified the behaviors of swarm robots into four classes: navigation behaviors, spatial organization behaviors, collective decision-making, and other collective behaviors. This classification is illustrated in Fig.3.

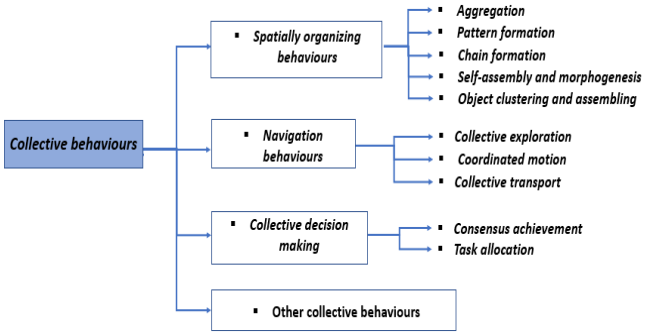


Fig. 3. Taxonomy of swarm behaviors [13]

The state machine diagram shown in Fig. 4 shows that the collective behavior of the swarm is established through a set of functions of individual robots [14]. The swarm member (a robot) begins with the first function such as "Moving forward" in most cases. Subsequently, it finds a choice. According to a selection criterion, for example, a probability value or a message, it will decide to choose one of two choices: either to establish function 2 (for example "Wait") or to establish the two simultaneous functions 3 and 4 (for example "Take the object" and "Go to the target"). And finally, the program is closed by a final state.

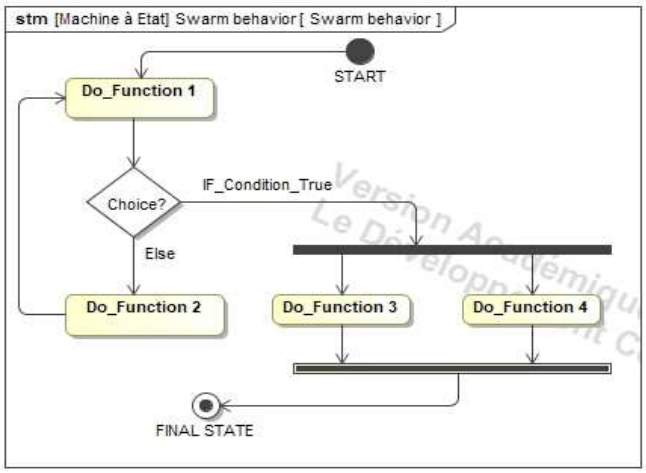


Fig. 4. Generic swarm behavior

## C. Functional level

In this phase the functions performed by the swarm robotic system must be determined. The main system functions are broken down into sub-functions that are connected in a way to transform input flows to output flows. Fig.5 shows that the collective behaviors of robots are defined through all the functions offered by the system. A function is an action performed by the system or by one of its parts. For example, in swarm robotics the functions offered by the system are the functions performed by each robot. A set of functions form a collective behavior of the robots. This behavior on the other hand satisfies a swarm requirement.

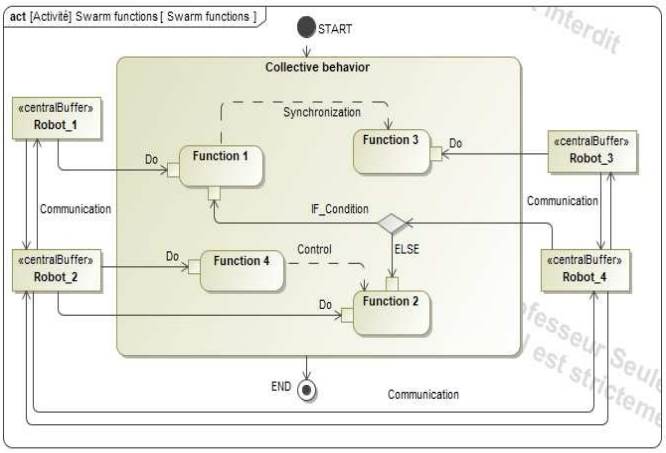


Fig. 5. Functional level of collective behavior

To ensure some behaviors such as aggregation, pattern formation, the robot walks forward, repels, waits, and approaches other robots autonomously. Another example, the collective transport of swarm robots requires that individual robots walk forward, grip, and deposit objects. Indeed, the search neighbors, detection, and avoidance of obstacles are still necessary to ensure the majority of swarm behaviors. Connecting and following other robots is essential to combine the data locally sensed by the robots in the swarm into a big picture. It allows the swarm to make collective

decisions in an informed way, e.g., to classify objects reliably or to determine the optimal solution to a global problem [15]. Besides, agree on or converge toward a single common choice from several alternatives assures the consensus of the individual robots in the swarm.

