

# Emergent Properties for Data Distribution in a Cognitive MAS\*

Andrei Olaru, Cristian Gratie, and Adina Magda Florea

**Abstract** Emergence is a key element in the research of multi-agent systems. Emergent properties provide higher level features of a system formed by simpler individuals. So far, emergence has been studied mostly by means of systems formed by reactive agents – that are more simple to design and implement. As computational power increases and even small devices become more capable, cognitive agents become a promising solution for the solution of more complex problems. This paper presents a simple example of a multi-agent system inspired from the Beliefs-Desires-Intentions model, designed to manifest emergent properties: a data storage system that assures distribution and replication of data without the need for these features to be controlled from the exterior.

## 1 Introduction

Emergence is a key concept in the recent development of multi-agent systems. Several definitions of emergence exist [3] but none is yet generally accepted. Many times emergence is defined by its effect – the formation of patterns in the structure or behaviour of certain systems.

In the context of a large number of agents, forming a complex system [1], emergence offers the possibility of obtaining a higher level function or property by using agents with lower level implementations. The use of emergence allows a considerable reduction in the complexity needed for the individual agents, therefore they can run on simpler and smaller devices.

---

\* The original publication is available at  
<http://www.springerlink.com/content/a616518k47114854/>

Andrei Olaru · Cristian Gratie · Adina Magda Florea  
University Politehnica of Bucharest, Splaiul Independentei 313, 060042 Bucharest, Romania  
e-mail: cs@andreiolaru.ro · cgratie@yahoo.com · adina@cs.pub.ro

Many recent papers on emergence and multi-agent systems deal with emergent properties and behaviour in systems formed of a large number of reactive agents. Reactive agents are used because they are simple and need very limited computational capacity, but emergent functions may be more complex and the structure is robust [13, 8, 2].

Nowadays, the capabilities of even very basic computing devices have considerably increased, allowing for a much more complex internal structure for agents – the possibility to hold reasonable amounts of data and to have a more nuanced behaviour. Cognitive agents have knowledge about the surrounding environment, have goals they desire to fulfill, make plans and take action in order to fulfill them.

The purpose of this paper is to study emergent properties in a multi-agent system formed of cognitive agents. A system of cognitive agents used for the storage of data has been designed and implemented. This system is, at this time, very simple and limited, but it makes a good framework for the study of emergent properties.

The system was designed with the purpose of manifesting emergent properties like uniform distribution and availability of data. Individual agents were given local goals that do not directly imply that the data is uniformly distributed or that data is not completely lost. No agent has the capability to have a global image on the system or even on a significant part of it. Agents' knowledge is limited to beliefs about its immediate or almost immediate neighbours. Yet the system as a whole keeps data well replicated and distributed.

The paper is organised as follows. Section 2 is dedicated to related work in the field of emergence and multi agent systems. Section 3 presents the main topic of this paper: a cognitive multi-agent system designed to manifest emergence. Section 4 describes the results obtained from experiments. The last section is dedicated to the conclusions.

## 2 Related Work

What we know is that emergence appears in the context of complex systems [1] – systems composed of a large number of interacting individual entities. Emergence needs two levels of perspective: the inferior, or micro level of the individual entities and the superior, or macro level of the whole system. A simple definition is that "emergence is the concept of some new phenomenon arising in a system that wasn't in the system's specification to start with" [11]. A more elaborated definition is that "a system exhibits emergence when there are coherent emergents at the macro-level that dynamically arise from the interactions between the parts at the micro-level. Such emergents are novel with respect to the individual parts of the system" [5]. An "emergent" is a notion that can represent a property, a structure or a behaviour that results from emergence.

The essence of emergence is not actually the novelty or the unexpectedness of the emergent – as these will fade at later experiments although the emergents will stay the same – but the difference between the description of the individual and

the description of the emergent [11]. If the minimal description of the individual is taken, it cannot be used to describe the emergents resulting from the system, therefore the emergent is considered as novel and, potentially, unexpected.

More formally, a better definition of emergence [2] would be that an emergent is, in the context of an interacting set of agents whose dynamics are expressed in a vocabulary  $D$ , a global phenomenon – static or dynamic, but nevertheless invariant – that is observed by the agents or by an external observer and can only be interpreted in a vocabulary  $D'$  that is different from  $D$ .

