II Facoltà di Ingegneria
Alma Mater Studiorum – Università di Bologna

Study on behaviour analysis and predictability of self-organizing systems as part of project SAPERE

Dott. Ing. Marco Alberti

Overview

- **Objectives**
  - Phases of the study
- **Background**
  - Analysis by Model Checking
  - Analysis of Swarm systems
  - Analysis by Data Mining
- **Initial proposals for project SAPERE**
  - MABLE
  - Tuning by Data Mining
- **Bibliography**

Objectives

Phases of the study

1) The first part focused on the acquisition of knowledge and documents that provide adequate image of the state of the art with regard to the analysis and prediction of the behavior of multi-agent systems and, in particular, of self-organizing systems

2) The second part focused on the analysis of techniques and methodologies highlighted, to see which could merit further investigation, having proved their worth and/or effectiveness

3) The third part focused on how the results achieved in the previous parts might be applied, with reasonable chances of success, to project SAPERE

Background

Analysis by Model Checking
(1 di 3)

- It focuses on (semi-)exhaustive analysis of the states the system will go through, often accompanied by a representation by a LTS (*Labelled Transition System*)
- It allows the verification that the system meets certain properties expressed using various types of logics (LTL, CTL, DLTL, GLTL, etc.)
- It is almost exclusively used for the verification of *safety* or *liveness* properties
- Very effective for targeted problems but the application is generally complex for generic analysis and often prohibitive from the computational point of view

Analysis by Model Checking
(2 di 3)

- LTL is usually preferred to CTL
- From LTL numerous extensions have arisen, including:
  - DLTL (*Dynamic Linear Temporal Logic*): focused on upgrading the *until op*erator to provide better support for expressing pre-and post-conditions
  - GLTL (*General Linear Temporal Logic*): allows the definiton of LTL properties for the entire system, as well as for the individual entities involved

Analysis by Model Checking
(3 di 3)

- Many multi-agent systems consist of BDI agents; BDI specifications can be translated into LTL specifications
- AgentSpeak(L) dominates among BDI structures; AgentSpeak(F) is a subset of AgentSpeak(L), for *finite state systems* only
- A system described by AgentSpeak(F) can be represented by a SLPN (*Simple Logic Petri Net*)

Analysis of Swarm 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 entities of the same type, it is possible to define a state diagram common to all entities
- Each state is then associated to the number of entities that actually are in that state at a particular moment using probabilistic analysis
- Now it is possible to define a set of expected behaviors associated with chances of effective implementation

Analysis by Data Mining

- Because of the complexity of a multi-agent system, approach is often attempted through posteriori analysis of the behavior
- Applying data mining tools to sets of data obtained through a series of simulations, it has been possible to predict the expected behavior of a system with a relatively low margin of error
- In particular, this approach has been used to predict the strategy developed in the field by the various robot teams of the RoboCup football tournament, to thereby prepare an effective counteroffensive

Initial proposals for project SAPERE

MABLE

(1 di 5)

- Model checking is an undoubtedly effective analysis, less often efficient
- The language used to model the system as well as the tool used for the analysis play a very important role
- The main issue, most of the times, is to be able to check global properties of multi-agent system, as well as properties of the internal state of the individual entities

MABLE

(2 di 5)

- MABLE is a *C-like* imperative language that allows the definition of multi-agent systems and the verification of system properties expressed in LTL, the system modeled using MABLE is then translated into a Promela specification that can be analyzed using compatible tools such as SPIN

- It is being currently developed a translation system to the Java language

- MABLE has two unique features compared to other similar languages: *claims* and *semantic specifications*

MABLE

(3 di 5)

```
shared int a;

claim exists ag : agent <>(believe ag
(intend agent1 (believe ag a == 10)));

agent agent1 {

  for (a = 0; a != 10; a = a + 1)

    print("agent1: a = %d \n", a);

  inform agent2 of (a == 10);

  for (a = 10; a != 15; a = a + 1)

    print("agent1: a = %d \n", a);

}
```

- *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
- Claims also allow an entity to make assumptions about the internal state of other entities

MABLE
(4 di 5)

```
i:inform(j,phi)
(believe i phi)
(believe j (intend i (believe j phi)))


i:inform(j,phi)
1
(believe j (intend i (believe j phi)))
```