Fig. 6 represents the possible allocations between the individual functions of each robot and the collective behaviors of the whole swarm. Generally, the individual functions form the overall aspect of the swarm behavior (i.e., the functions that produce each behavior).


Fig. 6. Function-Behavior allocation

## *D. Logical level*

In this step, functions (activities) are allocated to logical components. So, the system breaks down into logical components with different alternative solutions. We therefore use the ROS platform as an operating system. The Robot Operating System (ROS) is a set of tools defined for writing robot software. It is a set of conventions and libraries for creating the complex and robust behavior of a robot on a wide variety of robotic platforms.[16]. Fig. 7 describes the ROS architecture of a swarm robots.

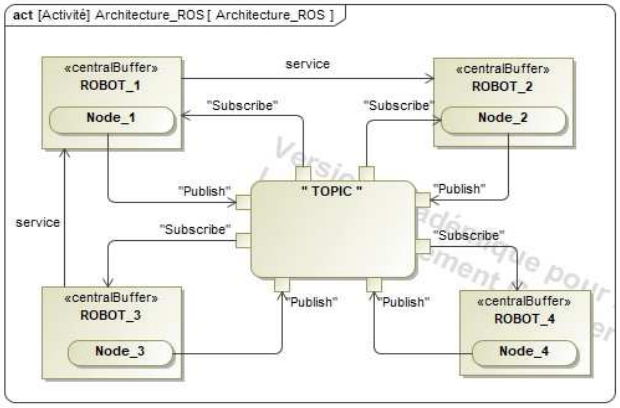


Fig. 7. ROS architecture of a swarm robots

Fig. 8 shows the traceability diagram between functions and swarm requirements.

