A Comparison of Self-Organization Mechanisms in Nature and Information Technology

André Miede, Steffen Braun, Julian Eckert, Dieter Schuller, Nicolas Repp, Ralf Steinmetz
Multimedia Communications Lab (KOM)
Technische Universität Darmstadt, Germany
Corresponding Author's E-Mail: Andre.Miede@kom.tu-darmstadt.de

ABSTRACT

Successful concepts of self-organization found in natural systems can enable enterprise information systems to address their complexity issues. In this paper, we propose an analysis of self-organization approaches found in natural sciences and information technology. Based on common classes both for application areas and mechanisms, these two fields are compared in order to identify successful concepts, which can be used for the adaptation in information systems research. For illustration purposes, we give a brief example for self-organization in the domain of Service-oriented Architectures, i.e., cooperation mechanisms for agents monitoring services.

Keywords

Self-Organization, Enterprise Application Architecture, Service-oriented Architectures

INTRODUCTION

Technological advancements and globalization are major drivers for enterprises in the modern world. Enterprises face constant challenges to react to rapidly changing market requirements in order to stay competitive. An important factor to achieve this goal is the underlying enterprise information technology (IT). It has to provide solutions to model, operate, and adapt business processes efficiently and effectively. In the last years, the integration of both company-wide and inter-company information systems (IS) has emerged as a big challenge in this context (Krafzig, Banke, and Slama, 2004; Melzer, 2007). Often, enterprise information systems have not been designed and developed as a holistic concept but they have rather grown in a heterogeneous organic fashion over the years. For example, this leads to vertically organized enterprise IT architectures (cf. Figure 1), where each application has its own database and has implemented its own functions for access and modifications. This scenario causes serious problems when changing things at the process-level, due to its high amount of redundant data and code.

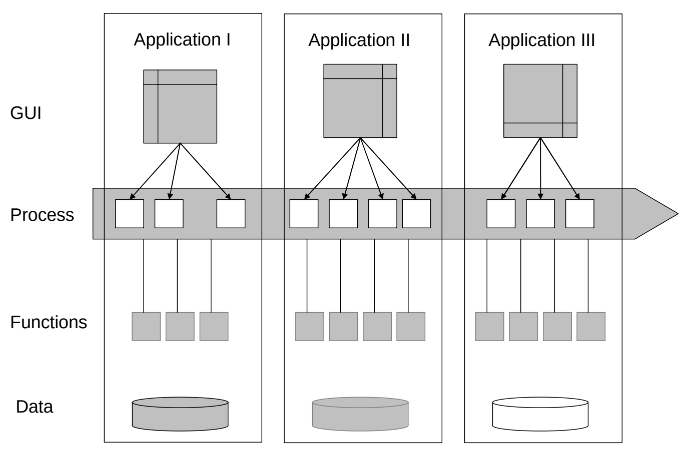

Figure 1: Vertically organized IT landscape (after Melzer (2007))

Paradigms such as Service-oriented Architectures (Papazoglou, 2003) offer technological and organizational possibilities to evolve towards a more horizontally organized IT landscape and, thus, to improve the alignment between the business and the IT side.

An important aspect in this context is the concept of Quality-of-Service (QoS), which serves as an umbrella for non-functional requirements of information systems such as performance, security, availability, scalability and so on (Gouscos, Kalikakis, and Georgiadis, 2003). The corresponding QoS-parameters have to be monitored and managed in order to operate the enterprise successfully. Due to the high business dynamics, this monitoring and management is a very difficult task and, therefore, very expensive and often too complex to be achieved manually (Hermann, Mühl, and Geihs, 2004). A possible solution is to reduce human intervention in these tasks and to strive for the property of *self-organization* which can be found in many complex natural systems.

The goal of this paper is to provide IS researchers with an analysis of self-organization's application scenarios both in IT and other academic disciplines. This is done in order to identify successful concepts and to lay the foundation for adapting these concepts to current and future research in IS.

The rest of this paper is structured as follows. In the next section we give an overview of related work in order to introduce the basic concepts. The successive section describes the methodology and goals of our preliminary analysis which classifies application areas and mechanisms. Thereafter, the results from analyzing self-organization in nature and IT are presented both separately and in comparison. In order to illustrate the importance and applicability of our results, we give an overview of applying self-organization concepts in the form of a cooperation model for monitoring agents for Service-oriented Architectures. The paper closes with conclusions and an outlook on future research.

