ANALYSIS OF AGENTS' BEHAVIOR IN MULTIAGENT SYSTEM

Kateřina Slaninová[(a)], Jan Martinovič[(a)], Pavla Dráždilová[(a)], Dominik Vymětal[(b)], Roman Šperka[(b)]

[(a)]VSB - Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Czech Republic
[(b)]Silesian University in Opava
School of Business Administration in Karviná
Czech Republic

[(a)]katerina.slaninova@vsb.cz, jan.martinovic@vsb.cz, pavla.drazdilova@vsb.cz,
[(b)]vymetal@opf.slu.cz, sperka@opf.slu.cz

ABSTRACT
Multi-agent systems are commonly used for simulation purposes. The authors focused on agent-based technology in the business process simulation, especially on the analysis of the agent-based simulation outputs in order to facilitate the verification of the used methodology. The paper deals with an analysis of agents' behavior in multi-agent model of business processes. The main goal of the paper is to find, how selected methods can influence finding of behavioral patterns of selected agents in the system and how the amount of extracted sequences can be reduced. Extraction of behavioral patterns was performed by process mining and pattern mining methods with the focus on the sequences. The authors present the comparison of selected methods for the definition of agents' behavior with the focus to selected characteristics of observed methods. Behavioral patterns and relations between them create complex networks; thus, the extraction of behavioral patterns is optimized by spectral graph partitioning. Moreover, the visualization of groups of agents with similar behavior is presented.

Keywords: system modeling, multi-agent systems, behavioral patterns, process mining

1. INTRODUCTION

The overall idea of proposed methodology is to simulate real business processes and to provide predictive results concerning the management impact. This should lead to improved and effective business process realization. The paper deals with the analysis and visualization of agents' behavior to facilitate the verification of the simulation model and to effectively observe the reaction of the model in relation to the initialization of its input attributes, or in relation to the influence of the model environment.

The presented approach deals with a lot of influences that are not able to be captured by using any business process model (e.g. the effects of the collaboration of business process participants or their communication, experience level, cultural or social factors). The statistical method has only limited capabilities of visual presentation while running the simulation. Finally, an observer does not actually see the participants of business process dealing with each other.

Agent-based simulations dealing with a company simulation can bring several crucial advantages (De Snoo 2005), (Jennings et al. 2000). They can overcome some of the problems identified above. It is possible to involve unpredictable disturbance of the environment into the simulation with the agents. All of the mentioned issues are the characteristics of a multi-agent system (MAS).

This paper is structured as follows. Section 2 briefly informs about the multi-agent framework developed. Section 3 contains the description of the proposed method used for the analysis of the agents' behavior in the multi-agent system, and finally, the experiments with real data collection from multi-agent system are presented in Section 4.

2. AGENT-BASED SIMULATION MODELING

The authors deal with the agent-based simulation model, which is described in (Macal and North 2005), with the focus to Business Process Management (BPM) (Yan et al. 2001). Usual business process simulation approaches are based on the statistical calculation, as can be seen for example in (Scheer and Nuttgen 2000, Islamov and Rolkov 2010). But only several problems can be identified while using the statistical methods. The model uses the advantages of the agent technology (communication, cooperation, social behavior), to describe some of the core business processes of a typical business company. The authors developed a business process simulation framework (Vymetal et al. 2012) in order to simulate the real behavior of the company on the market. JADE (Bellifermine 2010) platform was used for the implementation, which

provides running and simulation environment allowing for the distribution with thousands of software agents.

The core of this paper is the analysis of agent-based simulation outputs. To ensure the outputs, above mentioned framework was used to trigger simulation experiments. The simulation framework covers business processes supporting the selling of goods by company sales representatives to the customers. It consists of the following types of agents: sales representative agents (representing sellers), customer agents, an informative agent (provides information about company market share, and company volume), and manager agent (manages the communication between seller and customer). All the agent types are developed according to the multi-agent approach. The interaction between agents is based on the FIPA contract-net protocol; see Figure 1 (FIPA 2002).