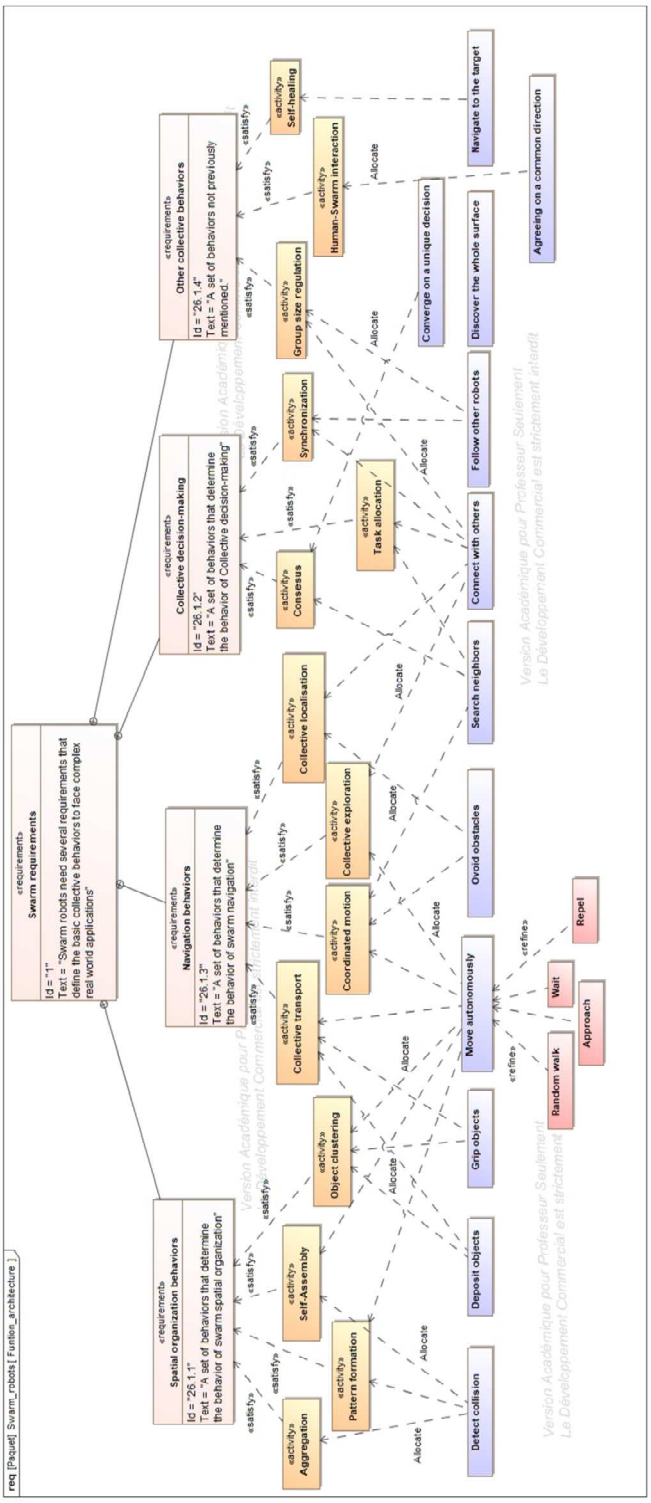


Fig. 8. Traceability diagram between functions and requirements.

In ROS, the node can correspond to a sensor, a motor, a processing algorithm, a robot monitoring. The exchange of information takes place either asynchronously via a topic or synchronously via a service exchanged between robots. A topic is a system-based information transport system (subscribe / publish). One or more nodes of the robots will be able to publish information on a topic and one or more nodes will be able to read information on that topic [17].

### *E. Structural level*

In the last step, the system functions are identified. These functions will be achieved by components. Fig. 9 represents the general architecture of the swarm.

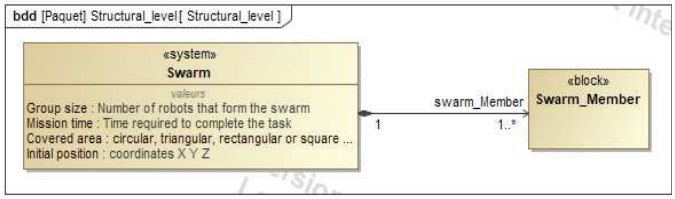


Fig. 9. The swarm architecture

First, the developer identifies the different alternative solutions that he wants to develop. Then, for each solution, he identifies the components that will achieve each function (or group of functions). Finally, he builds the system breakdown structure by creating the components and links each component to the system through a composition link. Fig.10 shows the different components of a robot in a swarm.

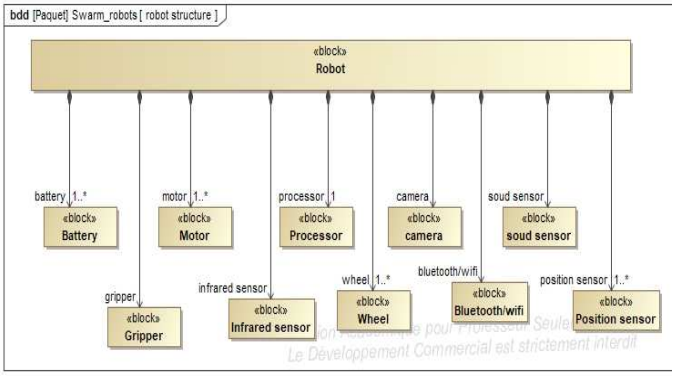


Fig. 10. The swarm member architecture

Swarm architecture and swarm member architecture are specified by using SysML Block Diagram Defition (BDD) [18]. The structure of the swarm member depends on the required functions. Generally, robots consist of batteries to supply power, motors to run, and wheels to ensure movement while other components are specified according to the appropriate behavior such as infrared sensors to explore the environment and position sensors to ensure localization.

Fig.11 represents the component-function allocations of the system.


Fig. 11. Component-Function allocations

In the next section, we apply our design approach to a collective behavior of swarm robots. We choose aggregation as collective behavior to study it using SysML diagrams and ROS platform in modeling.

