A Multilevel Agent-Based Approach to model and simulate Systems of Systems

Jean-Baptiste Soyez, Gildas Morvan, Rochdi Merzouki, Daniel Dupont

A Multilevel Agent-Based Approach to model and simulate Systems of Systems

Jean-Baptiste Soyez, Gildas Morvan, Rochdi Merzouki, Daniel Dupont

***Abstract*—This article proposes a generic modeling approach of systems of systems (SoS) using agent-based modeling (ABM). SoSs are large scale systems including numerous - possibly heterogeneous - interacting component systems (CS) evolving in a dynamic environment. The aim of this article is to provide generic formalism allowing to represent and control the whole complexity of a SoS using agent-based simulations. Models generated using this formalism encompass static and dynamic aspects of SoSs. They consider reorganization of SoSs caused by changes of goals or sub-system capacity. All these elements are illustrated in this article using a SoS case study of Intelligent Automated Vehicles (IAV) initiated by the InTraDE (Intelligent Transportation for Dynamic Environment) European project to automate the port container logistic.**

***Keywords*—Agent-based modeling ; System of Systems ; Complex system engineering**

I. INTRODUCTION

This paper deals with the design and control of artificial complex systems, i.e., large scale systems composed of numerous communicating components [1]. Such systems can be distinguished from other systems by the importance of the number of entities, the number of interactions between entities, the size of the environment and the level of detail of all these represented elements. All these parameters can evolve over time. System of Systems (SoS) can be considered as a particular type of Federation of Systems (FoS): systems that are themselves composed of complex systems. Moreover, in SoS component systems (CS) operate independently and are driven by their local goals, nonetheless they have to cooperate in order to fulfil global goals potentially contradicting local ones [2]. Indeed, it is a promising concept that should be able to capture the whole complexity of such real systems.

Initially, in the beginning of 21st century, SoS were focused to military applications in engineering systems [3], [4], Currently, the concept has extended to other areas such as education [5], transport [6], security [7], service [8] . . .

Our work takes place in the context of the European project InTraDE [9]. It deals with automated container loading and unloading operations in large-scale container port terminals of northwest Europe (Dublin, Oostende, Le Havre and Rouen), using Intelligent Autonomous Vehicles (IAV) controlled by a central operator. IAVs are semi-autonomous vehicles, in charge of container transport tasks. IAVs can react to changes in their environment or their attributed goals, they can also reason about their internal dynamic and communicate with human operators or other IAVs. The container terminal system consists in several IAVs, cranes, operators and boats. It can be seen as a SoS with numerous heterogeneous elements and several organizations present at different levels and communicating with each other. To illustrate the proposed SoS modeling and simulation approach, a single practical case, taken from the InTraDE project will be considered throughout this article. A SoS model seems adapted to consider organizational and functional aspects unfolded on several levels of an IAV fleet. Thus, this article modeling example concerns an IAV fleet.

Our goal is to create a modeling tool for SoSs in order to engineer and operate such systems. Controlling a SoS means insuring that it will fulfil its global goals through its CSs or indicating that it is impossible. To do so, we need a formalism able to define a SoS in terms of the behaviour of its CSs. However, it does not exist any generic, rigorous and complete enough modeling method for SoSs. In this article we introduce a formalism able to capture the static and dynamic aspects of SoSs. This formalism allows to divide the complexity of a system by splitting it into a set of levels and organizations that represent the system at different scales and structured by groups of CSs.

A. Related Works

Because SoSs have been introduced in various domains, dedicated tools have been proposed to model and operate them. Nevertheless, until recently, SoSs remained a theoretical concept with no generic simulation formalism. Several modeling approaches of SoSs are non generic and focus on domain-related issues: Parker proposed a classical modeling framework [10], defining interfaces between existing CSs. Huhn et al. used the agent-based modeling (ABM) paradigm to represent situated CSs constrained by their place in a discretized environment [2]. Sloane et al. introduced a model representing the effects of climate policy also using an ABM [11]. Zhou et al. proposed a modeling method for SoSs in the manufacturing domain using ABM [12].

Held also developed a formalism to represent SoS models, using metrics and attributes [13]. But this representation focuses on data more than on SoS entities. The proposed tool is designed to evaluate and predict the performances of a SoS rather than controlling one.

Khalil et al. proposed a formalism based on hypergraphs to represent directed SoSs [14] but they did not provide generic mechanisms to ensure that the constitutive SoS characteristics are preserved during their evolution (adding or deleting CS, change of general goal).

This literature survey shows that the most generic attempts to model SoSs rely on ABM. Indeed, there are many similarities (and some differences though) between ABMs and SoSs, such as the autonomy property of agents and CSs.

Jean-Baptiste Soyez and Gildas Morvan are with the LGI2A laboratory, University of Artois. E-mail: jbaptiste.soyez/gildas.morvan/@univ-artois.fr

Rochdi Merzouki is with the LAGIS laboratory, University of Lille 1. E-mail: rochdi.merzouki@polytech-lille.fr

Daniel Dupont is with the HEI engineering school of the Catholic University, Lille. E-mail: Daniel.DUPONT@hei.fr

B. Contributions

This article offers two main contributions. 1) It selects and enhances a **SoS definition** generic enough to be accepted by the SoS community and concrete and precise enough to be operational in a modeling and simulation context. 2) It proposes a generic **SoS formalism** exploiting the previous definition. In this formalism, organizational aspects of SoSs are managed with Agent-Group-Role (AGR) model [15]. Functional aspects, guiding SoSs to accomplish their global goals, are handled via a functional specification. And Multi-level aspects are modelled with the Influence Reaction Model for Multi-Level Simulation (IRM4MLS) agent-based meta-model [16]. IRM4MLS makes model generated using this formalism easy to apprehend and to divide its computational complexity cutting the SoS by scales or independent aspects. Such models allow to represent IAV fleet of the InTraDE project which are SoSs directed by a global goal.