- It's possible to add a semantic specification for the communicative actions to the multi-agent system model
- The specification enriches communications with preconditions to be met to allow the execution, and post-conditions to be put into effect after the communicative action has occurred

MABLE
(5 di 5)

If it were to choose a language for modeling a SAPERE infrastructure, choosing MABLE would have some benefits:

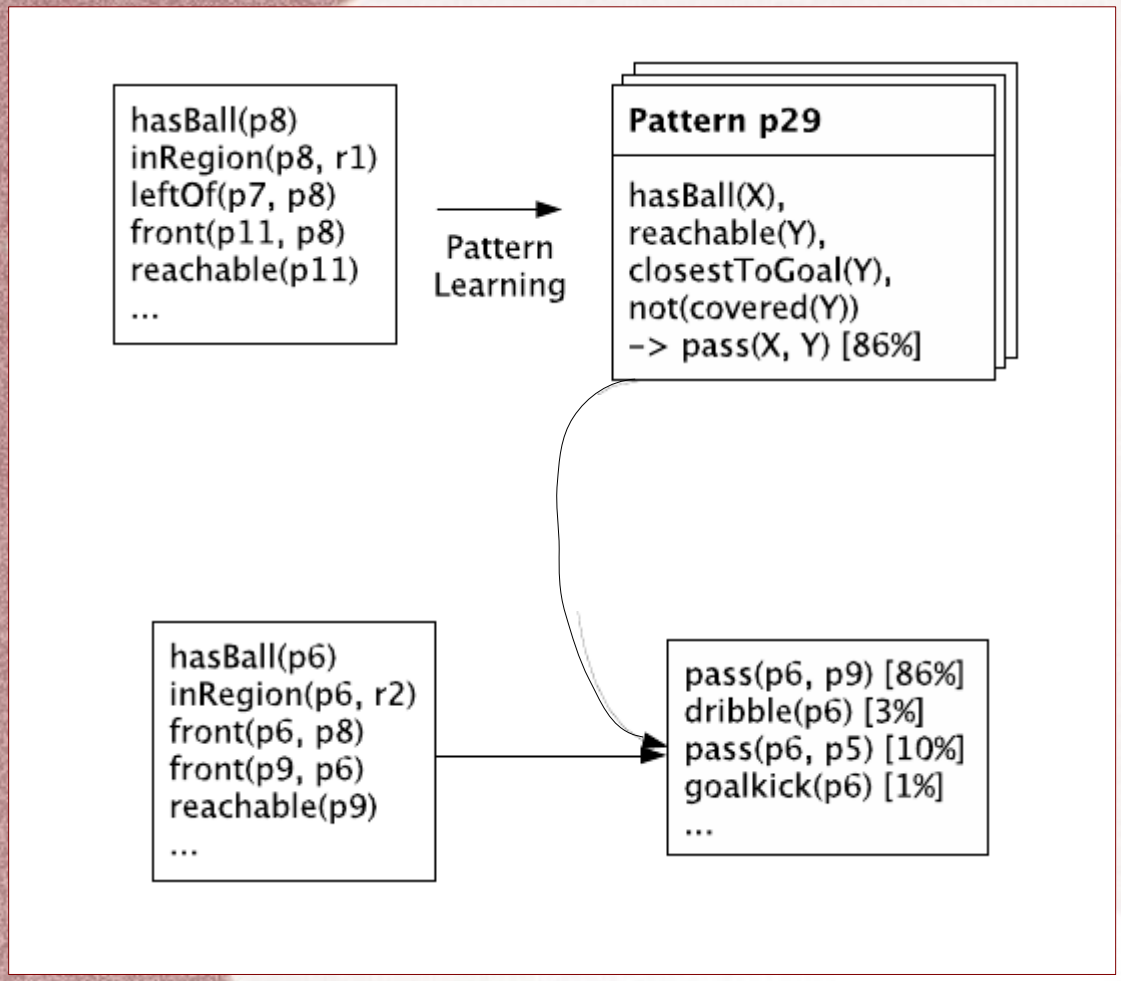
- the mechanism of *claims* would allow to verify global properties which are much more significant for a self-organizing system, compared to properties of state of the individual entities
- the *semantic specifications* supported by MABLE find many similarities to the mechanism of *EcoLaws* used by the SAPERE infrastructures
- Promela is a language widely used so it's possible to find many model checking tools, SPIN first of all; furthermore it's already under development a porting to the Java language

