Towards the Analysis & Prediction of Complex System Behaviour in SAPERE

Marco Alberti & Andrea Omicini
marco.alberti3@studio.unibo.it, andrea.omicini@unibo.it

Alma Mater Studiorum—Università di Bologna

ASENSIS 2012
1st International Workshop @ SASO 2012,
Lyon, France, 10th of September 2012

Outline

Complexity & Unpredictability

Complex computational systems are typically unpredictable

- for both practical & theoretical reasons
- such as their situatedness within unpredictable environment, or the non-compositionality of their inner components

Predictability is anyway an issue

- again, for both theoretical & practical reasons
- our ability to model complexity defines more or less the level of our scientific achievements
- our ability to use models to predict behaviours is an essential premise to the engineering of artificial systems

SAPERE Complex Systems

SAPERE systems are *complex*

- nature-inspired, adaptive, pervasive, self-organising, multi-agent systems
- SAPERE models are articulated
- SAPERE methodology, too
- analysis & prediction tools are required, to be integrated in the SAPERE SE process

SAPERE systems are *coordinated* systems

- . . . so, we could have taken that route. . .
- . . . instead, we choose to forget about it for a while – actually, it the subject of another research stream – and to focus on the many approaches available in the literature

Outline

Survey

Analysis and prediction of complex computational systems

- we survey the most relevant approaches
- we focus on those "reducible" to the SAPERE systems, in some ways

Analysis and prediction of SAPERE computational systems

- we identify the two most promising techniques: *model checking* and *data mining*
- we derive two potentially-effective approaches to the analysis and prediction of the behaviour of SAPERE pervasive ecosystems

Outline

MAS & Pervasive Systems

Many approaches, limited results

- The most effective approaches deal with very narrow portions of the system behaviour, perform quite limited sorts of analysis, and are typically applicable to quite specific domains

A possible classification for MAS

a priori approaches dealing with compliance to the rules steering system evolution

a posteriori approaches aimed at predicting system behaviour focusing on specific system features

Analysis by Model Checking I

Focus

- (semi-)exhaustive analysis of the states the system will go through, typically along with by a representation by a LTS (Labelled Transition System)
- verification of system meeting given properties, expressed though various logics (LTL, CTL, DLTL, GLTL, etc.)

Uses

- mostly used for the verification of safety or liveness properties
- quite effective for "paper" problems, its application is however complex for generic analysis, and often prohibitive from the computational point of view

Analysis by Model Checking II

Logics

- LTL is usually preferred to CTL
- From LTL numerous extensions have arisen, including

DLTL *(Dynamic Linear Temporal Logic)* focussed on enhancing the *until* operator to better express *pre-* and *post-conditions* [Giordano et al., 2004]

GLTL *(General Linear Temporal Logic)* allows the definition of LTL *properties* for the *whole system*, as well as for the individual entities involved [Pokorny and Ramakrishnan, 2006]

Analysis by Model Checking III

BDI agents

- BDI specifications can be translated into LTL specifications
- `AgentSpeak(L)` dominates among BDI languages, and `AgentSpeak(F)` is a subset of `AgentSpeak(L)`, for finite state systems only
[Bordini et al., 2003, Bordini et al., 2004, Bordini et al., 2006]
- a system described by `AgentSpeak(F)` can be represented by a SLPN (Simple Logic Petri Net), and model-checked [Behrens and Dix, 2007]

Analysis of Swarm Systems

Homogeneous systems

- a detailed model checking of individual entities is unthinkable because of the explosion in the number of states
- under the hypothesis of a system composed of homogeneous entities [Konur et al., 2010, Dixon et al., 2011], it is possible to define a state diagram common to all entities
- a single LTS can be used to represent all the entities, by assigning a number to each admissible state for the individual entities, which represents at any instant of time the number of the system entities currently in that particular state
- each state is then associated to the number of entities that actually are in that state at any given instant of time, using probabilistic analysis
- such a number is computed by using probabilistic analysis, so it represents an *estimation* of the real number of entities that will be in the state at runtime
- however, noise tolerance of swarm systems ensures the reliability of the analysis

Analysis by Data Mining I

A posteriori

- given MAS complexity – and *opaqueness* of agents – *a posteriori* analysis of the behaviour is often attempted
- the most satisfactory techniques exploit data mining algorithms to analyse *past behaviours* of systems, to understand the rules actually governing the system evolution

