Simulation of complex emergent structures in Presage

R. Frei[1], S. Macbeth[2], J. Pitt[2], and G. Di Marzo Serugendo[3]

[1]EPSRC Centre for Through-life Engineering Services, Cranfield University, Bedforshire MK43 0AL, UK
[2]Intelligent Systems and Networks Group, Department of Electric and Electronic Engineering, Imperial College London, SW7 2AZ, UK
[3]Institute of Services Science, Faculty of Social and Economics Science, Univ. of Geneva, 1227 Carouge, CH
work@reginafrei.ch, samuel.macbeth06@imperial.ac.uk, j.pitt@imperial.ac.uk, giovanna.dimarzo@unige.ch

Abstract
Systems with self-* properties are often governed by certain rules. This article reflects on the underlying concepts, principles and mechanisms with the goal to make them accessible to the engineering of socio-technical systems composed of humans, agents, robots and other entities. Simulations in Presage, a platform for prototyping agent societies, have been set up to demonstrate how a set of simple and mostly local rules governs the agents' behaviours, and how system level behaviour results without anybody having control over others, a global view or knowing the system's goal. Complex structures, such as bridges between seeds, emerge according to the chosen rules and parameters. This paper presents preliminary results as well as ongoing / future work.

Keywords:
Complexity; Complex adaptive systems; Emergence; Multi-agent systems; Presage.

1 INTRODUCTION

Complex adaptive systems in nature often exhibit a range of self-* properties, such as self-organisation, self-reconfiguration, self-healing and others, as well as emergence. Engineers are increasingly trying to understand, apply and use the underlying concepts, principles and mechanisms to provide the systems they are building with similar properties. However, most of the work is still intuitive trial-and-error style; generic architectures and systematic methodologies are largely missing, although some have started to appear. An example is MetaSelf [4], which served as an inspiration for the work presented in this paper. More investigations and implementations are, however, necessary. To persuade other researchers as well as interested industrials, demonstrators with real-world functionalities are needed.

The work presented in this article contributes both on a conceptual level – explaining why and how nature-inspired concepts, principles and mechanisms of complex systems are useful for engineering, and how they could be used – as well as on an application level, providing the results from simulated experiments with complex emergent structures.

The systems which we address with our work are typically composed of a medium number of agents, meaning a few dozens to a few hundreds. The agents can be purely software, or software associated with a technical device, or human. Such socio-technical ecosystems dynamically change and evolve. The agents may interact in multiple and multi-lateral ways, dynamically compete and collaborate with diverse peers, request and provide services, leave or join the open system at any time. Generally, their behaviours follow certain rules, policies and norms; these very depending on the type of interaction (machine to machine, human to machine, or human to human). Rules, policies and norms assure that the interactions have certain standards, and that the system or parts of the system maintain certain properties, and that they converge towards the desired properties [13].

Organisation of this article: Section 2 elaborates on the engineering of self-* properties. Section 3 introduces the Presage simulation platform. Section 4 explains how the simulations were designed and for what reasons. Section 5 details the experiments. Section 6 specifies the complex emergent structures system in a formal model. Section 7 discusses their outcome. Section 8 presents related work. Section 9 draws conclusions and indicates further steps to be taken.

2 RULES FOR SELF-* PROPERTIES

The objective of this research is to investigate how entities of various nature can interact and achieve something together; this may be fulfilling a task or providing a service. Humans and technology will increasingly work hand-in-hand, and they will need new ways of communicating with each other.

Various types of interactions are required:

- Machine – machine
- Machine – human
- Human – human

Both machines (here meaning technology such as software agents, robots, devices, etc.) and humans have their characteristic strengths and weaknesses. Among others, humans are sometimes irrational, whereas technology lacks the typically human understanding of unforeseen situations. Technology, however, has a higher level of repeatability and precision. Humans are often more dexterous.

In simulation, the human irrationality can be represented by giving the representing agents a variable probability of doing what they are supposed to do. Artificial agents usually do what they rationally should, but their likelihood to fail may also be reflected by a certain probability.

The mechanisms used in this type of research are of three types:

- Self-organisation between technical (artificial) units [6], guided by a variety of features such as gradients and aggregation.
- Self-management of technical systems [4, 3, 15], guided by rules and policies.

Note that the combination of self-organisation and self-management in the same system is rather rare. An example where this has been envisioned are self-organising assembly systems (SOAS) [7, 10] (in spite of their name).

