Simulation on the Web with Distributed Models and Intelligent Agents

András JÁVOR, Attila FŰR

McLeod Institute of Simulation Sciences
Hungarian Center

Budapest University of Technology and Economics
H-1111 Budapest, Műegyetem rkp. 3-9.
Hungary
E-mail: javor@eik.bme.hu, fuat@eik.bme.hu
Tel.: +36 1 463 1987

Keywords: agent-controlled simulation, distributed models, web-based simulation

I. INTRODUCTION

The application of simulation as the up-to-date tool for problem solving in a widening range of applications is growing exponentially in recent years. The Society for Modeling & Simulation International (SCS) played a leading edge role in the field. Within the society the McLeod Institute of Simulation Sciences (MISS) has been established to form a network of research centers where simulation tools and methodologies have been developed and applied for solving problems by simulation in various fields. In the present paper we intend to outline a very important R&D work undertaken within the MISS network. The aim of the work was to introduce intelligent actors controlling and modifying simulation experiments with regard to the operation as well as the structure of the models and the experimental environments. The main concepts of the methodology have been elaborated and introduced in the Hungarian MISS Center. Joint EU [1], Transatlantic [2] and other [3] projects have been realized with the participation of other MISS Centers of the network.
The novel concept enabling to use objects with AI controlling simulation in various ways and in a great variety of fields was called *demon controlled simulation.* Later the intelligent objects mentioned above were quoted as *agents* [4].

Early history of simulation controlled by intelligent objects

The procedure of problem solving by simulation is actually the building of models representing the real world (or an anticipated one), executing its dynamic operation, evaluating its performance and modifying it if necessary. This procedure is repeatedly executed until the acceptable results obtained from the dynamic simulation model are reached. In this procedure the dynamic operation of the model can be obtained fast since this is controlled by the computer that can be rather fast in comparison with the process of evaluation and the consequent modification of the model performed by the human investigator. This latter process is relatively slow in the man-machine system described above.

An early development to control the simulation experiments by AI was the investigation of digital logic networks using simulation that was implemented in the LOBSTER (Logic Operational Behavior Simulator for Time and Effort Reduction) simulation system [5].

The introduction of *intelligent demons* (called intelligent agents today) [6][7][8] to control simulation experiments was introduced by the McLeod Institute of Simulation Sciences Hungarian Center as it was stated by Professor Tuncer Oren who invited the director of the MISS Hungarian Center to present an invited paper about the intelligent demons the forerunners of intelligent agents [4]. The agents have been implemented in a general purpose object oriented simulation system and applied for problem solving in a wide range of fields of applications as it is described in details in section II. Concerning the development and application of the intelligent agents (demons) since 1996 ten papers were published at SCS conferences by the McLeod Institute of Simulation Sciences Hungarian Center.

<u>Various developments in the field [9][10[11][12]</u>

To overcome the problems of the inefficiency of the human part of the man-machine system in simulation attempts have been made to replace the human actor of the system mentioned above by artificial intelligence. An early result achieved in this field was the MAGEST system developed by Tuncer Oren and his associates [13]. After that the combined application of AI and simulation methodologies has started.

Martin Wildberger already in 1995 in the journal Simulation dealt with AI & Simulation describing the importance of applying the two disciplines in combined form for solving problems in various fields of application. The topic has been discussed continuously in the columns of AI & Simulation continuously in the next years.

Since 1995 a few publications concerning intelligent agents applied in modeling and simulation have appeared at SCS conferences [14][15] and their number increased continuously. Since 2001 special sessions dealing with *agents in simulation* have been held at SCS conferences. A special issue of the journal Simulation was dedicated to "Software Agents and Simulation" in June 2001. In the journal Transactions of the Society for Computer Simulation International contributions concerning agents in simulation have also been published [16].

<u>The introduction of agent controlled simulation</u>

Agent controlled simulation provided essentially new and dramatically more efficient means. This enhanced the efficiency of simulation in a number of ways by using AI.

The main possibilities which can be implemented are the following.

(i) Building object oriented models that may form a network of elements where the individual blocks can be intelligent entities with data bases and inference engines.

(ii) The intelligent entities may modify other objects within the model or the network structure of a whole.

(iii) External intelligent objects may monitor the operation of the simulation model and modify its structure (i.e. the model network) or the individual elements in it.

(iv) Mobile intelligent objects moving within the model network during simulation may interact with the model structure or each other [17]. The latter possibility may represent human actors or messages in an information network [18][19].

The principles mentioned above have been implemented in the CASSANDRA (Cognizant Adaptive Simulation System for Applications in Numerous Different Relevant Areas) 3.0 simulation system by means of which numerous problems have been solved. Among others it has been applied for problem solving in projects in numerous different fields [1][3][20]. The new version of the simulator CASSANDRA 4.0 is under development. The main feature of the new version is a simulator structure that can be distributed of the web.

Simulation is aimed very often to solve problems of great complexity requiring – beyond using the advanced simulation software tools – platforms which enable the implementation of such software systems. In recent years the concept of *cloud computing* emerged and is applied more and more widely for solving such problems [21].

II. THE MAIN CONCEPTS OF AI CONTROLLED SIMULATION

Let us briefly assume the main benefits which can be gained by applying the concepts of AI controlled simulation mentioned above.

II.1. Intelligent Agents (Demons)

In Figure 1 the basic structure of an agent and its interconnection with a simulation model can be seen. The agent collects information from the model during the dynamic simulation run and after possible preprocessing and using its knowledge base intervenes into the model modifying its structure and/or parameters in accordance with its knowledge base and inferencing procedure.

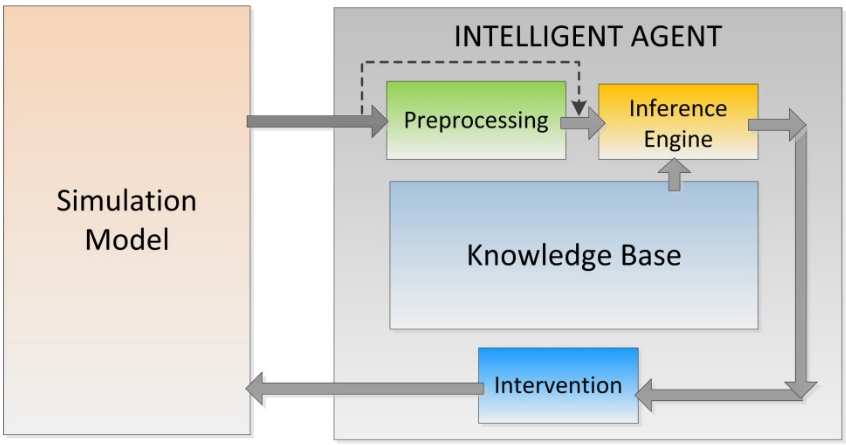

Figure 1: The interconnection of an agent with the simulation model

In Figure 2 two different applications are depicted. One depicts the case when an external intelligent agent influences the model in accordance with the aims provided externally by the user. The other agent is part of the model. This means that the model contains an internal control realized by an intelligent entity (or possibly more of them). The modifications may effect the individual elements of the model (their parameters and/or algorithms), the topology of the network (i.e. the interconnection of the individual elements) or global information concerning the model as a whole.