## III. CASE STUDY: AGGREGATION

The swarm robot aggregation means that the individual robots meet spatially in a specific zone of the environment. This allows individuals in the swarm to come spatially close to each other for further interaction [15], [19], [20].

### *A. Requirement Level*

In this step, a requirements diagram is used to specify the requirements of our case study. As indicated in Fig.12, the aggregation of a swarm of robots requires that the robots must choose the target where they regroup. Besides, robots must be autonomous without any external intervention. Finally, the robots must avoid collision to ensure the formation of an aggregate.

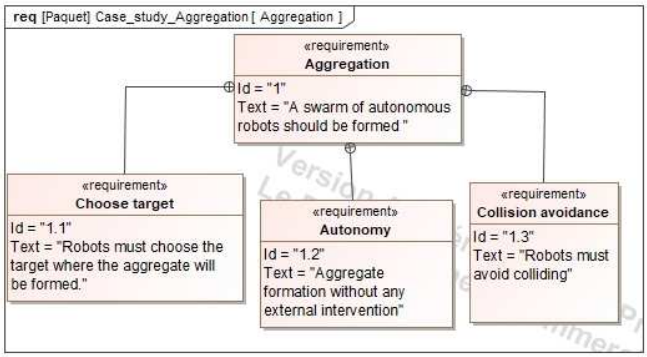


Fig. 12. Robots aggregation requirements

### *B. Behavioral level*

The state machine diagram shown in Fig. 13 gives a detailed description of the aggregation behavior of swarm robots [21]. The robot begins to move forward. It checks if there is a collision with another robot: if there is a collision with another robot, it calculates the driving feedback vector and the collision feedback

vector and merge them to find the global feedback vector. Using this global feedback vector, the wheel speed manager sets the speed to return. If there is no collision, it returns randomly and continues to move forward.

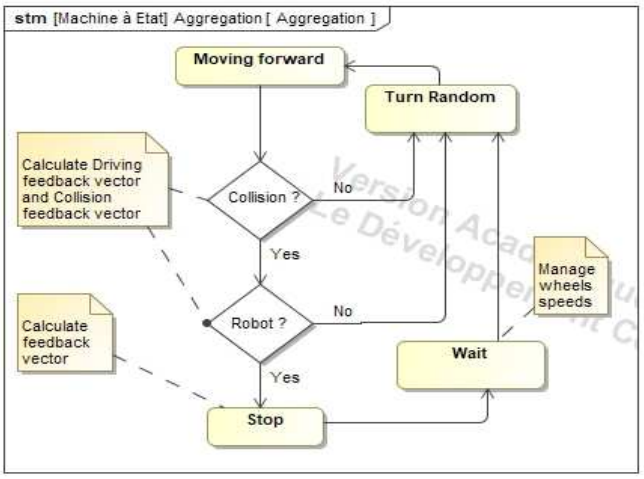


Fig. 13. A state machine diagram of an aggregation behavior

## C. Functional level

In this step, each robot performs all of the individual functions identified in the previous section. The activity diagram shown in Fig. 14 shows the different functions of the individual robots to form the aggregate and the links between them.

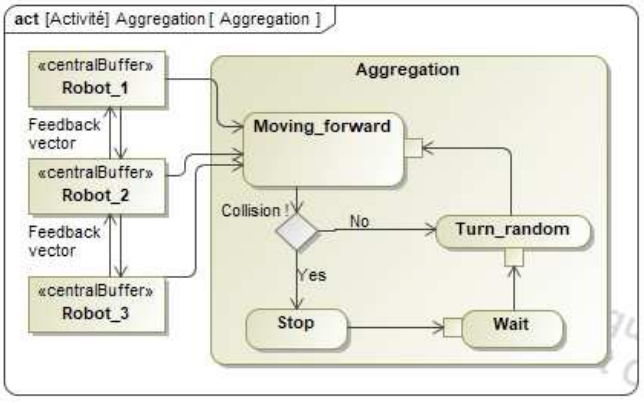


Fig. 14. Functional level of an aggregation behavior

## D. Logical level

Fig. 15 represents the ROS architecture that performs the perception, treatment, navigation and control functions. The navigation is performed by using the navigation stack of ROS [22]. It detects the position and orientation of obstacles and neighboring robots through the integration of infrared sensors, position sensors and camera. It calculates the driving feedback vector and collision feedback vector. Then, it merges the two vectors to determine the global feedback vector. The navigation service sends service request of wheel velocities to the control service to manage the function of the motor and wheels [23].

