Multi-agent Simulation based Control of Complex Systems

(Extended Abstract)

Tomas Navarrete Gutiérrez
CRPHT, CRTE
6A avenue des Hauts-Fourneaux
L-4362 Esch-sur-Alzette, Luxembourg
tomas.navarrete@tudor.lu

Laurent Ciarletta
Université de Lorraine
LORIA,
Vandoeuvre-lès-Nancy
F-54506, France
laurent.ciarletta@loria.fr

Vincent Chevrier
Université de Lorraine
LORIA,
Vandoeuvre-lès-Nancy
F-54506, France
vincent.chevrier@loria.fr

ABSTRACT

The challenge we address in this work is the control of complex systems based on the multi-agent paradigm.

Our focus is on identifying what specific questions arise when using the multi-agent paradigm in the control of complex systems context. We present here a solution under the form of an equation-free control architecture based on multi-agent model simulation.

The results of the implementation on a free-riding problem in p2p networks demonstrate that the proposed architecture can control such a network. Our contribution is to identify key questions that rise when using the multi-agent paradigm in the context of control of complex systems, concerning the relationship between the model entities and the target system entities.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence—*multiagent systems*

General Terms

Complex systems management

Keywords

1. INTRODUCTION

The objective of our work is to make a concrete link between multi-agent model simulation and external control of complex systems. We consider that the control of complex systems is defined by overcoming a series of difficulties given by the following characteristics of complex systems: local interactions produce the global outcomes of the system, complex systems are made of decentralized systems made of autonomous entities, they are not easily (or at least usefully) modeled by analytical models, preexisting complex systems may not be legally, or technically stopped or tampered with in order to control them.

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

One major difficulty is modeling the evolution of the system behavior. Overcoming this difficulty means to characterize the evolution of the behavior of the target system, taking into consideration: the different levels of a complex system (local and global for example) and the emergence of global outcome from local interactions.

In this article, we consider the special case of complex techno-social systems [5] made of humans and artificial entities (cars, routers, servers, telephones, electricity lines). Specifically, we focus on techno-social systems out of control, that cannot be "stopped" to modify their behavior and guide them to a particular state. Controlling a system means applying (control) actions to modify the course of its behavior. The choice of the action to apply is made by using a predictive model of the system. In the case of complex systems, these actions are at local level but their effects are global. Controlling a complex system implies to have a predictive model that includes both aspects in order to assess the impact of a local action at global level. Additionally, we consider the case where endogenous control mechanisms (if any) of a given system are insufficient.

Our work is guided by the question: **How can we use multi-agent model simulation to control a techno-social system from the outside?**

Approaches that include the multi-agent paradigm to control a complex system (explicitly) include organic computing [7], self-organization [6] and emergent engineering [1]. Despite considering the specifics of complex systems, they all propose example applications that tackle the problem of control from a "design" or "engineering" point of view. As such, they are hardly applicable without major modifications to preexisting systems.

Our proposal is to envisage a control mechanism from an exogenous perspective, in the form of a control architecture.

2. THE ARCHITECTURE

Our architecture (called C) is external to the system we wish to control (the target system called T). The objective of C is to keep T in a given state. The architecture C works in a feedback loop with the following flow of execution (summarized in figure 1.).

General overview

For space reasons, we only give here a broad description of the different elements of the architecture. The interested reader may consult [2] for its full generic definition.

Feedback loop. System C will influence system T in order to make the output y of T be as close as possible to a reference value. The output of system T is used as input for system C which in turn will produce an output that will become the input of system T.

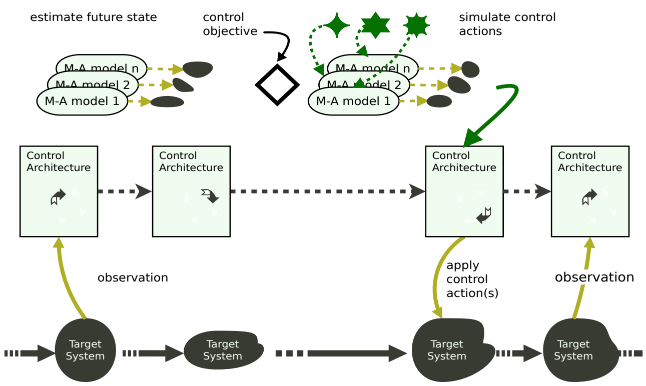

Figure 1: Item flow of the blocks in the architecture.