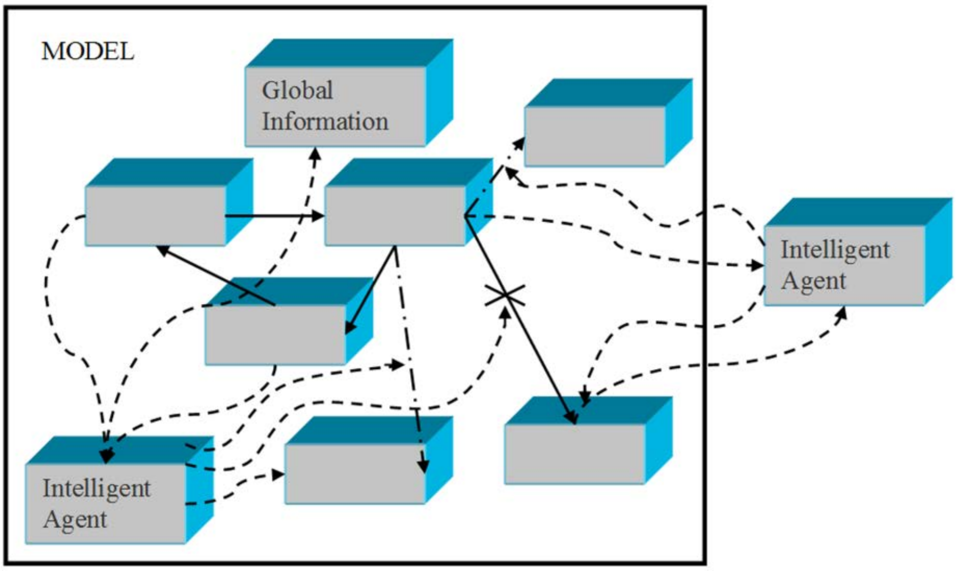

Figure 2: The possible interconnections of the model and the agents

II.2. Identification by reconstruction

In a number of cases the structure, parameters and the algorithms describing the operation of the system to be investigated are only vaguely known. These are to be found – among others – in micro- and macroeconomic systems. Sometimes even the behavior of the systems themselves are not strictly defined [22][23] In these cases a methodology of model identification by reconstruction applying intelligent agents (demons) can be applied. Initially a model describing the soft system – based on the available theories and assumptions – is built (see Figure 3). It is assumed that the model is built as a network of elements (i.e. objects) thereby it describes the real system in a natural way. After that an intelligent agent is applied that observes the trajectory of the operation of the model during the simulation run. As an input to the model historical data is provided and the agent compares it with the historical behavior of the system.

In case the agent finds significant deviations it may change the model until acceptable correspondence between reality and the model is achieved. The changes which the intelligent agent can execute are the following.

(i) Change the topology of the model network
(ii) Change the parameters of the model elements
(iii) Change the functions describing the effects of one model element on an other

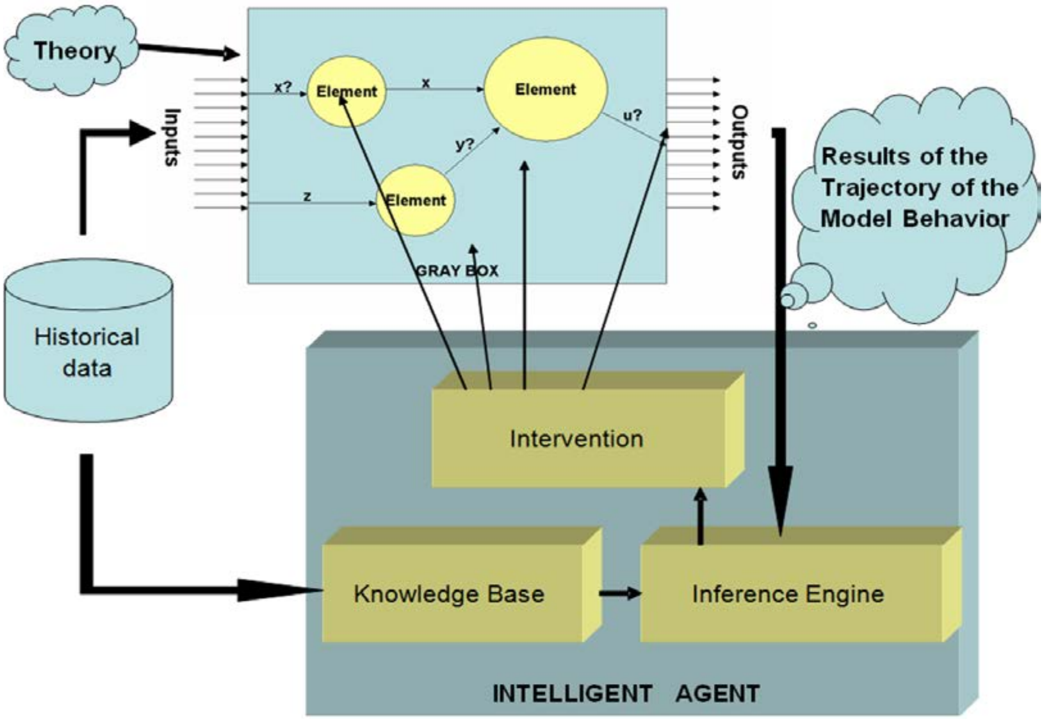

Figure 3: Identification by reconstruction using an intelligent agent

After having reconstructed the initial model to represent the system to be investigated adequately the problem identification by reconstruction has been solved. Thereafter the forecasting of the results of further actions to be taken by the decision makers can be undertaken according the structure shown in Figure 4. Here the intelligent agent represents the decision maker, the anticipated data the future effects anticipated and the decision maker obtaining the aims and the responses of the model after having intervened may achieve optimal solutions.