RELATED WORK – THE CONCEPT OF SELF-ORGANIZATION

This section serves as the foundation for the paper's main concepts. It provides a definition of basic terms such as self-organization and gives an overview of typical so-called "self-X" properties.

Definition

The term „self-organization" is used in a variety of academic disciplines, which makes it very hard to find a general and comprehensive definition. In our previous work (Miede, Repp, Eckert, and Steinmetz, 2008), we have given the following definition of self-organization for the use in IS and a technological context in general, which follows definitions by Capra (1997), Goldstein (1999), Camazine, Deneubourg, Franks, Sneyd, Theraulaz, and Bonabeau (2003), and Bonabeau, Dorigo, and Theraulaz (1999):

Self-organization is the capability of a system to spontaneously create new structures and forms of behavior on the global level through internal interactions of its lower-level components and without external intervention.

Figure 2 visualizes the relationship between the major self-organization concepts, i.e., using special mechanisms to transform a system from one state into another to achieve a particular emergent behavior.

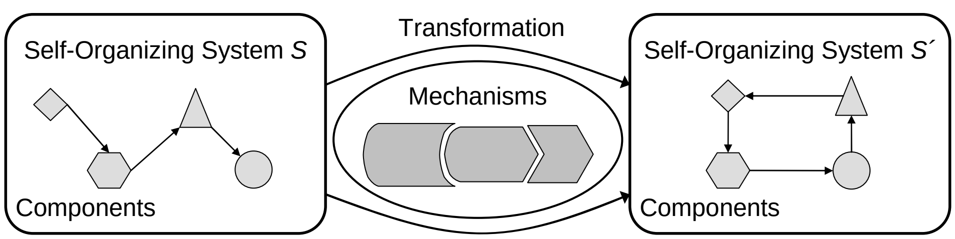

Figure 2: Relationship of self-organization concepts (Miede et al., 2008)

An introduction to self-organization and related concepts can be found in the works of the pioneers of this field, e.g., Kauffman (1996), Camazine et al. (2003), Holland (1996), and Bonabeau et al. (1999).

Self-Properties

There is a variety of subclasses and specialized capabilities which are generally referred to as "self-*" or "self-X" (Hinchey and Sterrit 2006). Four variants initially introduced by IBM (2006), which are known as "CHOP" are the most famous among many other possibilities for "X" and "*":

- self-*c*onfiguration (adapt to changes in the system)
- self-*h*ealing (recover from detected errors)
- self-*o*ptimizing (improve the use of resources)

- self-*p*rotecting (anticipate and cure intrusions)

In this area, even some meta-research has been conducted, e.g., Fromm and Zapf (2005) assembled and approximated the occurrences of self-X terms in papers and conferences (cf. Figure 3).

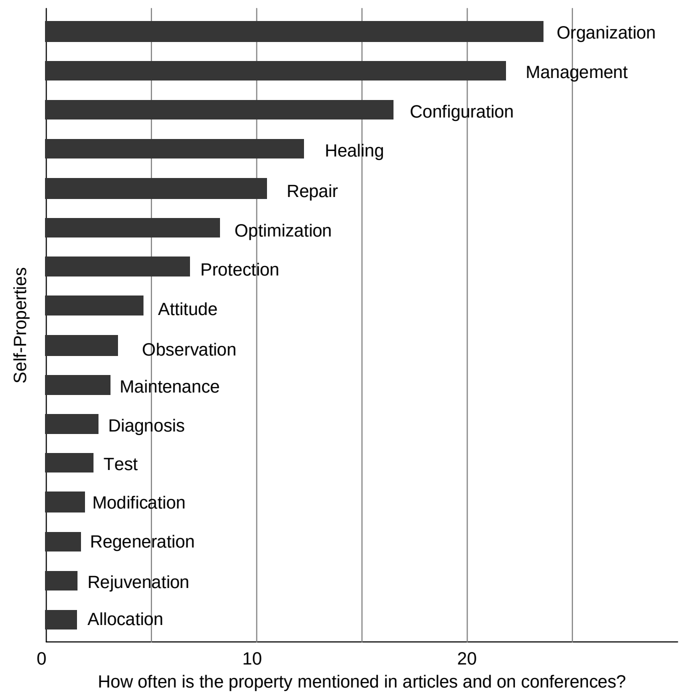

Figure 3: Frequency of self-properties in scientific literature (after Fromm and Zapf (2005))

The two most frequently occurring variants are self-organization and self-management, which are be used synonymously and which are rather general terms containing a multitude of the other self-X properties as subclasses.

