An approach for the verification of Systems of Systems’ (SoS) behaviors and properties

Mustapha Bilal, Nicolas Daclin, Vincent Chapurlat

ENS Mines Alès, France

{firstname.lastname}@mines-ales.fr

A System of Systems (SoS) is a complex system (Chapman and Bahill, 1995) based on the collaboration and interaction between existing subsystems (of technical or sociotechnical nature) in order to fulfill a common mission for a limited duration of time (*e.g.* coalition forces, air transportation system, networked enterprises, etc.). SoS design or re-design is distinguished from classical system design (Blanchard *et al*., 2010). Indeed, systems, where most of them already exist, are selected according to their relevance, capacity and self-interest to fulfill the SoS mission. They are assembled in a way to respect the requirements of SoS’ stakeholders and that their interaction allows them to fulfill this mission. During this assembly, interfaces are required whether physical (hardware), informational (model and data exchange protocols) or organizational (rules, procedures and protocols) in order to ensure the necessary interoperability of subsystems (Mallek *et al*., 2012). Moreover, their behavior, decision-making autonomy and their own organization should not be impacted or influenced more than necessary by risky situations or undesired effects resulting from the interaction between these subsystems. Moreover, the resulting behavior of SoS is not necessarily the expected one due to the unpredictable emergent behaviors caused by these interactions (Maier 1998). Similarly, some of SoS properties cannot be directly deduced and linked to the set of subsystems properties. SoS Designers are facing a major challenge: How to have a better control over these behaviors and these properties in a relatively short time without making an extra effort and without the need of additional knowledge in terms of modeling both behavioral scenarios and properties?

The presented research focuses on the complementary role that the different verification techniques can play. The goal is to ensure and subsequently improve “real-time design”, e.g. in early stages of architectural and interfaces design (Dhillon, 1987), SoS’ architecture and behavior verification, no matter what may be the size or the complexity of its subsystems.

On the one hand, it’s about a technical formalization starting from stakeholders’ requirements and proof of properties on a model describing the SoS architecture. On the other hand, it is about the use of an advanced simulation technique of the architecture’s behavior in order to achieve two goals:

1. Establishing an evaluation approach of several non-functional characteristics (De Weck et al. 2012) as proposed in the case of SoS engineering of (Blanchard *et al*., 2010) or systems engineering (SeBok 2012) (ISO 2008) to meet the Model Based System Engineering (MBSE) hypothesis and principles (Estefan 2007). We keep here the characteristics of robustness of SoS architecture. The robustness is defined here as the ability aggregation to ensure its stability (it is always able to fulfill its mission despite the different emergence phenomena and external events that threaten its behavior), its integrity (it is always able to fulfill its mission despite the various emergence phenomena and internal events causing an important dysfunction of these subsystems), its control (it is always able to maximize its performance).
2. To facilitate the detection of possible errors, omissions or uncertainties and to judge the behaviors’ plausibility and credibility which are considered as unexpected but they occur during the simulation.

This work should lead to the proposal of a modeling framework for the SoS architecture based on a mathematical formalization underlying the concept of SoS. This will allow the use of formalization technique and proof of properties, as well as the formalization of essential non-functional characteristics expected in a SoS. This framework will then be equipped with operational semantic and formal rules of transformation to a Multi-Agent System (MAS) (Wooldridge 2009) which allows simulating the subsystems’ behavior that form the architecture, and characterize the plausibility and credibility behavior, met during the simulation, without the need to pre-specify evolution SoS scenarios. The use of multi-agent systems (MAS) is a natural and effective solution to deal with complex situations in distributed environments (Khosla et Dillon 1997). They allow modeling and simulating the parallel evolution and the interactions of various complex entities (subsystems) independently (Brandolese et al. 2000). Moreover, the MAS can answer to the individual failure of one of the elements without degrading the system in its totality and thus, it is able to deal with this type of behavior. Finally, the BDI technology (Rao 1995) (Beliefs, Desire, Intention), used in some MAS, allows to model, and more accurately, the knowledge and rules of behavior to be exhibited by the agents which model each subsystem. Environments for modeling and simulation for design complex systems have been proposed (ID4CS 2009) except that they were reserved to particular areas and domains (e.g. aerospace) and they do not allow to address to any other type of SoS.