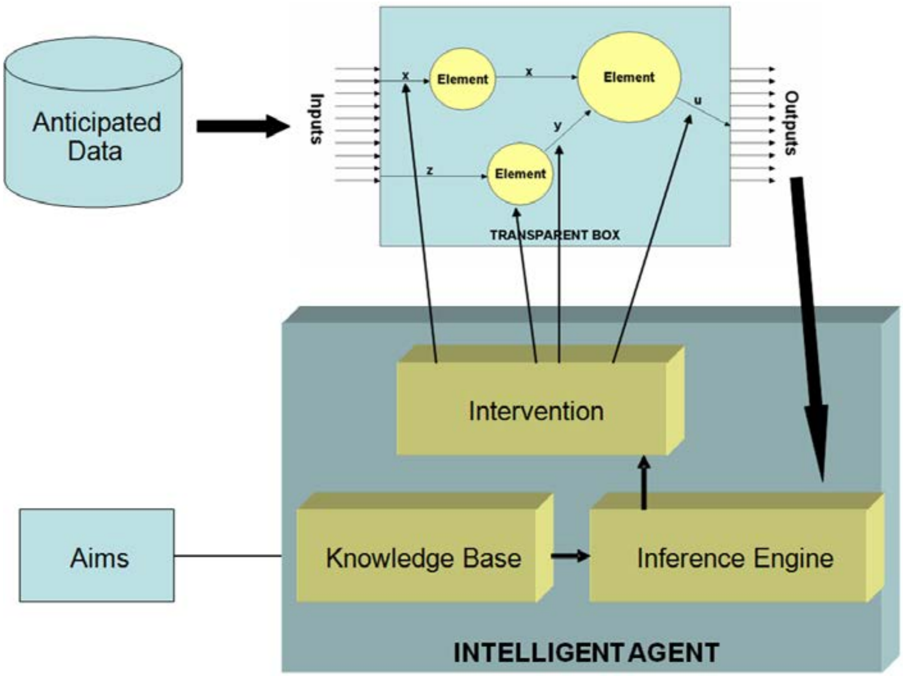

Figure 4: An intelligent agent representing the decision maker to find optimal solutions

II.3. Mobile intelligent entities (KAPN)

A methodology to enhance the efficiency of simulation was the introduction of Knowledge Attributed Petri Nets [17]. Here knowledge attributed tokens are used. In these cases one or more knowledge base(s) – represented mostly as frame structures – are assigned to the tokens. The transition may look into the knowledge bases of the tokens residing in the input places and may write into the knowledge bases of the tokens created in the output places. Thus information residing in the knowledge bases of the tokens may move in the model network represented by the Knowledge Attributed Petri Nets.

The propagation of information within the model can be simulated effectively by using the above methodology. The modeling and simulation of communication networks e.g. can be undertaken this way. Another important possible application is the simulation of systems characterized by human communication networks as e.g. advertisements, marketing, politics, etc. The investigation of such systems may result in large financial benefits. This can also be solved using simulation by building and operating models representing the generation and propagation of memes applying the principles of memetics [18][19]. There are numerous other fields where this methodology can be used effectively.

II.4. The main structure of the CASSANDRA system [24]

The CASSANDRA simulation system enhances the effectivity of simulation by automating the control of simulation experiments using demons. The structure of the system is shown in Figure 5.

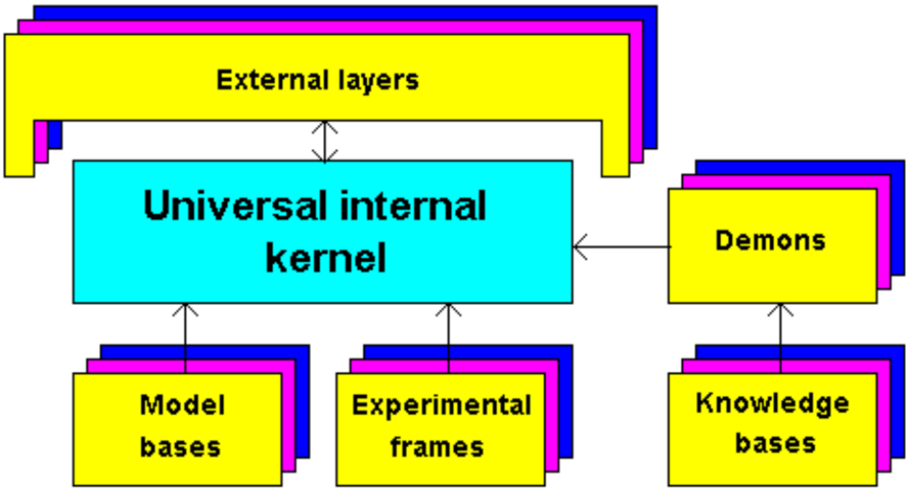

Figure 5: Structure of the CASSANDRA simulation system

The most important features of the CASSANDRA system are the following:

- A universal kernel is provided based on which problem oriented simulators specially customized for special requirements and man/machine communication style of the given application fields can be supplied within a short time and at reasonable cost.

- Automated control of modeling. The entire recursive process of the simulation experiment is controlled by methods of artificial intelligence. Demons using knowledge bases and inference engines evaluate the results of dynamic simulation continuously and modify the experimental conditions, model structure, model parameters as required to achieve the goals intended.
- Non-procedural models revealing the topological structure of the model described as a network of model elements (and not in the form of the conventional procedural program description). This means that the model is not represented as a single sequence of instructions, but rather as such a structure in which parallel interacting procedures occur the handling of which in simulated time the simulator automatically takes care of. In this way a model directly conforming to the real modeled system is obtained, revealing the original one in a natural way.
- Separate model building and dynamic simulation phases ensure the investigations of the same model under different conditions as well as investigating different models under the same conditions.
- Handling of both level type events that can be described by variables and entity type events that can be described by the location of mobile elements within the model structure. In other words, in a real system the events can be classified into these two subclasses. The output level of a logic circuit or the traffic light can be regarded, for example, as level type, while the location of a lorry in a transport system or a message in a computer network can be regarded as an entity type event.
- Monitoring the process of dynamic simulation with demons can modify the experimental conditions, the model structure and parameters.

III. SIMULATION ON THE GRID

III.1. Basic concepts and taxonomy of distributed simulation

Due to increasing complexity of models computational capacity is required that can be effectively provided by distributed resources on computer networks. During the last decades several approaches have been introduced according to the development of software framework solutions and hardware network infrastructure. Also several standards had been created aiming at the description the methodology of distributed simulation systems and their components. The most prevalent approach called HLA (High Level Architecture) is defined under the IEEE 1516-2010 Standard [25].

HLA provides robust frame for the development of distributed simulation systems – enabling the establishing of so called *federates*[1] from simulation entities – and gives a common architecture to the integration over a grid.