However, one of the next logical steps, i.e., to compare the classification and application of the mechanisms has not been performed yet.

PRELIMINARY ANALYSIS

In this section, the results of our preliminary analysis are presented, i.e., the classification of self-organization mechanisms and the identification of common application areas for them. This analysis is the foundation for the main part of our paper: the comparison of self-organization in natural sciences and information technology, which is subject of the next section. Before we discuss the results, we give an overview of the goals of the analysis and methodology used.

Goals

In scientific literature, there is a multitude of examples for self-organizing systems both in natural sciences and in IT. Our analysis is focused on the following questions:

- Which mechanisms are used to achieve self-X properties and how can we classify and structure them?
- Which are the common application areas for the mechanisms?

Based on these questions, this paper gives a first answer to the question whether there are any new ideas to be learned from the comparison of self-organization in natural sciences and IT.

Methodology

To answer the above questions, we used the following iterative approach, which can be performed ad infinitum and which can, therefore, be the basis for further research in this field (cf. Figure 4).

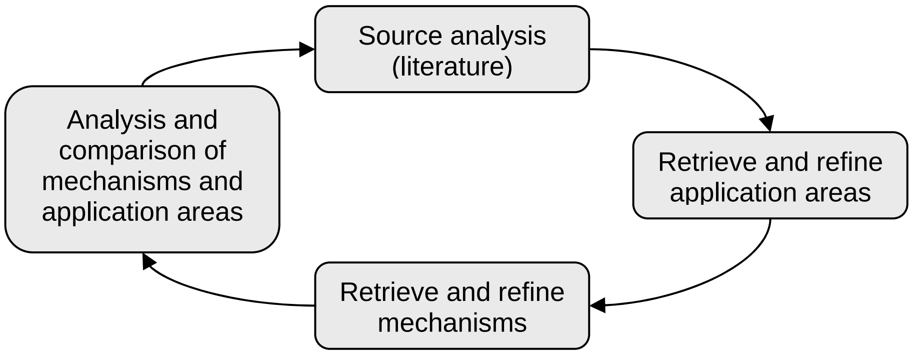

Figure 4: Analysis and comparison methodology

A comparison of self-organization in natural sciences and information technology can be facilitated by identifying and defining application areas, which can be found in both areas – any approaches examined subsequently either fit into the identified application areas or lead to new ones. After that, we have examined the approaches concerning the self-organization mechanisms they use. From this, we assembled classes for the found mechanisms, where mechanisms either fit into the already identified classes or extended the list of classes. Both, the application areas and the mechanism classes lead to a matrix, which is the foundation and choice of presentation for the comparison (cf. Table 1 and 2 later on).

In the following, we briefly outline the findings regarding the identified application areas and mechanisms for self-organization.

Application Areas

Self-organization mechanisms can be found in different areas of nature and science. For our comparison, we structure and classify the application areas. De Wolf and Holvoet (2006) describe a variety of application areas. Building on this portfolio and using three meta application areas for organization purposes, we propose the following classification:

- Collective coordination:
  - Spatial movement
  - Source-drain routing
  - Pattern and structure formation
  - Recruitment for tasks
  - Team formation (avoiding egoistic behavior)
  - Specialized skills within a team
- Information
  - Integration of information
  - Integration of contextual information
  - Information about resources
  - Assessment of resources
- Decentralized control
  - Local control
  - Trust and security
  - Control of access to resources
  - Allocation of resources

This classification is not concluding but a first starting point, which incorporates all our examined self-organization approaches.