Structuring the Multi-Agent models in order to model the SoS architecture is defined according four dimensions (Figure 1): *Agent models, Environment models, Interaction models,* and *Organization models* (Demazeau 1995):

1. ***Agent models*** represent the active entities (subsystems). However, the subsystems can be of various natures which raises the importance of defining various kinds of Agents, such as: Intentional Agents (Rao 1995), Rational Agents (Russel 1991) and Situated Agents (Agre 1987) (Maes 1990).
2. ***Environment models*** *The* SoS (as well as the subsystems) exist in an environment which they interact with. This environment representsthe common space between the previous defined Agents. It is here question of an active entity that performs the interactions mediation between Agents and between Agents and resources.
3. ***Interaction models***as stated previously, the subsystems interact together in order to fulfill a mission of SoS during a limited period of time. Interaction models manage this interaction by defining and structuring the dynamic linking of two or more Agents through a set of reciprocal actions during this period of time. These models define also the communication languages (KQML, ACL, etc.) and the interaction protocols that control this communication.
4. ***Organization models*** they define the constraints on the interaction models. They define how the agents should cooperate in order to attend a goal. These constraints are defined by a designer or by the Agents themselves for a specific purpose. Among these constraints, we mention: the Role Based Access Control model (RBAC) for the Agents, norms, obligations, permissions, laws and many others. This organization model is based on sociological sources, physiological, social and Computer-Supported Cooperative Work (CSCW).



Fig 1: Structure of MA model specific to SoS.

REFERENCES

(Agre 1987) Agre, P. E. and D. Chapman (1987). " Pengi: An Implementation of a Theory of Activity" . AAAI-87. The Sixth National Conference on Artificial Intelligence, Menlo Park, CA., Morgan Kaufman, Los Altos, CA

(Blanchard *et al*., 2010) Systems Engineering and Analysis by Benjamin S. Blanchard, Wolter J. Fabrycky 2010

(Brandolese, 2000) Brandolese A., Brun A., Portioli-Staudacher A., "A Multi-Agent approach for the capacity allocation problem" International Journal of Production Economics, Vol. 66, pp. 269-285, 2000.

(Chapman et Bahill, 1995) Chapman, W.L., Bahill, A.T.: Complexity of the system design problem.

(Demazeau 1995) Y.Demazeau: From interactions to collective behaviour in agent-based systems.In Proc. of the 1st European Conf. on Cognitive Science, Saint Malo, France, April, 1995, p. 117-132

(Dhillon,1987) Dhillon, B, S.: Reliability in Computer System Design. Alex Publishing Corporation (1987)(Estefan 2007)Estefan, J.A., « Survey of Model-Based Systems Engineering (MBSE) Methodology ». INCOSE MBSE Focus Group Report, 2007(ID4CS 2009) ANR project 2009, see http://www.irit.fr/ID4CS(Khosla et Dillon 1997) R. Khosla, T. Dillon, “Intelligent hybrid multi-agent architecture for engineering complex systems,” Proceedings of the 1997 IEEE international Conference on Neural Networks, vol. 4,pp. 2449-2454

(Maes 1990) Maes, P. (1990). "Situated Agents Can have Goals." Designing Autonomous Agents . Maes, P. (Ed.). Cambridge, MA., MIT Press: 49-70

(Maier 1998) Maier, M.W.: Architecting principles for systems-of-systems. Systems Engineering 1(4) (1998) 267-284

(Mallek *et al*., 2012) The application of interoperability requirement specification and verification to collaborative processes in industry

(Rao 1995) A. S. Rao and M. P. Georgeff, BDI-agents: from theory to practice, Proceedings of the First

Intl. Conference on Multiagent Systems, 1995

(Russel 91) Stuart Russell and Eric Wefald. Do The Right Thing. The MIT Press, Cambridge, Massachusetts, 1991

(Sebok 2012) <http://www.sebokwiki.org> (dernier accès le 23/04/2013)

(De Weck *et* *al*. 2012) Olivier L. de Weck, Adam M.Ross, Donna H. Rhodes, Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (-ilities), third International Engineering Systems Symposium CESUN 2012, Delft University of Technology, 18-20 June 2012

 (Wooldridge and al., 1995) Wooldridge M., Jennings N., Intelligent Agents: Theory and Practice, Knowledge Engineering Review, 1995.

(Wooldridge 2009) An introduction to Multi-agent Systems (2009)