Analysis by Data Mining II

Examples

- a technique aimed at reducing the complexity of the behaviour of RoboCup team down to atomic actions is presented in [Kaminka et al., 2003]
- good results based on the same sort are reported in [Lattner et al., 2006] for strategy prediction in football teams of robots
- in [Mnif et al., 2007], a similar approach was used to identify the different phases of the behaviour of a swarm system modelling the interactions between chickens inside their cages

Outline

Self-aware Pervasive Service Ecosystems I

SAPERE project

- European Project FP7 – 2010-2013[a]
  - http://www.sapere-project.eu
  - http://apice.unibo.it/xwiki/bin/view/SAPERE/
- under the hat of the Proactive Initiative AWARENESS
  - http://www.aware-project.eu/

[a] This work has been supported by the EU-FP7-FET Proactive project SAPERE – Self-aware Pervasive Service Ecosystems, under contract no.256873

Self-aware Pervasive Service Ecosystems II

SAPERE model

- based on chemical coordination for pervasive computing [Viroli et al., 2012, Zambonelli et al., 2011]
- LSA (Live Semantic Annotation), as chemical tuples representing individuals, components, services in pervasive scenarios, and triggering eco-laws governing self-organisation of pervasive services
- LSA are reified into *LSA-spaces*, representing contexts for distributed nodes
- *LSA-bonding* allows an LSA to link to other LSA in the same space, and SAPERE entities to inspect the state of their peers and act accordingly
- system rules – the *eco-laws* – are instead in charge of managing the global behaviour of the whole system, by manipulating – deleting, updating, moving, bonding – LSA in the system, in the way of chemical reactions

Outline

MABLE I

Tools for model checking

- model checking is an undoubtedly effective analysis, less often efficient
- which language is used to model the system as well as which tools are used for the analysis play a very important role
- the main issue is the ability to check global properties of MAS, as well as properties of the internal state of the individual entities

MABLE II

MABLE

- MABLE [Wooldridge et al., 2002] is a C-like imperative language that allows the definition of MAS and the verification of system properties expressed in LTL
- the system modelled using MABLE is translated into a Promela specification that can be analysed using compatible tools such as SPIN
- currently developed a translation system to Java
- MABLE has two unique features compared to other similar languages: *claims* and *semantic specifications*

MABLE III

Claims

- claims allow the definition of properties to be verified during model checking
- these properties are not limited to individual entities, but they affect the system in its entirety

Semantic specifications

- a semantic specification allows the definition of pre- and post-conditions for a given communication in the MAS[Huget and Wooldridge, 2004]
- similarly to SAPERE eco-laws

MABLE IV

Assumptions on internal agent states

- another feature of MABLE is the capability of an agent to make assumptions on the internal state of other agents—on their beliefs, for instance
- assumptions can also be expressed inside *claims*, which is particularly useful in MAS contexts, where entities are essentially *blind* to the internal state of others
- this could be used to model the SAPERE mechanism of LSA-bonding SAPERE, which would change an external piece of information (LSA) into a internal property of the agent (belief)
- a bond between LSA would simply be represented as a a connection of the beliefs of the agents

MABLE V

Crowd steering in SAPERE

- two possible ways
  - modelling everything represented by a LSA as a MABLE agent
  - modelling only nodes and user handheld devices as agents, and other LSA as internal beliefs of the *node agent*
- in the former approach, fields of the LSA would become beliefs of the corresponding agent, and bonds between LSA would become beliefs on the internal state of the coupled agents
- in the latter, a new LSA corresponds to a communication act towards the node agent, followed by a change of its internal beliefs
- in both ways, the eco-laws would be modelled as MABLE communication acts, with pre- and post-conditions
- *claims* could be used to check whether the *crowd level* in the nodes, specified by an internal belief of the agent corresponding to the node, evolves in a satisfactory way
- a SAPERE model could be represented in MABLE, then translated into a Promela specification[a], and ideally analysed using SPIN[b]—or, any other model checker supporting Promela specifications

[a] Promela verification language: http://www.dai-arc.polito.it/dai-arc/manual/tools/jcat/main/node168.html