C. Plan

This article is organized as follows. Section II defines what is a SoS, and gives a graphical representation for SoSs. Section III introduces the concept of Multi-Agent Systems (MAS) and presents ABM tools, adapted to represent complex systems like SoS. In section IV, we give static and dynamic elements necessary to describe Multi-Agent SoS (MASoS) accounting for their organizational and multi-level aspects. Subsequently, we show how to link the previous elements to achieve system-level goals to allow the execution of a MASoS. Finally, we give some conclusions in section V and discuss future perspectives.

II. System of Systems (SoS)

In this section we present the organizational concept of SoS and we explain the need of a solid formalism to represent SoSs. Then, a graphical notation is introduced to represent SoSs in the rest of this article.

A. SoS Definition

Many definitions of SoS have been proposed in the literature. This article leans on Sage and Coppan's definition [17], [18], inspired by Maier's one [19], that can be summarised as follows:

> A SoS is a set of autonomous Component Systems (CS) endowed with a global goal. CSs can be heterogeneous. They manage theirs own resources and under CSs in an independent way and can coexist and cooperate to accomplish a mission that a single CS cannot realize. CSs are geographically distributed without any physical link. A SoS has to be robust and adaptive: its environment, goal or structure (by adding/deleting CS) can evolve without modifying its capacity to fulfil its global goal.

The following lexicon is used in this article. A group of CSs forming another CS, in a higher level, views this CS as their **super CS** and are viewed by this CS as its **under CSs**.

This definition respects the five following characteristics illustrated with an example from the InTraDE project.

1) **Operational independence**: Each CS possesses its own resources necessary to accomplish its mission. For example, an IAV has its own stock of energy and its personal status diagnosed online.
2) **Managerial independence**: Once a mission is attributed to a CS, it manages itself and its under CSs, on its own, to accomplish it. For example, a fleet of IAVs attached to a quay decides on its own how to organize its IAVs to accomplish its allocated missions.
3) **Geographic distribution**: CSs can exchange information but there is no physical exchange of energy. For example, when two IAVs are attached to form a platoon to be able to carry 40 feet containers, they lost their geographic distribution and are not considered as CSs anymore.
4) **Emergent behaviour**: The global goal of a SoS can only be reached by the joint action of more than one CS. For example, if one IAV is sufficient to reach the SoS goal to perform all transport tasks in the port, the SoS has no reason to exist.
5) **Evolutionary development**: A SoS can adapt itself to the addition or deletion of CSs or evolution of the environment or change of the global goal to stay able to reach it by the actions of CSs. For example, when an IAV endowed with a mission loses its capacity to fulfil it, it informs its super CS. This super CS has means to overtake this loss of capacity by allocating the mission to another CS agent with intact capacities.

Similar characterizations like Bordman and Sauser's one [20] exist, which provides the following characteristics differentiating a SoS from monolithic systems: *autonomy*, *belonging*, *connectivity*, *diversity* and *emergence*. [21] gave a SoS definition acknowledged by the SoS community:

> "Systems of systems are large-scale integrated systems that are heterogeneous and independently operable on their own, but are networked together for a common goal".

Other definitions exist but can not be exploited here due to their vagueness or to the restrictive nature of the interaction between their CSs [19], [22]. A more comprehensive definition, in relation with the MAS paradigm, is given in section IV.

B. Graphical Representation and CS Taxonomy

In this section a graphical representation of the real entities of a SoS is introduced. For more details on CS notation see section IV. Let us consider an example from the InTraDE project. A whole fleet of IAVs, $CS_{1,3}$, is composed of quay fleets from quay 1, $CS_{1,2}$, and quay 2, $CS_{2,2}$. The $CS_{1,2}$ fleet is composed of a platoon $CS_{1,1}$ (including IAVs $CS_{1,0}$ and $CS_{2,0}$) and an IAV $CS_{3,0}$. fleet $CS_{2,2}$ is composed of three IAVs: $CS_{4,0}$, $CS_{5,0}$ and $CS_{6,0}$.

Some CSs or set of CSs can temporarily violate some of the SoS characteristics explained further. We call these entities **virtual component systems**. For example $CS_{1,0}$ and $CS_{2,0}$ are IAVs physically linked to form a platoon. They have partially lost their independence, but still exist. In addition a CS that cannot be divided into CSs respecting these characteristics is called an **elementary component system** (like $CS_{1,1}$, $CS_{3,0}$, $CS_{4,0}$, $CS_{5,0}$ and $CS_{6,0}$). Basically, the physical individual entities of a system are

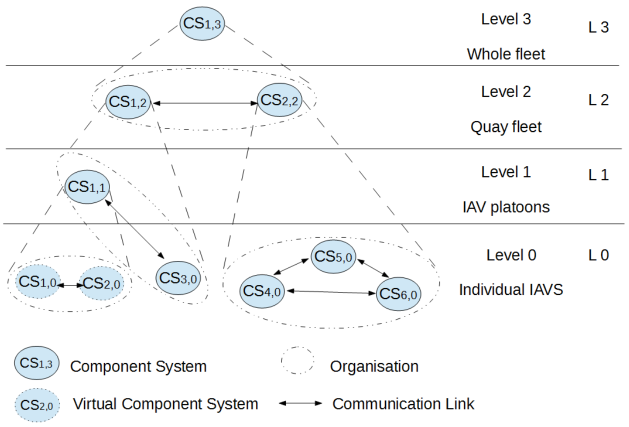

Fig. 1. Multi-level graphical representation of a SoS.

considered as elementary CSs (like $CS_{3,0}$, $CS_{4,0}$, $CS_{5,0}$, $CS_{6,0}$ and potentially $CS_{1,0}$ and $CS_{2,0}$). Opposed to elementary CSs, the only CS present at the highest level, $CS_{1,3}$, which is equivalent to the SoS can be named the **top component system**.

In the following sections, we will model CSs using CS agents taking advantage of autonomy and independence inherent to agents. CS agents are agents representing CSs in SoSs and exhibiting the same characteristics as CS.