- For socio-technical systems as discussed in this paper, human factors need to be added. For instance, what incentives does a person need to interact with a machine? What happens in case of communication difficulties or other failures?

The simulation framework introduced below is abstract enough to take all three types of mechanisms into account: self-organisation, self-management, and human factors. The details of these mechanisms, and especially the third type, are subject to ongoing research and will be discussed in subsequent papers.

3 PRESAGE: A SIMULATION PLATFORM FOR SELF-* SYSTEMS

While there is wide consensus that self-* properties are desirable in many engineered systems, it is much less clear how to create them in engineered systems. A major challenge is how to show sufficient evidence that the desired property is achieved under all considered constellations, and that the system does never behave in undesired or harmful ways. One way to show this is by systematic simulation. This paper contributes to the research in this area by showing examples of abstract experiments in a suitable simulation software. The results shown are preliminary and merely an indication of the direction of the ongoing research.

Based on general interaction principles, which could represent any of the mechanisms discussed in section 2, we derived the simulation scenarios presented in section 5. After discussion, they will contribute to the conclusions about the engineering of complex emergent structures in general (section 9).

Presage [21] is a simulation platform for rapid prototyping of agent societies. Particular attention was given to its suitability for networks that are open, dynamic and decentralised.

This approach enables designers to investigate the effect of agent design, network properties and the physical environment on individual agent behaviour and long-term collective global performance, as much as the verification of specific properties. Presage can be used for the simulation and animation of agent interaction models, allowing a system designer to investigate the complex social behaviour of the agents, the evolution of network structures, and the adaptation of conventional rules. Previous experiments have been done in areas including e-commerce, ad hoc networks, and colored trails.

Presage is currently being improved in many aspects, particularly adding the capability of running simulations on a distributed cluster, scaling up to much larger agent societies than are currently possible.

Further information as well as the software itself is available on:
`https://sites.google.com/site/presageproject`

4 SIMULATION SYSTEM DESIGN

The objective pursued with the current research is to investigate the creation, maintenance and change of complex emergent structures that may be composed of humans, organisations, software agents, robots, and other technical devices of almost any imaginable nature.

In concrete terms, the creation of such systems will consist of **self-assembly** and **self-organisation**; their maintenance will rely on **self-reinforcement, self-management, self-adaptation, self-protection, self-diagnose, self-repair** and **self-healing**; these systems' change will be based on some of the previously mentioned capabilities in addition to **self-disassembly, self-reassembly, self-reconfiguration**, and potentially **self-reproduction** in the wider sense of a system being able to create a copy of itself.

The MetaSelf framework serving as a conceptual inspiration and the Presage simulation platform being available, a simple yet powerful abstraction of such a system was required. Given that the mechanisms to be investigated shall be usable for a plentitude of different **socio-technical systems**, we decided to use simple agents that could represent anything. The agent society is heterogeneous, in this initial case consisting of only two types, but more diversity can be introduced at any time.

Also the functionality of the agents should be generic and could be anything that may be described as '*requesting a service*' or '*providing a service*'; **ad-hoc service composition** based on emergent structures is the objective. These emergent structure shall have many properties of living systems or self-* systems. They may spontaneously emerge from the local interactions of the agents, and may or may not disassemble themselves after the task at hand has been fulfilled. In some cases, there may be an incentive for idle agents to participate in other emerging structures, and in other cases, the agents may prefer to stay where they are and wait for a certain time.

The ultimate challenge is to find a way in which the assembling and dis-assembling systems retain their history, which would enable the agent societies to learn from past experience. The difficulty lies in the fact that a self-* system will have only very limited global knowledge, no central commander, and no system-wide building plans. This means that knowledge may be preserved within the individual agents or implicitly within the agent society, possibly in the form of norms or policies.

Based on this idea, we started preliminary simulations with the intention to determine which type of agents, which capabilities and characteristics, and which type of rules provide us with the most interesting results. In this quest, contributions from other research groups would be highly welcome.

5 EXPERIMENTS

Subsection 5.1 explains the basic features of the emergent structures simulation; subsection 5.2 introduces structure functionalities, and subsection 5.3 introduces rules for self-optimisation and self-repair.

5.1 Emerging bridge structures – basic features

The goal of the conducted experiments was to show how complex structures – for instance 'bridges' which are strings of functional connections between agents – can emerge on the basis of mostly local rules. For this purpose, a multi-agent system consisting of static and mobile agents was created in Presage, as follows.