[b] The SPIN model checker: http://spinroot.com/spin/whatispin.html

Outline

Data Mining I

Issues

- putting together the techniques presented in [Kaminka et al., 2003] with the results described in [Lattner et al., 2006], one might be able to face the issue of how the parameters of a simulation could be tuned up, and how to set the relation between the values of these parameters and the corresponding emergent behaviour
- the key-issue in data mining are primarily the identification of the values that define the emergent behaviour, and the creation of a significant and adequately representative data-set

Data Mining II

Goals

1. the correlation between parameter values and emergent behaviour
2. possibly counter-intuitive correlations between some parameters and some aspects of the emergent behaviour could be pointed out
3. identification of the ideal assignments of values, to either enforce desired behaviours, or discourage undesired ones
4. identification of correlations between micro- and macro-events, as well as between macro-events themselves, such as the identification of recurrent patterns of the emergent behaviour

Data Mining III

Crowd Steering

1. apart from the many algorithms that could be useful—partitional (*K-means*), if the number of clusters is known beforehand, hierarchical (*single linkage*), if it is not, bayesian, density based (*DBSCAN*), neural networks, or SVM (*Support Vector Machine*)
2. the first step would be to identify the parameters and values to be analysed—for instance, the crowd factor that penalises the calculation of distances, the rate of the eco-laws, and the effect of changing those values on the crowd levels in every node
3. to build a meaningful training set, a variety of *simulation* should be executed, varying both the values of the parameters and the topology of the museum, so to prevent pathological cases to affect the analysis
4. Alchemist would be obviously the first choice here

Position paper, of course. . .

- we just started facing the issue of analysing and predicting the behaviour of complex pervasive, self-organising systems like SAPERE coordinated systems
- we shortly survey the literature on the general subject along with the available tools, then we select two promising techniques and sketch their application to the analysis and prediction of the behaviour of SAPERE pervasive ecosystems

Future works

- actually experimenting the two approaches here devised upon selected SAPERE applications – such as crowd steering and pervasive display applications
- trying to integrate different techniques into a single articulated yet coherent approach capable of providing reliable prediction of the most meaningful and useful features of a SAPERE system

Bibliography I

Behrens, T. M. and Dix, J. (2007).
Model checking multi-agent systems with logic based Petri nets.
Annals of Mathemathics and Artificial Intelligence, 51(2-4):81–121.
From the issue entitled "Special Issue in honor of Professor Jack Minker's 80th birthday".

Bordini, R. H., Fisher, M., Pardavila, C., and Wooldridge, M. J. (2003).
Model checking AgentSpeak.
In Rosenschein, J. S., Wooldridge, M. J., Sandholm, T., and Yokoo, M., editors, *2nd International Joint Conference on Autonomous Agents and Multi-Agents Systems (AAMAS 2003)*, pages 409–416, New York, NY, USA. ACM Press.

Bordini, R. H., Fisher, M., Visser, W., and Wooldridge, M. J. (2004).
Verifiable multi-agent programs.
In Dastani, M. M., Dix, J., and El Fallah-Seghrouchni, A., editors, *Programming Multi-agent Systems*, volume 3067 of *LNCS*, pages 72–89. Springer.
1st International Workshop (PROMAS 2003), Melbourne, Australia, 15 July 2003. Selected Revised and Invited Papers.

Bordini, R. H., Fisher, M., Visser, W., and Wooldridge, M. J. (2006).
Verifying multi-agent programs by model checking.
Autonomous Agents and Multi-Agent Systems, 12(2):239–256.
EUMAS'04 Special Issue.

Bibliography II

Dixon, C., Winfield, A., and Fisher, M. (2011).
Towards temporal verification of emergent behaviours in swarm robotic systems.
In Groß, R., Alboul, L., Melhuish, C., Witkowski, M., Prescott, T. J., and Penders, J., editors, *Towards Autonomous Robotic Systems*, volume 6856 of *LNCS*, pages 336–347. Springer.
12th Annual Conference (TAROS 2011), Sheffield, UK, 31 August – 2 September 2011. Proceedings.

Giordano, L., Martelli, A., and Schwind, C. (2004).
Verifying communicating agents by model checking in a temporal action logic.
In Alferes, J. J. and Leite, J., editors, *Logics in Artificial Intelligence*, volume 3229 of *LNCS*, pages 57–69. Springer.