III. Multi-Agent Systems (MAS)

This section presents MAS modeling tools and how they are applied to represent complex systems like SoS.

A. MAS Definition

Most authors generally agree to define a MAS as a system composed of communicating and collaborating agents, that have objectives (personal or collective) and resources to achieve them. Communication implies the existence of a shared space to support it. This space is generally called environment. Ferber defined agents as follows [23].

We call agent a physical or virtual entity

1) which is able to act in an environment,
2) which can communicate (directly) with other agents,
3) which is driven by a set of tendencies (under the form of individual objectives or a satisfaction or survival function that it tries to optimize),
4) which possesses its own resources,
5) which is able to perceive (in a limited way) its environment,
6) which disposes only of a partial representation (eventually none) of this environment,
7) which possesses competencies and offers services,
8) which can eventually reproduce itself,
9) whose behaviour tends to satisfy its objectives with its resources and its competencies and according to its perceptions, representations and communications it receives).

We call MAS a system composed by an environment in which a set of agents can act, to perceive create or modify other agents and the environment and a set of relations linking these agents together.

B. Agent-Based Modeling of Complex Systems

MAS are widely used to simulate interactions between complex, autonomous entities acting in parallel. This kind of entities are easily found in complex systems like SoS. First of all, a SoS should be considered as an organizational concept, and therefore it can be compared to MAS organizational models such as AGR (Agent-Group-Role) [15]. This formalism organizes a system into groups depending in which agent(s) play which role(s).

Gaud et al. used a hierarchical multi-agent meta-model (CRIO) [24] based on the concept of holon. A holon represents both an individual entity and an organization composed of a structured set of entities. CRIO contains also an organizational decomposition aspect.

Hübner et al. studied organization oriented MAS and to insure that the system will complete its global goal while guaranteeing agents autonomy in terms of behaviour and creation of organizations, they used deontic specification to link groups and allocation of missions in their model: $\mathcal{M}oise^{+}$. In a way this is similar to our approach to represent SoSs [25]. But in our approach, the existence of groups is strongly constrained by the necessity to fulfil the global goal of the system.

Ribino et al. created a meta-model mixing holonic organizational aspects of ASPECS and deontic specification (renamed norms) issued from $\mathcal{M}oise^{+}$ to instantiate holonic MAS (HMAS) [26]. The created HMAS are designed to reach a global goal, coordinating independent entities and respecting the system integrity during its changes. Our approach is inspired through consideration of similarities and differences between SoSs and holonic systems.

Scerri et al. presented a MAS model to represent systems with the High Level Architecture (HLA) [27]. HLA is a meta-model that allows the coexistence and coordination of different simulations in the same platform. Even if HLA is more concerned with implementation than our approach, the proposed model deals with coexisting independent entities forming different levels of organization.

Picault et al. also presented a MAS model, PADAWAN, whose specificity is to include agents and their nested environment in other agents [28]. This way of viewing a system with aggregate organizations at a level creating individuals at a higher level is common with SoS.

Morvan proposed a bibliography to list the existing multi-level agent-based modeling approaches [29]. This article identifies the different theoretical issues, meta-models, platforms and applications related to that domain.

Our formalism is based on the multi-level agent-based model IRM4MLS. It allows the representation of multiple entities in interaction situated in different levels. A level is not necessarily a scale [30]. Moreover interactions between entities in different levels and their results are represented without any bias owing to the influence reaction model used in IRM4MLS [31].

IV. Agent Formalism to Represent SoS

Organizational and hierarchical definition and global goal and missions expressed in a multi-level way are crucial concepts. Because they are needed to express the characteristics of managerial independence, evolutionary development, cooperation and coexistence.

In this section, we give all static and dynamic elements necessary to represent a SoS and that are not basically included in all MAS. Then, we give some elements to prove that multi-agent models, created using our formalism, respect SoS fundamental characteristics.

A. Static Aspects

SoS is an organizational concept in which CSs are entities that can be individuals or aggregations of CSs in lower level(s). A good way to represent these groups and provide the super CS a way to reason about the mission distribution in order to reach its allocated goal is to use the Agent-Group-Role model (AGR) [32].

From the AGR point of view, an agent is an entity playing a set of roles in several groups, with its knowledge and capacities. An agent or group *capacity* is the possibility for this entity to accomplish a goal, i.e., reaching a target level state, by the means of its actions, in a precise given level state.

An *agent role* describes the behaviour in context defined by a group. It confers a status and means in this group to interact with other agent(s) playing role(s) in that group. These groups possess capacities determined by the roles played by agents and their individual capacities.

1) CS Modeling: To designate a CS in a SoS in a unique way and without any ambiguity we can use the following formalism: $CS_{n,l}$ designates the nth CS and its representing agent in the level l. Here is a definition inspired by Gaud [24]. A CS, CS_n of a level l, can be noted as follows:

$CS_{n,l} =< CS_{n',l+1}, R_{n,l}, \mathcal{CS}_{n'',l-1}, OP, \Psi, \Lambda >$, where :

- $CS_{n',l+1}$: a super CS which is directly composed of $CS_{n,l}$ and possibly other under CSs. This CS has to be in a level higher than l.
- $R_{n,l}$: the set of roles played by $CS_{n,l}$, $R_{n,l} = 2^{Roles(O_{n,l})}$ with $O_{n,l}$ the set of groups in which $CS_{n,l}$ plays at least one role.
- $\mathcal{CS}_{n'',l-1}$: the set of under CSs constituting the super CS: $CS_{n,l}$. These CSs have to be in level(s) lower than l.
- OP: the set of groups taking part in the life and functioning of $CS_{n,l}$ CS and contributing to accomplish the objectives linked to the $R_{n,l}$ roles.
- $\Psi : \mathcal{CS}_{n'',l-1} \rightarrow 2^{Roles(OP)}$: A function associating an under CS member to the set of roles that it plays in groups defined in $CS_{n,l}$, such as $\forall csi \in \mathcal{CS}_{n'',l-1}, \Psi(csi) \neq \emptyset$. The role function provides the set of all roles defined in OP groups.
- $\Lambda : \Sigma_l * \Sigma_l \rightarrow \{0, 1\}$: A capacity function indicating if a goal state in l is reachable, by the action of $CS_{n,l}$, knowing the actual state of l. This function depends on the evolving capacities of $CS_{n,l}$ and also on the capacities offered by OP to $CS_{n,l}$ when $CS_{n,l}$ is not an elementary CS.