There are a few important features of emergence [5, 6, 7]:

- emergence occurs out of the interactions between the parts of a complex system.
- emergence is defined in relation with two levels – it is manifested at the higher level, arising from interactions at the lower level.
- the representation of the emergents cannot be reduced to the specifications of the individuals.
- emergents only arise after a certain time in which the system has evolved.
- once they occurred, emergents will maintain identity over time.
- emergents arise without any centralised or exterior control.
- the emergent phenomena is robust and flexible, i.e. it is not influenced by damage on the system (even if the emergent is temporarily not observable, it will arise again as the system evolves).

In most examples of emergence developed so far in the field of multi-agent systems, reactive agents have been used. They are easier to implement and control, and they are suitable for small devices with low computational power.

Notable examples of emergents in reactive agent systems relate to the formation of a certain geometrical or geometry-related structure or behaviour: arrangement of agents in a circular or almost circular shape [2, 8, 13]; detection of areas in an image [4]; gathering of resources in a single area [10]; foraging of food or transportation of loads [12]; emergence of a traffic direction [9].

Although reactive agent systems may be very useful, there are many advantages that a cognitive agent has over a reactive agent. First, it is proactive. Even if there are no signals, perceptions or stimuli from the environment, a cognitive agent may act by itself, taking action according to its objectives. Second, a cognitive agent is aware of its situation and may reason about it. It is aware of what it is suppose to fulfil as final goal and is capable of making plans and taking action towards the realisation of its goal. The cognitive agent can use its experience and information about the environment and the consequences of past actions to develop a better plan every time a similar situation occurs.

### 3 A Cognitive Multi-Agent System for the Distribution of Data

A simple application has been developed and experiments have been performed in order to demonstrate how emergents may arise in a system formed of cognitive

agents. The system models a data storage network. Its purpose is to store pieces of data (of equal size) injected from the environment and to provide the environment with the data that is requested. There is a limited set of pieces of data, with certain similarities between them. The desired emergents are the replication, distribution and availability of the data across the system.

The experiments were performed with a set of agents placed on a rectangular grid. Each agent can communicate only with its 8 neighbours. Each agent has a limited capacity of 4 data chunks (6 different data chunks are used in the experiments). The agents must always have some capacity left, ready for data chunks that might be injected from the environment. However, agents must try to store as much data as possible, in order to not waste the capacity of the system.

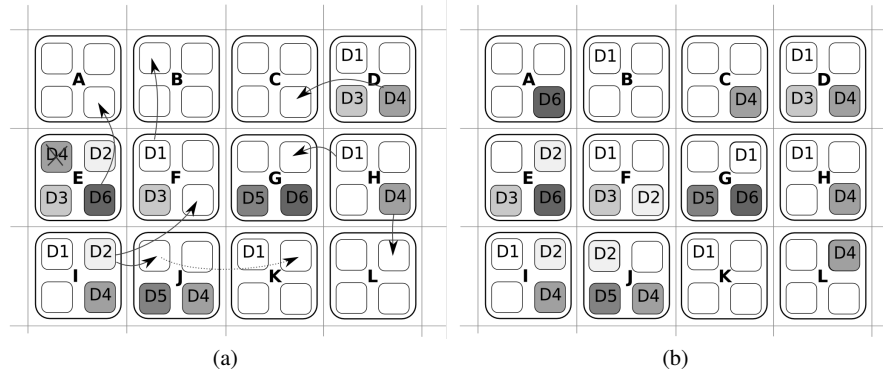
The desired emergent properties of the system are replication of data, distribution (possibly uniform) and availability. That means that, after letting the system evolve for some time, the following properties should exist: there should be more than one agent holding one piece of data; in any area of the grid there should exist copies of all pieces of data; if requested to, any agent should be able to get hold of any piece of data.

Following the ideas in [13, 8, 3, 2], in order to obtain the desired emergents the individual agents were designed with selfish objectives that reflect the spirit of the global objectives. In order to obtain data replication and distribution, an agent is "curious" and, if capacity is available, it will be interested in and it will request a copy of any piece of data it does not hold and that is similar to what it already holds. Using similarity is good because it is more likely for subsequent external requests to be made for similar content. In order to obtain variation and uniformity in the data distribution pattern, an agent will "lose interest" in pieces of data that are already held by most of the neighbour agents. In order for an agent to be able to get hold of pieces of data not held by itself or any of the neighbours, agents will be able to collaborate for the common goal of finding a certain piece of data that, if data is well distributed, cannot be very far away.