Each of the static agents ("seeds") has a unique token represented by a letter (A, B, C, ...); the mobile agents ("cells") move randomly under certain conditions and according to certain behaviour rules connect to both peers and seeds. When connecting, the tokens carried by each agent are copied to the other agent[1]. For instance, cell 1 which already carries token A may connect with cell 2 that carries B; consequently, token B is copied to cell 1 and token A is copied to cell 2. When a seed receives a foreign token, it recognises that a bridge to another seed must have been established, as illustrated in Figure 1.

[1]Concretely, each agent implicitly gains the tokens held by those peers it connects with. Holding a token therefore means being connected to a seed with that token. Upon disconnection, this information disappears.

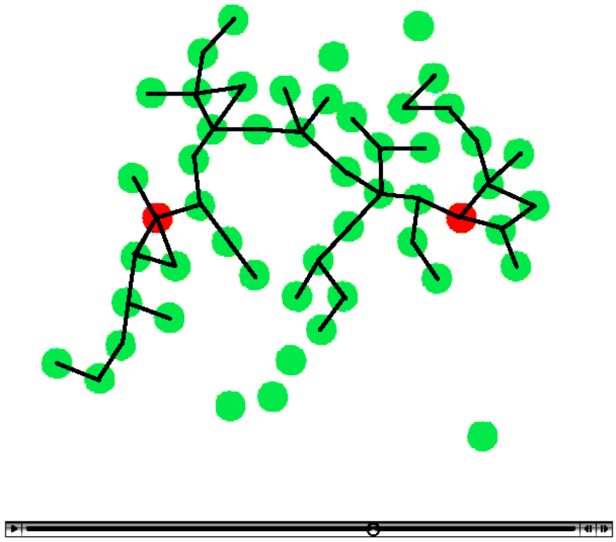

Figure 1: Construction of an emergent bridge. The dark dots are static seeds and the lighter dots are (previously) mobile cells. Strings represent established connections.

Presage also provides a dynamically updated tree structure of the established connections, as shown in Figure 2. We use a *breadth first tree search* algorithm, which skips repeated nodes. Therefore, the algorithm will always find the shortest path to each node and will ignore loops in the tree.

All cells have a certain probability to make connections (attractive force), and a certain probability to dissolve connections (repulsive force). Additionally, there is a time-out for unsuccessful strands: once connections have existed for over a certain time without completing a bridge, the cells will dissolve the connections and continue to roam randomly.

The subsequently listed rules and metadata are necessary at this stage of the simulations.

Cell rules:

- Move randomly.
- Connect to peers and seeds if they do not hold identical token(s).
- Copy token(s) of connected peers or seeds.
- Dissolve connections when time-out expired without success.
- Remove tokens received from peer when connection dissolved.

Seed rules:

- Connect to cells if they do not hold identical token(s).
- Copy token(s) of connected cells.
- Remove tokens received from cell when connection dissolved.

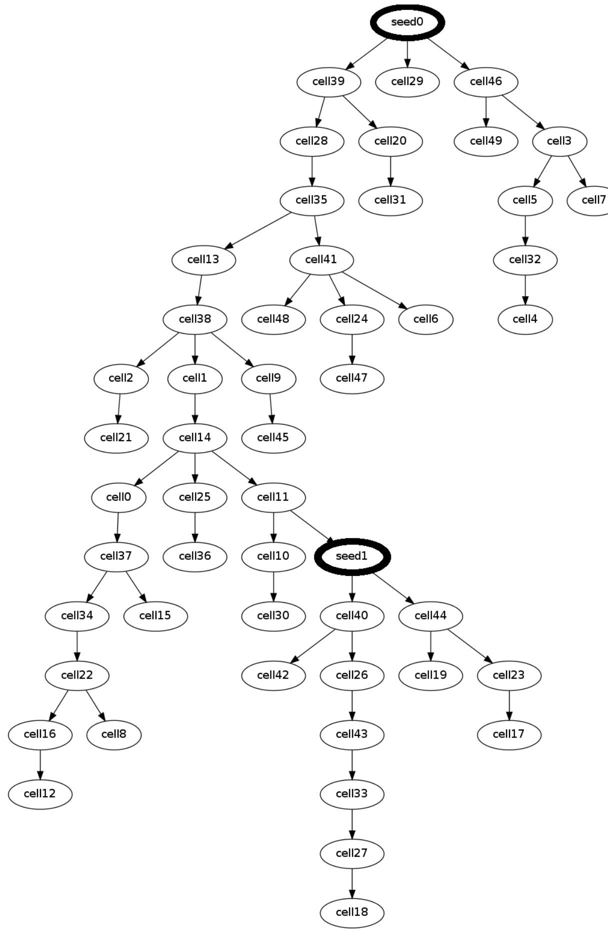

Figure 2: Tree structure of the bridge shown in Figure 1

Metadata of each cell and each seed:

- Established and dissolved connections
- Time to success
- Probability to connect and its development over time
- Probability to dissolve connections and its development over time
- Number of tokens held over time

5.2 Functional structures – ongoing work

As previously mentioned, the ultimate goal of these simulations is to lay the foundations for creating emergent functional structures in a physical environment. These functions may be based on services provided by pure software agents or agentified physical devices.

As the bridges are instances of *functional* structures, the seeds have a goal which uses the bridges and wants to maintain them - for instance, periodically transmitting something (for instance, a packet) to another seed. Whenever a transmission is successful, the cells on the direct connection receive a reward, and those in second grade receive a minor reward because they provide increased stability to the bridge. The cells in the second grade are then in a game theoretical dilemma: should they stay and receive minor rewards or should they move and try to be a first grade member of a new bridge?

The cells which participate in building a successful bridge will receive a reward based on functionality; that is, each time a packet is transmitted from one seed to the other over the corresponding bridge, the cells receive a reward. Rewards are accumulated and cells with high amounts of rewards are ‘happy’ and thus have a higher probability to continue in their current state / configuration.

Additional cell rules:

- When receiving a packet, transmit it to the other connected agent(s).
- After receiving a reward, decrease probability of disconnection.
- When receiving a reward, transmit it to the other connected agent(s).

Additional seed rules:

- When receiving a foreign token (we assume: a bridge has been established), send a packet over the bridge in regular intervals.
- Upon receiving a packet, send a reward to the delivering cell.

Additional metadata of each cell and each seed:

- Packets sent and received over time
- Rewards received over time
- Number of cells in first grade and its development over time
- Number of cells in second grade and its development over time

5.3 Self-optimising and self-repairing bridges: future work

The probabilities of connecting and dissolving can be dynamically adjusted by the agents themselves, for instance through learning. Cells record the time between the beginning of the game or the last time they dissolved their connections and the next success, as well as their connection and disconnection probabilities since then. Correlating the probabilities for connection and dissolution with their time to success, the agents will be able to optimise their probability parameters.

Cell also have a behaviour which allows them to shorten the bridge they participate in; in other words, the bridge self-optimises. Key to this behaviour will be the connection tree and the cells’ grade. Cells in the second grade will attract new peers and test if they might be able to establish a shorter path to the seeds. If this is the case, the new structure will replace the old one; otherwise, the additionally attracted cells will be released.

To demonstrate the ability of the bridge to self-repair, the cells also have a certain probability to die and ‘disappear’. A dead cell will leave a gap in the strand it belonged to. Its neighbours will try to reconnect and close the gap, potentially by attracting new peers or by moving themselves towards the gap without disconnecting existing links.

The additional rules listed subsequently are preliminary; they will be refined once the simulations have reached this phase.

Additional cell rules:

- (1) Every cell calculates the number of cells to next seed.
- Cells in the second grade attract new peers and do (1).
- If the probability of a cell to die reaches the value of 1, the cell is deleted from the system.
- If a peer dies, the neighbouring cells try to close gap by moving closer towards the gap and connecting with any cell nearby, or by attracting new peers.

Additional metadata of each cell:

- Number of cells to next seed over time and its development over time
- Probability to die and its development over time
- Strategies used to repair a gap and their successes over time

6 Formal model of the complex emergent structures system

The system is composed of a set of agents which belong to either static seeds or mobile cells. Each agent has a location and may have a number of tokens.

Set of cells C; Set of seeds S. $C \cap S = \phi$

Set of possible positions within the system A.

Set if possible tokens T.

Each cell and seed $k \in C \cup S$ has a location at time t given by $L_k(t)$.

$$\forall k \forall t : L_k(t) \in A, k \in C \cup S \quad (1)$$

$$\forall l \forall t : L_l(t) = L_l(t+1), l \in S \quad (2)$$

Each cell's position is limited by it's movement speed V_j.

$$\forall j \in C : |L_j(t+1) - L_j(t)| \leq V_j \quad (3)$$

Each seed $i \in S$ has set of tokens $T_i \subseteq T$.