In the case of SoS engineering there is a hierarchy of levels of increasing abstraction, approximation and scale. These levels can be represented by a hierarchical graph, as shown above, which indicates the level imbrication. When a level l_i is directly included in a level l_j, CSs in l_i compose higher scale CS in l_j. A CSs can directly belong to only one CS of a higher level, therefore, here, a level can be directly included in at most one level. When a CS $CS_{i',j'}$ is an under CS of $CS_{i,j}$ it is noted $CS_{i',j'} \in CS_{i,j}$

2) Group of CSs Modeling: To decide how to form super CS from organization of under CSs and how CSs are organized to reach goals it is necessary to endow our model with set of group specification defined as follows:

$gs =_{def} < R, L_R, \mathcal{L}, C^{intra}, C^{inter}, nr, nc >$, where

- R: the set of non abstract roles playable by agents in groups created from gs.
- $L_R : R_{gs} \rightarrow L$: a function indicating the level of an agent which can play a given role in gs.
- $\mathcal{L}$: indicates the level(s) of the group created from gs. if $\mathcal{L}$ contains only one level l and l is higher than any level of $L_R(R)$ then the group defined by gs has to be instantiated by a CS agent situated in l. CS agents playing role(s) in R are the under CSs of this new CS agent.
- C^{intra}: a compatibility function which indicates if two roles are compatible in the same group. By default, two roles are not compatible. $\rho_a \Delta \rho_b$ says that agents playing role ρ_a are authorized to play role ρ_b. This relation is reflexive and transitive.
- C^{inter}: a compatibility function which indicates if two roles are compatible in two different groups. It possesses the same formalism than C^{intra}.
- $nr : R_{gs} \rightarrow N * N$: a function which specifies the number (minimum, maximum) of roles which has to be played in a group, ex., $nr_{gs}(carrier) = (1, 3)$ means that groups created using gs have to possess at least one and at most 3 agents playing the $carrier$ role.
- $nc : Rgs * C \rightarrow N * N$: a function which specifies for a given role and capacity the quantity (minimum, maximum) of this capacity which has to be available for each agent playing this role and for the whole population of agents playing this role.

Here a capacity should be expressed in a functional way and has to be quantified, i.e., not expressed in terms of goal states but in terms of concrete results of agent actions. $nc_{gs}(carrier, carryOneContainer/hour) = (1, \infty, 3, \infty)$ means that an agent playing the role carrier in a group produced by gs has to possess at least one unit of the $CarryOneContainer/hour$ capacity and the whole population of agents playing this role in the group has to possess at least three cumulative units of the $CarryOneContainer/hour$ capacity with in both cases no maximum limit for this quantified capacity.

3) Multi-level Environment and Goal Decomposition: IRM4MLS provides a support to define global goals. A global goal is a world state to be reached by the system, which can be expressed as a description of the system and its environment variables. A global goal can be divided into a set of missions. This cut out can be functional, geographic or other. Because the modeling of SoSs include multi-level representation, this cut out can be done according to levels. Thus missions of a level can be expressed as states of this level, i.e., the state of CSs and environment of this level. Moreover the global goal of a SoS can be expressed as a goal state of the highest level L_H. It is used to define the global goal of the SoS : $Gg = \delta_{L_H}(t') =< \sigma_{L_H}(t'), \gamma_{L_H}(t') >$. t' can be considered as a time limit for the SoS to reach Gg.

Because it is the most abstract and approximate, the wished state $\delta_{L_H}(t')$ of the highest level L_H can be divided into a set of states in every lower level corresponding to local goals which describe more in details the accomplishment of the global goal.

B. Dynamic Aspects

In this section we present the mechanisms that insure the feasibility of the global goal during SoS evolution. The first one concerns the creation of groups with capacity matching accomplishment of the global goal and the second concerns the change of capacities for CS.

1) SoS Creation: In our approach we try to incite CS agents to adopt an appropriate organizational structure to cooperate and fulfil a global goal. At the outset of the SoS, outside the top CS and all CSs with a physical existence, like most of the elementary CSs, all intermediary CSs have to be instantiated. The situation is quite similar when the global goal or even a non elementary CS goal is modified: all its under CSs except those with no under CS (virtual or not) have to be created again. This is done in order to design organizations with capacities adapted to reach the new goal and its associated sub-goals. To form organizations of CSs matching the SoS global goals, our model has to lean on a functional specification.

Missions can be seen as sequences of CS agent actions required to meet a goal. Missions in that model display the fact that a CS agent can engage simultaneously to fulfil several goals within the same mission. The fact that a mission $\mathcal{M}x$ is allocated to the $CS_{n,l}$ CS is noted $\mathcal{M}x_{n,l}$. The *global goal* is decomposed as a graph tree. Each time a goal is divided into sub-goals with a change of scale, it shows that these goals will be accomplished by a CS in one level and its under CSs in at least another level. A goal g_x can be noted with its corresponding level l: $g_{x,l}$.

Here follows a graph tree presenting a possible global goal allocated to the SoS presented in fig. 1. CS agent $CS_{1,3}$ would be engaged on $\mathcal{M}1$, $CS_{1,2}$ on $\mathcal{M}2$, $CS_{2,2}$ on $\mathcal{M}3$ and so on. So the set of missions for this goal decomposition should be equal to $\mathcal{M} = \{\mathcal{M}1_{1,3}, \mathcal{M}2_{1,2}, \mathcal{M}3_{2,2}, \ldots\}$. It has to be noted that $CS_{1,0}$ and $CS_{2,0}$ cannot have any allocated mission because they are not considered as CSs but as virtual CSs.