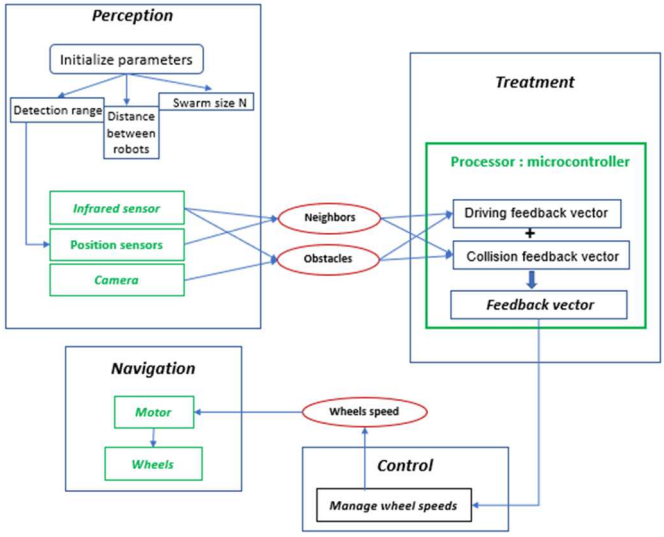


Fig. 15. ROS architecture for robot perception, navigation and control

## E. Structural level

In this step, the developer should identify the different components of the selected architecture to be developed and model the components interactions, their interfaces, and the exchanged flows while being consistent with the functional architecture. The block definition diagram illustrated in Fig. 16 represents the different components of our system.

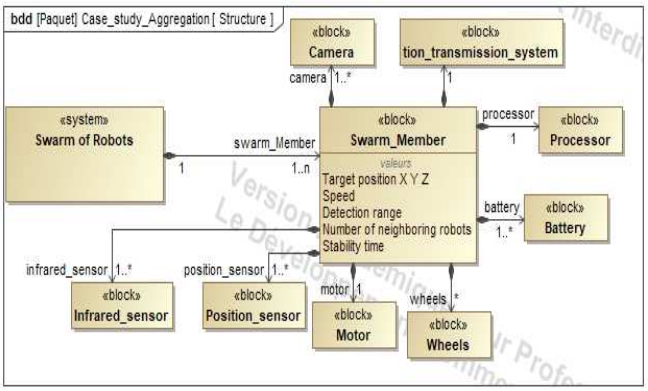


Fig. 16. The swarm architecture

The internal block diagram indicated in Fig. 17 represents the interactions between the various components identified above.

Finally, after identifying the functions of the system that forms the aggregation behavior and creating the structure of the system, it is necessary to allocate each component to its functions to ensure continuity between design levels.

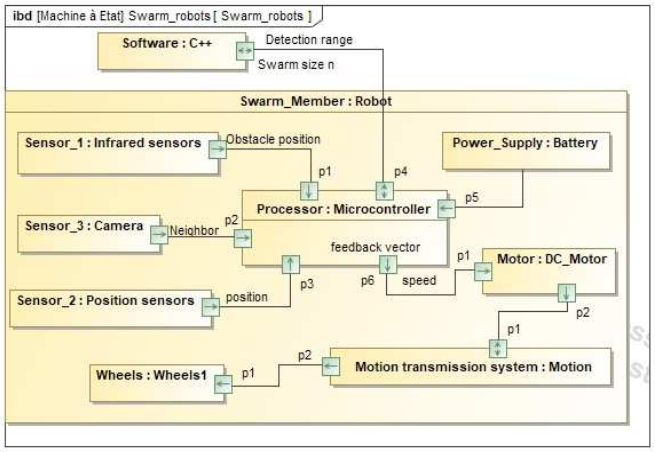


Fig. 17. Internal Block Diagram

Fig. 18 defines the allocation matrix of the components of the system with its functions.

