A DECISION SUPPORT SYSTEM FOR INTERMODAL TRANSPORTATION NETWORKS MANAGEMENT

Maria Pia Fanti[(a)], Giorgio Iacobellis[(b)], George Georgoulas[(c)], Chrysostomos Stylios[(d)], Walter Ukovich[(e)],

[(a)]Dept. of Electrical and Electronic Engineering - Polytechnic of Bari
[(b,d,)]Dept. of Informatics and Telecommunications Technology, Technological Educational Institute of Epirus
[(c)]Teorema Engineering S.r.l., Area Science Park Basovizza, Trieste, Italy
[(e)]Dept. of Engineering and Architecture- University of Trieste

[(a)]fanti@deemail.poliba.it, [(b)]iacobellis@deemail.poliba.it, [(c)]georgoul@teleinfom.teiep.gr, [(d)]stylios@teiep.gr,
[(e)]ukovich@units.it,

ABSTRACT

This paper specifies a Decision Support System (DSS) devoted to manage Intermodal Transportation Networks (ITN). With the aim to support decision makers in the management of the complex processes in the ITN, we describe the architecture of a DSS and its main components. In order to obtain a generic DSS, we employ the Unified Modeling Language (UML) to describe the components and the architecture of the DSS and we apply the solutions based on Service Oriented Architecture, the simulation drivers and window services.

Keywords: Decision Support System, Intermodal Transport Network, Unified Modeling Language, Simulation.

1. INTRODUCTION

An Intermodal Transportation Network (ITN) is defined as a logistically linked system integrating different transportation modes (rail, ocean vessel, truck etc.) to move freight or people from one place to another in a timely manner (Boschian et al. 2011; Crainic and Kim 2007; Macharis and Bontekoning 2004). There is a great availability of various alternative means and routes and so the complexity of ITN is continuously increased. On the other hand the availability of the new Information and Communication Technologies (ICT) that could be used to support the Decision Makers (DM) and the huge amount of acquired information, require the development of new models and methods for decision support (Coronado et al. 2009, Giannopoulos 2004). It is widely accepted that ITN decision making is a very complex process, due to the dynamical and large-scale nature of the intermodal transportation chain, the hierarchical structure of decisions, as well as the randomness of various inputs and operations. Researchers have followed similar approaches from other application areas (e.g., production processes), and they have identified different hierarchical/functional levels for transportation systems (Dotoli et al. 2009, Caris, Macharis, and Janssens 2008). More precisely the levels are the following: a) the strategic level, related to the long-term definition of the transportation network, to the selection of the different transportation modes and to the evaluation of the feasible flows (capacities of the nodes and of the arcs); b) the tactical level, related (on a middle-short term) to the management of logistic flows connected to the information flow and to the transportation network that is topologically and dimensionally defined at the higher (strategic) level; c) the operational level, including real-time decisional processes that concern the resource assignment, the vehicle routing definition, and so on. In particular, assuming a real time availability of the information regarding the conditions of the network (like unexpected requests of transportation, variations in the availability of the transportation system, road conditions and traffic flows), operational decisions should be taken in a dynamic context.

Since intermodal transportation is more data-intensive than conventional transportation means, the modern ICT tools help to produce, manipulate, store, communicate, and/or disseminate information and provide useful information about the state of the system in real time and therefore manage and change on-line paths, vehicle flows, orders and deliveries. Dotoli et al. (2009) proposed an integrated system that is based on a reference model and a simulation module to support the decision making process. The integrated system tracks the state changes from the real ITN and evaluates performance indices typical of the tactical and real time management, such as utilization, traffic indices and delivery delays (Viswanadham, 1999).

This paper takes into account the integrated system proposed in (Dotoli et al. 2009), and it specifies a Decision Support System (DSS) devoted to manage ITNs for taking tactical decisions, i.e., in an off-line mode, and operational decisions, i.e., in real time. Indeed, due to the complexity of the system, the DM needs support during the decision making process. There are proposed typical methodologies for decision making, which depend on the type of ITN problems that are addressed. Moreover, suitable optimization algorithms can be used for the definition of new long term policies, and employ the treatment of massive volumes of data to address multicriteria problems. In any case, the utilization of computer-based approaches to support the decision making procedures is a necessity. Different type of computer-based systems have been developed that are used widely according to the activity and the type of problem that they support.