In case of existing system, the representation of legacy systems are encapsulated into CS agents and integrated in the SoS model, like any other CS.

A functional specification adapted to our formalism can be defined as follows:

$fs =< \mathcal{G}, \mathcal{M}, \mathcal{P}, mo >$, where

- $\mathcal{G}$: the set of SoS goals.
- $\mathcal{M}$: the set of labels for all missions. The missions can be allocated ($\mathcal{M}x_{n,l}$) or not ($\mathcal{M}x$).
- $\mathcal{P}$: the set of global plans resulting from the goal decomposition tree.

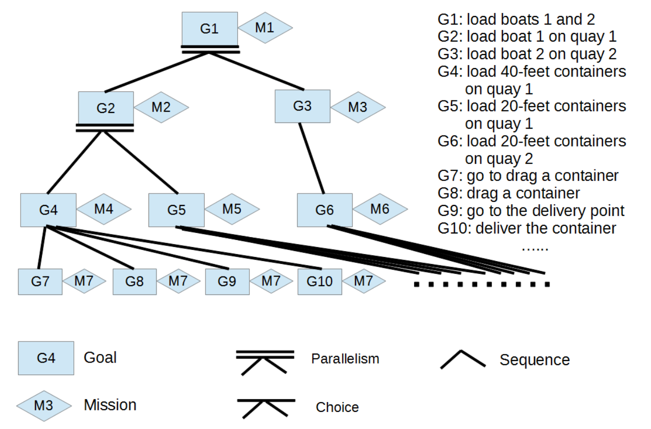

Fig. 2. The functional decomposition of SoS global goal.

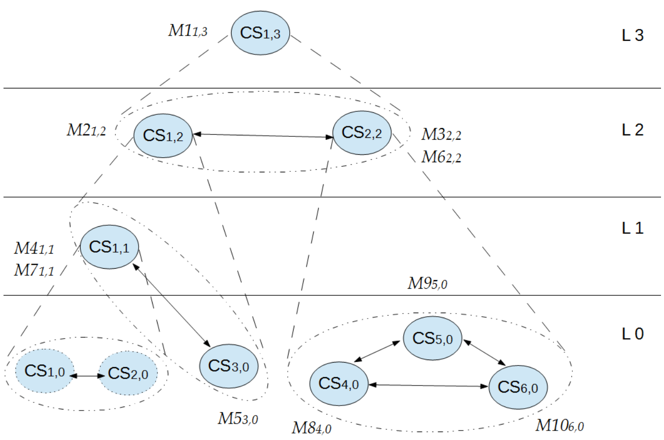

Fig. 3. Partial allocation of missions over the CSs SoS.

- $mo : \mathcal{M} \rightarrow 2^{\mathcal{G}}$: a function specifying for each mission the set of attached goals.

2) Capacity Evolution: As said previously CS agent capacities evolve through time. Sometimes CSs can lose capacities which compromise their missions and possibly the missions of their super CS. When the capacities of a CS are modified, its super CS capacities are calculated thanks to CS and group specification. The capacity evolution is spread through levels due to influences. That means each CS agents with under CS(s) can calculate its own capacity from its under CS capacities and group specification. So when a CS loses some of its capacities, it first tries to reorganize itself to remain capable of fulfilling its allocated mission and if it is impossible, it informs its super CS which acts in a similar way until the top CS agent is informed.

If the top CS agent is not able to reorganize itself anymore, by adopting a new organizational structure, it is able to inform the modeler that the SoS has failed and there is no available means to reach the global goal.

This mechanism can be illustrated by the following example. The IAV $CS_{1,0}$ has a failure and its capacity to carry a 40-feet container is compromised. So the capacity of the IAV platoon $CS_{1,1}$ is also modified and it cannot reorganize itself only with its under CSs $CS_{1,0}$ and $CS_{2,0}$.

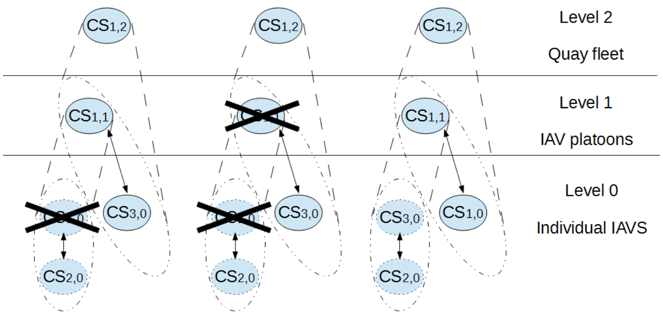

Fig. 4. Reorganization of a SoS guided by the CS capacity evolution.

It cannot accomplish its mission to deliver a series of 40-feet containers to the vessel anymore. So the failure is spread to $CS_{1,2}$, the IAV fleet in quay 1. $CS_{1,2}$ can reorganize itself by switching IAVs $CS_{3,0}$ and $CS_{1,0}$. This is done, maintaining the structure of $CS_{1,2}$.

C. Respecting Fundamental Characteristics

In this section we show that directed SoS characteristics are respected in agent-based models generated using our formalism. It is important to consider that in the first three characteristics the CS agents $CS_{i',j'}$ and $CS_{i'',j''}$ of a system $CS_{i,j}$ are not related by a direct hierarchical relation:

$$\forall CS_{i',j'}, CS_{i'',j''} \in CS_{i,j},$$
$$CS_{i',j'} \notin CS_{i'',j''} \wedge CS_{i'',j''} \notin CS_{i',j'}$$

Here are the five characteristics that any SoS must respect (see section I) and their translation in a MAS model representing a SoS:

1) **Operational independence**: CS agents $CS_{i',j'}$ and $CS_{i'',j''}$ of a system $CS_{i,j}$ are independent in an operational way iff they are instantiated by CS agents, $\mathcal{A}$, and possess their own private variables, s_a, representing their resources.

$$\forall CS_{i',j'}, CS_{i'',j''} \in CS_{i,j},$$
$$CS_{i',j'}, CS_{i'',j''} \in \mathcal{A} \wedge$$
$$s_{CS_{i',j'}} \neq \emptyset \wedge s_{CS_{i'',j''}} \neq \emptyset \wedge$$
$$s_{CS_{i',j'}} \cap s_{CS_{i'',j''}} = \emptyset$$

Fig. 5 shows that $CS_{4,0}$ is operationally independent, but $CS_{1,0}$ is not, because it shares a part or its resources with $CS_{2,0}$.