Each cell $j \in C$ may connect with any cell or seed $k \in C \cup S$ within a range R_j at time t.

$$|L_j(t) - L_k(t)| \leq min(R_j, R_k) \quad (4)$$

For a connection made at time t' the connection remains until a time $t' \geq t'$ given that the above condition is not broken at any time t where $t' \leq t \leq t''$.

For each cell $j \in C$ it's set of tokens at time t, $T_j(t) \subseteq T$ is the union of the all the tokens sets of it's connected neighbours $N_j(t) \subseteq C \cup S$ at that time.

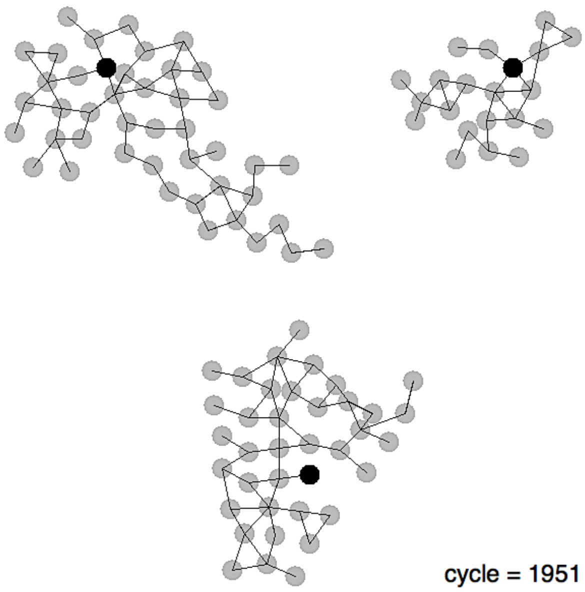

Figure 3: Initial cell "clumps" around seeds

$$T_j(t) := \cup_{m \in N_{jt}} T_m(t) \quad (5)$$

This basic model serves as the basis for a more comprehensive model that is currently being formulated.

7 RESULTS AND DISCUSSION

As the implementation and experiments are ongoing, we only present initial results at this stage. The preliminary experiments with the system described in section 5 were intended to test whether, using purely the basic rules described previously, functional structures would emerge and what the nature of their formation would be.

The experiments were based on an environment with three seeds evenly distributed in the considered area, and focused on two cases: one with 100 cells and the other with 200. We investigated how certain static values of the probabilities of connection and disconnection affected the performance of the system.

As the cells only follow a very basic set of local rules, as a collective they form "clumps" around each seed, as Figure 3 shows. To build a functional bridge a cell must make a connection in the gap between two "clumps" when they are close enough together, as pictured in Figure 4. Therefore, at this stage the structures are largely random occurrences. Rules to modify this behaviour are currently being elaborated.

Investigating how many bridges are created in the described scenarios, given different connection probabilities of the cells in the system, we see in Figure 5 that the range from 0.6 to 1 seems to lead to more bridges, with a peak at 0.6 connection probability. The probability of disconnection was set to 0

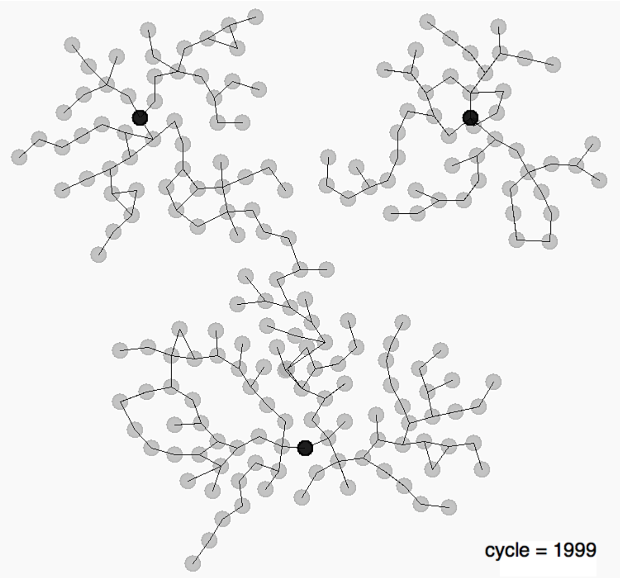

Figure 4: Successful establishment of a bridge between two seeds

for these simulations. Interestingly, the results vary a great deal between repeat runs of the experiments. With 20 repetitions run for every value, we conclude that randomness is still a large factor in comparison to the connection probability.

