Type of the Paper (Article, Review, Communication, etc.) ***ARTICOL IN FAZA DE EVALUARE***

Strategic decision system for Crisis Management using multi-agent algorithm.

Catalin Dumitrescu [1,*], Brindusa Radu [1], Man Mariana[2], Matei Ducu[1], Radu Gheorghe[1] and Liliana Manea [1]

1 Athenaeum University of Bucharest, Giuseppe Garibaldi 2 A, 20223, Bucharest, Romania; catalindumi@yahoo.com, bmradu@yahoo.com, ducumatei@yahoo.com, Gheorghe_radu@yahoo.com, lilyanamanea@yahoo.com
2 University of Petrosani; man_mariana2006@yahoo.com
* Correspondence: catalin.dumitrescu@upb.ro

Abstract: Humanity is exposed to crisis situations generated by panic situations of the population (public demonstrations, performances/competitions in stadiums), natural/industrial disasters and wars. In these situations, the decision-making and intervention factors must control the situation and control the crowds by taking appropriate decisions to reduce the potential victims and damages to a minimum. Usually, in crisis situations, the decision-makers do not have complete information, which is, most of the time, uncertain, ambiguous, or missing. That is why software applications such as panic simulators for training play an important role in managing crisis situations. In this article we will present the implementation of a panic simulator and prediction of evacuation and intervention routes using multi-agent Artificial Intelligence algorithms.

Keywords: strategic decision systems, panic situation, multi-agent algorithm

Citation: To be added by editorial staff during production.

Academic Editor: Firstname Lastname

Received: date
Revised: date
Accepted: date
Published: date

1. Introduction

In the study of the design process of strategic decision support systems, the notion of system is essential. The system is defined by modules that are in a mutual dependency, for the fulfillment of a common purpose using predetermined rules [1]. Depending on how detailed the analysis is, a system has nine characteristics [2], [3].

Each system accepts several inputs, processes the information provided by them and sends the results to the outputs and from here, in the environment in which the system evolves. The limits of the system separate it from the environment in which it manifests itself. Through the interfaces a system communicates with the environment. A system exists only in the environment that contains it and will interact with it through the data and information received.

A system is made up of components. These, also called subsystems between which various relationships are established, can consist of indivisible component elements or groups of aggregate elements. The design of the subsystems is very important because, if the system no longer functions properly, its "repair" should be possible by simply replacing the subsystem that caused the failure. System limitations refer to the constraints imposed by its parameters (eg storage capacity or speed) according to which the system must operate in order to achieve the purpose for which it was achieved.

There are some important notions regarding the study of the system. Decomposition is very useful when trying to understand how a complex system works. It will be broken down into simpler subsystems, the structure of which can be analyzed and understood in turn, without considering interference between subsystems [4]. Modularity derives directly from decomposition and facilitates the troubleshooting steps of systems. Coupling refers to the fact that the systems are interdependent although they should be as independent as possible. Thus, the failure of a subsystem would lead to the chain damage of all subsystems and, ultimately, the malfunction of the main system. If independence is ensured, only the defective component can be replaced, making the troubleshooting and repair process much shorter. Cohesion is the extent to which a subsystem can perform a single function [5].

The computer and informational system are based on data. These informational data include all internal and external information that is used within the organization. To this primary information will be added the data regarding the personnel involved in crisis management, dissemination of information, processing, storage and transmission of information to decision makers.

In the military field, the information system ensures the connection between the decisional and the operational system. An information system is shown in the figure 1.