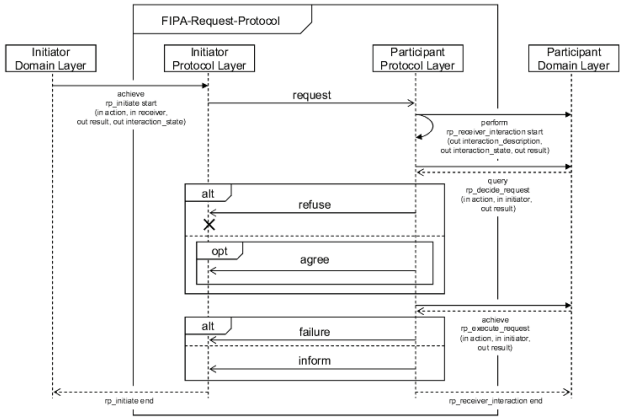

Figure 1: FIPA Contract-Net Protocol

The number of customer agents is significantly higher than the number of sales representative agents in the model because the reality on the market is the same. The behavior of agents is influenced by two randomly generated parameters using the normal distribution (amount of requested goods and a sellers' ability to sell the goods). In the lack of real information about the business company, there is possibility to randomly generate different parameters (e.g. company market share for the product, market volume for the product in local currency, or quality parameter of the seller). The influence of randomly generated parameters on the simulation outputs while using different types of distributions was presented in (Vymetal et al. 2012). The simulation uses random values instead of real data.

The overall workflow of the system proposed can be described as follows (see Figure 2).

The customer agents randomly generate the requests to buy some random pieces of goods. The sales representative agents react to these requests according to their own internal decision functions and follow the contracting with the customers. A production function is used to define the value of the negotiated price. The purpose of the manager agent is to manage the requests exchange. The contracting results into the sales events. More indicators of sale success like revenue, amount of sold goods, incomes, and costs are analyzed as well. Each action running in the simulation framework was recorded in the log file. The log file serves as a dataset for further analysis described in Section 4.

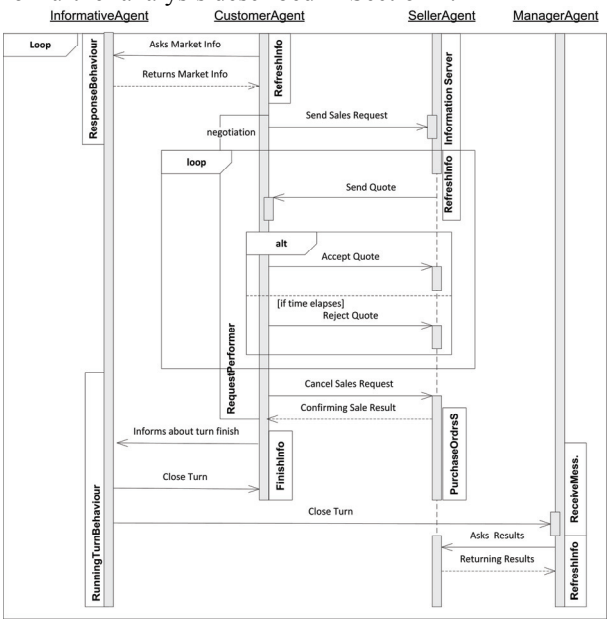

Figure 2: Workflow of Proposed System

3. ANALYSIS OF AGENTS' BEHAVIOR

The verification of the business process model is based on the analysis of recorded outputs. The approach described in this paper is focused on the analysis of agents' behavior in the model, which is recorded in a log. The log is a simple text file, where each row represents one event. The main structure, which is valid for all agents, consists of TimeStamp (date and time when the event was performed), AgentID (each agent has its unique ID), AgentClassName, TypeOfAction, ActionAttribute.

Definition of the agent behavior was performed using the methods of process mining (Aalst 2011; Aalst et al. 2004). Then, the agents' behavior in the model can be described by a set of event sequences. The paper is oriented to comparison of methods used for finding behavioral patterns of the agents and to their description on the basis of similarity of extracted sequences.

Behavioral patterns were extracted by the methods used for the sequence comparison; their description was provided using methods of text processing. Two basic groups of algorithms for the comparison of two or more categorical sequences are generally known. The first group divides the algorithms by the fact, whether the sequences consist of ordered or unordered elements. The second group of algorithms focuses on the comparison of the sequences with the different lengths and with the possible error or distortion.