Agents were implemented using the Beliefs-Desires-Intentions model. Beliefs of an agent are associations of the type  $\langle DataID, AgentID \rangle$  specifying the pieces of data held by itself and by each agent in the vicinity. Knowledge about other agents is also associated with the *ID* of the neighbour agent that provided that knowledge. Beliefs that have not been refreshed or revised for a long time will be discarded, or "forgotten", as the probability for them to still be true decreases.

According to the ideas above, the goals of an agent are:

- In case there is an external request for data, provide that data or try to find it in the vicinity.
- Maintain 25% of the capacity free, ready for potential data coming from the environment.
- In case there is available capacity (over 25%), request interesting data from a neighbour.
- Process and respond to messages from neighbour agents.
- If all other objectives are complete (capacity 75%, no messages), discard some data that is already contained by most of the surrounding agents.



**Fig. 1** Group of agents, before (a) and after (b) the data transfers indicated by the arrows. There are 6 data pieces in the system: D1 to D6.

In order to fulfill its objectives, an agent has the following available actions:

- Send a request for data to a neighbour.
- Send a piece of data to a neighbour (as reply to a request only).
- Receive data or knowledge from a neighbour.
- Discard a piece of data.
- Broadcast to the all neighbours own knowledge and intentions.

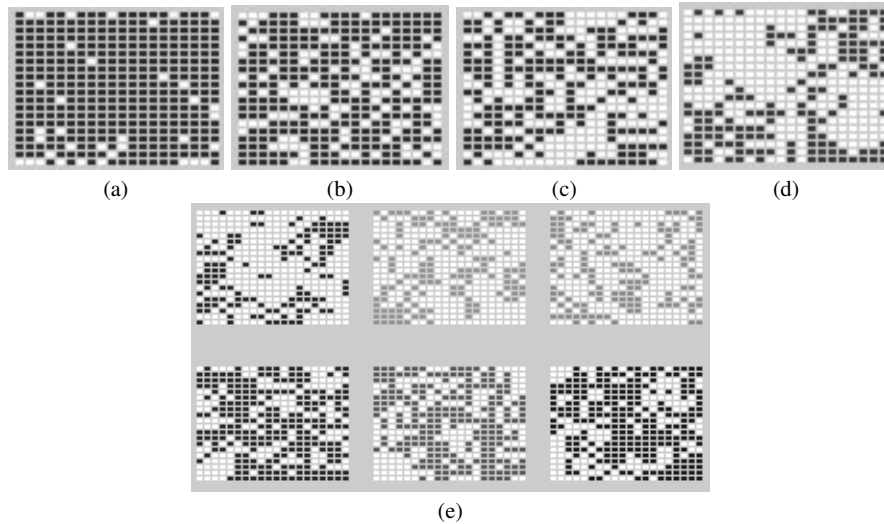
All agents in the system work simultaneously. At each step, one agent chooses a goal that is neither already completed (like already having 25% of the capacity free) nor impossible to attain (the agent cannot retrieve specified data from any of the neighbours). After choosing a goal, the agent makes a plan formed of the actions it needs to take to fulfill the goal, and starts performing those actions.

The example in Figure 1 presents a group of agents in a system where data did not get to spread very much. Consider every agent has sufficient knowledge about his vicinity. For example, agent **I** has the following beliefs:

$\langle I : D1 \rangle \langle I : D2 \rangle \langle I : D3 \rangle$   
 $\langle J : D4 \rangle \langle J : D5 \rangle$   
 $\langle E : D4 \rangle \langle E : D2 \rangle \langle E : D3 \rangle \langle E : D6 \rangle$   
 $\langle F : D1 \rangle \langle F : D3 \rangle$   
 $\langle G : D5 \rangle$  (knows that from *F*)

Each agent that has less than 75% of its capacity full will ask for some piece of data from one of its neighbours. Agent **E** will discard data *D4*, as both **I** and **J** also have it. Supposing that agent **K** needs *D2* (most likely because there is an external request for it) it will communicate its intentions to its neighbours and **J** will collaborate with it and retrieve *D2*.

There are several tweaks that had to be performed on the system in order for it to function well and for the agents not to be overwhelmed by messages and/or by actions to perform. These tweaks were implemented in the agents' policies. First, there must exist a good balance between how fast the agent can learn new knowledge



**Fig. 2** The distribution of one piece of data in a system with agents of capacity 4 in the context of a total number of pieces of data of: (a) 3, (b) 4, (c) 6, (d) 8. The simultaneous distributions of 6 pieces of data (e).