This paper proposes a generic DSS that is devoted to manage complex ITN systems, based on the Unified Modeling Language (UML) (Miles and Hamilton 2006), a visual modeling language that is suitable to describe and specify software engineering. Moreover, we specify the components and the architecture of the DSS and the corresponding software tools. In addition, all solutions included in the proposed DSS are based on a Service Oriented Architecture (SOA) approach that guarantees the interoperability of the simulation and optimization drivers.

2. THE DSS STRUCTURE

In this section, we describe the main components of the proposed DSS.

Usually DSSs are categorized on the basis of different characteristics of the systems, e.g. whether they are for personal or group oriented decision making (Gorry and Scott Morton 1971; Keen and Scott Morton 1978; Delen et al. 2010; Power 2002). Based on the type of the application, DSSs are divided in desktop and web based applications. Despite the different categories of DSS, all of them share common characteristics; i.e., a typical DSS should include four main components: the data component, the model component, the decision component, and the interface component. Here, a UML class diagram is used to describe the main structure of the DSS. In particular, UML has standard notation and syntax and is composed of thirteen main types of diagrams, each one serving a different purpose and describing the system from different points of view. We consider two views of the DSS: *the structural view* is described by the package and class diagrams that illustrate the different types of objects which the system consists of and their relationships; *the behavioral view* is described by activities diagrams that describe the rules that the system follows to operate in a complete and correct manner, to avoid misunderstanding on both the user side and the developer side.

2.1. The DSS Components

Figure 1 shows the UML class diagram of the proposed DSS architecture. More precisely, each class is represented by a rectangular box divided into compartments. The first compartment holds the class name, the second holds attributes and the last holds operations. More precisely, attributes are qualities and named property values that describe the characteristics of a class. In addition, operations are features that specify the class behavior. Moreover, classes can exhibit relationships that are represented by different graphic connections: association (solid line), aggregation (solid line with a clear diamond at one end), composition (solid line with a filled diamond at one end), inheritance or generalization (solid line with a clear triangle at one end), realization (dashed line with a clear triangle at one end) and dependency (dashed line with an arrow at one end). By class diagrams each component can be modeled as a different class illustrating the different types of objects that the system can have and their relationships.

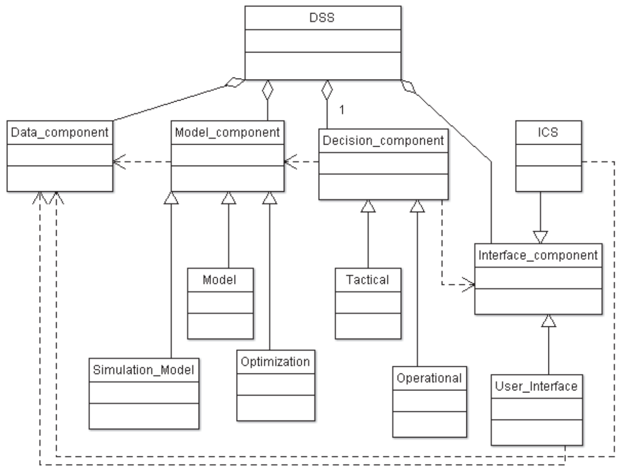

Figure 1: The DSS Structure

In the following we briefly specify the DSS components shown in Fig. 1.

Data component. This module is a database that can be denoted as internal and external data base. The data are internal if they come from organization's internal procedures and sources such as products and services prices, recourse and budget allocation data, payroll cost, cost-per-product etc. Moreover, external data are related to the competition market share, government regulations etc. and may come from various resources such as market research firm, government agencies, etc. In some cases the DSS can have its own database or it may use other organizational databases that can be connected directly with them.

Model component. This component mainly includes a simulation model, a mathematical model, and a set of optimization algorithms suitable to analyze effects of choices on the system performances. The models describe the operations at different management levels and the type of functions varies with the operation that they support.

Interface component. This module is the part of the DSS that is responsible of the communication and interaction of the system with the DM. Such a component is very important because regardless of the quality and quantity of the available data; the accuracy of the model is based on this interface. Indeed, this component includes an Information Communication System (ICS) that is able to interact with the real system and maintains the consistency between the stored data and the real system.

Decision component. This component consists of two second level classes: the operational decision class and the tactical decision class. Moreover, such classes include the performance indices that have to be considered in order to take the decisions. In addition, in relation with the performance indices and the object of the decisions, the DSS has to collect the decision rules and the optimization procedures that are used by the model and the simulation component.