2) **Managerial independence**: CS agents $CS_{i',j'}$ and $CS_{i'',j''}$ of a system $CS_{i,j}$ are independent in a managerial way iff they do not share any part of their own missions, $\mathcal{M}_{i',j'}$ and $\mathcal{M}_{i'',j''}$.

$$\forall CS_{i',j'}, CS_{i'',j''} \in CS_{i,j},$$
$$\mathcal{M}_{i',j'} \cap \mathcal{M}_{i'',j''} = \emptyset$$

$CS_{4,0}$ of fig.5 is independent in an managerial way, but $CS_{1,0}$ is not, because it shares its missions with $CS_{2,0}$.

3) **Geographic distribution**: CS agents $CS_{i',j'}$ and $CS_{i'',j''}$ of a system $CS_{i,j}$ are geographically distributed iff their physical state, ϕ_a, are totally distinct and not directly dependent. This last property can be formulated by the fact that it is not possible to compute directly all or any part of the physical state of a CS agent (using a function f) from all or any part of the physical state of another CS agent.

$$\forall CS_{i',j'}, CS_{i'',j''} \in CS_{i,j},$$
$$\phi_{CS_{i',j'}} \cap \phi_{CS_{i'',j''}} = \emptyset \wedge$$
$$(\forall \phi'_{CS_{i',j'}} \subset \phi_{CS_{i',j'}}, \forall \phi'_{CS_{i'',j''}} \subset \phi_{CS_{i'',j''}},$$
$$\nexists f : \Phi \rightarrow \Phi, f(\phi'_{CS_{i',j'}}) = \phi'_{CS_{i'',j''}}$$
$$\vee f(\phi'_{CS_{i'',j''}}) = \phi'_{CS_{i',j'}})$$

Fig. 5 shows that $CS_{4,0}$ is geographically dispersed, but $CS_{1,0}$ and $CS_{2,0}$ are not. Because, in the case of the platoon $CS_{1,1}$ composed of the leader IAV $CS_{1,0}$ and the follower IAV $CS_{2,0}$, it exists a function f and $f(pos_{CS_{1,0}}) = pos_{CS_{2,0}}$, where $pos_{CS_{1,0}}$ and $pos_{CS_{2,0}}$ designate the part of the physical state of these two CS agents that represent their position.

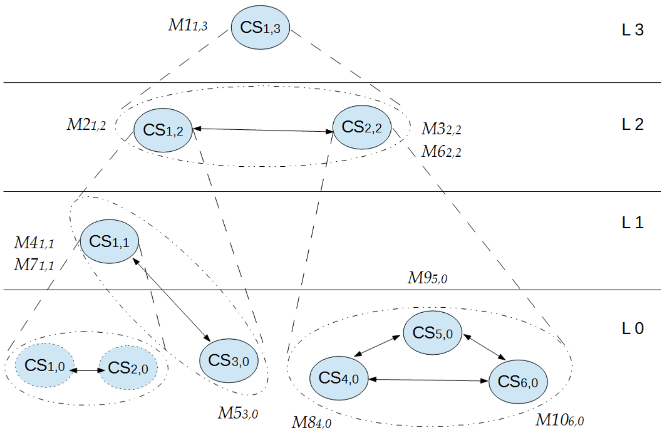

Fig. 5. Characteristics of a SoS.

4) **Emergent Behaviour**: A set of CS agents $CS_{i',j'}$ of a system $CS_{i,j}$ cooperates iff CS agents cooperate to realize a global mission $\mathcal{M}_{i,j}$ of the system formed of a set of missions $\mathcal{M}_{i',j'}$, that each CS agent of $CS_{i,j}$ cannot realize alone.

$$\forall CS_{i',j'} \in CS_{i,j},$$
$$\bigcup \mathcal{M}_{i',j'} = \mathcal{M}_{i,j}$$

CS agent $CS_{2,2}$ of fig. 5 can fulfil its missions $\mathcal{M}3_{2,2}$ and $\mathcal{M}6_{2,2}$ only through the cooperation of its under CSs $CS_{4,0}$, $CS_{5,0}$ and $CS_{6,0}$ which fulfil missions $\mathcal{M}8_{4,0}$, $\mathcal{M}9_{5,0}$ and $\mathcal{M}10_{6,0}$.

5) **Evolutionary development**: This characteristic is different from the others because it can be noticed only during the SoS execution. So, it is an emerging characteristic. In particular, this characteristic is expressed in three situations: a) a CS agent is added to the SoS, b) a CS agent of the SoS is removed or c) a CS agent notices that one of its missions is impossible to fulfil according to its capacity function.

A system $CS_{i,j}$, formed by several CS agents $CS_{i',j'}$ follows an evolutionary development iff when one of its CS agents allocated mission $\mathcal{M}x_{i',j'}$ is detected as impossible to fulfil according to its capacity function, $\lambda_{i',j'}$, and the actual state of the level j', $\delta_{j'}(t)$, this mission is no more allocated to any CS agent, but is set free in the set of missions $\mathcal{M}$ in the functional specification of the SoS.

$$\forall CS_{i',j'} \in CS_{i,j},$$
$$\lambda_{i',j'}(\delta_{j'}(t), \delta_{j'}(\mathcal{M}x_{i',j'})) = 0$$
$$\rightarrow \mathcal{M}x_{i',j'} \notin \mathcal{M} \cup \mathcal{M}x \in \mathcal{M}$$

Once the mission is set free an algorithm has to allocate it to another CS agent or reorganize the SoS to fulfil it, in a feasible way. Khalil et al. presents such SoS reorganisations without evoking a formalized algorithm [14].