Based on the various ways of usage and the HLA Standard the following main concepts of distributed simulation appeared [26]:

- Grid-Facilitated Approach – This architecture ensures services over the grid that facilitate the execution of the HLA-based simulation model

[1] Federate is set of simulation entities that participates the execution of simulation

- Grid-Enabled Approach – In that case a client-server architecture is present, where the server side joins other HLA compliant simulation systems through vendor-specific RTI[2] (Run-Time Infrastructure)
- Grid-Oriented Approach – In that approach RTI is defined through web-services and all communications of simulation federates are realized through direct invocations addressed to them
- Non-HLA Approaches – Distributed systems that do not follow the IEEE 1516-2010 Standard.

Currently one of the most advanced approaches is the SOHR (Service Oriented HLA RTI) framework that implements the RTI entirely using Grid services [27].

III.2. Web-based simulation

In case when the grid of simulation system elements overgrows the size of local or wide area networks (LAN or WAN) and the domain of interpretation can be extended to the whole internet Web-based simulation can be suggested. The Web (originally World-Wide Web, or W3 [28]) – often confused with the Internet that stands for a network of networks – has been developed at CERN (European Organization for Nuclear Research) aiming at the collaboration on remote sites to share data in a standardized form. The Web is a system of interlinked hypertext documents that can be accessed via Internet therefore the Web can be understood as a subset of the Internet. The Web has been standardized and transformed since its appearance and new technologies have been introduced in order to assist the client-server architecture considering role-centric and load-balancing approaches. Actually in our days the Web is understood in conjunction with its complementary technologies.

Web-based simulation – having respect for theoretically equal entities on the Internet – can take place either on the server side or on the client side. In the classic case, in server-side simulation the execution of simulation experiment e.g. the numerical calculations, timing, event monitoring is carried out on the simulator server (server-cluster) located in a remote place.

The client-side has to fulfill some visualization requirements – including the interactive graphical user interface (GUI) that ensures model managing functions or visualization of simulation results (e.g. graphs, plots, tables) – and also some other interfaces connected to client devices, or local databases.

The classic client-server approach, the *simulation system as service* can be seen in Figure 6. The clients are connected to the simulation system service through a secure tunnel with proper authentication opening simulation sessions. On multiple client connection each simulation session owns its separated threads of data exchange and state of interactions. Users connected can share the resources of the server-side fully functional simulation system with centralized or distributed computational units (e.g. server computers), data, or procedures. The simulation system contains the simulator core, the knowledge bases (databases) and inference engines for agent prototypes and the system also provides interface to users (generating the GUI website) and gives connections to other applications.

[2] RTI is a middleware (e.g. interface) that ensures coordination of data exchange and synchronization of operations during simulation run

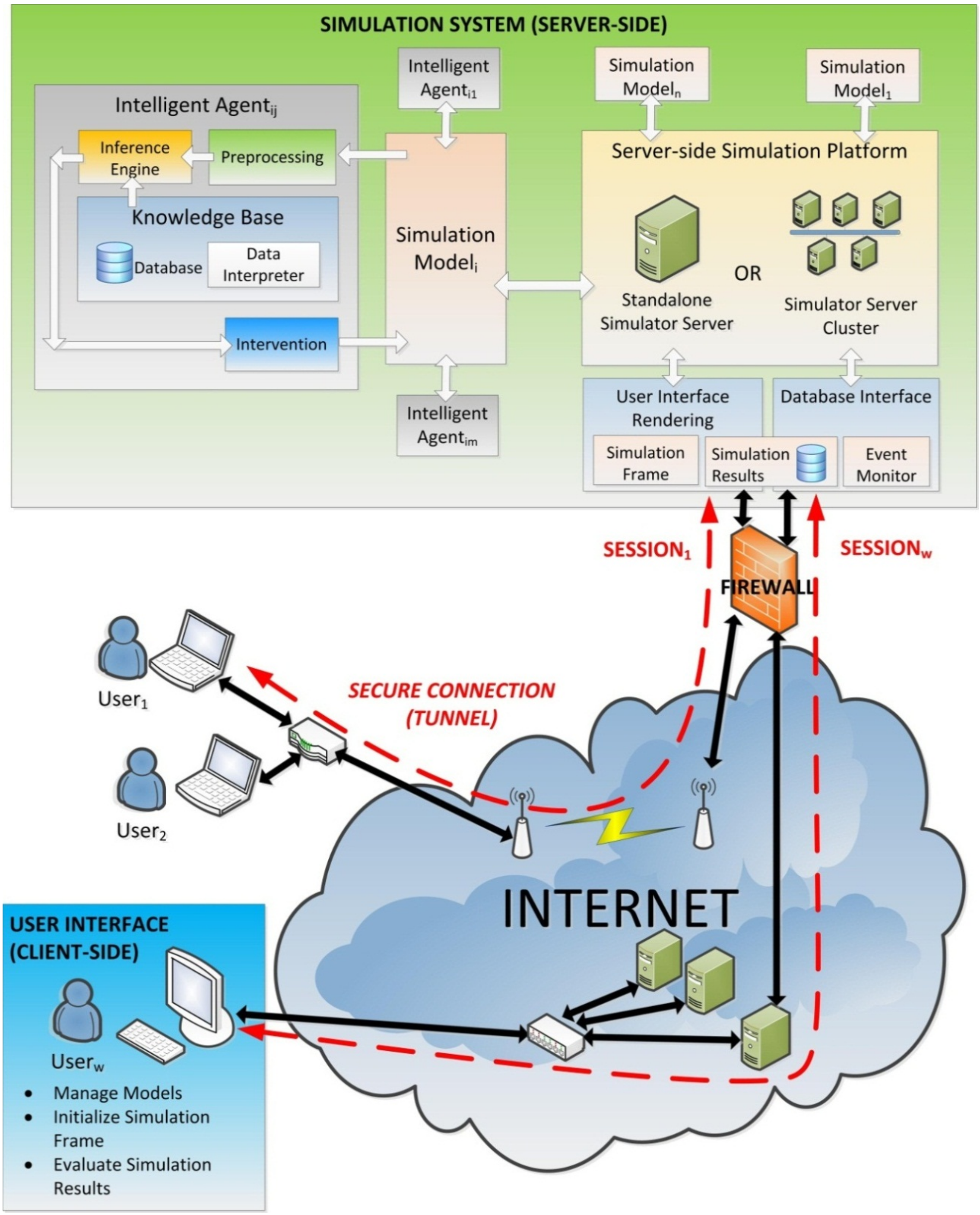

Figure 6: Classic client-server approach of a simulation service

There are several ways how clients can access the server resources:

- offline (outside simulation run)
- online (read, write, modify during simulation run)
- having exclusive access (unique instance of resources assigned to each simulation session, e.g. experimental datasets are cloned)

- having concurrent access (two or more clients accessing a common resource e.g. work database of the simulation model)
    - optimistic locking (System is caching data and version information of current state, before write/modification operation is checking version information. If version information is changed in between then the planned changes are dropped else write-back to resource is carried out.)
    - pessimistic locking (During write/modification cycle only one user can access the given resource, other users have to wait in execution queue until the resource becomes accessible.) – This method has better calculation performance but can cause dead-lock situations.
- through roles (To each user one (or more) well defined role determining the accessibility to declared resources can be assigned)
    - static roles
    - dynamic roles (can change during the simulation run)
    - time-frame roles (are only valid during defined sets of intervals of simulation run)

The classic approach can easily be implemented using prevalent development of frameworks of client-server architecture. During the last decade the following preferred frameworks, languages has been used:

- CGI (Common Gateway Interface) [29]
- PHP (Hypertext Preprocessor) [30]
- ASP.NET (.NET based Active Server Page) [31]
- Interactive services based on AJAX or (Asynchronous JavaScript and XML) [32]
- Scriptable remote access services

In some cases it becomes necessary that the client-side also takes part in the execution of the simulation. In that kind of the client side-simulation, the executable simulator program is downloaded from the server side and is installed – on demand – on the client computer. The simulation run takes place on local recourses of the client. Main client-side technologies that can be used in simulation are:

- Java Applets [33]
- Flash animations [34]
- JavaScript [35]
- Web-browser plug-ins

III.3. Decomposition of simulation models and web-agents

The previous chapter highlighted how the first level of decomposition of a simulation system can be realized through the classic client-server approach. The second level aims at the functional decomposition of a simulation model and the delegation of the agents outside the simulation server to dedicated web-agent services. (see Figure 7.)

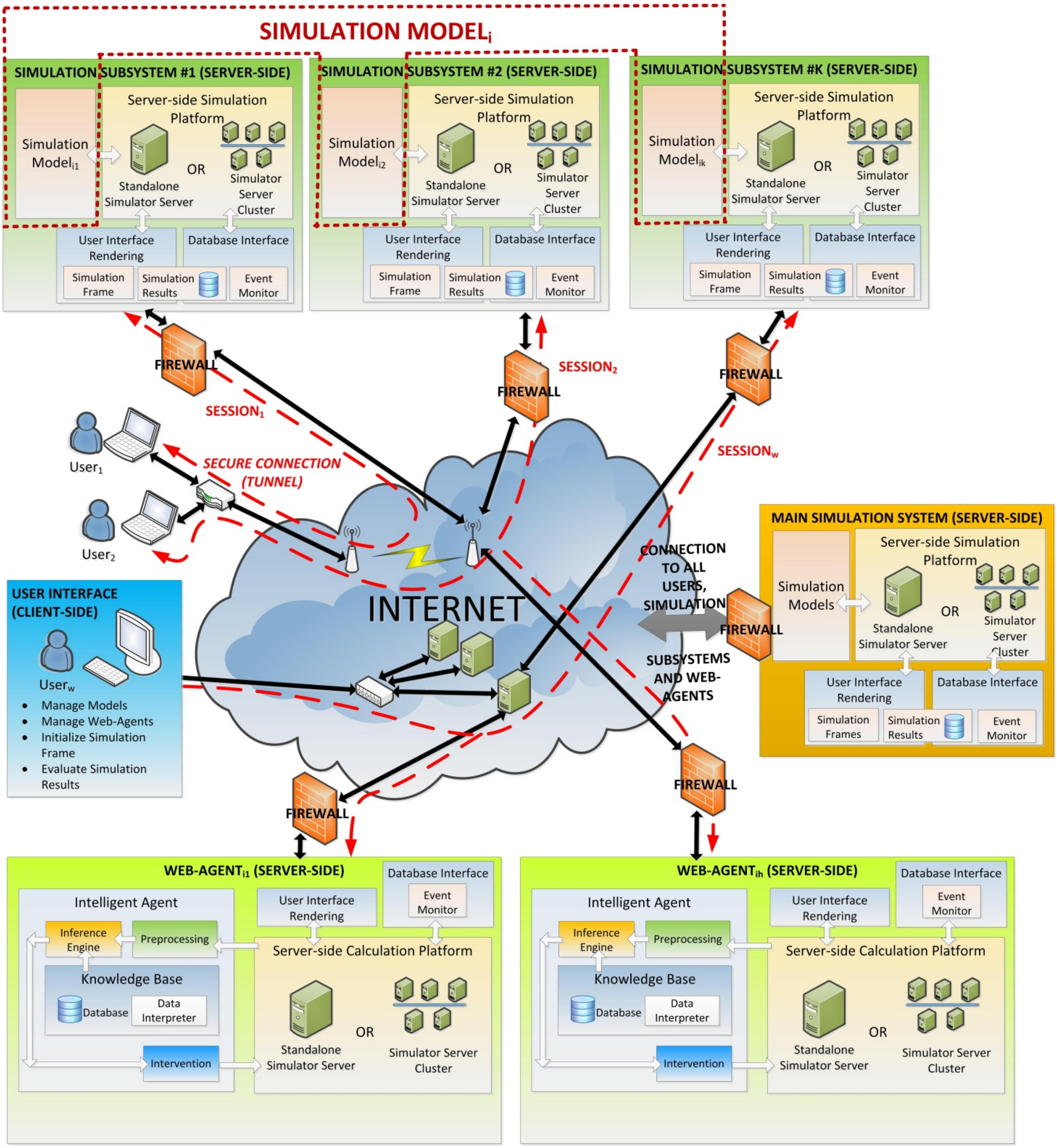

Figure 7: Decomposition of simulation models and web-agents

Often there is a need to partition the model, when:

- the model cannot be effectively simulated on one server-side simulation system (based on load-balancing or financial considerations)
- the parts of the model are owned by different organizations
- the parts of the model are physically separated in great distances
- the model is originally distributed by its nature

The existence of a *main simulation system* (see Figure 7) – that joins all partitions and web-agents – is suggestible (to accomplish synchronizing, coordinating, data routing tasks), but is not necessary [36].

On the other side there are also some cases, when the agent itself is located outside the server-side simulation system establishing *web-agents* (see Figure 7). Web-agent is a simulation entity that fulfills all the requirements of being an agent – having input data collection, preprocessing, inference engine based on knowledge base, structural and parametric intervention abilities on the output side – but works on dedicated resources, and behaves like a web-service. Web-services are sets of protocols and web-standardized software that can be accessed via Internet aiming at data exchange and processing between interoperable machines.