3. THE DSS ARCHITECTURE

This section describes in detail the DSS components presented in Section 2. Moreover, Fig. 2 shows the architecture of the realized system by enlightening the connections among the modules.

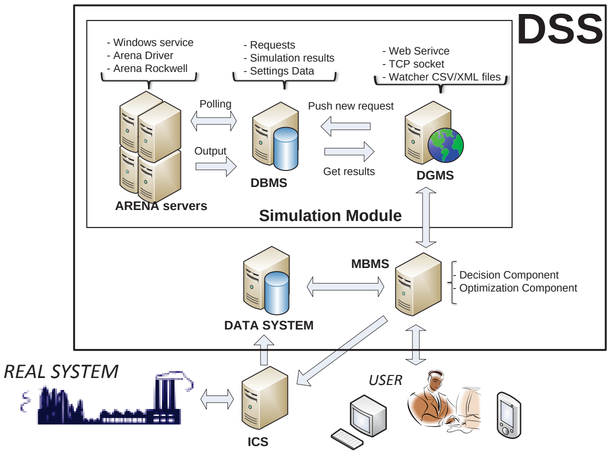

Figure 2: The DSS architecture.

3.1. The Data Component

We distinguish three different kinds of data. The first data are managed by the Data Base Management System (DBMS) that stores the internal data used by the decision and the simulation components. More precisely, the DBMS stores the requests of a new simulation with its input data, the data related to internal variables of the simulation model, and the outputs produced by simulations. Furthermore, the DBMS contains the queue of the requests to be sent to the simulation server, the state of each request and the results of the simulation. Moreover, it stores the description of all the simulation models that are available for the simulation runs.

The second and third kind of data are stored in the Data System: the internal data and the external data. The internal data represent all data necessary to describe the internal procedures, e.g. the time required for each activity, the number of available resources, the capacity of parking areas and safety levels, etc. On the other hand, the external data are information coming in real time from the system: the current number of vehicles, the information about the conditions of the roads, the accidents, the road maintenance works, the weather conditions, etc.

3.2. The Model Component

The model component is the core of the DSS: it consists of the simulation model and the optimization component.

3.2.1. The simulation model

The simulation model mimes the system, applies the optimization strategies proposed by the optimization module and provides the performance measures.

In the proposed solution, the simulation model is implemented by the Arena servers. In particular, each Arena server consists of the following main elements:

- **The ARENA software.** In such a server the simulation models are implemented and executed in the Arena Rockwell environment (Kelton 2009).
- **The Windows service.** The service is automatically started, it monitors the DBMS, as soon as a new simulation request is loaded, the service executes the simulation by the arena driver and loads results on the DB.
- **ARENA driver.** This component is the driver that connects the system to the software ARENA: it is in charge of starting and stopping simulations.

By the proposed architectural solution, we can provide a set of Arena servers and each server is independent from the others. Then, the number of operative Arena servers can dynamically change on the basis of the computational effort that is required in real time. Moreover, such an approach allows a parallel execution of the simulation requests by reducing the total execution time.

3.2.2. The Optimization Component

The second basic module of the model component is the optimization module (see Fig. 1) that combines a variant of Particle Swarm Optimization (PSO) (Kennedy 1995) with an Optimal Computing Budget Allocation (OCBA) scheme (Chen 2000). In particular, the original PSO algorithm was introduced in 1995 by Eberhart and Kennedy (Kennedy 1995). Its main concept includes a population, called a swarm, of potential solutions of the problem at hand, called the particles, probing the search space. The particles iteratively move in the search space with an adaptable velocity, retaining in a memory the best positions they have ever visited, i.e., the positions with the lowest function values (considering only minimization problems). The exploration capability of PSO is promoted by information exchange among particles. More specifically, each particle is assigned to a neighborhood. In the global PSO variant, also known as *gbest*, the neighborhood of each particle is the whole swarm and the overall best position is the main information provider for all particles. On the other hand, in the local PSO variant, also known as *lbest*, the neighborhoods are strictly smaller, usually consisting of a few particles. In such cases, each particle may have its own leader that influences its velocity update. In real life problems we usually accept good enough solutions instead of the globally optimal solution. Therefore, in case of "noisy" functions, "following" the "best" particle becomes a bit involved since the actual value of a particle is obscured by noise and repeated function evaluations are required in order to more accurate estimate the true value. Especially in situations where a function evaluation is a costly process, a compromise should be reached between the need for an accurate estimate of the true value and the need to have as few as possible unnecessary function evaluations.