Mechanisms

The following is a structured summary of the self-organization mechanisms we found during our analysis. Both the list and the classes are not concluding but an attempt towards organizing the field:

- *System behavior* (i.e., realizing non-linear behavior):
    - *Positive feedback and negative feedback*: actions of system elements can have an amplifying or alleviative impact on the system. Examples are pheromones left by ants during food foraging.
    - *Random processes and dynamic instability*: the system's equilibrium is deliberately disturbed to allow for a new and probably better equilibrium.
    - *Threshold*: the system's state is changed if a certain parameter of the system reaches a particular value.
- *Individual behavior* (i.e., of the elements of the system)
    - *Differentiation and stem cells*: different elements of the system arise from the same basic elements, whose development is not pre-determined. This is a concept to be found in cells of humans, animals, and plants.
    - *Replication*: the exact duplication of elements and their information, e.g., as genetic information in DNA.
    - *Mutation and cross breeding*: random or deliberate interventions can cause deviations from the exact replication of elements, leading to new and atypical properties.
- *Collective behavior* (i.e., between the elements of the system to achieve emergence)
    - *Cooperation and rivalry*: achieving common or egoistic goals through individual actions of the system's elements.
    - *Local aggregation*: clustering of system elements to increase the density, i.e., to assemble patterns.
    - *Building blocks*: assembly of multiple re-usable, basic elements to achieve a more sophisticated functionality.
    - *Natural selection*: elements considered „fitter" according to specialized metrics are selected for survival while others are not.
- *Communication* (i.e., between the elements of the system for coordination purposes)
    - *Direct communication*: from element to element.
    - *Pheromones*: volatile, chemical substances are used to mark the environment and are recognized by other elements of the system.
    - *Gradients*: a difference in the concentration of a substance, e.g., to describe a path for movements. The difference to pheromones is the non-volatility of gradients which have to be removed explicitly.
    - *Stigmergy*: elements of the system communicate by modifying their environment. An example for this is the nest building process of many insects.
    - *Tags*: elements attach special signals to other elements, e.g., to mark them for a particular treatment.
- *Control* (i.e., to influence the decisions made by the system's elements)
    - *Local monitoring*: elements monitor other elements in their direct neighborhood regarding their properties or behavior, e.g., like cells of the human immune system.
    - *Token*: elements share special objects which grant the respective owner special rights, the passing of the objects is determined by a dedicated algorithm. This mechanism is found communication networks, for example.
    - *Assessments*: properties or behavior of elements is assessed and the result influences the behavior of the other elements in the system, e.g., regarding communication or cooperation.

- *Contracts*: rights and responsibilities have a high impact on the behavior of the system's elements, i.e., like in different markets of our human society.
- *Templates*: in order to replicate a certain setup or configuration, templates are used.
- *Checkpoints*: particular activities or decisions can only be performed if other activities have been finished. These can be seen as checkpoints for the whole process and are mandatory for synchronizing and aligning different actors.

It is important to understand in this context that not all mechanisms for self-organization are known yet and that not all known are completely understood (Kube and Bonabeau, 2000; Camazine et al., 2003). In addition, a clear separation of mechanisms is often not possible, e.g., regarding pheromones, gradients, and stigmergy. It is also often difficult to assign a mechanism strictly to one category, e.g., tags can be used for communication but also for control purposes.

However, in spite of these minor difficulties, the current setup of our classification is in accordance with the majority of the examined approaches.

A COMPARISON OF SELF-ORGANIZATION MECHANISMS IN NATURE AND INFORMATION TECHNOLOGY

The comparison of self-organization mechanisms in nature and IT is based on how mechanisms are applied within the identified application areas. This aims at providing insight into questions such as, which mechanisms are particularly successful and whether there might be a correlation between special application areas and mechanisms. As a first step to answer these questions, we counted how often the mechanism classes occurred in the examined literature. The results for this are shown in Table 1 and 2. In these tables, a number denotes the occurrences of a mechanism in an application area and is increased for each one found. An assignment of the application area is often not precisely possible, thus, a mechanism can be assigned to more than one application area. Furthermore, if an application area could not be assigned at all, it is left out of the table in order to increase clarity.

