Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

A Complexity Model for Networks of Collaborating Enterprises

Balázs Csanád Csáji * **László Monostori** *,**

* *Computer and Automation Research Institute, Hungarian Academy of Sciences*
** *Faculty of Mechanical Engineering, Budapest University of Technology and Economics*
{*balazs.csaji, laszlo.monostori*}*@sztaki.hu*

Abstract: Theoretical study of complex systems receives more and more attention as most sciences broaden their perspectives. The paper first briefly overviews a few important complexity approaches, then it presents a triple-level model for describing and analyzing collaborating enterprises. The environment is treated as a stochastic process, the core topology of the collaboration is represented by a graph and, finally, the dynamic behavior of collaborating enterprises is modeled as a Complex Adaptive System (CAS). Complexity measures for the different sub-models are suggested, some complexity drivers are investigated and it is argued that the resulted model can be effectively analyzed by simulation. *Copyright © 2008 IFAC*

Keywords: collaborative networked organizations principles, methodologies and tools for analysis of complexity, network centric collaboration, enterprise modeling, agent-based models

1. INTRODUCTION

The need to be able to *measure the complexity* of a system, structure or problem and to obtain bounds and quantitative relations for complexity arises in more and more sciences: besides computer science and engineering, the traditional branches of mathematics, physics, chemistry, biology and social sciences are also confronted more and more frequently with this problem (Lovász and Gács, 1999).

The paper focuses on the complexity of collaborative enterprises. Though, *collaborations* promise a better survival in the globlazied market, the lack of problemoriented comprehension concerning the required systems setup often cause these collaborations to fail. This results in a very high failure rate that is near 50% (Schuh *et al.*, 2006).

The main aim of the paper is to present a model of *collaborating enterprises*. Such systems are usually concurrent and distributed the components of which are complex systems themselves, thus, they are "systems of systems" (Kotov, 1997). If a model can describe and measure complexity in them, it could help identifying *complexity drivers* in diverse types of collaboration structures. Such a model could also serve as a basis and common ground for further *complexity management* research and, hence, it could help to develop *decision support* systems for designing and controlling enterprise networks. For example, these kinds of models could help the theoretical foundations of managing collaborations, such as supply chains, working in uncertain, dynamic environments with often incomplete data. Consequently, they are of high practical importance.

Since there is a huge literature on complexity studies in different sciences (mostly in mathematics and computer science) and these researches have well-established models and theories, we try to apply as much from their contributions as possible: our description model is separated into three different abstraction levels and they are modeled by fundamental theories, such as stochastic processes, network theory, and the theory of complex adaptive systems.

The structure of the paper is as follows. First, some important complexity approaches will be briefly considered. Then, we turn our attention to the multi-level complexity model. In order to model the dynamic and uncertain behavior of the environment we will apply *stochastic processes* (series of multivariate random variables). Next, we will investigate quasi-static models of collaboration networks. We will apply *network theory* in order to model the basic network connections between enterprises that describe the core topology of collaborations. Finally, we will give a dynamic model of collaborative enterprises with the aid of *Complex Adaptive Systems* (CAS). This CAS model could be applied to design a simulation system that could help analyzing the dynamic structures of collaborations.

2. WHAT IS "COMPLEXITY"?

The meaning of the word "complexity" is vague, ambiguous, there is no universal, precise (e.g., formal) definition of it. Yet, there are approaches especially in mathematics and computer science which aim at defining special forms of complexity. Every serious, long-term research must have strong theoretical basis, therefore, if we want to define, analyze and manage complexity arising in collaborative enterprise networks we should consider the previous definitions of complexity. In this section we provide a brief overview of some important classical complexity approaches mostly from mathematics and computer science.