The OCBA was proposed by Chen (2000) as a procedure to optimal allocate a predefined number of trials/replications in order to maximize the probability of selecting the best system/design: allocate replications not only based on the variance of the different designs but also taking into account the respective means. More specifically, the noisier the simulation output (larger variance), the more replications are allocated while more replications are also given to the design that its mean is closer to that of the best design.

The global version of the PSO uses the mean values for each performance measure that are evaluated by a low number of replications. Then using the OCBA procedure more replications are allocated in order to increase the probability of correctly identify the best particle, whose value is used to guide the search of the swarm.

3.3. Interface Component

In the presented architecture we implement two interfaces: the first interface connects the simulation module with the Model Base Management Server (MBMS) by means of the Dialog Generation/Management Server (DGMS) module; the second one connects the DSS and the real system through the MBMS.

In particular, the ICS module (see figure 2) represents the information system of the whole infrastructure and is the interface between the real system and the information system. It updates the system status stored in the Data System in real time.

The DGMS allows the communication between the decision component and the simulation component. It receives requests from the decision component such as running a new simulation or retrieving the results of a finished simulation. All requests coming from the MBMS are loaded into the DBMS by the DGMS. In order to guarantee the modularity of the architecture and the possibility to execute the DSS components under different platforms, we implement the following three different ways to communicate with the DGMS:

- **Web Service (WS).** The WS can trigger a new simulation, provides information about the state of the request and gives the simulation results. A web service is a process running on a server and allowing a client to execute a process on the server. This kind of application communicates by the Simple Object Access Protocol (SOAP).
- **TCP Socket.** This application is a process running on the server that creates a listener on a preconfigured communication gate. The MBMS sends messages to DGMS by the TCP channel. These messages are strings codified by a prefixed communication protocol. By this application the MBMS can request a new simulation or access to stored data.
- **File Watcher**. This process monitors the files on a shared folder. As soon as a new file is loaded by an FTP client, it analyses the contents of the file and executes the corresponding operations. When a simulation goes to an end, the file watcher creates a report file in the same folder so that all data are available to the client.

Furthermore, the MBMS shows data to the DM: if the current performance of the system is not satisfactory, then the DM can decide to evaluate the impact of some decision using a "what if" approach. The MBMS allows the DM to run a simulation directly without the decision module control. This proposed solution is very useful for the DM and contributes to generalize all the features offered by the DSS.

3.4. The Decision Component

The MBMS server implements the decision component. In particular, it monitors the system state stored in the Data System component and decides if it is necessary to run a new simulation or not. Indeed, if the performances of the system decrease, then the decision component starts the optimization and the decision procedure. Figure 3 shows the used decision approach that is based on the optimization module and the validation by the simulation. More precisely, the optimization algorithms propose some solutions that are sent to the simulation module. The ICS provides the current state of the system by returning the values of the variables that cannot be determined by the DSS. Then the DSS invokes the simulation module: the simulation starts and applies the proposed management strategies. The obtained performance indices allow evaluating the impact of the proposed solution on the system. Then a new set of candidate variables are passed to the simulation model and the process continues till the algorithm leads to a satisfactory decision described by a set of candidate variables. In this model, the candidate variables are the controllable inputs that may change in the tactical or operational decision making process.

In the proposed DSS, the hybrid optimization module (PSO+OCBA) is connected by a web service to the simulation module. The optimization module sends inputs and the number of replications for each candidate solution/design and it records the outputs, again through the web service.

The average values of the outputs are sent to the OCBA procedure that decides if a number of extra replications are required. The average outputs are used to guide the PSO procedure: the process is repeated till a good number of available replications is obtained.

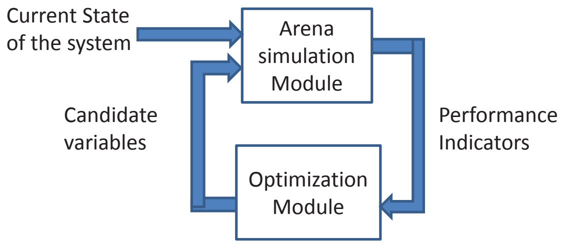

Figure 3: The Decision approach

On the basis of the current state of the system and the design variables, the DSS can be used to take decision both at the tactical and operational levels. In particular, at the tactical level the input variables are generated stochastically, while at the operational level the inputs come from the ICT tools of the real systems.