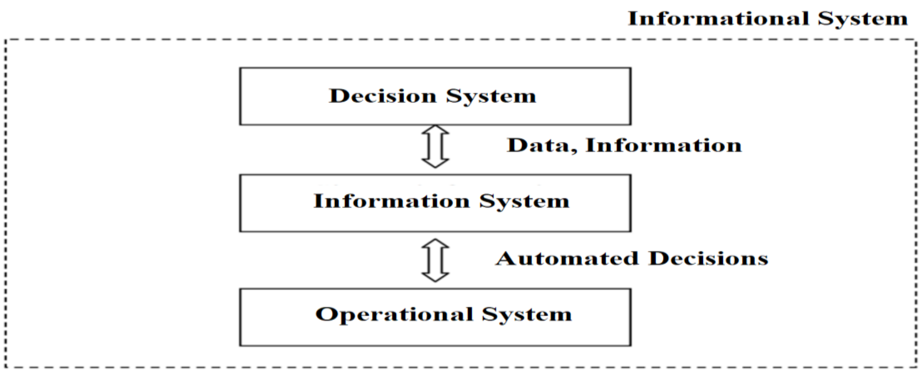

Figure 1. The relationship between computer system and information system.

According to figure 1, the functions of the system are:

- acquisition of information from the internal environment (operational and decision-making), but also information from the external environment after the occurrence of the panic event.
- the acquired information and the results obtained after processing will therefore be stored in the database.
- the system must ensure access to the database for their transmission to the intervention teams.
- the system will process data at the request of the decision-making and intervention factors.

The use of information systems (IS) within organizations leads to the implementation of the Automated Information System (AIS) based on Artificial Intelligence algorithms [6]. Thus, the SI - SIA relationship is represented in figure 2.

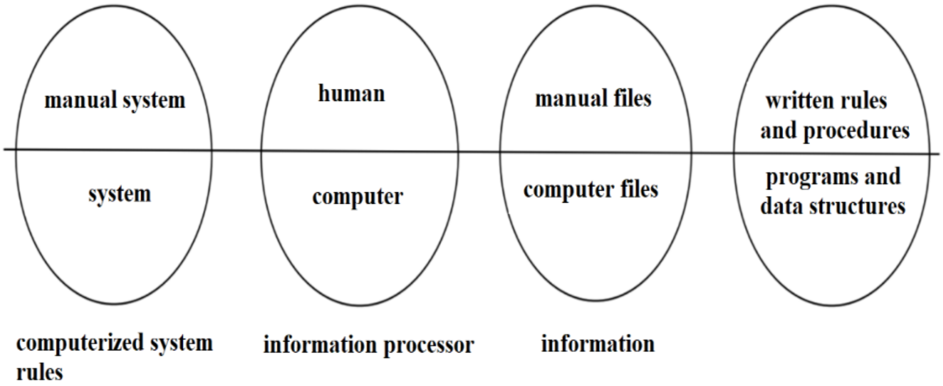

Figure 2. SI - SIA relationship.

Panic simulator computer system is a system that will provide predictive information for decision-makers, interventionists, and management activities in an organization, mathematical models for analysis and decision making (planning, control) [7]. The elaboration of the information systems imposes the modeling of the information system of the organization with the help of a formalism through which the reality within the information system can be represented as suggestively and faithfully as possible. For organizations of low complexity, computerization can mean the realization of a single computer application also referred to as a computer system [8]. Computer systems that use AI algorithms can be divided into application modules for certain users, which can be made in different programming languages. This structure is shown in Figure 3.