Web-agents can be part of the simulation model – realizing internal control – connecting to the simulation model from outside. This scenario can only be suggested if the implementation of an agent inside the simulation system would be less effective than the delegated one – considering the time-cost of the whole execution round of Web-agent (network latency, processing time, etc.) [37].

The other application of a Web-agent (see Figure 2) is the external intelligence that influences the model in accordance with some external criteria, boundary conditions, or optimization aims.

Web-agents are suggested to apply when:

- a simulation model is decomposed and an agent collects information from different simulation model parts
- if the database or knowledge base that is required to solve the job of the agent is extremely large and/or is confidential or contains sensitive data therefore cannot be moved or cloned or accessed in a normal way [38]
- the agent contains intervention modules that have strong connection to special software elements (e.g. traffic controllers, financial or Enterprise Resource Planning Systems)

III.4. Service-based simulation on the Web, Cloud-agents

In our days simulation models become more complex overgrowing simple knowledge or databases. Also interpretation of data and information requires expert teams and possible intervention methodologies can be carried out in numerous different ways. Investigating e.g. interdisciplinary problems by simulation methodologies often let us face the fact that the integration of agent elements (large databases, different novel concepts of inference engines, etc.) into one simulation entity is not always possible – and in some cases also not necessary.

In case when using of different knowledge bases, various inference techniques and different types of modeling and simulation is needed service-oriented approach can be suggested.
In service-oriented approach each decomposable part of the agent and also the simulator core are provided entirely through web-services. These services are maintained by different stake-holders such as owners of data-warehouses, research institutes, the government or concerned companies. According to the business model some of the web-services can be provided in

ASP[3] (Application Service Provider) approach, others stand under GPL (General Public License).

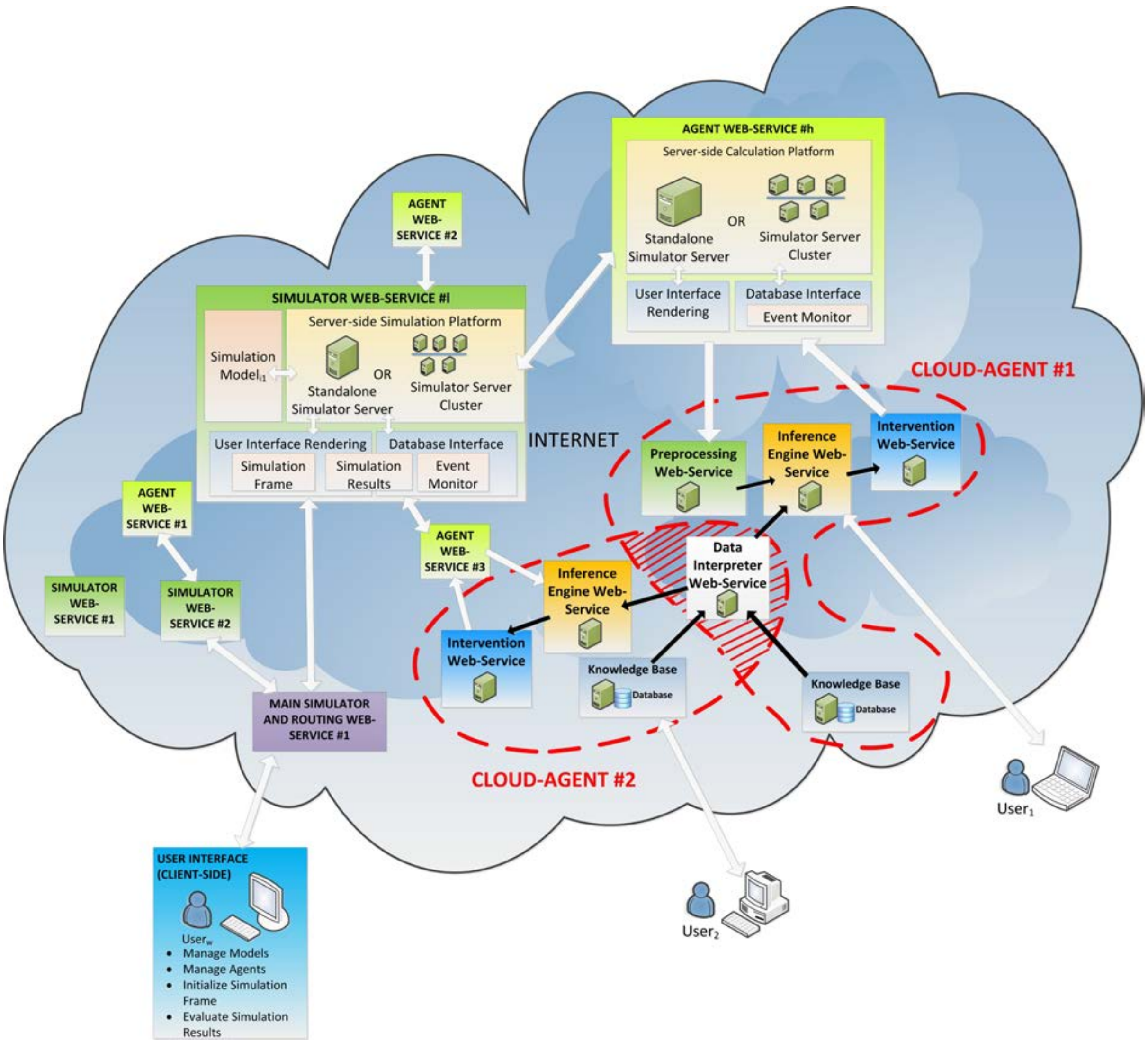

Figure 8: Simulator web-services and cloud-agents

As it is illustrated in Figure 8 that kind of decompositions have several advantages over the centralized approach:

- Services can be shared between virtual simulation entities (e.g. *cloud-agents*[4])
- Experts can separately access virtual parts of simulation model
- The simulation model is based on the most adequate knowledge of each scientific field therefore knowledge transfer from the origin of knowledge to the model can be easily realized.

The main disadvantage of the service-based distributed simulation is that the interconnected services have to follow web and simulation standards (HLA) strictly in order to manage their communication. As a remarkable challenge it can also be mentioned that in case of large

[3] ASP is a business that provides computer-based services to customers over a network
[4] Cloud-agents are sets of interacting web services behaving virtually like Web-agents

networks the handling of synchronization can become complex and sometimes it can overshadow the previously detailed advantages of the service-based simulation on the Web.