The agent behavior in the business process simulation can be described by a set of event sequences. A *sequence* is an ordered list of elements (in this approach events), denoted $s = (e_1, e_2, ..., e_m)$. Given two sequences $s = (e_1, e_2, ..., e_m)$ and $u = (f_1, f_2, ..., f_l)$. Sequence s is called a subsequence of u, denoted as

$s \subseteq u$, if there exist integers $1 \le j_1 < j_2 < ... < j_m \le 1$ such that $e_1 = f_{j1}, e_2 = f_{j2}, ..., e_m = f_{jm}$. Sequence u is than a super sequence of *s*.

3.1. Comparison of Sequences

For easier description of the agents' behavior, behavioral patterns were extracted with the focus to the similarity between the sequences.

The basic approach to the comparison of two sequences, where the order of elements is important, is The longest common substring (LCS) method (Gusfield 1997), see example in Table 1. As obvious from the name of the method, the main principle of the method is to find the length of the common longest substring. Given the two sequences $s = (e_1, e_2, ..., e_m)$ and $u = (f_1, f_2, ..., f_l)$, we can find such subsequence $v = (g_1, g_2, ..., g_p)$, where $g_k = e_{(i+k-1)} = f_{(j+k-1)}$ for all $k = 1, ..., p$ and $p \le m, l$.

The LCS method respects the order of elements in the sequence. However, the main disadvantage of this method is that it can find only the identical subsequences, where no extra element is presented in the sequence. For some domains, typically where a large amount of different sequences exists, this fact gives too strict limitation. As a solution of this problem can be considered The longest common subsequence method (LCSS), described for example in (Hirschberg 1977), see example in Table 1. Contrary to LCS method, this LCSS method allows (or ignores) the inserted extra elements in the sequence, and therefore, it is immune to slight distortions.

	1.	2.	3.
Sequence *s*	EABCF	EAEBCE	ABBCC
Sequence *u*	ZABCT	FABCF	EABCE
LCS	ABC	BC	AB
LCSS	ABC	ABC	ABC
T-WLCS	ABC	ABC	ABBCC

Table 1: Results of Algorithms for Subsequence Detection

Whether the similarity of compared sequences is defined as a function using a length of common subsequence, one characteristic of this method can be found. The length of the common subsequence is not immune to the recurrence of the identical elements, which can occur only in one of the compared sequences. We can find such situations, for example due to inappropriate sampling or due to any kind of distortion.

In some applications, it is suitable (or sometimes even required) to eliminate such type of distortions and to work with them like with the equivalent elements. The solution is in another method, The time-warped longest common subsequence (T-WLCS) (Guo, A. and Siegelmann, H. 2004), see example in Table 1. The method combines the advantages of LCSS method with dynamic time warping (Müller, M. 2007). Dynamic time warping is used for finding the optimal visualization of elements in two sequences to match them as much as possible. This method is immune to minor distortions and to time non-linearity. It is able to compare sequences, which are for standard metrics evidently not comparable.

The method emphasizes recurrence of elements in one of the compared sequences. Due to this fact, the length of the common subsequence can be longer than the shorter length of the compared sequences. In the experiments described in the paper, the authors compare the impact of LCS, LCSS and T-WLCS methods to finding the behavioral patterns and to construct the agents' profiles based on their behavior in multi-agent system.

3.2. From Log to Agents' Network

As the first step of the approach, a set of agent profiles was generated. In the agent profile, each agent has assigned its set of sequences performed by it. The sequences were extracted using similar principle as extraction of traces in business process analysis (Aalst 2011). However, in our approach, the trace is defined in relation to one agent, not to one case. Given a set of agents $A = \{A_1, A_2, ..., A_n\}$, where each agent is described by its set of sequences representing the agents' behavior during the business process simulation in the MAS. Thus, we have for all A_i a set $S_i = \{s_{1i}, s_{2i}, ..., s_{mi}\}$. Set of all possible sequences, which can occur in the log is given by $S = \cup S_i$ for all $i = 1, 2, ..., n$.