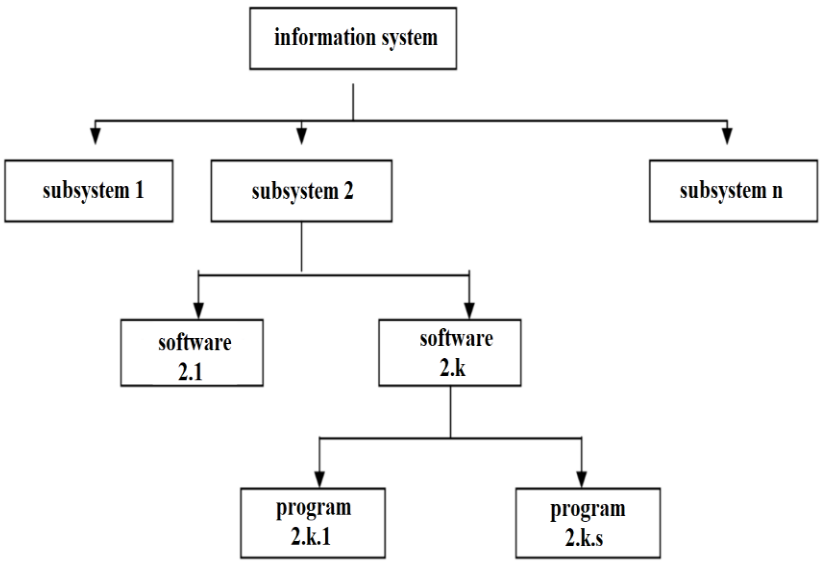

Figure 3. Computer system, subsystems, applications, programs.

A computer product consists of programs that access the database and the necessary documentation for the use and maintenance of the programs. These are based on methodologies and require the completion of stages starting with the specification of requirements and ending with their implementation, operation, and maintenance [9]. The military information system is a structured set of functionally intercorrelated elements to automate the process of obtaining information and to substantiate decisions [10]. The information system is included in the scope of the information system if within the information system there will be a series of activities that cannot be automated.

2. Materials and Methods

Crowds constitute some of the most complex and fascinating aspects of social reality, intriguing philosophers, historians, writers, psychologists, sociologists, or political scientists in equal measure. Here, numerous psychosocial phenomena are manifested that assume strong interferences between conscious and unconscious, between the most primitive drives and the noblest starts, under the incident of which the human individual discovers new facets of his soul [11].

Crowds and crowd phenomena represent one of the essential dimensions of reality and social life, without which the understanding and interpretation of some of their most important and interesting aspects would not be possible. Just as it would not be possible to develop a coherent theory on social action, or on the transformations of social systems [12], [13]. In this article we present the software development of a multi-agent-based algorithm for characterizing crowd behavior in panic situations, predicting collective human behavior when they are in a dynamic action. The results obtained from the simulations have a practical applicability and can help minimize the number of victims, collateral damage and reduce the time of evacuation and intervention in panic situations. By using the multi-agent type algorithm [14-20], a model was implemented as close as possible to the dynamic model that characterizes the crowds.

In the case of panic simulators using multi-agents, the environments are complex and require different variants of search algorithms than in closed environments. In these open environments, actions may be nondeterministic, the effects of actions are not guaranteed, and/or the agent cannot fully perceive the state of the environment. The resulting plan, through the sequence of actions, can be used, for example, in conjunction with the incremental Delaunay triangulation algorithm.

Usually in non-deterministic environments, similar actions can lead to different results. By simplifying the reasoning presented in figure 4, we have a person (agent) and an environment with two cells, A and B, in which the actions are movement (Left, Right), push (Push), and the state of the cells can be clean (Clean) or dirty (Dirty).

In this example, the plan can be represented as an AND-OR search tree, where we distinguish between cell-nodes where the agent decides which action to perform, called OR nodes, and nodes that represent the possible effects of an action, that is, the states that can result from applying an action to a state, called AND nodes.

```
%% Agent
% place the agent
agentCoord = rand(nrPeople,2) .* repmat([xmax, ymax],nrPeople, 1);
prefDoor = ceil(rand(nrPeople,1) .* size(doorCoord,1));
rad = peopleRad * ones(nrPeople,1);
v = zeros(nrPeople, 2);

% test if the people have chosen a valid door
for I = 1:nrPeople
    while (doorW(prefDoor(i)) == 0)
        prefDoor(i) = ceil(rand(1) * size(doorCoord,1));
    end
end

% set value and direction of the initial velocities
% of the people
for I = 1:nrPeople
    dir = doorCoord(prefDoor(i),☺ - agentCoord(I,☺;
    v(I,☺ = (dir./norm([xmax,ymax])) * norm([15,15]);
end

end
```

Figure 4. Implementation of the algorithm for agents in the Panic Simulator.