III.5. An application – Fuzzy Web Service[5]

As a sample application to a service-based inference engine *Fuzzy Web Service* (FWS) has been developed by the McLeod Institute of Simulation Sciences Hungarian Center. This service is an ASP.NET 4.0 and MS SQL[6] 2008R2 based web-service that can be accessed via web-browser or on demand secure VPN[7] tunnel. FWS has GUI through that the parameters of the fuzzy reasoning can be adjusted.

FWS ensures an authenticated multi-user environment providing security roles to determine rights to access to different projects or to parts of those. Projects contain all information about the Fuzzy Inference Services hosted in FWS.

Due to its implementation under SQL server framework FWS fulfills all the requirements of being able to interact with other web-services on the Internet respecting standards, and hosting other knowledge or databases therefore it can be considered as a part of a Cloud-agent as well as an independent fully functional Web-agent.

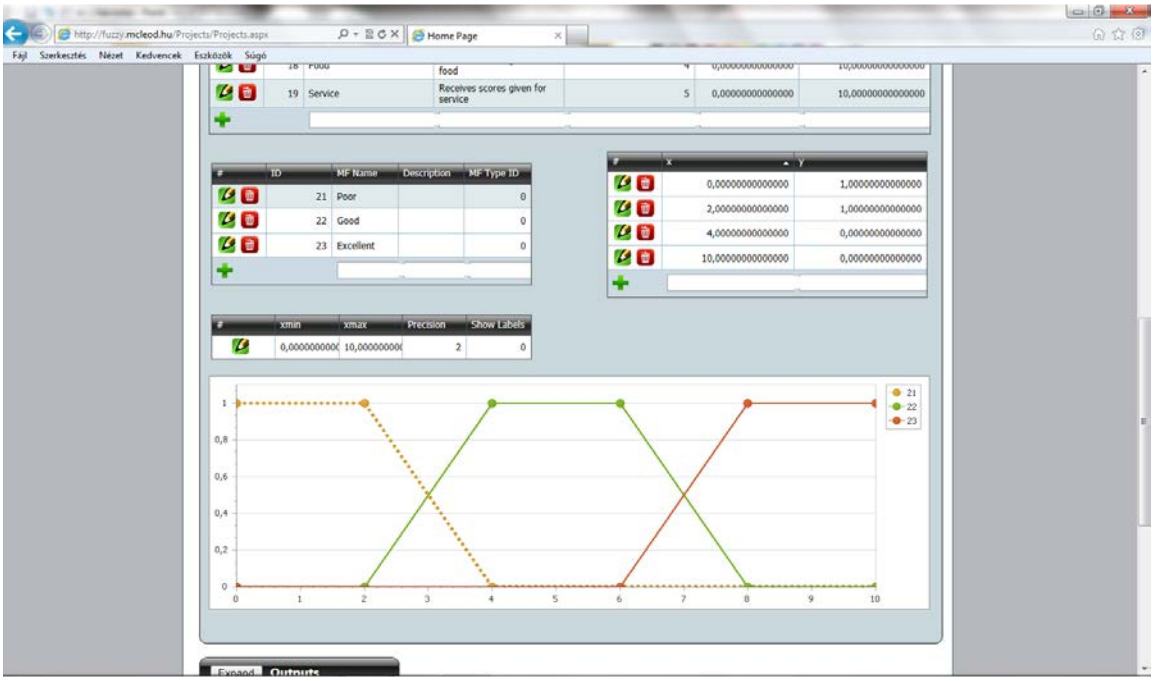

Figure 9: Screenshot of GUI of Fuzzy Web Service

The main areas of application of FWS are the followings:

- improvement of availability of services
- monitoring production, financial and logistic processes

[5] http://fuzzy.mcleod.hu
[6] Microsoft SQL relational database system
[7] Virtual Private Network – ensures secure remote access to computers on the Internet like local area network

- medical applications (managing medical databases)
- decision support system for governmental organizations or enterprises in different multidisciplinary questions

IV. CONCLUSIONS

As this paper delineates that there are several ways how distributed simulation systems can be organized over the Web. One of the most effective methodology is the service-oriented approach that provides flexible, scalable and robust frame for implementing distributed simulation systems.
The McLeod Institute of Simulation Sciences Hungarian Center is committed to develop a world-wide network of cooperating simulation web services based on the CASSANDRA 4.0 simulation system to investigate significant questions of sustainability, economy and industrial systems. As a sample application a new Fuzzy Web-agent (and Cloud-agent) has been introduced that is currently used related to measuring renewable energy potential in regions of Hungary and enabling to support local resource plans for governmental organizations. A research related to the adaptation to climate change applying Web-agents is also undertaken in the framework of the project "Talent care and cultivation in the scientific workshops of BME". This project is supported by the grant TÁMOP-4.2.2.B-10/1—2010-0009.
The distributed version of the Knowledge Attributed Petri Net and intelligent agent based simulator the CASSANDRA 4.0 system is under development following the recommendations of IEEE 1516-2010 Standard.

We do hope that the R&D work outlined above will contribute to strengthen the recognition of the McLeod Institute of Simulation Sciences within the Society for Modeling & Simulation International.

ACKNOWLEDGEMENTS

The authors of this paper i.e. Prof. András Jávor director of the McLeod Institute of Simulation Sciences as well as the Hungarian Center and Attila Fűr vice director of the Hungarian Center would like to acknowledge to those colleagues in the Center who participated in R&D work developing the methodology, tools and its applications.

REFERENCES

[1] Jávor, A., Szûcs, G.: Simulation and Optimization of Urban Traffic using AI, Mathematics and Computers in Simulation 46 (1998) 13-21.

[2] Roy Crosbie, Dietmar Moeller, David Murray-Smith, Hamid Vakilzadian, Bernard Schroer, Andras Javor: An International Network for Collaboration in Engineering and Science Education, International Conference on Engineering Education – ICEE 2007, Coimbra, Portugal, September 3-7, 2007

[3] Jávor, A., Szűcs, G.: AI Controlled Simulation of the Complex Development of Regions, Summer Computer Simulation Conference, Chicago, Illinois, July 11-15, 1999. 385-390.

[4] Jávor, A.: Demons as Forerunners of Software Agents in Simulation – Invited paper 2006, SCS International Conference on Modeling and Simulation - Methodology, Tools, Software Applications (M&S-MTSA'06), Calgary, Canada, July 31- August 2. 151-155.

[5] Jávor, A.: Applications of Expert Systems Concepts to Adaptive Experimentation with Models, Chapter III.3. in M.S. Elzas, T.I. Ören, B.P. Zeigler (eds.): Modelling and Simulation Methodology in the Artificial Intelligence Era, Elsevier, 1986. 153-163.