Since *Alan Turing* introduced his mathematical machines (viz., the Turing-machines) in 1930s, they became a fundamental tool for analyzing algorithms and combinatorial optimization problems. According to the theory of *computational complexity*, complexity is measured by the quantity of computational resources used up by a particular task. There are several complexity measures which are associated with algorithms (Lovász and Gács, 1999), e.g., time-complexity, space-complexity and, for distributed systems, communication-complexity. According to complexity of problems the two most important classes of problems are P and NP. By definition, a P-problem is a decision problem that can be solved by a deterministic Turing-machine in polynomial time . An NP-problem can be solved by a nondeterministic Turing-machine in polynomial time. Roughly, problems in P are "easy" problems, while problems in NP are considered as "hard". Naturally, this classification is a simplification. It is easy to see that any P-problem is also an NP-problem. However, the question whenever $P = NP$ is currently the most important undecided problem of computational complexity theory. This problem is named as one of the seven "millennium problems" of the Clay Mathematics Institute in the memory of *Hilbert's* celebrated open mathematical questions.

Information complexity, viz. *entropy*, tries to measure the randomness or disorder of objects. This approach was suggested by *Claude Shannon*, who in 1948 introduced entropy to communication-theory. Entropy provides a measure of the amount of information associated with the occurrence of given states. It has key importance in information- and code-theory, however, it can be also applied to measure other complex systems, e.g., graphs or networks. Note that Shannon himself borrowed the concept of entropy from physics (viz. thermodynamics and statistical mechanics), and used the *Boltzmann-Gibbs* formulation of entropy. An intuitive understanding of information entropy relates to the amount of uncertainty about an event associated with a given probability distribution.

In 1960s *Solomonoff*, *Kolmogorov* and *Chatin* (independently) introduced a complexity concept which is often called algorithmic information complexity. The name itself indicates that it has close connections to computational complexity and entropy. Given a universal Turing-machine, the *Kolmogorov complexity* of a (bit)string (description) is the length of the shortest program that generates the description and halts. In other words Kolmogorov defined the complexity of a structure as the length of its shortest description (namely, on a universal Turing machine). A structure is simple if it can be described by a short program, and is complex if there is no such short description, e.g., a random string whose shortest description is specifying it bit-by-bit. If some, fairly simple, assumptions are made on the used universal Turing-machine then the complexity of a structure (or string) will only slightly depend on the used reference machine.

Krohn and *Rhodes* introduced a complexity definition in the 1960s that aims at measuring the complexity of abstract algebraic structures, such as groups and semigroups with the concepts of homomorphisms and wreath products. In computer science, the Krohn-Rhodes theory gave new, unexpected methods to build any finite state automaton using series-parallel emulation by simple components.

In the second half of the 20th century it became more and more important to measure the complexity of structures in natural sciences (e.g., in chemistry and biology). The theory of *topological complexity* addresses this problem and applies graph theory as its basis. There are several measures to define the complexity of a graph: e.g., there are symmetry-based measures, which often apply the concept of entropy, other measures include: average- or normalized-edge complexity, subgraph count, overall connectivity, total walk count, and others based on adjacency and distance. Some of them are investigated in Section 3.2.

One of the newest complexity approaches is the theory of *Complex Adaptive Systems* (CASs). It has deep roots in the interdisciplinary field of multi-agent systems, however, the term "complex adaptive systems" itself was coined at the interdisciplinary Santa Fe Institute (SFI), by *John H. Holland*, *Murray Gell-Mann* and others (Holland, 1992; Holland, 1995). John H. Holland is one of the inventors of evolutionary computation and genetic algorithms. Nobel Prize laureate Murray Gell-Mann discovered quarks. As CASs are especially important for our enterprise network model, they are investigated more deeply in Section 3.3.

The recent EU funded project "Coll-Plexity" (Collaborations as Complex Systems) was called to life with the purpose of defining a Generic Model of Complexity (GeMoC) that is applicable for modeling collaborating enterprises in production industry and R&D (Schuh *et al.*, 2006).