Delaunay diagram is a partition of space based on a nearest neighbor. Given a collection of vertices, the Delaunay triangulation is a collection of convex polygons such that each point inside a region is closer to the vertex in that region than any other vertex.

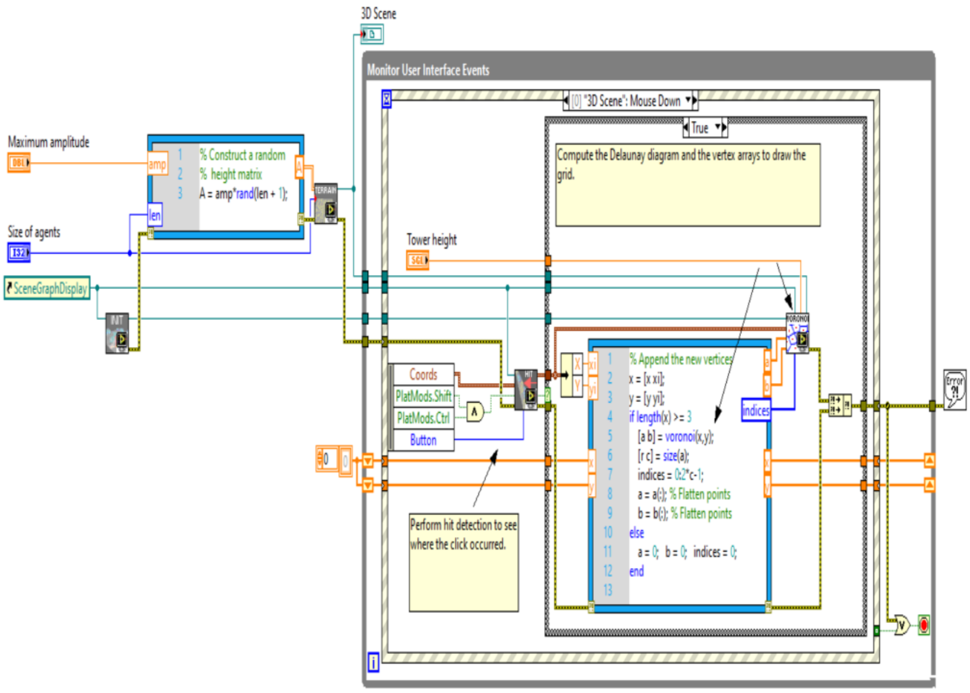

Figure 5. Implementation of the Delaunay triangulation algorithm.

The agent must find a plan for each possible state. The solution is represented by a sequence of actions and tests, for example [*Push*, if State = 5 then [*Right, Push*] else []], which ensures that the action state (goal) is reached in all possible ways had the agent in the tree, due to the non-deterministic environment. When nondeterminism can cause an action to sometimes have no effect, there are cyclic solutions, for example [while State = 5 do Right]. For the simulation presented together with the Delaunay algorithm, we also applied the A* search algorithm, because the environment is deterministic and completely observable, since diagrams for real situations can be used in the simulation. A* uses a heuristic estimation to estimate the best routes by classifying the nodes, starting from a given initial node to a given objective node.

3. Results

The solution search algorithm is:

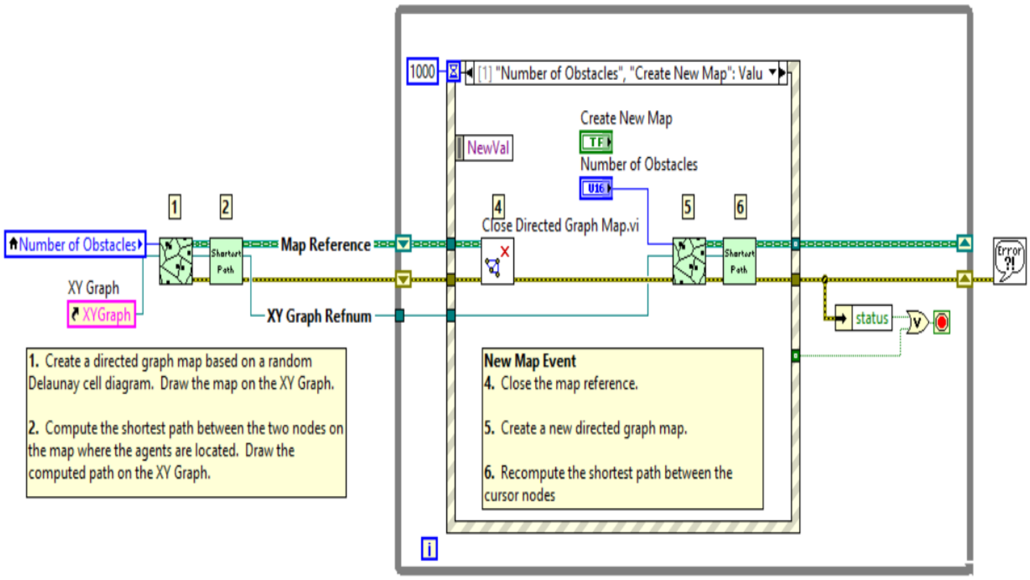

Figure 6. Implementation of the escape route search algorithm.

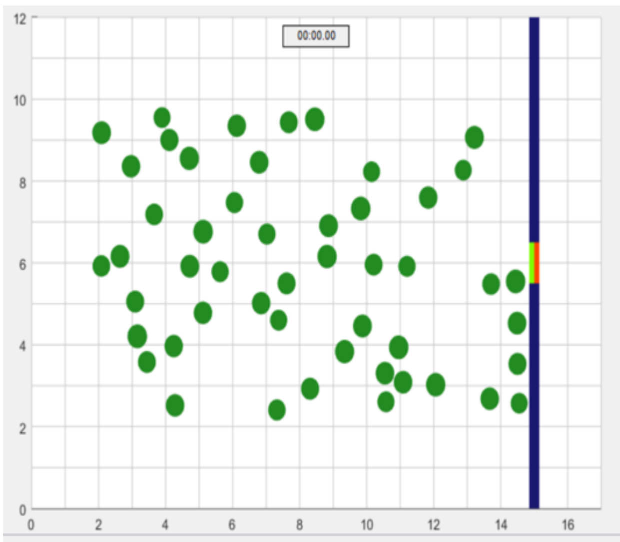

Figure 7. The results obtained by implementing the Panic Simulator.

To build a simulation of agent-based panic management systems (figure 7), three components must be developed. First, models of individual agents must be developed that are able to emulate the relevant behaviors within the system. Second, an environmental model must be developed that provides agent models with the information they need about the physical and procedural aspects of their context. Third, mechanisms must be provided for agents to act and interact, including mechanisms for synchronizing the simulation and transmitting data within it.

These developments require both conceptual models and their software instances. The conceptual models of the first two developments are closely related to the analyzed domain and capitalize on the structures of conservation of abstractions. The third development is crucial for the architecture of the simulation engine and for the fidelity of the simulation as it governs the dynamics of the full simulation. While software instances of the first two can be verified and validated for the concepts they model, they should also conform to the architecture of the third development.

The main graphical user interfaces of the development platform consist of a series of graphical tools that assist the developer in performing the necessary tasks of designing and creating a MAS [21]. The developer in generically determining the behavior of the template is assisted by the Behavior Design Tool. The agent's behaviors are modeled as they would exchange messages, and if necessary, they arrive at decisions using decision engines deduction. The data and control dependencies between these blocks are required by a MAS.