Huget, M.-P. and Wooldridge, M. J. (2004).
Model checking for ACL compliance verification.
In Dignum, F., editor, *Advances in Agent Communication*, volume 2922, pages 75–90. Springer.
International Workshop on Agent Communication Languages (ACL 2003), Melbourne, Australia, 14 July 2003. Revised and Invited Papers.

Bibliography III

Kaminka, G. A., Fidanboylu, M., Chang, A., and Veloso, M. M. (2003).
Learning the sequential coordinated behavior of teams from observations.
In Kaminka, G. A., Lima, P. U., and Rojas, R., editors, *RoboCup 2002: Robot Soccer World Cup VI*, volume 2752 of *LNCS*, pages 111–125. Springer.

Konur, S., Dixon, C., and Fisher, M. (2010).
Formal verification of probabilistic swarm behaviours.
In Dorigo, M., Birattari, M., Di Caro, G. A., Doursat, R., Engelbrecht, A. P., Floreano, D., Gambardella, L. M., Groß, R., Sahin, E., Sayama, H., and Stützle, T., editors, *Swarm Intelligence*, volume 6234 of *LNCS*, pages 440–447. Springer.
7th International Conference (ANTS 2010), Brussels, Belgium, 8-10 September 2010. Proceedings.

Lattner, A. D., Miene, A., Visser, U., and Herzog, O. (2006).
Sequential pattern mining for situation and behaviour prediction in simulated robotic soccer.
In Bredenfeld, A., Jacoff, A., Noda, I., and Takahashi, Y., editors, *RoboCup 2005: Robot Soccer World Cup IX*, volume 4020 of *LNCS*, pages 118–129. Springer.

Bibliography IV

Mnif, M., Richter, U., Branke, J., Schmeck, H., and Müller-Schloer, C. (2007).
Measurement and control of self-organised behaviour in robot swarms.
In Lukowicz, P., Thiele, L., and Tröster, G., editors, *Architecture of Computing Systems —- ARCS 2007*, volume 4415 of *LNCS*, pages 209–223. Springer.
20th International Conference, Zurich, Switzerland, 12-15 March 2007. Proceedings.

Pokorny, L. R. and Ramakrishnan, C. R. (2006).
Modeling and verification of distributed autonomous agents using logic programming.
In Leite, J. A., Omicini, A., Torroni, P., and Yolum, P., editors, *Declarative Agent Languages and Technologies II*, volume 3476 of *LNAI*, pages 313–319. Springer.
2nd International Workshop (DALT 2004), New York City, NY, USA, 19 July 2004. Revised Selected Papers.

Viroli, M., Pianini, D., Montagna, S., and Stevenson, G. (2012).
Pervasive ecosystems: a coordination model based on semantic chemistry.
In Ossowski, S., Lecca, P., Hung, C.-C., and Hong, J., editors, *27th Annual ACM Symposium on Applied Computing (SAC 2012)*, Riva del Garda, TN, Italy. ACM.

Bibliography V

Wooldridge, M. J., Fisher, M., Huget, M.-P., and Parsons, S. (2002).
Model checking multi-agent systems with MABLE.
In Castelfranchi, C. and Johnson, W. L., editors, *1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002)*, volume 2, pages 952–959, New York, NY, USA. ACM.

Zambonelli, F., Castelli, G., Ferrari, L., Mamei, M., Rosi, A., Di Marzo, G., Risoldi, M., Tchao, A.-E., Dobson, S., Stevenson, G., Ye, Y., Nardini, E., Omicini, A., Montagna, S., Viroli, M., Ferscha, A., Maschek, S., and Wally, B. (2011).
Self-aware pervasive service ecosystems.
Procedia Computer Science, 7:197–199.
Proceedings of the 2nd European Future Technologies Conference and Exhibition 2011 (FET 11).

Towards the Analysis & Prediction of Complex System Behaviour in SAPERE

Marco Alberti & Andrea Omicini
marco.alberti3@studio.unibo.it, andrea.omicini@unibo.it

Alma Mater Studiorum—Università di Bologna

ASENSIS 2012
1st International Workshop @ SASO 2012,
Lyon, France, 10th of September 2012