Afterwards, finding the behavior patterns of agents was performed. Since obtained sequences are often similar with inconsiderable differences, the next three steps of proposed approach are oriented to finding behavioral patterns, which can better describe the agents' behavior during the simulation. Finding of behavioral patterns is based on a weight of the sequence relation. In the paper the results for three methods: LCS, LCSS, and T-WLCS are compared.

A weighted graph $G = (V, E, W)$ describing relations between the sequences was constructed, where the nodes represent sequences ($V = S$) and edges are evaluated by the weight of the relation between two sequences (depending on the selected method), and which is higher than selected threshold ($w_{ij} > \theta$).

The obtained graph is then divided into clusters with the similar sequences using spectral clustering, improved by our proposed Left-Right Oscillate algorithm. For more details, see our previous work (Martinovic et al. 2012). Thus, set of sequences *S* is divided into clusters C_j, where $S = \cup C_j$ and $j = 1, ..., c$, which is count of the clusters.

Afterwards, selected clusters are described by its representatives, called behavioral patterns. The main principle is that each representative is the most similar to the other sequences in the cluster. Finding a representative sequence of a cluster C_j is based on

finding a sequence with the maximal sum of relation weights (depending to the selected method) to the other sequences s_k within the cluster C_j.

For each agent A_j its own reduced profile P_j is then created, which determines the agent using behavioral patterns. The reduced agent profile is a binary vector of a dimension c (count of clusters), which gives information about characteristic behavioral patterns of the given agent.

A weighted graph describing the relations between agents was generated at the last step. The nodes of the graph represent the agents identified by its unique ID, vertices represent the relations between the agents. The relation between agents was defined by the similarity of their behavior using cosine measure.

4. EXPERIMENTS ON SIMULATION MODEL

In the experiments, the log file generated by MAS for the simulation of one year with 52 turns (one turn is equivalent to one week) was used. The log file contained 557760 records of agents' activities. Four types of agents: ManagerAgent, InformativeAgent, CustomerAgent and SellerAgent were defined in the simulation model. The basic description of the log file is presented in Table 2.

Agent Class	Records	AgentInstances
ManagerAgent	54	1
InformativeAgent	107	1
CustomerAgent	406 482	498
SellerAgent	151 098	51
Sum	557 760	552

Table 2: Log File Description

4.1. Sequence Extraction

The records from the log file were used for the extraction of sequences. As mentioned in Section 3.2, the methodology of Aalst was used, modified to our approach based on the multi-agent system requirements. During the extraction, each sequence was related to one agent; its start and end points were detected by the start and the end of one turn. For described data collection, we have obtained set *S* consisted of 1974 sequences, and 24 types of different sequences used in the simulation. We assume more diverse output in future experiments, due to planned simulations with more variable initialization of input parameters of agents.

4.2. Finding of Behavioral Patterns

This phase of the experiments was oriented to an exploration, how the different methods for measurement of relation weight between two sequences can influence the finding of the behavioral patterns. The following methods have been tested: LCS, LCSS and T-WLCS. All the methods can find the longest common subsequence α of compared sequences β_x and β_y, where $\alpha \subseteq \beta_x \wedge \alpha \subseteq \beta_y$, where $\beta_x \in S$ and $\beta_y \in S$. For description of used methods, see Section 2. The relation weight R_w was counted by Equation 2.

$$R_w(\beta_x, \beta_y) = \frac{(l(\alpha) * h)^2}{l(\beta_x) * l(\beta_y)}, \quad (1)$$

where $l(\alpha)$ is a length of the longest common subsequence α for sequences β_x and β_y; $l(\beta_x)$ and $l(\beta_y)$ are analogically lengths of the compared sequences β_x and β_y, and

$$h = \frac{Min(l(\beta_x), l(\beta_y))}{Max(l(\beta_x), l(\beta_y))} \quad (2)$$

Table 3, Table 4 and Table 5 describe obtained components with similar sequences for each method and for different threshold θ (level of relation weight R_w).