During the Coll-Plexity project several abstract *complexity drivers* were identified that could cause problems in collaboration networks. These drivers are as follows

(1) Uncertainty (e.g., limited information)
(2) Dynamics (e.g., sudden or constant changes)
(3) Multiplicity (e.g., a large number of participating elements and influencing factors)
(4) Variety (e.g., many types of elements)
(5) Interactions (e.g., communication loads)
(6) Interdependencies (e.g., feedback loops)

According to the approach of the Coll-Plexity project, a system is called *complicated* if it has any of the six properties above, e.g., if it has a large number of elements or it has significant uncertainty. On the other hand, a *complex system* must have the first two properties and at least one of the last four. Formally, it can be written as

$$\big[(1) \wedge (2)\big] \wedge \big[(3) \vee (4) \vee (5) \vee (6)\big]. \tag{1}$$

Note that these complexity drivers are abstract, for example, the issues of people, culture, politics, geography or weather can all be regarded as different influencing factors.

3. THE COMPLEXITY MODEL

This section aims at describing our abstract, multi-level model for collaborative enterprises. It contains three sub-models: the model of the environment (applying stochastic processes), the enterprise network model (built upon network and graph theory) and the collaboration model (modeled as a complex adaptive system). We also investigate which complexity measures can be applied and which complexity drivers are addressed by the sub-models. The conceptual overview of the model can be found in Fig. 1.

3.1 Environment Model

Our main aim is to create an effective complexity model for collaborative enterprises. In order to do this we also need an environment model, since the environment affects and drives the collaboration. According to our view, the environment includes all factors that somehow influence the collaboration, e.g., the customers, the political and economical situation, the geography, the weather, etc. However, to keep the complexity of the model in a manageable level, we apply a very abstract environment model: the uncertain behavior of the environment is described by a multivariate random variable and, since the environment can change over time, we consider a sequence of such variables, one for each observable time step. These sequences are called stochastic processes (Papoulis and Pillai, 2001).

Stochastic processes are standard models used in statistics, signal processing, machine learning and financial mathematics. They consist of a sequence of random variables,

$$X_1, X_2, \dots, X_{t-1}, X_t, X_{t+1}, \dots, \tag{2}$$

where each X_t is a *random variable*, a measurable function $X_t : \Omega \to S$ from the sample space Ω of a probability measure space $\langle \Omega, \mathcal{F}, P \rangle$ to a measurable space of possible outcomes S. They describe an event which is uncertain from the viewpoint of the observer. *Multivariate* random variables have vector output, namely, they render several values to an element of the sample space. Random variables can be adequately described by their *distributions*.

The analysis of a stochastic process can be simplified if we make different assumptions on the distributions of the random variables. For example, a stochastic process can be assumed to be *stationary* which means that each X_t has the same distribution. One of the most simplifying, but often applied, assumptions is if we treat all variables as *independent* from each other but we also assume that they all have the *same distribution*. This property is abbreviated as "i.i.d." (independent and identically-distributed).

Another potential simplifying assumption concerning stochastic processes is the *Markov property*. In probability theory, a stochastic process has the Markov property if the conditional probability distribution of future states of the process is conditionally independent of the past states given the present state. More precisely, a stochastic process $(X_t)_{t=1}^{\infty}$ has the Markov property if for all t we have

$$P(X_t = x \mid X_{t-1}) =$$
$$= P(X_t = x \mid X_{t-1}, X_{t-2}, \dots, X_1). \tag{3}$$

Even if the process that describes the behavior of the environment is non-Markov, it is appropriate to consider it as an approximation to a Markov process, since it greatly facilitates the theoretical analysis of the process.