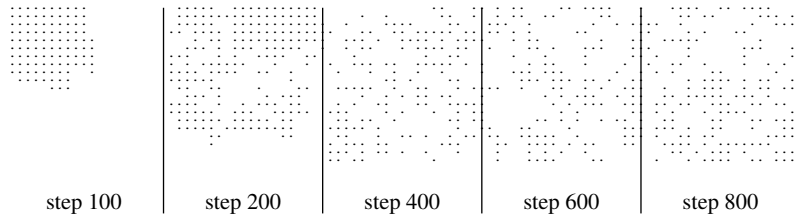
and revise its beliefs and how often an agent broadcasts its own knowledge and intentions to its neighbours. If an agent broadcasts all its knowledge each time it learns something new, all agents will be flood with messages they don't have time to process. If the agents broadcast their knowledge too rarely, this might result in the agents having outdated beliefs about their neighbours. The settings used for the experiments were to broadcast all new knowledge about every 10 to 20 steps.

Another element that needs good balance is the rate at which an agent forgets old beliefs. If an agent forgets too quickly, it might have already forgotten some important information at the time it is supposed to be using it. If an agent forgets too slowly, it will have a lot of information that is outdated and of no use anymore.

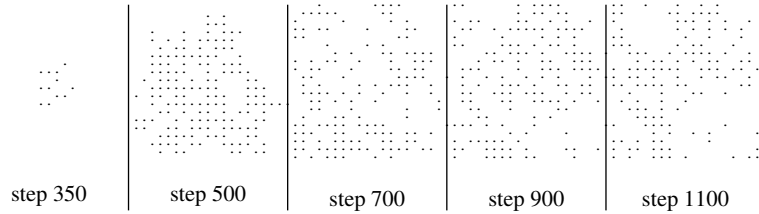
## 4 Results and Discussion

Many experiments were carried out throughout the development of the system, first for assuring that the agents were not overwhelmed by messages and actions to perform, second for the actual study of the evolution of the system. The presented experiments involved a square grid of 400 agents.

Several cases were studied, according to the number  $ND$  of pieces of data injected into the system:  $ND = 75\% \cdot agent\_capacity$ ,  $ND = 100\% \cdot agent\_capacity$ ,  $ND = 150\% \cdot agent\_capacity$  and  $ND = 8(\text{number of neighbours})$ .



**Fig. 3** The distribution of one piece of data injected at moment 0.



**Fig. 4** The distribution of one piece of data injected at moment 300.

**Case 1 (75%)** The system acted as expected: every agents contains approximately all data (see the distribution for one piece of data in Figure 2 (a)). Here, "approximately" means that, due to the "loss of interest" in certain pieces of data, they are discarded and, later, acquired again. Data is almost perfectly distributed. The "holes" that are present (agents not holding that data) "move" around the system, however there is always a neighbour that has that data.

**Cases 2 and 3 (100% and 150%)** After the system stabilises, data remains evenly distributed throughout the system (Figure 2 (b, c)). The "holes" present in the distribution of a certain piece of data are larger as more data is stored in the system, however the uniform distribution makes it easy for agents to quickly obtain data, if necessary.

**Case 4 (200%)** Closer to reality, the case when more data is present in the system yields a less even distribution (Figure 2 (d)). However, no data is farther than two neighbours away. In this case, when the pressure on agents is high (each agent only holds 30% to 50% of the data available in the system), similarity between data becomes important and observable. Figure 2 (e) shows the data distributions for 6 of the 8 pieces of data, of which the first three have a high similarity between them, as well as the last three. It is easy to observe that the distributions of similar data are resemblant, showing that similar data tends to be found in the same areas.

It is also interesting to follow how the distribution of one piece of data evolves as the system runs. Figure 3 presents the evolution of data injected at moment 0 – when the system was started – in the top left corner. At first the distribution is solid (step 100). When most of the agents start to hold that data, agents in the central part start discarding it (step 200). Next, the phenomenon spreads throughout the system and the distribution becomes poor (steps 400, 600). Finally (step 800), the distribution

stabilises at a normal value of 179 out of 400 agents, with the system holding 5 different pieces of data.

A slightly different evolution is observed in Figure 4, that shows the distribution of a piece of data injected at a later time, together with other two (making a total of 8). Initially, the distribution grows slower (the system is already full) and no solid distributions are found. An interesting fact is that the number of agents holding the data stabilises much faster, steps 500 to 1100 having an almost constant 160 agents holding the data.