The system for extracting useful information, for logic agents, is based on the application of data mining techniques against the background of specific data requests. Agent-Oriented Programming (AOP) is a new methodological programming model regarding the construction of multi-agent systems. From an engineering point of view, agent-centered programming can be viewed as a specialization of the object-oriented programming model. Exposure in such a model consists of the actions of these agents by which they request or provide information, participate or compete with agents in the system. Communication between agents is different depending on the intended mode of communication, and its results are varied depending on individual intentions. Models for coordination arrange the knowledge, availability, and projects of intelligent agents in such a way that they can assemble their actions or solve a problem [22].

Coordination is essential to a multi-agent system, without coordination the benefits of interaction between agents cannot be discussed, which will quickly degenerate into a chaotic group of individual agents [23-24]. The reasons why the agencies need to be coordinated would be:

- To prevent disorder;
- They may encounter global constraints;
- In a multi-agent system, agents have different capabilities and expertise possibilities;
- Their actions are often interdependent, that is, an agent has to wait for the completion of the tasks of another agent before executing its own task.

An easy way to ensure orderly behavior and resolve conflicts is to give the group an agent with a broad view of the system through an organizational or hierarchical conformation. The simplest coordination technique is in the classical architecture, coordinator - executor or client - server for assignments and resource sharing for executor agents by a coordinator agent. To ensure global consistency [25], the coordinating agent can gather data from group agents and assign tasks to individual agents and make plans. In an agent network, agents are most often identified with its nodes, thus there is a hierarchy with two types: system management agents and safety agents. By this hierarchical relationship it is defined as a mode of server-client interaction. For execution activities, higher-level agents select lower-level agents in order to perform independent activities, there is no inter-agent cooperation for decision-making purposes, and each agent shows an option on system control.

The division of tasks and resources is one of the essential areas of MAS and one of their important contributions to computer science. The assignment of tasks is achieved through the definition of the organizational systems through which the agents operate, their competences aiming at the achievement of the common objective. In this context, we can talk about the presentation of the division of tasks, since the potential of an agent depends on its structural competences and the possibilities it possesses, external reserves and environmental conditions [26-27]. Tasks that require more means of either work or skill, which cannot be provided by one agent, must be divided into several subtasks and distributed to different agents.

Limitations of multi-agent systems

A static architecture is one in which all the components of the multi-agent system such as its inputs and outputs are determined in the design specification. In dynamic architecture, not all components are known, the specification, source of inputs and destination of outputs are mobile for each component. What differentiates the two architectures is the fact that in a static architecture the presence of the elements is necessary for the system to work, and in a dynamic architecture it is not necessary for the agents to participate for a certain period of time, and they can enter or leave participation in the system.

Coordination is essential to a multi-agent system, without coordination it is impossible discuss the benefits of interaction between agents, which will quickly degenerate into a chaotic group of individual agents.

The reasons why the agencies need to be coordinated would be:

- To prevent disorder;
- They may encounter global constraints;
- In a multi-agent system, agents have different capabilities and expertise possibilities;
- Their actions are often inter-conditioned, meaning an agent is forced to wait to terminate another agent's tasks before performing their own task.

An easy way to ensure orderly behavior and resolve conflicts is to give the group an agent who has a broad perspective on the system, through an organizational or hierarchical conformation.

The simplest coordination technique is found in classical architecture, coordinator - executor or client - server for assignments and resource sharing for executor agents to a coordinating agent.

To ensure global consistency, the coordinating agent may aggregate data from agencies group and can assign tasks to individual agents and make plans. The assignment of tasks is achieved by defining the organizational systems through which the agencies they work, their competences aiming at the achievement of the common objective. In this context it can talk about the presentation of how to divide the tasks because the potential of an agent depends on his structural competences and the possibilities he possesses, external reserves and environmental conditions.

Assignments that require more means of work or skill, which cannot be provided by one agent, must be divided into several subtasks and distributed to different agents.

4. Conclusions

Strategic decision support systems are integrated systems of policies, procedures, organizational structures, personnel, equipment, facility, and communications designed to support the exercise of the command-and-control process in all phases (states) of missions in the tactical field.