Exogenous implementation. The exogenous principle of the architecture is twofold. First, it is exogenous because system C is meant to be an independent system. That is, if it stops working, it shall not prevent system T from working. Second, it is exogenous because it is built under the hypothesis that the preexisting system T cannot be stopped to add the architecture as a control mechanism.

Equation-free modeling. Equation-free refers to a paradigm for multiscale computation and computer-aided analysis [4]. The main idea behind the equation-free approach is the use of microscopic models to do macroscopic analysis of a model. In our case, we use microscopic models to determine the state of the target system as well as the effects of control actions. These models are simulated models given at a local description level namely multi-agent models.

Structure of the architecture. The architecture is composed of blocks charged of the different functions of the architecture. These blocks are:

1. Observe Target System: provide the architecture with information observed from the target system.
2. Estimate Future State: execute the multi-agent simulation of models used to estimate the target system.
3. Simulate Control Actions: execute the multi-agent model simulation of possible control actions effects.
4. Apply Control Actions: effectively apply control actions.

Experimental validation

We have implemented our architecture to control a free-riding phenomenon in a simulated peer-to-peer network. Results are reported in: [3, 2]. We have conducted two series of experiments.

In the *first* series of experiments, we demonstrated the possibility of controlling the system with our architecture through the illustration of the implementation of our architecture in a concrete case.

In the *second* series, we focus on issues related to multi-agent simulation. The objective in these experiments is to identify what is the influence of the specific issues of multi-agent model simulation in the performance of the architecture. We concentrated on the implications of three different issues: i) having multiple models producing multiple future state predictions ii) changing the time horizons of the simulations and iii) the amount of information gathered from the target system.

3. CONCLUSIONS

We outlined the principles of a control architecture where multi-agent models are used to predict the behavior of the system and simulate the consequence of control actions. This architecture was applied on a p2p problem in which it succeed to attain the control objective.

Introducing multi-agent simulation implies that issues concerning how to establish the relationship between the model and the target system come up, namely: the validity and calibration of models, and the translation of entities from the target system to the elements of the model. Our modular design of the architecture is convenient to investigate such issues and to assess the decision made.

As perspectives, the generic and modular design of the architecture would allow to further investigate the example application along different dimensions to better assess the architecture: open target system, heterogeneous peer behaviors, noise on the observations.

The different operating regimes of the target system are another aspect. They lead to an implementation where different models would be used for different regimes.

Also we consider worthy to study different techniques (like learning techniques, or genetic algorithms) to make evolve the models used in the architecture.

4. REFERENCES

[1] R. Doursat and M. Ulieru. Emergent engineering for the management of complex situations. In *Proceedings of the 2nd International Conference on Autonomic Computing and Communication Systems*, 2008.

[2] T. Navarrete Gutierrez. *Une architecture de contrôle de systèmes complexes basée sur la simulation multi-agent.* Ph. D Thesis (in english), Université de Lorraine, 2012.

[3] T. Navarrete Gutierrez, L. Ciarletta, and V. Chevrier. Multi-agent simulation based governance of complex systems : architecture and example implementation on free-riding. In *Mexican International Conference on Computer Science*, Morelia, Mexico, 2013. IEEE Computer Society Press.

[4] Y. K. G. Samaey. Equation-free modeling. *Scholarpedia*, 5(9):4847, 2010.

[5] A. Vespignani. Predicting the behavior of techno-social systems. *Science*, 325(5939):425, 2009.

[6] T. D. Wolf, T. Holvoet, and G. Samaey. Engineering Self-Organising Emergent Systems with Simulation-based Scientific Analysis. In *Proceedings of the Fourth International Workshop on Engineering Self-Organising Applications*, Utrecht, 2005.

[7] R. P. Würtz. *Organic Computing*, volume 21 of *Understanding Complex Systems*. Springer Berlin / Heidelberg, 2008.