θ	Components (Size)	Isolated Nodes
0.9	1 (2)	22
0.8	2 (2,2)	20
0.7	3 (2,2,2)	18
0.6	4 (2,2,2,2)	16
0.5	4 (4,4,2,2)	12
0.4	5 (6,6,2,2,2)	6
0.3	3 (11,7,2)	4
0.2	**3 (11,8,2)**	**3**
0.1	3 (11,8,2)	3

Table 3: Components of Sequences, LCS

θ	Components (Size)	Isolated Nodes
0.9	1 (2)	22
0.8	2 (2,2)	20
0.7	3 (4,2,2)	16
0.6	4 (5,2,2,2)	13
0.5	3 (9,4,2)	9
0.4	4 (9,6,2,2)	5
0.3	3 (11,7,2)	4
0.2	**3 (11,8,2)**	**3**
0.1	3 (11,8,2)	3

Table 4: Components of Sequences, LCSS

θ	Components (Size)	Isolated Nodes
0.9	2 (7,2)	15
0.8	3 (7,3,2)	12
0.7	3 (8,7,2)	7
0.6	3 (9,8,2)	5
0.5	2 (11,8)	5
0.4	**3 (11,8,2)**	**3**
0.3	3 (11,8,2)	3
0.2	3 (11,8,2)	3
0.1	3 (11,8,2)	3

Table 5: Components of Sequences, T-WLCS

As we can see from Table 3 and Table 4, we have obtained for level θ=0.2 the same 3 components of similar sequences for both methods LCS and LCSS, while for method T-WLCS the same 3 components have

been obtained even for level θ=0.4, see Table 5. Which is important, each method has for higher levels of θ different outputs.

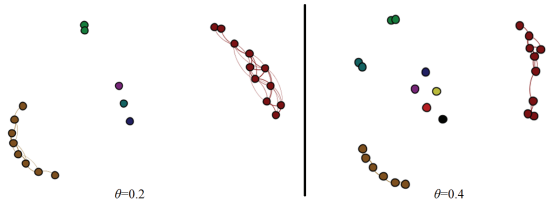

Figure 3: Sequence Components - LCSS, θ=0.2 and θ=0.4

In Figure 3 and Figure 4, we can see the examples of visualized graphs with colored components of similar sequences for LCSS and T-WLCS method with selected threshold θ.

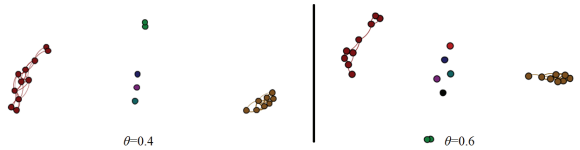

Figure 4: Sequence Components – T-WLCS, θ=0.4 and θ=0.6

The detailed description of obtained clusters of sequences for the method T-WLCS (θ=0.4) is presented in Table 6. The same clusters with the same sequences were obtained for the methods LCS and LCSS with θ=0.2 as well.

Cluster	Seq. ID	Sequence Activities
C0	9	1;2;3;4;5;7;8;7;8;7;8;7;8;7;8;7;8;7;8;7;8;7;8;7;8;10;
	10	2;3;4;5;7;8;7;8;7;8;7;8;7;8;7;8;7;8;7;8;7;8;7;8;7;8;10;
	13	2;3;4;5;7;8;12;10;
	14	2;3;4;5;7;8;7;8;7;8;7;8;7;8;12;10;
	15	2;3;4;5;7;8;7;8;12;10;
	16	1;2;3;4;5;7;8;12;10;
	17	2;3;4;5;7;8;7;8;7;8;12;10;
	18	2;3;4;5;7;8;7;8;7;8;7;8;7;8;7;8;7;8;7;8;12;10;
	19	2;3;4;5;7;8;7;8;7;8;7;8;12;10;
	20	2;3;4;5;7;8;7;8;7;8;7;8;7;8;7;8;12;10;
	21	1;2;3;4;5;7;8;7;8;7;8;7;8;12;10;
C1	1	6;6;6;6;6;6;6;6;6;6;6;6;
	2	6;6;6;11;
	3	6;11;
	4	6;6;11;
	5	6;6;6;6;6;6;11;
	6	6;6;6;6;11;
	7	6;6;6;6;6;11;
	8	6;6;6;6;6;6;6;6;11;
C2	11	2;9;10;
	12	1;2;9;10;

Table 6: Detailed Description of Sequence Clusters, T-WLCS method, θ=0.4

For selected weight level θ, the obtained clusters of similar sequences were assigned as behavioral patterns. Then, each behavioral pattern was described by its representative, see Section 3.2. The example of one cluster obtained using LCS method (θ =0.3) can be seen in Table 7, where the sequence with ID 19 has been selected as a representative of this cluster.