4. THE PILOT CASE

The proposed DSS has been realized to manage at tactical level an ITN composed of the port of Trieste (Italy), the dry port of Fernetti, and the railway station of the Roll on-Roll off freight trains. Mainly, the DSS deals with the decisions about the import and export flow of goods and the containers management and transportation. The DSS is designed in the framework of the SAIL Project, sponsored by the European Commission under the 7th Framework Program, Specific Program PEOPLE - Marie Curie Actions.

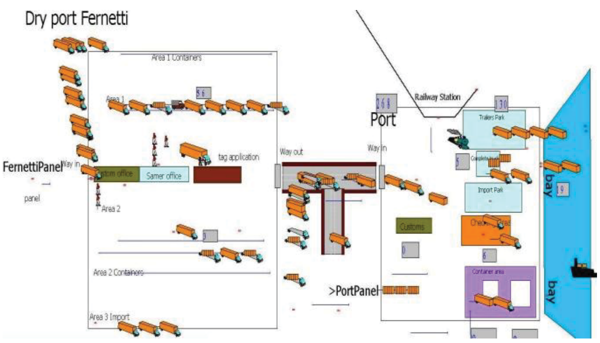

Figure 4: ARENA MODEL

In order to specify the simulation model in Boschian et al. (2011), the authors identified the main events occurring in a transportation system and described the main classes of the system: the port area, the inland port area, the road connection system and the rail network system. Furthermore, some interviews are previously performed to the actors involved in the process and the necessary data and the relations are collected. The involved resources and their availability are identified and the following performance indices are considered:

- throughput, i.e., the number of units boarded in a time units;
- average lead time, i.e., the time elapsed from the arrival of a new units till its system output;
- the average resources utilization;
- the average queue lengths of the users that have to acquire resources.

Figure 4 shows a sketch of the simulation model realized in the ARENA environment. The DSS allows us to identify all the weakness and the bottlenecks of the system. Moreover, some optimization strategies are proposed and analyzed at tactical level.

5. CONCLUSIONS

This paper specifies a Decision Support System devoted to manage Intermodal Transportation Networks and to take tactical and operational decisions. We describe the DSS architecture and, in particular, the decision process that is based on two main modules: the optimization and the simulation module. Moreover, we show how the DSS is designed by applying SOA and web server approaches. Hence, the obtained DSS can be realized in a distributed framework in order to face the system complexity.

Future work will describe in more detail the decision procedures of the DSS.

ACKNOWLEDGMENT

This work is supported by the E.U FP7-PEOPLE-IAPP-2009, Grant Agreement No 251589, Acronym: SAIL.

REFERENCES

Boschian, V., Dotoli, M., Fanti, M.P., Iacobellis, G. and Ukovich, W., 2011. A metamodeling approach to the management of intermodal transportation networks. *IEEE Transactions on Automation Science and Engineering*, vol. 8: pp. 457-469.

Caris, A., Macharis, C., and Janssens, G.K., 2008. Planning Problems in Intermodal Freight Transport: Accomplishments and Prospects. *Transportation and Planning Technology*, vol. 31: pp. 277-302.

Chen, C.H., Lin, J., Yücesan, E. and Chick, S.E., 2000. Simulation Budget Allocation for Further Enhancing the Efficiency of Ordinal Optimization. *Journal of Discrete Event Dynamic Systems: Theory and Applications* , Vol. 10: pp. 251-270.

Coronado, A.E., Coronado, E., and Lalwani, C.S., 2009. Wireless Vehicular Networks to Support Road Haulage and Port Operations in a Multimodal Logistics Environment. *Proceedings of IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics*, pp. 62-67, July 22-24, Chicago (IL, USA).

Crainic, T.G. and Kim, K.M., 2007. Intermodal transportation. In: C. Barnhart and G. Laporte, ed. *(Transportation) Handbooks in Operations Research and Management Science*, vol. 12. Amsterdam: North-Holland/Elsevier, 467–537.

Delen, D., Sharda, R., and Turban, E., 2010. *Decision Support and Business Intelligence Systems. 9th Edition*. New Jersey, USA: Prentice Hall.