Any increase in the probability of disconnection leads to less bridges, or they take longer to form. Values in the range of 0.0001 to 0.0512 (logarithmically) lead to no bridges at all within the 2000 considered simulation steps; other disconnection probability values have not been tested yet. 2000 steps seemed enough that in every case there was a very high chance of every cell connecting.

Notice that a static disconnection probability is slightly flawed as a rule, as it will break up useful structures, too. A more elaborate rule for having an increased probability of disconnection in certain situations only will be introduced presently.

Figure 6 illustrates how quickly bridges were established, again with a probability of disconnection equal to zero, and diverse values of connection probabilities. Unsurprisingly, the greater chance the cells have on connecting, the faster they can form bridges.

In conclusion, the current findings of these experiments confirm that the simple entities (cells) used in these simulations are able to build emergent structures based on simple local rules and local knowledge, without any central control or building plan. They are thus a suitable basis for further research towards more sophisticated self-* and emergent properties. The ongoing work particularly investigates how the heterogeneity of the cells, that is, agents, robots, devices, humans, institutions, etc., may be represented in a generic yet meaningful way, and how their different interaction characteristics will influence the system level behaviour.

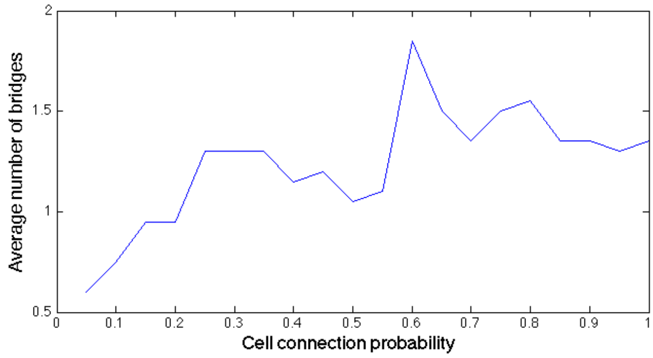

Figure 5: Average number of bridges formed

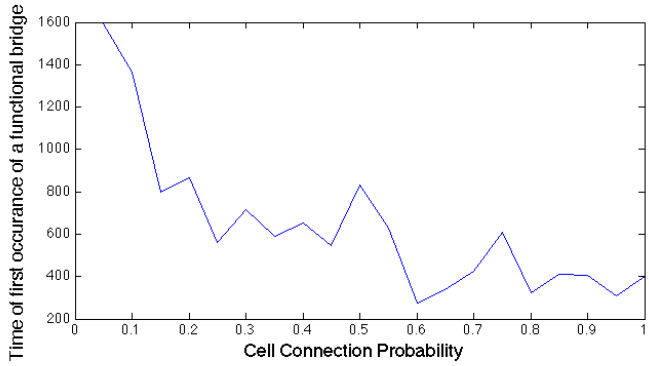

Figure 6: Average time taken for the formation of a functional bridge

8 RELATED WORK

Over several decades, *self-reinforcing structures* and phenomena have been studied in various contexts, such as social systems [17], organisations [27], child psychology [1], population migration [19], business administration (power and status) [18], animal behaviour [12] and the extension of the human life span [2].

In the context of ecology and economy, self-reinforcing and self-organising structures have been observed across various levels and scales [22], often displaying a hierarchy of systems that span food chains, communities and landscapes / regions. For instance, both real world experience and models indicate that processes such as hydrology and the propagation of disturbance can be strongly self-reinforcing (i.e. the landscape structure supports the process, and vice versa).

In multi-agent systems, the agreement of conventions may be hindered by self-reinforcing structures [26]. Social tools such as observation, rewiring based on previous success, and particularly their combination, allow the MAS to overcome these difficulties.

A way to decompose software engineering methodologies and recompose them while introducing elements of self-organisation is presented in [23]. A variety of existing methodologies for self-organisation is discussed, explaining their characteristic features, advantages and limitations. Using the SPEM (Software & System Process Engineering Meta-Model) approach, method fragments can be combined in many different ways according to the designers wish and the desired self-organisation mechanism.