R_w	Seq. ID	Sequence
0.44	17	2;3;4;5;7;8;7;8;7;8;12;10
1	**19**	**2;3;4;5;7;8;7;8;7;8;7;8;12;10**
0.81	21	1;2;3;4;5;7;8;7;8;7;8;7;8;12;10
0.49	14	2;3;4;5;7;8;7;8;7;8;7;8;7;8;12;10
0.35	20	2;3;4;5;7;8;7;8;7;8;7;8;7;8;7;8;12;10

Table 7: Example of representative sequence, LCS method (θ=0.3)

The behavioral patterns has been used for the description of agents' behavior, see Section 4.3.

4.3. Agents' Network Based on Behavioral Patterns

The last step of proposed approach was the generation of agents' network based on the obtained behavioral patterns. Example of such network is presented in Figure 5, which demonstrates the agents' communities based on similar behavioral patterns extracted using LCS method (θ=0.3).

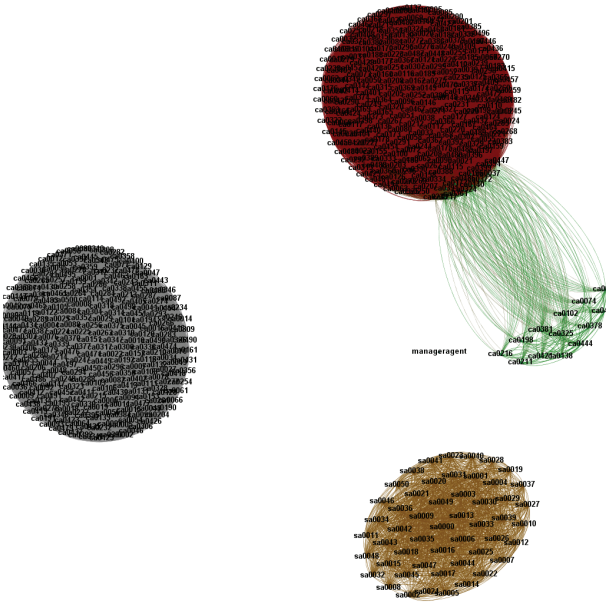

Figure 5: Agents' Communities Based on Behavioral Patterns, LCS (θ =0.3)

CONCLUSION

The paper deals with the analysis and visualization of the agents' behavior to facilitate the verification of the simulation model and to effectively observe the reaction of the model in relation to the initialization of the input attributes, or to the influence of the model environment. The proposed method was used for the determination, whether the agents which are included in the multi-

agent system have the appropriate behavior like has been programmed.

The agents' behavior was described using the behavioral patterns, obtained via sequential pattern mining methods and methods for the determination of relations between the sequences. Concretely, LCS, LCSS, and T-WLCS methods were compared in the experiments to discover, how the selected method can influence the finding of the agents with the similar behavior.

The detailed description of the outputs obtained using each method was presented in Section 4.2. From the obtained outputs we can see, that each method has its specific approach to the determination of the relations between the sequences. Due to this fact we can say, that each method can be used for different type of data collection. Moreover, the selection of the method depends on what process with selected behavioral patterns we intend to do.

In the cases, where we need more detailed division of sequence clusters to find more accurate behavioral patterns like in this data collection from the multi-agent system, LCS method is the most suitable. In other cases, where we have large data collections with various types of sequences (typical domains are analysis of users' behavior on the web, or analysis of business processes), method T-WLCS is better to use. This method is immune to minor distortions and to time non-linearity, and emphasizes the recurrence of the elements in one of the compared sequences. Finally, LCSS method generates the outputs more similar to LCS method, but more compact.

Visualized groups of agents with similar behavior during the simulation, extracted using LCS method is presented in Figure 5

As presented in the described experiments, LCS method is the most suitable for the presented data collection. However, we assume more diverse output in future experiments with proposed simulation model, due to planned simulations with more variable initialization of input parameters of agents. Therefore, T-WLCS or LCSS method will be more suitable in the future experiments with MAS.