Although all the examples presented here involve the same number of agents, this number can be changed easily and the system scales with no problem, as every agent interacts only with and has knowledge only about its close vicinity.

What is important to point out is that very good distributions are obtained although none of the agents has that as objective. Moreover, agents are not capable of measuring in any way the distribution of the data. Their objectives are local and mostly selfish, but a different global result is obtained. As shown in the previous subsection, the system and the agents have been specifically designed to produce the emergents, by translating the idea of the global result to the local scale of the individual.

The developed system makes a good platform for the study of emergent properties in a simple cognitive agent system. Although the designed system is still very simple, there are important differences between the implemented agents and reactive agents. It must be pointed out that the agents reason permanently about what data they should get and what they should keep. A simple form of collaboration has been implemented – agents are able to share intentions. Reactive agents would not have been able to have beliefs about their neighbours and would not have been able to reason about what action would be best in the context.

## 5 Conclusion

Emergence is an essential notion in the field of multi-agent systems. It provides the possibility of obtaining a more complex outcome from a system formed of individuals of lower complexity. Although there are many recent studies concerning emergence, there is yet no clear methodology on how to specifically design a system so that it would manifest emergence. Moreover, most implementations use reactive agents, that have limited cognitive and planning capacities.

The paper presents a cognitive multi-agent system in which the agents' interactions allow the emergence of specific properties required for solving the problem, by giving the agent local goals that naturally lead to the global, desired, goal of the system.

As future work, the system will be improved, so that the emergents will be more nuanced. Agent interests will have a stronger influence and collaboration will be used in a greater measure.

## References

1. Amaral, L., Ottino, J.: Complex networks: Augmenting the framework for the study of complex systems. *The European Physical Journal B-Condensed Matter* **38**(2), 147–162 (2004)
2. Beurier, G., Simonin, O., Ferber, J.: Model and simulation of multi-level emergence. *Proceedings of IEEE ISSPIT* pp. 231–236 (2002)
3. Boschetti, F., Prokopenko, M., Macreadie, I., Grisogono, A.: Defining and detecting emergence in complex networks. *Lecture notes in computer science* **3684**, 573–580 (2005)
4. Bourjot, C., Chevrier, V., Thomas, V.: A new swarm mechanism based on social spiders colonies: From web weaving to region detection. *Web Intelligence and Agent Systems* **1**(1), 47–64 (2003)
5. De Wolf, T., Holvoet, T.: Emergence versus self-organisation: Different concepts but promising when combined. *Engineering Self Organising Systems: Methodologies and Applications* **3464**, 1–15 (2005)
6. Goldstein, J.: Emergence as a construct: History and issues. *Emergence* **1**(1), 49–72 (1999)
7. Heylighen, F.: The science of self-organization and adaptivity. *The Encyclopedia of Life Support Systems* pp. 1–26 (2002)
8. Mamei, M., Vasirani, M., Zambonelli, F.: Self-organizing spatial shapes in mobile particles: The TOTA approach. In: S. Brueckner, G.D.M. Serugendo, A. Karageorgos, R. Nagpal (eds.) *Engineering Self-Organising Systems, Methodologies and Applications* (after ESOA 2004 workshop), *Lecture Notes in Computer Science*, vol. 3464, pp. 138–153. Springer (2004)
9. Picard, G.: Cooperative agent model instantiation to collective robotics. In: *Engineering Societies in the Agents World V: 5th International Workshop, ESAW 2004, Toulouse, France, October 20-22, 2004: Revised Selected and Invited Papers*. Springer (2005)
10. Randles, M., Zhu, H., Taleb-Bendiab, A.: A formal approach to the engineering of emergence and its recurrence. *Proc. of EEDAS-ICAC* pp. 1–10 (2007)
11. Standish, R.: On complexity and emergence. *Arxiv preprint nlin.AO/0101006* pp. 1–6 (2001)
12. Unsal, C., Bay, J.: Spatial self-organization in large populations of mobile robots. *Proceedings of the 1994 IEEE International Symposium on Intelligent Control* pp. 249–254 (1994)
13. Zambonelli, F., Gleizes, M., Mamei, M., Tolksdorf, R.: Spray computers: Frontiers of self-organization for pervasive computing. *Proceedings of the 13th IEEE Int'l Workshops on Enabling Technologies, WETICE* pp. 403–408 (2004)