An overview on concepts, principles and mechanisms used in *complexity engineering* is provided in [9], together with a survey on existing work [8]. Most of the existing literature refers to software systems; work on socio-technical, robotic or mechanical systems remains rather rare. Complexity engineering is closely related to *emergent engineering* [25], which aims at solutions that can be selected through an evolutionary adaptation process to produce progressively better (and continuously improving) solutions.

Morphogenetic systems provide us with powerful mechanisms for systems design and architecture. Just in the same way as biological systems grow, expand and diversify their bodies and functions, also engineered systems and structures may rely on on local mechanisms for growing, expanding and diversifying [5].

The principles of *degeneracy – functional redundancy and function plasticity* – are frequently observed in natural complex adaptive systems, and they also apply to engineering [11]. Functional redundancy means that a system has several different components which can fulfil the same function, whereas functional plasticity refers to components which can execute several different functions.

Autonomic Computing [14] was conceived because complex technological systems must start to take care of themselves, to hide complexity from the user, and to offer intuitively understandable services. Different types of policies are used to guide the system [16, 15]: action-based policies, goal-based policies and utility-function- based polices. Policies are a form of guidance used to determine decisions and actions. Hence, analogously, *reflex-based / action-based* agents use if-then action rules. *Goal-based* agents can determine which actions to take to achieve specified goals. Finally, *utility-function-based* agents choose the actions to maximise their utility function, which allows a finer distinction between the desirability of different states than goals do. The implementation of such policies can take various forms, if (...) then (...) else being the most intuitive one (but limited to relatively simple applications).

Organic computing (OC) aims at a favourable trade-off between autonomy and controllability, which machines may never lose; i.e. freedom and security. A similar trade-off is required between robustness versus sensitivity [28]. One of the main goals of OC is to find ways of controlling emergence [20]. OC is oriented towards the user's needs: to be more user-friendly, systems are designed to be *life-like*. They dynamically adapt to the changing environment. To achieve this, OC previews the *controller/observer architecture* [24], which is very generic. It consists of three layers: a top-down layer (high level) with reasoning, simulations, and observation capabilities, which can give feedback; a middle layer; and a bottom-up layer (low level) with reflexes. The low level assertions are similar to action policies (if-then rules). The counterpart in the system which receives violation messages is the observer; the one which takes measures accordingly is the controller [20]. When the observer decides that reconfiguration is necessary, the controller will execute it. Notice that MAPE, from autonomic computing, can be considered as a type of observer/controller architecture.

9 CONCLUSION

In nature, including biological and social systems, complexity often goes hand-in-hand with adaptivity, flexibility, evolvabilty and emergence. Systems with such characteristics can cope with changing conditions, failing components and other perturbations. Engineers strive to build systems with similar intrinsic features, and thus study the principles and mechanisms found in nature.

Simple experiments as presented in this article contribute to the demonstration of self-* properties in multi-agent systems that are guided by rules and policies.

Further experiments will be done in Presage to investigate the potential of dynamically changeable policies. Several learning algorithms will be implemented, and ways to preserve history in the system even after the dissolution of formed structures.

ACKNOWLEDGEMENT

At the time when this work was done, Regina Frei was receiving a fellowship for prospective researchers from the Swiss National Science Foundation.

REFERENCES

[1] C. Ames. Children's achievement attributions and self-reinforcement: Effects of self-concept and competitive reward structure. *J. of Educational Psychology*, 70(3):345–355, 1978.

[2] J.R. Carey and D.S. Judge. Life span extension in humans is self-reinforcing: A general theory of longevity. *Population and Development Review*, 27(3):411–436, 2001.

[3] D.M. Chess, A. Segal, I. Whalley, and S.R. White. Unity: Experiences with a prototype autonomic computing system. In *Int. Conf. on Autonomic Computing (ICAC)*, pages 140–147, New York, USA, 2004.

[4] G. Di Marzo Serugendo, J. Fitzgerald, and A. Romanovsky. MetaSelf - an architecture and

development method for dependable self-* systems. In *Symp. on Applied Computing (SAC)*, pages 457–461, Sion, Switzerland, 2010.

[5] R. Doursat. Programmable architectures that are complex and self-organised: From morphogenesis to engineering. In S. Bullock, J. Noble, R. Watson, and M. Bedau, editors, *11th Int. Conf. on the Synthesis and Simulation of Living Systems*, pages 181–188. MIT Press, 2008.