Our proposed model for the environment is that we should consider it as stochastic process. This process can be assumed to be stationary or Markovian. In that case we do not concern with the inner structure and the internal dependencies of the environment, we mostly treat the environment as a black box, however, we still have a formal statistical model to work with. According to our model, at each observable time t the state of the environment can be described by a multivariate random variable, X_t, such as

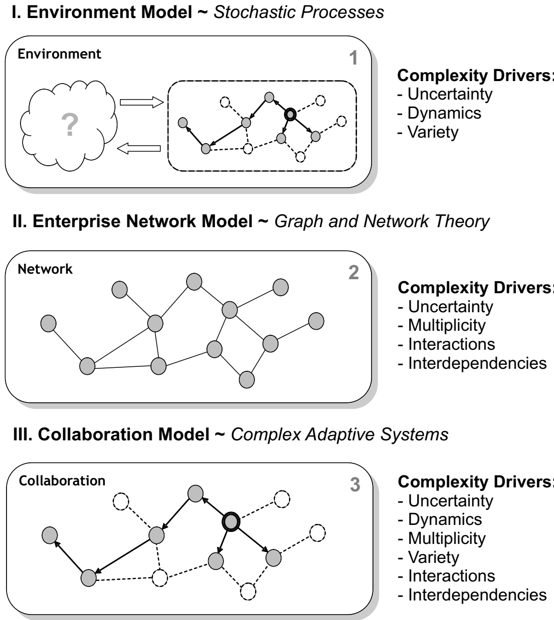

Fig. 1. Conceptual overview of the proposed model

$$X_t : \Omega \to \begin{bmatrix} X_{t,1} \\ X_{t,2} \\ \vdots \\ X_{t,n} \end{bmatrix}, \tag{4}$$

where each $X_{t,i}$ is a single valued random variable that describes a particular aspect of the environment that we want to take into account. For example, the number of requests, the number of products that the customers have ordered, due dates, the trustiness of the customers, external costs, the economical situation (e.g., interest and currency rates, asset prices), the weather (e.g., temperature) or even the social, the cultural and the political situation.

For example, a classical approach to model the evolution of *asset prices* V_t over time is to treat them as a continuous-time diffusion process, which can be written in the form of the following stochastic differential equation

$$\frac{dV_t}{V_t} = \mu dt + \sigma dW_t, \tag{5}$$

where $V_0 > 0$, μ is a drift parameter, $\sigma > 0$ is a volatility parameter, and W is a standard Brownian motion. Price V_t can be calculated directly by applying Ito's lemma,

$$V_t = V_0 \exp(mt + \sigma W_t), \tag{6}$$

where $m = \mu - \frac{1}{2}\sigma^2$, which simplifies the investigations.

Regarding *measuring* the complexity of random variables, the concept of *entropy* can be applied (Shannon, 1948). The (differential) entropy of random variable X is

$$h[X] = -\int_{-\infty}^{\infty} f(x) \log_2 f(x)\, dx, \tag{7}$$

where f is the probability density function of X. The concept of entropy measures how much information (or disorder) there is in a signal or in an event.

Note that the probability distribution of a random variable can be estimated using accumulated historical, statistical data (at least if the stochastic process was stationary).

3.2 Enterprise Network Model

Now, we turn our attention to model the *core topology* of an enterprise network. We will assume that this topology is *quasi-static* or slowly varying, hence it can be adequately modeled by network theory. The dynamic elements of the network are investigated in the collaboration sub-model.

The basis of modern network theory (Barabási and Albert, 1999; Barabási, 2002; Newman, 2003) and, hence, the basis of network- or topological-complexity is graph-theory, which is one of the fundamental theories in discrete mathematics. Its history goes back to *Euler*'s celebrated solution of the Königsberg bridge problem in 1735.

Formally, a graph $G = \langle V, E \rangle$ consists of a set of *vertices* (or *nodes*) denoted by $V = \{v_1, \ldots, v_n\}$ and a set of *edges*, $E \subseteq V \times V$. We call two vertices $v_i, v_j \in V$ *adjacent* if $\langle v_i, v_j \rangle \in E$. Sometime *weights* and *labels* (e.g., colors) are also associated with the vertices and the edges.