Dotoli, M., Fanti, M.P., Iacobellis, G., and Mangini, A.M., 2009. A First Order Hybrid Petri Net Model

for Supply Chain Management. *IEEE Transactions on Automation Science and Engineering*, vol. 6: pp. 744-758.
Giannopoulos, G.A., 2004. The Application of Information and Communication Technologies in Transport. *European Journal of Operational Research*, vol. 152: pp. 302-320.
Gorry, A.G. and Scott Morton, M.S., 1971. A framework for management information systems. *Sloan Management Review*, vol. 13: pp. 55-70.
Keen, P.G.W. and Scott Morton, M.S., 1978. *Decision support systems: An organisational perspective*. Reading, MA: Addison-Wesley.
Kelton, W.D., Sadowski, R.P., and Swets, N.B., 2009. *Simulation with Arena. 5th Edition*. Boston (MA): McGraw-Hill.
Kennedy, J. and Eberhart, R.C, 1995. Particle swarm optimization. *IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth (Australia).
Macharis, C. and Bontekoning, Y.M., 2004. Opportunities for OR in Intermodal Freight Transport Research: a Review. *European Journal of Operational Research*, vol. 153: pp. 400-416.
Miles, R. and Hamilton, K., 2006. *Learning UML 2.0*. Sabastopol, CA (USA): O'Reilly Media.
Power, D., 2002. *Decision Support Systems: concepts and resources*. Westport, USA: Quorum Books.
SAIL project. Available from: http://www.teorema.net/it/UM/ReD/HarborLogistics/SAIL.aspx
Viswanadham, N., 1999. *Analysis of manufacturing enterprises: An approach to leveraging value delivery processes for competitive advantage*. Boston: Kluwer Academic.

AUTHORS BIOGRAPHY

Maria Pia Fanti received the Laurea degree in electronic engineering from the University of Pisa, Italy in 1983. Since 1983 she has been with the Department of Electrical and Electronic Engineering of the Polytechnic of Bari (Italy), where she was Assistant Professor from 1990 till 1998 and associate professor till April 2012. Now she is full professor of Control Systems in the same department. Her research interests include discrete event systems, Petri nets, modeling and control of automated manufacturing systems, supply chains, transportation systems, health care systems. She has published around 160+ papers and one textbook on these topics.

Giorgio Iacobellis received the Laurea degree in Electronic Engineering in 2004 from Polytechnic of Bari and the Ph.D. degree in Computer Science Engineering in 2009 from the same University. From February 2005 to February 2006, he was a research fellow and from December 2010 to November 2011 he was post-doc at Polytechnic of Bari. He currently works as an Experienced Researcher for the SAIL Project. His research interests include modeling, simulation, and control of discrete event systems, Petri nets, supply chains and urban traffic networks, distribution and internal logistics, management of hazardous materials, management of drug distribution systems and healthcare systems.

George Georgoulas received his diploma and PhD degrees from the Department of Electrical and Computer Engineering of the University of Patras in 1999 and 2006, respectively. He has worked as a postdoctorate fellow in the Intelligent Control Systems Laboratory at The Georgia Institute of Technology, USA, from 2006 till 2008. He currently works as an Experienced Researcher for the SAIL Marie Curie Project. He has published over 70 journal and conference papers, book chapter and technical reports. His main scientific interests include: Computational Intelligence Techniques, Evolutionary methods, Global optimization methods and Decision Support Systems.

Chrysostomos D. Stylios is an Assistant Professor at Dep. of Informatics and Telecommunications Technology, Technological Educational Institute of Epirus; he is a senior researcher at Telematics Center Department of Computer Technology Institute & Press. He was a visitor assistant professor at Computer Science Dep., of University of Ioannina. He received his Ph.D from the Department of Electrical & Computer Engineering University of Patras in 1999. He received the diploma in Electrical & Computer Engineering from the Aristotle University of Thessaloniki in 1992. He has published over 100 journal and conference papers, book chapter and technical reports. His main scientific interests include: Fuzzy Cognitive Maps, Soft Computing techniques, Computational Intelligence Techniques, Evolutionary methods, Neural Networks, Support Vector Machines, Knowledge Hierarchical Systems and intelligent data fusion and Decision Support Systems for Medical application.

Walter Ukovich was Assistant Professor of Electrical Engineering at the Engineering Faculty of the University of Trieste, Italy, from 1970 to 1985, and Associate Professor of Operations Research at the same University from 1985 to 1999. He is now full professor of Operations Research at the same University. Prof. Ukovich was from 2008 till 2011 director of the Department of Electrical Engineering, Electronics and Computer Science and was responsible for the curriculum in Management Engineering. He has been a member of the Evaluation Boards of the National Research Council, the Universities of Trieste, the Polytechnic of Turin, and the University Institute of Architecture of Venice. His main research interests are in Optimization, Logistics, Production Planning and Control and Health Management Systems and he is the author of over one hundred scientific papers.