[6] Jávor, A.: Adaptive FMS Analysis and Design by Demon Controlled Simulation, Proc. of the Fourth International Conference on CAD, CAM, Robotics and Factories of the Future, New Delhi, India, Dec. 19-22, 1989, Vol. I. 157-172.

[7] Jávor, A.: Demons in Simulation: A Novel Approach, Systems Analysis, Modelling, Simulation 7(1990)5. 331-338.

[8] Jávor, A.: Demon Controlled Simulation, Mathematics and Computers in Simulation 34(1992)3-4, 283-296.

[9] Elzas, M.S., Ören, T.I., Zeigler, B.P. (eds.): Modelling and Simulation Methodology in the Artificial Intelligence Era, Elsevier, 1986. 153-163.

[10] Klir, G.J., Way, E.C.: Reconstructibility Analysis: Aims, results, Open Problems, System Research 2(2), Pergamon Press, 1985, 141-163.

[11] Unger, B., Fujimoto, R. (eds.): Distributed Simulation, Proc. of the SCS Multiconf., 28-31 March, 1989, Tampa, Florida

[12] Yilmaz, L. and Ören, T. (Eds.): Agent-Directed Simulation and Systems Engineering, Wiley Series in Systems Engineering and Management. ISBN: 978-3-527-40781-1. Wiley, 2009.

[13] Ören, T.I., Aytac, K.Z.: Architecture of MAGEST: A Knowledge-Based Modelling and Simulation System, In Jávor, A. (ed.), Simulation in Research and Development, North-Holland, Amsterdam 1985, 99-109.

[14] Ören, T.I., Abdullah, B.: Integration of Visual Intelligent Software Agent Gauges (VISAG) into a Simulation Environment, Proc. of the Summer Computer Simulation Conference, Ottawa, Canada, July 24-26, 1995. 269-274.

[15] Conti, C.M.: Practical Use of Agents and Brokering in DIS, Proc. of the Summer Computer Simulation Conference, Ottawa, Canada, July 24-26, 1995. 621-626.

[16] Yilmaz, L. and Ören, T. (Eds.): Agent-Directed Simulation, Special Issue of the Simulation: Transactions of the Society for Computer Simulation International. vol. 81, no. 7. July, 2005.

[17] Jávor, A.: Knowledge Attributed Petri Nets, Systems Analysis, Modelling, Simulation, 13(1993)1/2, 5-12.

[18] Heylighen, F., Memetics, Principia Cybernetica, 1994.

[19] Jávor, A.: Simulation of Memetics by means of Knowledge Attributed Petri Nets, Summer Computer Simulation Conference (SCSC'08), Edinburgh, U.K., June 16-18, 2008, 487-493.

[20] Jávor, A.: Simulation with Embedded AI for Transdisciplinary Problem Solving, International Journal of Modeling, Simulation, and Scientific Computing 1(2010)1, 85-98.

[21] Armbrust, M. et.al.: Above the Clouds: A Berkeley View of Cloud Computing Electrical Engineering and Computer Sciences, University of California at Berkeley, Technical Report No. UCB/EECS-2009-28, February 10, 2009.
http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html

[22] Zadeh, L.A.: Fuzzy Sets Inf. Contr. 8, 338 (1965)

[23] Jávor, A.: A Representation of Fuzzyness in Discrete Simulation, Problems of Control and Information Theory 7(2) (1978) 111-114.

[24] Benkõ, M., Jávor, A., Vigh, Á.: Cognizant Adaptive Simulation System for Applications in Numerous Different Relevant Areas, IMACS European Simulation Meeting on Problem Solving by Simulation, Esztergom, Hungary, August 28-30, 1990, 13-16.

[25] http://standards.ieee.org/findstds/standard/1516-2010.html

[26] Pan, K., Turner, S.J. Cai, W. and Li, Z.: A Service Oriented HLA RTI on the Grid, Proc. 2007 International Conference on Web Services, 2007. pp. 984-992.

[27] http://pdcc.ntu.edu.sg/sohr/

[28] http://www.w3.org/

[29] http://www.w3.org/CGI/

[30] http://www.php.net/

[31] http://www.asp.net/

[32] http://www.w3schools.com/ajax/default.asp

[33] http://www.java.com/en/

[34] http://www.adobe.com/products/flashplayer.html

[35] http://www.w3schools.com/js/

[36] Park and R. M. Fujimoto, Parallel Discrete Event Simulation on Desktop Grid Computing Infrastructures, International Journal of Simulation and Process Modeling, Vol. 5, No. 2, 157-171, 2009.

[37] Malik, A.W., Park, A.J., Fujimoto, R.M.: An Optimistic Parallel Simulation Protocol for Cloud Computing Environments, SCS Modeling and Simulation Magazine, Society for Modeling and Simulation, Intl., Vol. 1, No. 4, October 2010.

[38] Chen, D., Theodoropoulos, G.K., Turner, J.S., Cai, W., Minson, R., Zhang, Y.: Large Scale Agent-based Simulation on the Grid, Future Generation Computer Systems, Vol. 24, Issue 7, July, 2008

Authors' Biographies

András Jávor holds the degrees M.Sc.E.E., Ph.D. and D.Sc. in computer science. His main research field is simulation. At the beginning of his carrier he started to work at the Central Research Institute for Physics of the Hungarian Academy of Sciences dealing with nuclear electronics, data transmission and leading the national project of the CAD of microelectronic systems. After that he was invited as professor to the Budapest University of Technology and Economics, and simultaneously to the Széchenyi István University, where he was also dean of the faculty for informatics. He is director of the McLeod Institute of Simulation Sciences (MISS) and also director of the MISS Hungarian Center. He is professor emeritus at the Budapest University of Technology and Economics. He is chairman of the Hungarian Simulation Society and of IMACS/Hungary, member of the Board of Directors of EUROSIM and the editorial boards of 6 international scientific journals. His publications exceed 170 items. He was visiting professor 3 times at the Aachen Technical University and invited lecturer several times at various universities in China, Austria, the Netherlands and Japan. In 2006 he was awarded the Gold Cross of Merit of the Hungarian Republic for his internationally acknowledged scientific results in simulation sciences.

Attila Fűr holds the degree M.Sc.E.E. He is research fellow at the Budapest University of Technology and Economics Department of Environmental Economics. His main research field – beyond planning sustainability by using AI controlled simulation – is developing novel concepts of agent controlled discrete simulation methodologies especially related to the fields of Petri Nets. He was awarded three times at Scientific Conferences. He is the vice director of the McLeod Institute of Simulation Sciences Hungarian Center.