The *adjacency matrix* of graph G is an $n \times n$ matrix, $A(G) = [a_{ij}]_{i,j=1}^{n}$ where a_{ij} is 1 if there is an edge between vertices v_i and v_j, otherwise it is 0. The *degree* of a vertex is the number of edges connecting it to other vertices. For directed graphs, a vertex has *in* and *out* degrees, as well. A *path* in the graph is a sequence of adjacent edges between two vertices without traversing any intermediate vertex twice. The length of a path is the number of edges that the path uses. A *distance* of two vertices is the length of the shortest path between them. The *distance matrix*, $D(G) = [d_{ij}]_{i,j=1}^{n}$, of graph G is defined as d_{ij} is the distance of vertices v_i and v_j. Naturally, in case of undirected graphs both $A(G)$ and $D(G)$ are symmetrical.

The elements of graphs can be naturally associated with the elements of an enterprise network, at least regarding its core topology, e.g., an association could be as follows

- Vertices ∼ companies or functionalities
- Edges ∼ connections between companies; functionality associations (e.g., supplier)
- Vertex labels ∼ production competences
- Vertex weights ∼ production capabilities
- Edge labels ∼ connection, contract type
- Edge weights ∼ collaboration strength
- Colors, labels ∼ roles, types, strengths

In Fig. 2, e.g., a directed, colored, vertex and edge labeled graph representation of a real production network is shown. The actual names of the companies are omitted.

An advantage of this approach is that there are plenty of *off-the-shelf* complexity measures available for graphs (Brochev and Rouvray, 2006). In what follows, we overview some of these measures that could be applied to measure the topological complexity of a collaboration network.

There are adjacency related measures, such as *total adjacency*, $Adj(G)$, or *average vertex degree*, $Avd(G) = Adj(G)/n$, or *connectedness*, $Conn(G) = Adj(G)/n^2$,

$$Adj(G) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}. \qquad (8)$$

Similarly to the above definitions, one can define *total graph distance*, $Dist(G) = \sum_i \sum_j d_{ij}$, *average vertex distance*, $Avdist(G) = Dist(G)/n$ and *average degree of vertex-vertex separation*, $Advvs(G) = Dist(G)/n^2$.

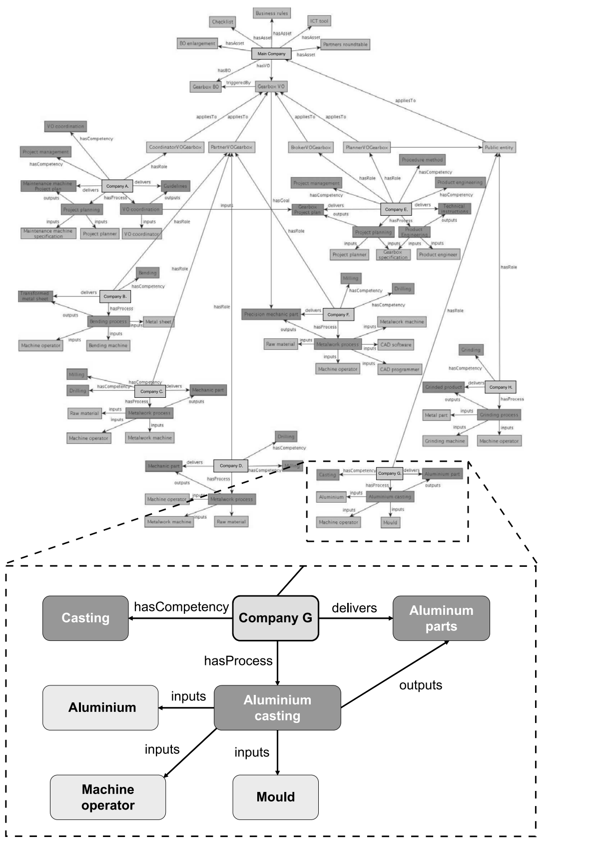

Fig. 2. Static representation of a real production network

The complexity of a graph can also be expressed by the total number of subgraphs. The number of all subgraphs containing two edges has an important role in chemistry and it has a special name, called *Platt's index*. This index was used to measure molecular complexity. The number of subgraphs containing three edges are called *Gordon-Scantleburry index*. Later, *total subgraph count* was introduced to measure the complexity of a graph, where subgraphs of all sizes (even the graph itself) was counted. In case of large networks, often subgraphs containing less then x, e.g., $x = 3$, edges are counted in practice, in order to avoid combinatorial explosion. Using subgraphs, one can also define the overall connectivity index of a graph as a sum of all adjacencies of all k-th order subgraphs.