[6] J.L. Fernandez-Marquez, G. Di Marzo Serugendo, S. Montagna, M. Viroli, and J.L. Arcos. Description and composition of bio-inspired design patterns: a complete overview. *Accepted for publication in Journal of Natural Computing*, 2012.

[7] R. Frei. *Self-organisation in Evolvable Assembly Systems*. PhD thesis, Department of Electrical Engineering, Faculty of Science and Technology, Universidade Nova de Lisboa, Portugal, 2010.

[8] R. Frei and G. Di Marzo Serugendo. Advances in complexity engineering. *Int. J. of Bio-Inspired Computation*, 3(4):199–212, 2011.

[9] R. Frei and G. Di Marzo Serugendo. Concepts in complexity engineering. *Int. J. of Bio-Inspired Computation*, 3(2):123–139, 2011.

[10] R. Frei and G. Di Marzo Serugendo. Self-organising assembly systems. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 41(6):885–897, November 2011.

[11] R. Frei and J. Whitacre. Degeneracy and networked buffering: principles for supporting emergent evolvability in agile manufacturing systems. *Accepted for publication in Journal of Natural Computing - Special issue on Engineering Emergence*, 2011.

[12] Charlotte K. Hemelrijk. Towards the integration of social dominance and spatial structure. *Animal Behaviour*, 59(5):1035–1048, 2000.

[13] D.H. Jones, R. McWilliam, and A. Purvis. Design of self-assembling, self-repairing 3d irregular cellular automata. In A. Lazinica, editor, *New Advanced Technologies*, pages 373–394. InTech, 2010.

[14] J.O. Kephart and D.M. Chess. The vision of autonomic computing. *IEEE Computer*, 36(1):41–50, 2003.

[15] J.O. Kephart and R. Das. Achieving self-management via utility functions. *IEEE Internet Computing*, 11(1):40–48, 2007.

[16] J.O. Kephart and W.E. Walsh. An artificial intelligence perspective on autonomic computing policies. In *Proc. 5th IEEE Int. Workshop on Policies for Distributed Systems and Networks (POLICY)*, pages 3– 12, New York, USA, 2004.

[17] P. Knapp. Path dependence in self-maintaining and self-reinforcing structures. In *Annual meeting of the American Sociological Association, TBA*, New York, USA, 2007.

[18] J. Magee and A. Galinsky. Social hierarchy: The self-reinforcing nature of power and status. *The Academy of Management Annals*, 2(791784088):351–398, 2088.

[19] D.S. Massey. Social structure, household strategies, and the cumulative causation of migration. *Population Index*, 56(1):3–26, 1990.

[20] C. Mueller-Schloer and B. Sick. Controlled emergence and self-organization. In R.P. Wuertz, editor, *Organic Computing, Understanding Complex Systems*, pages 81–104. Springer Berlin Heidelberg, 2008.

[21] B. Neville and J. Pitt. PRESAGE: A programming environment for the simulation of agent societies. In K. Hindriks, A. Pokahr, and S. Sardina, editors, *Programming Multi-Agent Systems*, volume 5442 of *LNCS*, pages 88–103. Springer Berlin / Heidelberg, 2009.

[22] D.A. Perry. Self-organizing systems across scales. *Trends in Ecology & Evolution*, 10(6):241–244, 1995.

[23] M. Puviani, G. Di Marzo Serugendo, R. Frei, and G. Cabri. A method fragments approach to methodologies for engineering self-organising systems. *Accepted for publication in ACM Transactions on Autonomous and Adaptive Systems (TAAS). JCR 5-year impact factor in 2011: 1.645*, 2011.

[24] T. Schoeler and C. Mueller-Schloer. An observer/controller architecture for adaptive reconfigurable stacks. In *Int. Conf. on Architecture of Computing Systems (ARCS)*, pages 139–153, Innsbruck, Austria, 2005.

[25] M. Ulieru and R. Doursat. Emergent engineering: A radical paradigm shift. *J. of Autonomous and Adaptive Communications Systems*, 4(1):39–60, 2011.

[26] D. Villatoro, J. Sabater-Mir, and S. Sen. Social instruments for convention emergence (extended abstract). In Tumer, Yolum, Sonenberg, and Stone, editors, *10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1161–1162, Taipei, Taiwan, 2011.

[27] K E Weick. *Cognitive Processes in Organizations*, volume 1, pages 41–74. JAI Press, 1979.

[28] R. Wuertz, editor. *Organic computing*. Understanding Complex Systems. Springer, Berlin Heidelberg, 2008.