Fig. 6 shows three steps in the life of a SoS. The second step is obtained by the suppression of $CS_{3,0}$. Step three represents the same SoS after the addition of $CS_{3,0}$ and $CS_{4,0}$ to the SoS. $CS_{3,0}$ from step one and three is not necessarily the same CS agent.

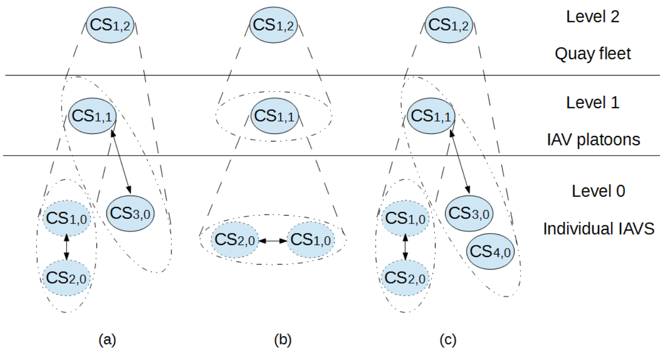

Fig. 6. Three steps in the life of a SoS: a) initial state, b) deletion of $CS_{3,0}$ and c) addition of $CS_{3,0}$ ans $CS_{4,0}$.

V. Implementation

To validate our work we implemented a co-simulation of SoS coping with a reconfiguration problem. We modelled a SoS whose elementary CSs are IAVs in InTraDE project [9]. The resulting simulation has been implemented on MaDKit multi-agent platform endowed with AGR model [33]. SCANeRstudio [34] is a simulation software which allows to pilot IAVs in realistic conditions. The resulting simulation is used to control direct system by interfacing a SoS model of IAV fleet in MaDKit and a realistic representation of IAVs in a container port terminal in SCANeRstudio as it is depicted in Fig. 7.

We implemented three main modules on MadKit. The first one integrates a simulation engine adapted to the multi-level concepts of IRM4MLS. The second one integrates the elements of the formalism we created to represent SoSs. And the last one communicates orders and receive data from SCANeRstudio via UDP communication. We also implemented an API on SCANeRstudio which communicates realistic data about the simulated vehicles and receives order to pilot them. These order are sent to the API TRAFFIC which applies these orders on the vehicles.

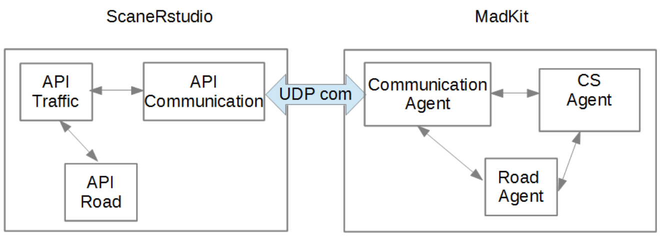

Fig. 7. Relations between SCANeRstudio and Madkit via UDP communications. SCANeRstudio is a network of communicating APIs and Madkit is a multi-agent platform.

VI. Conclusion and Future Perspectives

In this article we presented an operational SoS definition based on the fundamental characteristics for CSs: operational independence, managerial independence, geographic distribution, evolutionary development and emergent behaviour. We proposed a generic multi-level multi-agent formalism to represent a SoS managed by a central authority. The proposed model can be applied to any SoS controlling it in a changing environment affecting CS capacities. Only repeated execution of Multi-Agent Based Simulations seems suited to test and validate algorithms to control, diagnose systems with numerous heterogeneous interacting sub-systems endowed of a changing hierarchical structure. Moreover, the system is divided into groups and levels. We also provide elements to prove the validity of our approach.

Fig. 8. A platoon of IAvs in SCANeRstudio (simulation platform).

A development of the applied part of this work is to couple the multi-level multi-agent based model in MaDKit with other simulation tools than SCANeRstudio. Flexsim and its Container Terminal library (Flexsim CT) is used to model traffic flows in container terminal environments [35]. It could figure the same SoS but represented at a higher level and provide semi-realistic inputs for that level. In the future, our work could be applied to other forms of SoSs but not only directed ones with addition of authority and organizational aspects. Another track is to improve the building group mechanism. It can be done by specifying strategies to optimize the group specification choice and/or the choice of CS agents to constitute these groups according to the robustness of the resulting SoSs or the quality of the accomplished mission.

Acknowledgment

The authors would like to thank all InTraDE project members, the Uiversity of Lille 1 and the HEI ingeenering school which supported and permitted this work during the phd training of the first author.

REFERENCES

[1] P. Aniorté, E. Cariou, and E. Gouardères, "Modélisation de systèmes complexes distribués : l'ingénierie des modèles pour l'intégration des paradigmes "agent" et "composant"," in *Proceedings of the* 2^{nd} *Journée Multi-Agent et Composant (JMAC)*, Nîmes, 21 Mars 2006.

[2] M. Huhn, J. Muller, J. Gormer, G. Homoceanu, N.-T. Le, L. Martin, C. Mumme, C. Schulz, N. Pinkwart, and C. Muller-Schloer, "Autonomous Agents in Organized Localities Regulated by Institutions," in *Digital Ecosystems and Technologies Conference (DEST), 2011 Proceedings of the 5th IEEE International Conference on*, 31 2011-june 3 2011, pp. 54 –61.

[3] J. Dahmann, "Systems Engineering for Department of Defense," *System of Systems, Wiley Series in Systems Engineering and Management John Wiley & Sons Inc*, 2008.

[4] US-Army, *Army Acquisition Policy, Research, Development, and Acquisition*, Army Regulation 70-1, U.S. Department of Defense, Department of the Army, Washington, DC, Decembre 31 2003.