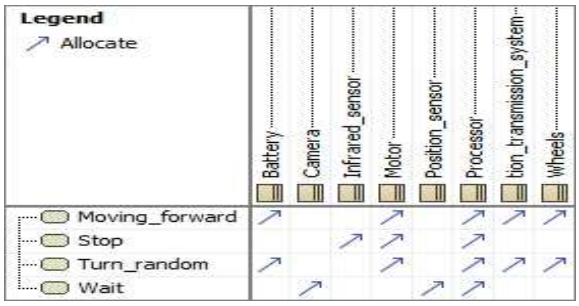


Fig. 18. Component-Function Allocation

## IV. ANALYSIS AND DISCUSSION

To validate the approach used in this work, we present in Table 1 a comparison of this design approach with other methods developed for the design of swarm robots [3], [13], [4]. The selection criteria used are as follows: flexibility which means the ability of a method to adapt to changes, a continuity which represents the passage from modeling to simulation towards implementation with real robots, the advantages which are the strengths of each method and finally the quality of the final product (ie whether the method ensures the robustness, flexibility, and extensibility of the final product).

Each design method has limitations. Our approach indeed facilitates the design of swarm robot systems, but modeling a swarm robotic system is a difficult task. For example, the temporal and spatial aspects of the system or the robot-robot interaction are always difficult to capture in a model. However, the simulation of the system can be complicated depending on the complexity of the system to be developed. it requires the ingenuity and expertise of the developer.

TABLE I. COMPARATIVE STUDY BETWEEN DESIGN METHODS

| Criterion / Method | *Flexibility* | *Continuity* | *Quality of final product* | *Advantages* |
|---|---|---|---|---|
| *Code-and-fix approach* | - unsuitable (based on testing)<br>- take time | - weak continuity<br>- models depends on materials | - not satisfied<br>- robust but not extendable | - a simple process |
| **Property Driven Design** | - flexible( check every phase) | - strong continuity<br>- simple transition between phases | - satisfied | - reduce the risk of developing wrong systems |
| **Hamiltonian method** | - brittle (based on mathematical modeling) | - weak continuity<br>- simple transition from simulation to implementation | - Produce perfectly symmetrical hexagonal networks | - applied for the design of complex systems |
| **Proposed Approach** | - flexible (based on SysML diagrams | - strong continuity<br>- simple transition between phases | - satisfied<br>- meet the requirements | - a simple process<br>- consistency<br>- continuity |

## V. CONCLUSION

To date, most swarm robot design approaches require continuity and consistency between these different design levels. In this article, we proposed a top-down method to ensure continuity from the level of requirements through the level of behavioral and functional to the structural level. This method is based on MBSE method using various SysML diagrams and ROS architecture.

In the first part, we developed a top-down design approach where we specified the generic requirements of a swarm. Subsequently, we identified the different collective functions and behaviors of swarm robots and we presented the allocations between them. In fact, a set of functions forms a collective behavior of robots. On the other hand, this behavior satisfies a swarm requirement. And finally, we described the structure of our system using the internal block diagram (IBD), the block definition diagram (BDD) and the ROS architecture. In the second part, we applied our developed approach to a robot aggregation case study. In the future, we aspire to know the high-level and low-

level ROS architecture of other collective behaviors such as pattern formation and collective transport and create a passing method of ROS logic architecture (high level) to the ROS architecture (low level).

## REFERENCES


[1] L. Bayindir et E. Şahin, «A review of studies in swarm robotics,» *Turkish Journal of Electrical Engineering & Computer Sciences,* vol. 15, p. 115–147, 2007.

[2] S. Stepney, F. A. C. Polack et H. R. Turner, «Engineering emergence,» chez *11th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'06)*, 2006.

[3] M. Brambilla, A. Brutschy, M. Dorigo et M. Birattari, «Property-driven design for robot swarms: A design method based on prescriptive modeling and model checking,» *ACM Transactions on Autonomous and Adaptive Systems (TAAS),* vol. 9, p. 1–28, 2014.

[4] S. Kazadi, J. R. Lee et J. Lee, «Artificial physics, swarm engineering, and the hamiltonian method,» chez *World congress on engineering and computer science*, 2007.