Vertex degree distribution applies the concept of *entropy*,

$$Vdd(G) = \sum_{i=1}^{n} a_i \log_2 a_i \qquad \text{where} \qquad a_i = \sum_{j=1}^{n} a_{ij}. \qquad (9)$$

This complexity measure has the property that it increases with the connectivity and with other complexity factors, such as, the number of branches, cycles or cliques.

An alternative way to define complexity is to count all paths from any vertex to any other vertex, called *total walk count*. Note that there could be only finite number of possible walks. A natural extension of this measure is when (edge or vertex) weights are also taken into account.

Networks with high complexity are characterized by both high vertex-vertex connectedness and small vertex-vertex separation, as well. The *A/D index* of a graph is defined as the ratio of total adjacency and the total distance,

$$(A/D)(G) = \frac{Adj(G)}{Dist(G)}. \tag{10}$$

A refinement of this measure is the *B-index* that is defined as the sum of all b-values, $b_i = a_i/d_i$. It has an advantage over the A/D index that it has less sensitivity to degredancy and more sensitivity to local topology.

3.3 Collaboration Model

Probably the most important elements of an enterprise collaboration are *dynamic* and, therefore, hard to model and analyze. In this section we suggest modeling the dynamic behavior of an enterprise network as a CAS.

Complex Adaptive Systems (CASs) constitute a new paradigm (Holland, 1992; Holland, 1995) with the goal to study the structures and dynamics of systems and the question, how the adaptability of the system creates complexity.

A CAS can be considered as a Multi-Agent System (MAS) with seven basic elements in which "a major part of the environment of any given adaptive agent consists of other adaptive agents, so that a portion of any agent's efforts at adaptation is spent adapting to other adaptive agents". Agents may represent any entity with self-orientation, such as cells, species, individuals, firms or nations.

Holland postulates seven basic elements that characterize a CAS, four of which are properties, the others are mechanisms (Holland, 1995): aggregation, flows, nonlinearity, diversity, tagging, internal models, building blocks. The first four concepts represent certain characters of agents, are very important in the adaptation and evolution process, while the other three concepts are mechanisms of agents for communicating with each other and their environment.

Environmental conditions are changing, due to the agents' interactions as they compete and cooperate for the same resources or for achieving a given goal. This, in turn, changes the behavior of the agents themselves. The most remarkable phenomenon exhibited by a CAS is the *emergence* of highly structured collective behavior over time from the interactions of simple subsystems, usually, without any centralized control (Ueda *et al.*, 2001). The emergence of a complex adaptive behavior from the local interactions is demonstrated in Fig. 3. Emergence concerning organizations was studied by (Kurtz and Snowden, 2003). The typical characteristics of a CAS include dynamics involving interrelated spatial and temporal effects, correlations over long length- and time-scales, strongly coupled degrees of freedom and non-interchangeable system elements, to name only the most important ones. Both the CAS and its environment simultaneously co-evolve in order to maintain themselves in a state of quasi-equilibrium.

CASs constitute a natural framework to model production structures (Monostori and Csáji, 2007), e.g., collaborative enterprises, and to investigate complexity drivers in them. An *enterprise* can be associated with an *agent* that interacts with other agents in an uncertain and changing environment. Note that there are already a variety of MAS based production control approaches (Csáji *et al.*, 2006).