Tuning by Data Mining
(1 di 4)

- At present, there is no analytical method to tune the parameters of the simulations
- Tuning is carried out trying to deduce the effect that changing the value of a parameter will have on the behaviour of the simulation
- Feedback of some kind on the effect that the new value has produced is achieved through observation of various subsequent runs

Tuning by Data Mining

(2 di 4)



- Behavior analysis of the football teams of RoboCup has given encouraging results
- Once identified the variables that characterize the behavior (actions performed by the robot) ...
- … it was possible to obtain predictions about the strategies that the team would put in place with associated probabilities

Tuning by Data Mining
(3 di 4)

- Making necessary changes and adaptations, the same procedure is applicable to the project SAPERE
- A key part will be the identification of significant variables in the simulation
- Once identified these, the data set will be obtained from different runs of the simulation itself
- At this point it will be possible to apply Data Mining analysis to the data sets generated this way

Tuning by Data Mining
(4 di 4)

The analysis should be able to put in evidence:

- the correlations between the values of parameters and the emergent behavior
- any non-intuitive correlations between the parameters of the simulation and the emergent behavior
- ideal policy of assignment of values to these parameters
- any correlations between macro-events of the emergent behavior

Bibliography

Bibliography
(1 di 2)

- Laura Giordano, Alberto Martelli, Camilla Schwind. *Verifying Communicating Agents by Model Checking in a Temporal Action Logic*. Proc. Logics in Artificial Intelligence, 9th European Conference, JELIA 2004, 2004
- L. Robert Pokorny, C. R. Ramakrishnan. *Modeling and Verification of Distributed Autonomous Agents Using Logic Programming*. Lecture Notes in Computer Science 3476, 2005
- Michael J. Wooldridge, Michael Fisher, Marc-Philippe Huget, Simon Parsons. *Model checking multi-agent systems with MABLE*. Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2, AAMAS '02, 2002
- Marc-Philippe Huget, Michael J. Wooldridge. *Model Checking for ACL Compliance Verification*. Lecture Notes in Computer Science 2922, 2004
- Rafael H. Bordini, Michael Fisher, Carmen Pardavila, Michael J. Wooldridge. *Model Checking AgentSpeak*. Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agents Systems, 2003
- Rafael H. Bordini, Michael Fisher, Willem Visser, Michael J. Wooldridge. *Verifiable Multi-agent Programs*. Lecture Notes in Computer Science 3067, 2004
- Rafael H. Bordini, Michael Fisher, Willem Visser, Michael J. Wooldridge. *Verifying Multi-agent Programs by Model Checking*. Lecture Notes in Computer Science 12(2), 2006

Bibliography
(2 di 2)

- Tristan M. Behrens, Jurgen Dix. *Model checking multi-agent systems with logic based Petri nets*. Annals of Mathematics and Artificial Intelligence, 2007
- Savas Konur, Clare Dixon, Michael Fisher. *Formal Verification of Probabilistic Swarm Behaviours*. Lecture Notes in Computer Science 6234, 2010
- Clare Dixon, Alan Winfield, Micheal Fisher. *Towards Temporal Verification of Emergent Behaviours in Swarm Robotic Systems*. Lecture Notes in Computer Science 6856, 2011
- Gal A. Kaminka, Mehmet Fidanboylu, Allen Chang, Manuela M. Veloso. *Learning the Sequential Coordinated Behavior of Teams from Observations*. Proceedings of the RoboCup-2002 Symposium, 2002
- Andreas D. Lattner, Andrea Miene, Ubbo Visser, Otthein Herzog. *Sequential Pattern Mining for Situation and Behaviour Prediction in Simulated Robotic Soccer*. Lecture Notes in Computer Science, 2006
- Franco Zambonelli, Mirko Viroli. *A survey on nature-inspired metaphors for pervasive service ecosystems*. International Journal of Pervasive Computing and Communications 7(3), 2011
- Mirko Viroli, Danilo Pianini, Sara Montagna, Graeme Stevenson. *Pervasive Ecosystems: a Coordination Model based on Semantic Chemistry*. Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC 2012) , 26-30 March 2012

End of presentation