The selection of data consists of 24 approaches for self-organization from nature and natural sciences and 44 approaches from information technology. As a complete listing of all examined sources would be beyond the allowed space for this paper, we list these sources in an appendix, which is available from the corresponding author upon request or from the authors' website noted at the end of this paper. This selection of examined literature is just a beginning as its significance is limited and the number of sources has to be increased in future investigations. In addition, the chosen approaches are not necessarily part of a representative sample. However, an educated guess about general trends and tendencies can be made.

Comparison

We found the following main differences between self-organization in nature and IT in our investigations.

Not all application areas could be found in natural systems and in information technology. The main differences can be shown by means of two special application areas:

- Self-organizing systems in IT seem to consider application areas concerning *resources* more relevant than in natural systems. For example, "information about resources", "assessment of resources", and "control of access to resources" were not found in the natural sciences sample. In addition, "allocation of resources" is the second-most-important application area for the IT sample. A reason for this could be that the scarcity of resources in the observed natural systems is less a problem than in the artificial systems. Looking at the ant foraging process, for example, self-organization focuses not on the food itself, but rather on the way to the food source. On the other hand, network resources are subject to bandwidth limits and traffic costs. This raises the question what mechanisms to use, if the fundamental problem does not exist in this form in natural systems and if, therefore, no natural solution is available as inspiration. Our analysis showed that IT then makes use of mechanisms, which are not wide-spread or not used at all in nature, i.e., contracts, assessments, tokens, or direct communication. Here, IT developed or adapted solutions on its own, which were not directly inspired by natural mechanisms.

Mechanism Classes	Application Areas	*Collective coordination*					*Information*		*Decentralized control*		Weighed sum
		Spatial movement	Source-drain routing	Pattern and structure formation	Recruitment for tasks	Specialized skills within a team	Integration of information	Integration of contextual information	Trust and security	Allocation of resources	
	Number of examples	5	1	8	2	1	1	2	2	2	24
System behavior	Positive feedback and negative feedback	1	1	2	1		1	1			3,45
	Random processes and dynamic instability			1						1	0,625
	Threshold			3	1			1			1,375
Individual behavior	Differentiation and stem cells			1		1	1				2,125
	Replication										0
	Mutation and cross breeding								1		0,5
Collective behavior	Cooperation and rivalry	1		4	2					1	2,2
	Local aggregation			5							0,625
	Building blocks										0
	Natural selection								1		0,5
Communication	Direct communication			1	1						0,625
	Pheromones	3	1	1	1		1				3,225
	Gradients	3		3				2	1		2,475
	Stigmergy	1		2	1						0,95
	Tags					1			1		1,5
Control	Local monitoring			1					1		0,625
	Token										0
	Assessments									1	0,5
	Contracts									1	0,5
	Templates										0
	Checkpoints			1					1		0,625

Table 1. Results for self-organization in nature and natural sciences

- The second special application area is *local control*, which does appear in our nature sample, but which is a dominant area in our IT sample, using the local monitoring mechanism. This seems to be due to a fundamental difference between self-organization in nature and IT: in IT, autonomous systems are often created for non-autonomous components (Calinescu, 2007), whereas natural systems are rather based on autonomous components. While the application area "local control" includes mainly approaches to implement autonomous behavior in a system, self-organizing natural systems do not use such control elements, because each element of the systems has itself the capability to act autonomously and to create emergent behavior. Investigating this issue can be seen as an important research task in order to clarify, which way for achieving control should be followed further.

The most important mechanisms in our nature sample are *feedback*, *pheromones*, and *gradients*. Non-linearity is present in nearly all applications, while pheromones and gradients are the major communication means.

In our IT sample, the most important mechanisms are *assessments*, *local monitoring*, and *natural selection*. In addition, *gradients* and *pheromones* are successfully adapted from nature, being covered by a multitude of literature. Feedback, for example, is already well-known and understood from systems theory.

The different relevance of mechanisms can again be traced back to the different control types in self-organizing systems, as discussed above. As one of the most important components of a self-organizing IT system is its local control structure, mechanisms realizing such structures are dominant.