ACKNOWLEDGMENT

This work was supported by SGS, VSB – Technical University of Ostrava, Czech Republic, under the grant No. SP2012/151 Large graph analysis and processing.

REFERENCES

Aalst, van der W. M. P., 2011. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Heidelberg: Springer.

Aalst, van der W. M. P., Reijers, H. A., and Song, M., 2005. Discovering social networks from event logs. *Computer Supported Cooperative.Work*, 14(6): 549–593.

Bellifemine, F., Caire, G. and Trucco, T., 2010. Jade Programmer's Guide. *Java Agent Development Framework*. Available from: http://jade.tilab.com/doc/ programmers guide.pdf. [Accessed 16 January 2012].

De Snoo, D., 2005. *Modelling planning processes with TALMOD*. Master's thesis, University of Groningen

FIPA 2002. FIPA Contract Net Interaction Protocol Specification. Available from: http://jadex.informatik.uni-hamburg.de/docs/jadex-0.96x/userguide/predef_cap.html [Accessed 12 July 2012].

Guo, A. and Siegelmann H., 2004. *Time-Warped Longest Common Subsequence Algorithm for Music Retrieval*. Universitat Pompeu Fabra.

Gusfield, D., 1997. *Algorithms on Strings, Trees and Sequences: computer Science and Computational Biology*. Cambridge University Press.

Hirschberg, D.S., 1977. Algorithms for the Longest Common Subsequence Problem. In *J. ACM,* 24(4), 664-675.

Jennings, N.R., Faratin, P., Norman, T.J., O'Brien, P. and Odgers, B., 2000. *Autonomous agents for business process management*. Int. Journal of Applied Artificial Intelligence 14, pp. 145–189

Islamov, R. and Olkov, A., 2010. The Use of Analytical-Statistical Simulation Approach in Operational Risk Analysis. In: *Enterprise Risk Management Symposium.* Society of Actuaries.

Macal, C.M. and North, J.N., 2005. Tutorial on Agent-based Modeling and Simulation. In *Proceedings: 2005 Winter Simulation Conference*.

Martinovic, J., Drazdilova, P., Slaninova, K, Kocyan, T., and Snasel, V., 2012. Left-Right Oscillate Algorithm for Community Detection used in E-learning system. In *IEEE Proceedings: 11th International conference Computer Information Systems and Industrial Management Applications, CISIM '12.* IEEE Computer Society. In print.

Müller, M., 2007. *Information Retrieval for Music and Motion*. Springer.

Scheer, A.W. and Nuttgens, M., 2000. ARIS architecture and reference models for business process management. In *Business Process Management, LNCS.* 1806: pp. 376–389.

Vymetal, D., Spisak, M. and Sperka, R., 2012. An Influence of Random Number Generation Function to Multiagent Systems. In *Proceedings: LNAI 7327: Agent and Multi-Agent Systems: Technology and Applications*. KES AMSTA 2012. Berlin Heidelberg: Springer-Verlag, Germany. pp.340-349. ISSN 0302-9743. ISBN 978-3-642-30946-5. DOI 10.1007/978-3-642-30946-5. Available from: <http://www.springerlink.com/content/g71k68505h76x1wx/>.

Yan, Y., Maamar, Z. and Shen, W., 2001. *Integration of Workflow and Agent Technology for Business Process Management*.

AUTHORS BIOGRAPHY

K. Slaninová is a PhD student at VŠB – Technical University in Ostrava, Faculty of Electrical Engineering, and an assistant professor at the Silesian University in Opava, School of Business Administration in Karviná, Department of informatics.

J. Martinovič is an associate professor at VŠB – Technical University in Ostrava, Faculty of Electrical Engineering.

P. Dráždilová is an assistant professor and PhD student at VŠB – Technical University in Ostrava, Faculty of Electrical Engineering.

D. Vymětal is a vice dean at the Silesian University in Opava, School of Business Administration in Karviná, and a head of department of informatics.

R. Šperka is an assistant professor and PhD candidate at the Silesian University in Opava, School of Business Administration in Karviná, Department of informatics.