The strategic decision support system will ensure the transmission of information between the organizational structures (entities participating in the action) and the technical means of processing in the space related to the tactical field, beyond the visible and invisible horizon. The missions that will be carried out by the strategic decision support system are:

1. Supporting command and control processes across the spectrum of tactical actions (providing decision-makers with the means to exercise authority and command of subordinate forces and those received in support to carry out the mission).

2. Support for planning, decision-making and rapid response in dynamically evolving tactical environments (rapid transition from offensive to defensive actions requires commanders to make frequent changes in conceptions and action plans).

3. Support for increasing the mobility of intervention teams and adapting them to different environments for carrying out specific actions.

4. Ensuring the exchange of information within the joint tactical actions, with the support of other participating entities.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used “Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, X.X.; writing—review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration, X.X.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.” Please turn to the CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: This research received no external funding

Data Availability Statement: Not applicable.

Acknowledgments: This work is supported by The Intermediate Organization for the Promotion of the Information Society (OIPSI), by the National Plan of R&D, Project NEUROGUV, cod: POC/875/2/2, Contract No. 142429/01.06.2022.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Alalwan, J.A. (2013) 'A taxonomy for decision support capabilities of enterprise content management systems', The Journal of High Technology Management Research, Vol. 24 No 1, pp. 10-17.
2. Mentzas, G. (1997) 'Implementing an IS Strategy- A Team Approach', Long Range Planning, Vol. 30 No 1, pp. 84-95.
3. Cao, Q. and Schniederjans, M.J. (2004) 'Empirical study of the relationship between operations strategy and information systems strategic orientation in an e-commerce environment', International Journal of Production Research, Vol. 42 No 15, pp. 2915-2939.
4. Bourletidis, K. and Triantafyllopoulos, Y. (2014) 'SMEs Survival in Time of Crisis: Strategies, Tactics and Commercial Success Stories', Procedia-Social and Behavioral Sciences, Vol. 148, pp. 639-644.
5. Alyoubi, B.A. (2015) 'Decision Support System and Knowledge-based Strategic Management', Procedia Computer Science, Vol. 65, pp. 278-284.
6. Fairbank, J.F., Labianca, G.J., Steensma, H.K. and Metters, R. (2006) 'Information Processing Design Choices, Strategy, and Risk Management Performance', Journal of Management Information Systems, Vol. 23 No 1, pp. 293-319.
7. Andersen, T.J. (2001) 'Information Technology, strategic decision making approaches and organizational performance in different industrial settings', Journal of Strategic Information Systems, Vol. 10 No 2, pp. 101-119.
8. Arnott, D. and Pervan, G. (2008) 'Eight key issues for the decision support systems discipline', Decision Support Systems, Vol. 44 No 3, pp. 657-672.
9. Kamariotou, M. and Kitsios, F. (2016) 'Strategic Information Systems Planning: SMEs Performance outcomes' in: Proceedings of the 5th International Symposium and 27th National Conference on Operational Research, Athens, Greece, pp. 153-157.
10. Brown, I. (2010) 'Strategic Information Systems Planning: Comparing Espoused Beliefs with Practice' in ECIS 2010: Proceedings of 18th European Conference on Information Systems, Pretoria, South Africa, pp. 1-12.
11. Brown, I.T.J. (2004) 'Testing and Extending Theory in Strategic Information Systems Planning Through Literature Analysis', Information Resources Management Journal, Vol. 17 No 4, pp. 20-48.
12. Giannacourou, M., Kantaraki, M. and Christopoulou, V. (2015) 'The Perception of Crisis by Greek SMEs and its Impact on Managerial Practices', Procedia-Social and Be-havioral Sciences, Vol. 175, pp. 546-551.
13. Maharaj, S. and Brown, I. (2015) 'The impact of shared domain knowledge on strategic information systems planning and alignment: original research', South African Journal of Information Management, Vol. 17 No 1, pp. 1-12.
14. Ilona Madalina COSTEA, Florin Codrut Nemtanu, Catalin Dumitrescu, Claudiu Virgil Banu, Geanina Silviana Banu; Monitoring System with Application in Road Transport; 2014 IEEE 20th International Symposium for Design and Technology in Electronic Packaging (SIITME); BDI; ISBN:978-1-4799-6961-6; pag. 145-148 , DOI: 10.1109/SIITME.2014.6967013- WOS:000358258300025
15. S Raboaca, C Dumitrescu, I. Manta, Aircraft Trajectory Tracking using Radar Equipment with Fuzzy Logic Algorithm, published Mathematics 2020, Volume 8, Issue 2, 207, doi.org/10.3390/math8020207, ISSN 2227-7390, 2020.
16. F. Nemtanu, Ilona Madalina COSTEA, Catalin Dumitrescu, Spectral Analysis of Traffic Functions in Urban Areas, PROMET - Traffic&Transportation, ISSN: 1848-4069, Vol. 27, 2015, No. 6; pag. 477-484 - WOS:000368321400003
17. M Minea, C Dumitrescu, I Moise, Non-Intrusive Driver Condition Monitoring in Highly Automated Vehicles with Medical Information Support for Emergency Calling, 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), 1-3 July 2019, IEEE Xplore, IEEE, DOI 10.1109/TSP.2019.8769094, **ISBN:** 978-1-7281-1864-2.