Mechanism Classes \ Application Areas		Collective coordination					Information			Decentralized control				Weighed sum
		Spatial movement	Source-drain routing	Pattern and structure formation	Team formation (avoiding egoistic behavior)	Specialized skills within a team	Integration of contextual information	Information about resources	Assessment of resources	Local control	Trust and security	Control of access to resources	Allocation of resources	
Number of examples		3	3	9	1	3	1	3	4	6	1	2	8	44
System behavior	Positive feedback and negative feedback		1										1	0,46
	Random processes and dynamic instability			1						1			2	0,53
	Threshold								1					0,25
Individual behavior	Differentiation and stem cells					1								0,33
	Replication					3				1				1,17
	Mutation and cross breeding			1		2								0,78
Collective behavior	Cooperation and rivalry				1					1			3	1,54
	Local aggregation			7										0,78
	Building blocks			1										0,11
	Natural selection	1		1	1				2			1	1	2,57
Communication	Direct communication			1				1					3	0,82
	Pheromones		3										1	1,13
	Gradients	3		3			1						1	2,46
	Stigmergy													0
	Tags			1				1			1			1,44
Control	Local monitoring			1				1	1	5	1		3	2,9
	Token											1	1	0,63
	Assessments		1	2	1				3	1		1	3	3,35
	Contracts												1	0,13
	Templates							1	1	2				0,92
	Checkpoints							1						0,33

Table 2. Results for self-organization in information technology

The results of the comparison are summed up in the following table (cf. Table 3):

	Nature and Natural Sciences	*Information Technology*
Important Application Areas	Spatial movement Pattern and structure formation	Pattern and structure formation Local control Allocation of resources
Key Mechanisms	Feedback Pheromones Gradients	Assessments Local monitoring Natural selection
System	Autonomous elements	Autonomous system

Table 3. Comparison result summary

APPLICATION AREA

Although the goal of this paper is to provide IS researchers with successful concepts for self-organization from nature and IT, we want to give a brief overview of a possible application for these concepts in the domain of Service-oriented Architectures (SOA) (Papazoglou, 2003).

Our approach is built on the AMAS.KOM architecture (Automated Monitoring and Alignment of Services), which is a decentralized monitoring and deviation handling architecture for Service-oriented Architectures based on agent technology (Repp, Eckert, Schulte, Niemann, Berbner, Steinmetz, 2008). It has been designed to monitor service-based workflows and to support the handling of Service-Level Agreements (SLA) violations autonomously. We have published the details of our agent cooperation concept previously (Miede, Behuet, Repp, Eckert, and Steinmetz, 2009), but here we give a short overview of the main concepts in order to illustrate its strong relationship with self-organization mechanisms.

In the AMAS.KOM architecture, one *Monitoring Agent* is monitoring the fulfillment of one SLA for one Web service. As a Web service is not dedicated to only one client and to use Web service technology to its full extend, a Web service serves several service consumers concurrently. Therefore, a Web service can be subject to multiple contracted SLAs and be monitored by several agents at the same time, where each agent monitors the fulfillment of one SLA in particular. Using all the agents monitoring the same Web service we create an *Agent Cluster* as the foundation for cooperation. Figure 5 gives an overview of this clustering concept. Within this cluster, the agents exchange their monitoring information and add as a whole specialized diagnosis capabilities to the cluster. For example, it becomes possible to distinguish a Web service crash from the inability to fulfill a SLA, or from problems in the monitoring infrastructure.