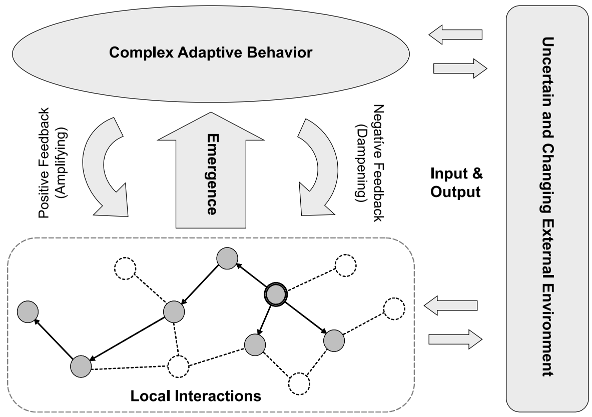

Fig. 3. Emergence in Complex Adaptive Systems

At the same time, a problem that we have to face is that even if we can describe a collaboration network as a CAS, it would be very hard to formally analyze a system like that. However, computer-based *simulations* can be applied to evaluate and test these systems. Simulations can help observing and investigating, e.g., how (potentially simple) individual behavior rules can emerge and give rise to complex (and often unpredictable) collective behavior. Additionally, the stability of these kinds of systems together with the effects of uncertainties (such as the lack of precise market forecasts, as well as personal contacts and cultural dependencies) could also be evaluated by simulations. This kind of experimental approach has several advantages, such as: it can effectively help a *what-if analysis*; it can support *statistical evaluation* of cooperation structures; hence, it can be used as a *design* or *decision support* tool.

In designing CAS, non-linear phenomena, incomplete data and knowledge, a combinatorial explosion of states, dynamic changes in environment and the frame problem are some notable examples of difficulties to be faced. The central question is realizing an artifactual system that achieves its purpose in unpredictable conditions. Complex systems, however, exhibit patterns of behavior that can be considered archetypal or prototypical. In order to manage such systems an appropriate balance between control and emergence must be found (Choi *et al.*, 2001).

The difficulty in understanding the effects of individual characteristics of the agents on their collective behavior underlines the importance of using simulation as primary tool for designing and optimizing such systems. In this respect, the proper balance between simulation and theory is to be aimed (Surana *et al.*, 2005). Our further research activities will also go in this direction.

4. CONCLUSION

Managing a network of collaborative enterprises is a hard task and, according to the current situation, the potential failure rate is very high (Schuh *et al.*, 2006). Therefore, it is important to experimentally and theoretically study collaborations, since later these researches can lead to effective complexity management techniques as well as design tools and decision support systems. In order to investigate networks of collaborative enterprises, first, an adequate model should be found. If the model could measure complexity it would also help identifying complexity drivers in different kinds of collaboration structures.

During the paper we have argued that, based on computer science and mathematics, a collaboration network description model can be made. We suggested three different submodels for modeling different parts of the problem.

The most abstract part of our system is the environment. In order to keep the complexity of the proposed model in a manageable level, we have to satisfy with a very rough model of those phenomena that we do not want to investigate in a detailed way, but still effect the collaboration. For example, the macro economy, the customers, the culture, the politics, the geography, the weather. We suggest modeling them as a multivariate random variable with potentially different marginal distributions. Each component of the variable can describe one particular aspect of the environment. Since the environment can change over time, it should be treated as a stochastic process, namely, as a sequence of (multivariate) random variables. In order to simplify the analysis, this process can be assumed to be stationary or Markovian. Even though, in this manner, the environment is almost treated as a black box, its complexity can still be measured, e.g., by information entropy.

Some parts of collaboration can be treated as quasi-static and, therefore, can be adequately described by static models, such as graph theory. Network and graph theory is well-developed and is one of the most important parts of discrete mathematics. Companies (or their functionalities) can be associated with graph vertices while the edges can represent connections or relations between these companies (such as potential cooperations). The properties of the companies can be encoded into the labels of the nodes; the features of the connections (such as physical distance or trustiness) can be incorporated in the edge weights. Non-numerical values can also be taken into account, e.g., in the edge labels. This approach also has the advantage that there are a lot of graph related complexity measures available that can be applied to measure the complexity of an enterprise network, at least the static parts of it.