[5] M. Brambilla, C. Pinciroli, M. Birattari et M. Dorigo, «Property-driven design for swarm robotics,» chez *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 2012.

[6] K. Aloui, M. Hammadi, T. Soriano, A. Guizani et M. Haddar, «On the continuity of the swarm robot design using MBSE method and simulation,» chez *13th International Conference on Modelling, Optimization and Simulation (MOSIM'20),*, 2020.

[7] A. Guizani, M. Hammadi, J.-Y. Choley, T. Soriano et others, «Multi-agent approach based on a design process for the optimization of mechatronic systems,» *Mechanics & Industry,* vol. 18, p. 507, 2017.

[8] A. Guizani, M. Hammadi, J.-Y. Choley, T. Soriano, M. S. Abbes et M. Haddar, «Agent-based approach for collaborative distributed mechatronic design,» chez *2014 10th France-Japan/8th Europe-Asia Congress on Mecatronics (MECATRONICS2014-Tokyo)*, 2014.

[9] K. Giles et K. Giammarco, «A mission-based architecture for swarm unmanned systems,» *Systems Engineering,* vol. 22, p. 271–281, 2019.

[10] F. Mhenni, J.-Y. Choley, O. Penas, R. Plateaux et M. Hammadi, «A SysML-based methodology for mechatronic systems architectural design,» *Advanced Engineering Informatics,* vol. 28, p. 218–231, 2014.

[11] R. Cloutier, B. Sauser, M. Bone et A. Taylor, «Transitioning systems thinking to model-based systems engineering: Systemigrams to SysML models,» *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 45, p. 662–674, 2014.

[12] A. Cavalcanti, A. Miyazawa, A. Sampaio, W. Li, P. Ribeiro et J. Timmis, «Modelling and verification for swarm robotics,» chez *International Conference on Integrated Formal Methods*, 2018.

[13] M. Brambilla, E. Ferrante, M. Birattari et M. Dorigo, «Swarm robotics: a review from the swarm engineering perspective,» *Swarm Intelligence,* vol. 7, p. 1–41, 2013.

[14] W. F. W. Othman, S. P. McKibbin, F. Caparrelli, J. R. Travis et B. P. Amavasai, «Pattern formation and organisation in robot swarms,» chez *Proceedings of the IEEE SMC UK-RI Chapter Conference on Applied Cybernetics (London, UK)*, 2005.

[15] M. Schranz, M. Umlauft, M. Sende et W. Elmenreich, «Swarm Robotic Behaviors and Current Applications,» *Frontiers in Robotics and AI,* vol. 7, p. 36, 2020.

[16] M. Quigley, B. Gerkey et W. D. Smart, Programming Robots with ROS: a practical introduction to the Robot Operating System, " O'Reilly Media, Inc.", 2015.

[17] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler et A. Y. Ng, «ROS: an open-source Robot Operating System,» chez *ICRA workshop on open source software*, 2009.

[18] M. Schranz, A. Bagnato, E. Brosse et W. Elmenreich, «Modelling a CPS Swarm System: A Simple Case Study.,» chez *MODELSWARD*, 2018.

[19] N. E. Shlyakhov, I. V. Vatamaniuk et A. L. Ronzhin, «Survey of methods and algorithms of robot swarm aggregation,» chez *Journal of Physics: Conference Series*, 2017.

[20] S. Ramroop, F. Arvin, S. Watson, J. Carrasco-Gomez et B. Lennox, «A bio-inspired aggregation with robot swarm using real and simulated mobile robots,» chez *Annual conference towards autonomous robotic systems*, 2018.

[21] Y. Liu et K. Lee, «Probabilistic consensus decision making algorithm for artificial swarm of primitive robots,» *SN Applied Sciences,* vol. 2, p. 95, 2020.

[22] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey et K. Konolige, «The office marathon: Robust navigation in an indoor office environment,» chez *2010 IEEE international conference on robotics and automation*, 2010.

[23] A. Ruiz-Larrea, J. J. Roldán, M. Garzón, J. del Cerro et A. Barrientos, «A UGV approach to measure the ground properties of greenhouses,» chez *Robot 2015: Second Iberian Robotics Conference*, 2016.