18. M Minea, C Dumitrescu, Unconventional Public Transport Anonymous Data Collection employing Artificial Intelligence, 11 th International Conference on Electronics, Computer and Artificial Intelligence – ECAI 2019, IEEE, vol 11 – No 1/2019, ISSN 1843-2115, ISBN 978-1-7281-1624-2
19. M Minea, C Dumitrescu, Enhanced public transport management employing AI and anonymous BT data collection, 23rd International Conference on Circuits, Systems, Communications and Computers (CSCC 2019), Athens, Greece, July 14-17, 2019, "MATEC Web of Conferences", E-ISSN: 2261-236X, Volume 292, 2019, https://doi.org/10.1051/matecconf/201929203006
20. Ilona Madalina COSTEA, C. Dumitrescu, C Banu. B. Soare, Trajectory estimation for transportation systems: Aircraft tracking applications, Proceedings of 38th International Spring Seminar on Electronics Technology - ISSE 2015, Eger, Ungaria; pag. 329-332, DOI: 10.1109/ISSE.2015.7248016 - WOS:000374113000067
21. Mirchandani, D.A and Lederer, A.L. (2014) '"Less is More:" Information Systems Planning in an Uncertain Environment', Information Systems Management, Vol. 29, No 1, pp. 13–25.
22. Roesch, M.; Linder, C.; Zimmermann, R.; Rudolf, A.; Hohmann, A.; Reinhart, G. Smart Grid for Industry Using Multi-Agent Reinforcement Learning. *Appl. Sci.* **2020**, *10*, 6900.
23. Yang, H.; Liu, X.Y.; Zhong, S.; Walid, A. Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy. *SSNR* **2020**.
24. Abbeel, P.; Darrell, T.; Finn, C.; Levine, S. End-to-End Training of Deep Visuomotor Policies. *J. Mach. Learn. Res.* **2016**, *17*, 1334–1373.
25. Konar, A.; Chakraborty, I.G.; Singh, S.J.; Jain, L.C.; Nagar, A.K. A deterministic improved q-learning for path planning of a mobile robot. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2013**, *43*.
26. Lin, J.L.; Hwang, K.S.; Jiang, W.C.; Chen, Y.J. Gait Balance and Acceleration of a Biped Robot Based on Q-Learning. *IEEE Access* **2016**, *4*.
27. Panagiaris, N.; Hart, E.; Gkatzia, D. Generating unambiguous and diverse referring expressions. *Comput. Speech Lang.* **2021**, *68*.