Finally, the dynamic parts of the cooperation can be modeled as a Complex Adaptive System (CAS). Since our main aim is to investigate collaborations between enterprises, to keep the model as simple as possible, we only roughly model the internal structure of the companies such as their resources (viz., factories, stores or transportation fleets) or their decision mechanisms (e.g., their managers). Therefore, agents are primarily associated with enterprises. Each agent can have its own goal and the ability to cooperate with other agents. Even if the strategy or behavior rule of each agent is simple, a complex adaptive behavior can emerge from local interactions. The analysis of such systems can be achieved through computer-based simulations that, later, could also become the basis of a collaboration network design or a decision support tool for managers.

Consequently, the paper proposed a triple-level model that offers a simple yet effective approach for modeling collaborative enterprises. The suggested model also offers complexity measures that could help investigating potential problems in networks of collaborating enterprises.

5. ACKNOWLEDGEMENTS

The work was supported by the Hungarian Scientific Research Fund (OTKA) "Production Structures as Complex Adaptive Systems" and by the EU-project "Coll-Plexity" 12781 (NEST). Balázs Csanád Csáji greatly acknowledges the scholarship of the Hungarian Academy of Sciences.

REFERENCES

Barabási, A.-L. (2002). *Linked: The New Science of Networks.* Perseus, Cambridge.

Barabási, A.-L. and R. Albert (1999). Emergence of scaling in random networks. *Science* **268**, 509–512.

Brochev, D. and D. H. Rouvray (2006). Quantitative measures of network complexity. *Complexity in Chemistry, Biology, and Ecology* pp. 191–235.

Choi, T.Y., K.J. Dooley and M. Rungtusanatham (2001). Supply networks and complex adaptive systems: Control versus emergence. *Journal of Operations Management* **19**, 351–366.

Csáji, B. Cs., L. Monostori and B. Kádár (2006). Reinforcement learning in a distributed market-based production control system. *Advanced Engineering Informatics* **20**, 279–288.

Holland, J.H. (1992). Complex adaptive systems. *Daedalus* pp. 17–30.

Holland, J.H. (1995). *Hidden Order: How Adaptation Builds Complexity.* Helix Books, Addison-Wesley.

Kotov, V. (1997). Systems-of-systems as communicating structures. Technical report. Hewlett Packard Computer Systems Laboratory.

Kurtz, C. F. and D. J. Snowden (2003). The new dynamics of strategy: Sense-making in a complex and complicated world. *IBM Systems Journal* **32**(3), 462–483.

Lovász, L. and P. Gács (1999). *Complexity of Algorithms.* Lecture Notes, Boston University, Yale University.

Monostori, L. and B. Cs. Csáji (2007). Production structures as complex adaptive systems. In: *Proceedings of the 40th CIRP International Seminar on Manufacturing Systems, May 30 - June 1, Liverpool, United Kingdom.* p. (CD version is available).

Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review* **45**(2), 167–256.

Papoulis, A. and S. U. Pillai (2001). *Probability, Random Variables and Stochastic Processes.* McGraw-Hill.

Schuh, G., A. Sauer and S. Döring (2006). Modeling collaborations as complex systems. In: *Proceedings of the 4th International Industrial Simulation Conference (ISC), The University of Palermo.* pp. 168–174.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal* **21**, 623–656.

Surana, A., S. Kumara, M. Greaves and U. N. Raghavan (2005). Supply-chain networks: A complex adaptive systems perspective. *International Journal of Production Research* **43**(20), 4235–4265.

Ueda, K., A. Márkus, L. Monostori, H. J. J. Kals and T. Arai (2001). Emergent synthesis methodologies for manufacturing. *Annals of the CIRP* **50**(2), 535–551.