[5] S. Luskasik, "Systems, systems of systems, and the education of engineers," *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, vol. 12(1), pp. 55–60, 1998.

[6] D. DeLaurentis, "Understanding Transportation as a System-of-Systems Design Problem," in *43rd AIAA Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, 10-13 January 2005, aIAA-2005-0123.

[7] D. Walker, "Realizing a corporate SOSE Environment," in *Keynote presentation, 2007 IEEE SoSE Conference*, San Antonio, TX, USA, 2007.

[8] J. Tien, "A System of Systems View of Services," *System of Systems Engineering, Wiley Series in Systems Engineering and Management John Wiley & Sons Inc*, 2008.

[9] [Online]. Available: http://www.intrade-nwe.eu/fr

[10] J. M. Parker, "Applying a System of Systems Approach for Improved Transportation," *S.A.P.I.E.N.S.*, vol. 3, no. 2, 2010.

[11] E. Sloane, T. Way, V. Gehlot, R. Beck, J. Solderitch, and E. Dziembowski, "A Hybrid Approach to Modeling SOA Systems of Systems Using CPN and MESA/Extend," in *Systems Conference, 2007 1st Annual IEEE*, april 2007, pp. 1 –7.

[12] B. Zhou, A. Dvoryanchikova, A. Lobov, J. Minor, and J. Martinez Lastra, "Application of the Generic Modelling Method for System of Systems to Manufacturing Domain," in *37th Annual Conference on IEEE Industrial Electronics Society*, nov. 2011, pp. 352 –358.

[13] J. M. Held, "Systems of Systems: Principles, Performances, and Modelling," Ph.D. dissertation, The University of Sydney, 2008.

[14] W. Khalil, R. Merzouki, B. Ould-Bouamama, and H. Haffaf, "Hypergraph Models for System of Systems Supervision Design," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 42, no. 4, pp. 1005 –1012, july 2012.

[15] J. Ferber, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley Longman Publishing Co., Inc, 1999.

[16] G. Morvan, A. Veremme, and D. Dupont, "IRM4MLS: the influence reaction model for multi-level simulation," in *Multi-Agent-Based Simulation XI*, ser. LNAI. Springer, 2011, vol. 6532, pp. 16–27.

[17] A. P. Sage and C. D. Cuppan, "On the Systems Engineering and Management of Systems of Systems and Federations of Systems," *Inf. Knowl. Syst. Manag.*, vol. 2, no. 4, pp. 325–345, Dec. 2001.

[18] B. Sauser, J. Boardman, and D. Verma, "Systomics: Toward a biology of system of systems," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 40, no. 4, pp. 803 –814, july 2010.

[19] M. Maier, "Architecting Principles for Systems-of-Systems," in *proceeding of the Sixth Annual Internationnal Symposium INCOSE*, INCOSE, Boston, 1996, pp. 567–574.

[20] J. Boardman and B. Sauser, "System of Systems the meaning of *of*," in *Proceeding of the 2006 IEEE/SMC International Conference on SoS Engineering*, Los Angeles, CA, USA, April 2006.

[21] M. Jamshidi, *System of Systems Engineering Innovations for the 21st Century*. Hoboken: Wiley, 2008, ch. Introduction to System of Systems, pp. 1–20.

[22] J. P. Dauby and S. Upholzer, "Exploring Behavioral Dynamics in Systems of Systems," *Procedia Computer Science*, vol. 6, no. 0, pp. 34 – 39, 2011, complex adaptive sysytems.

[23] J. Ferber, *Les Systèmes Multi-Agents: Vers une Intelligence Collective*. InterEditions, 1995.

[24] N. Gaud, S. Galland, F. Gechter, V. Hilaire, and A. Koukam, "Holonic Multilevel Simulation of Complex Systems : Application to Real-Time Pedestrians Simulation in Virtual Urban Environment," *Simulation Modelling Practice and Theory*, vol. 16, pp. 1659–1676, 2008.

[25] J. F. Hübner, J. Simão, and O. Boissier, "Moise+: Towards structural, functional and deontic model for MAS organization," in *Proceeding of the first International joint conference on Autonomous Agents and MultiAgent Systems (AAMAS'02)*, Bologna, Italy, July 15-19 2002, p. 502.

[26] P. Ribino, C. Lodato, S. Lopes, V. Seidita, V. Hilaire, and M. Cossentino, "A Norm-Governed Holonic Multi-Agent System Metamodel," in *Proc. 13th International Workshop on Agent-Oriented Software Engineering*, Valencia, Spane, June, 4 2012.

[27] D. Scerri, S. Hickmott, A. Drogoul, and L. Padgham, "An Architecture for Distributed Simulation with Agent-Based Models," in *Proc. of 9th Int. Conf on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck, and Sen, Eds., Toronto, Canada, May, 10-14, 2010, pp. 541–548.

[28] S. Picault and P. Mathieu, "An interaction-Oriented Model for Multi-Scale Simulation," in *the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, 2011.

[29] G. Morvan, "Multi-level agent-based modeling - Bibliography," *CoRR*, vol. abs/1205.0561, 2012.

[30] J.-B. Soyez, G. Morvan, D. Dupont, and R. Merzouki, "A Methodology to Engineer and Validate Dynamic Multi-level Multi-agent Based Simulations," in *MABS'12 - 13th International Workshop on Multi-Agent Based Simulation*, Valencia, Spain, 4th-5th June 2012.

[31] F. Michel, "The IRM4S Model: The Influence/Reaction Principle for Multi-Agent Based Simulation," in *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM, 2007, pp. 1–3.

[32] J. Ferber, O. Gutknecht, and F. Michel, "From Agents to Organizations: an Organizational View of Multi-Agent Systems," in *Agent-Oriented Software Engineering (AOSE)*, vol. 4, 2004, pp. 214–230.

[33] [Online]. Available: http://www.madkit.org

[34] [Online]. Available: http://www.scanersimulation.com/

[35] [Online]. Available: http://www.flexsim.com/