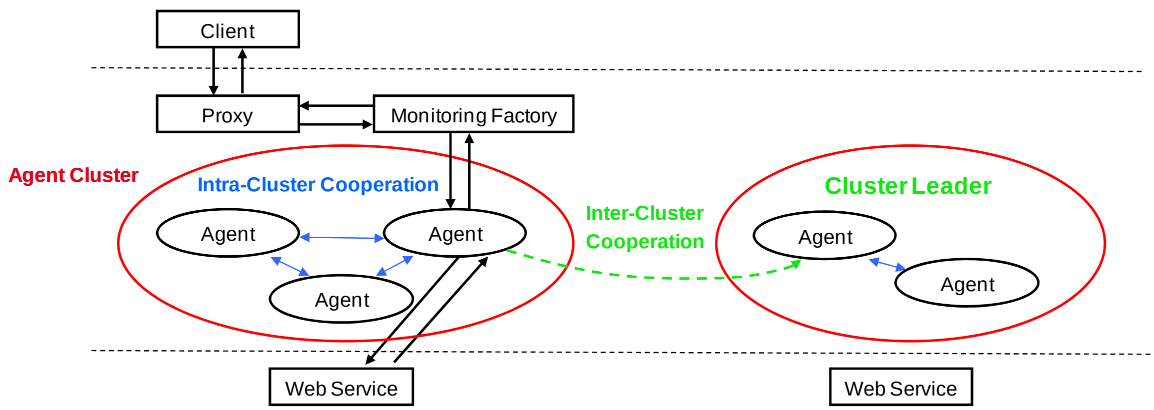

Figure 5: Self-organized agents in a Service-oriented Architecture (Miede et al., 2009)

Using this more enhanced perception of the monitored Web service, the Agent Cluster is able to initiate suitable reactions to detected violations automatically. These reactions may include, e.g., the renegotiation of SLAs, the invocation of alternative Web services, or the delegation of monitoring to other agents. In addition the cluster is able to manage itself by handling problems occurring to the agent infrastructure inside the Agent Cluster.

For special tasks one agent is elected from the Monitoring Agents in the cluster: the *Cluster Leader*. It acts still as a regular Monitoring Agent after its election, but it has also additional responsibilities, e.g., receiving requests from other Agent Clusters. The Cluster Leader is mainly responsible for the management of its Agent Cluster. To achieve this, the Cluster Leader is the centre of the heart-beat mechanism inside the cluster. It sends heart-beats periodically to the other Monitoring Agents in the cluster to inform them about its presence, upon which the other agents send heart-beats back. Thus, the Cluster Leader is aware of the presence of all the Monitoring Agents inside its cluster and can detect agent failures. In such a case, the leader tries to delegate the monitoring tasks of the damaged agent to another one in the cluster, or it creates a new agent, which replaces the problematic one and takes over its tasks. The Cluster Leader stores only basic information about the other agents in the cluster, thus, its centralized storage is no point of failure. In case the Cluster Leader fails, a dedicated election mechanism ensures the election of a new leader, which will retrieve and regenerate this information as soon as it is elected.

Choosing one agent per cluster to treat incoming communication from other Agent Clusters is in general a scalable way to design inter-cluster cooperation. In addition, electing one agent per cluster for extra responsibilities is a robust way of dealing with central non-critical functionality in the clusters, i.e., intra-cluster cooperation.

CONCLUSIONS AND OUTLOOK

Modern enterprise information systems have to fulfill a multitude of difficult requirements. An important aspect is the high complexity of such distributed, heterogeneous systems. Natural systems offer the successful concept of self-organization which enables autonomous management of complex structures without external control.

The goal of this paper is to provide IS researchers with an analysis of self-organization's application scenarios both in IT and other academic disciplines. This is done in order to identify successful concepts and to lay the foundation for adapting these concepts to current and future research in IS.

To achieve this, several approaches for self-organizations found in scientific literature were examined in order to identify both classes for application areas and classes for self-organization mechanisms. Based on this, approaches from the natural sciences and IT were compared.

The identified mechanisms are manifold and not yet completely understood in their respective fields. As part of the comparison, it could be observed that spatial movement and pattern formation are dominant application areas in nature and natural sciences, often using pheromones and gradients as mechanisms.

Opposed to that IT featured local control and resources allocation as dominant application areas with assessments and local monitoring as major mechanisms.

In addition, a major difference could be observed, as in the examined approaches artificial self-organization included a dedicated control entity. Such or similar control mechanisms are not necessary in natural, decentralized systems.

It is important to note that our approach is only a first attempt to organize the subject and it is not focused on statistics. The reasons for this are that self-organization approaches and their mechanisms are mainly qualitatively oriented and, thus, are difficult to compare quantitatively. However, based on our first results, future work could attempt to draw conclusions with a higher confidence level. This could eventually lead to an extended and refined list of the application areas and used mechanisms.

FURTHER RESOURCES

An appendix lists the sources for the examined approaches from both natural sciences and information technology and is available from the authors' website:

http://www.kom.tu-darmstadt.de/~miede/amcis2009/AMCIS_Self-Organization___2009.08_AM_Appendix.pdf

ACKNOWLEDGMENTS

This work is supported in part by E-Finance Lab e.V., Frankfurt am Main, Germany and BearingPoint Management and Technology Consultants.

REFERENCES

1. Calinescu, R. (2007) Model-Driven Autonomic Architecture, in *Fourth International Conference on Autonomic Computing* (ICAC'07), IEEE.
2. Camazine, S., Deneubourg, J., Franks, N. R., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2003) Self-Organization in Biological Systems: (Princeton Studies in Complexity), Princeton University Press.
3. Capra, F. (1997) The Web of Life: A New Scientific Understanding of Living Systems, Anchor.
4. De Wolf, T. and Holvoet, T. (2006) A Catalogue of Decentralised Coordination Mechanisms for Designing Self-Organising Emergent Applications. Universität Leuven, Report CW458, August 2006.
5. Fromm, J. and Zapf, M. (2005) Selbst-Eigenschaften in verteilten Systemen, in *Praxis der Informationsverarbeitung und Kommunikation*, no. 28, pp. 189-198.
6. Goldstein, J. (1999) Emergence as a Construct: History and Issues, Emergence **1**(1), 49-72.
7. Gouscos, D., Kalikakis, M., and Georgiadis, P. (2003) An Approach to Modeling Web Service QoS and Provision Price, in *4th International Conference on Web Information Systems Engineering Workshops* (WISEW 2003), Rom, Italien, 2003, pages 121-130.
8. Hermann, K., Mühl, G., and Geihs, K. (2004) Self-Management – Potentiale, Probleme, Perspektiven, in *Praxis der Informationsverarbeitung und Kommunikation*, no. 27, p. 74.
9. Hinchey, M. G. and Sterritt, R. (2006) Self-Managing Software, Computer **39**(2), 107.
10. Holland, J. H. (1996) Hidden Order: How Adaptation Builds Complexity (Helix Books), Addison Wesley Publishing Company.
11. International Business Machines Corp. (IBM) (2006) An Architectural Blueprint for Autonomic Computing, http://www-03.ibm.com/autonomic/pdfs/AC_Blueprint_White_Paper_4th.pdf.
12. Kauffman, S. (1996) At Home in the Universe: The Search for the Laws of Self-Organization and Complexity, Oxford University Press, USA.
13. Krafzig, D., Banke, K., and Slama, D. (2004) Enterprise SOA: Service-oriented Architecture Best Practices (The Coad Series), Prentice Hall PTR.

14. Kube, R. and Bonabeau, E. (2000) Cooperative Transport by Ants and Robots, in *Robotics and Autonomous Systems*, no. 30, pp. 85-101.
15. Melzer, I. (2007) Service-orientierte Architekturen mit Web Services. Spektrum Akademischer Verlag.
16. Miede, A., Behuet, J.B., Repp, N., Eckert, J., and Steinmetz, R. (2009) Cooperation Mechanisms for Monitoring Agents in Service-oriented Architectures, *9. Internationale Tagung Wirtschaftsinformatik 2009*, vol. 1, pp. 749–758..
17. Miede, A., Repp, N., Eckert, J., and Steinmetz, R. (2008) Self-Organization Mechanisms for Information Systems - A Survey, in *Proceedings of the Fourteenth Americas Conference on Information Systems*, Toronto, ON, Canada August 14th-17th 2008, Aug 2008.
18. Papazoglou, M. P. (2003) Service-oriented Computing: Concepts, Characteristics and Directions, *4th International Conference on Web Information Systems Engineering* (WISE 2003), IEEE, Rome, Italy, pp. 3-12.
19. Repp, N., Eckert, J., Schulte, S., Niemann, M., Berbner, R., Steinmetz, R. (2008) Towards Automated Monitoring and Alignment of Service-based Workflows, in *IEEE International Conference on Digital Ecosystems and Technologies 2008* (IEEE DEST 2008).
20. Sterritt, R. and Hinchey, M. G. (2006) Biologically-Inspired Concepts for Self-Management of Complexity, in *ICECCS '06: Proceedings of the 11th IEEE International Conference on Engineering of Complex Computer Systems*, IEEE Computer Society, Washington, DC, USA, pp. 163-168.