

# iDecisionSupport – a Web-based Framework for Decision Support Systems

Ciprian Candea\*, Gabriela Candea\*\*,  
F.G. Filip\*\*\*

\* *Ropardo SRL, Sibiu, 550129 Romania (e-mail: ciprian.candea@ropardo.ro).*

\*\* *Ropardo SRL, Sibiu, 550129 Romania (e-mail: gabriela.candea@ropardo.ro)*

\*\*\* *the Romanian Academy - BAR and INCCE, and the Nat. ins.*

*Informatics-ICI, Bucharest(e-mail: ffilip@acad.ro)}*

---

**Abstract:** This paper describes iDecisionSupport, a collaborative decision-making support system that is characterized by and provides safety, usefulness, efficiency and usability. Its development is based on the principles of Decision Support Systems and is designed as a framework that can integrate third party applications as decision support tools. Routine procedures are facilitated by software agents and an internal workflow engine, while there is the advantage that the system can be accessed from anywhere at any time through a friendly web-interface.

**Keywords:** Decision Support Systems (DSS), Collaborative Decision Making, Workflows, Intelligent Agents, Shared Plans

---

## 1. INTRODUCTION

We well know that nowadays business processes change rapidly, especially when a company is involved in manufacturing highly customised products or it is simultaneously running different projects. (Câdea, Câdea, 2011).

This leads to changes in the manufacturing processes and hence to reengineering business processes. To be able to execute such changes in accordance with the customer's requirements there is a need of permanent communication and constant decision making.

Another aspect of contemporary manufacturing enterprises is their need for interdisciplinary teams to collaborate and team members are frequently distributed in different locations (i.e. plants, research departments); sometimes enterprises make use of external specialists whose responsibilities consist of decision based activities.

When a project team is located at more than one site at a time, it is very difficult to synchronise their tasks and optimise the usage of these distributed resources. Therefore, it is essential that all participants contribute to the creation of the process models during the early stages of the project. In addition, such geographically dispersed teams need a suitable interactive environment to model business processes concurrently. The objective of such collaborative model development is to enable teams to improve their business processes and shorten product lead-times.

A digital factory depicts a hybrid persistent community where a rich virtualized environment, representing a various activities and processes of the factory, will facilitate the sharing of factory resources, manufacturing information and knowledge and it will help with the simulation of

collaborative design, planning, production and management among different participants and departments.

As we previously said, this paper presents the iDecisionSupport (iDS) system that was developed by Ropardo S.R.L. It is a collaborative decision-making support system designed to avoid the need for lots of training and especially to prevent user rejection. Basically it provides a collaborative working environment where team members attend different type of meetings, or work sessions. Also, it has an internal workflow engine which allows the creation of meeting chains in order to automate different routine procedures.

Following the above aspects we consider a factory hub with corresponding functionalities that can support project oriented activities on digital factory. Because users are dispersed geographically we consider a web based solution and, since we are willing to equally support SME, we also consider an open source solution as technology base.

Another aspect to mention here is that, from a technical point of view, iDecisionSupport is developed as a "framework for decision support tools that provides a collaborative environment where different software tools for decision making can be easily integrated while the users can access them remotely and asynchronously" (Georgescu, Candea, Zamfirescu, 2007). Also it provides a default set of tools such as the discussion list (a forum-like tool for discussions), vote (a tool that permits grading or expressing the agreements over a set of issues), brainstorming (electronic brainstorming based on the IBIS approach) and mind map. It is worth knowing that the decisional model that we implemented is based on the Shared Plans theory (Grosz, Kraus, 1996) and it was tested for the first time in a software prototype by our group in 2001, as described in Zamfirescu, Candea, Luca, 2001.

Besides its internal use at Ropardo S.R.L. the system also has other important implementations, some of them described in Chapter 6.

## 2. OVERVIEW

Basically iDecisionSupport creates a collaborative working environment where team members attend different type of meetings, or work sessions. The meeting type is set by the decision support tool that is used to support the meeting. These can be brainstorming, voting, discussion tools/meetings or any other type that results from any third party application that is integrated. It is meant to be a highly anthropocentric solution (Filip, Donciulescu, Filip, 2002).

### 2.1 Using the system

iDecisionSupport provides access to the meetings (decision making sessions) via web-browsers avoiding the need to install any desktop components, process that is usually cumbersome. The main idea of the design is to eliminate any constraint of time and space for the users. Basically what they can do is to access the system from any location at any time during the meeting time span.

The standard way to access iDecisionSupport is by using a web-browser, logging in, selecting the meeting and starting the work. However, procedures that require many steps usually lead to frustration and ultimately to user rejection especially when they only need to write down an idea or comment some issues. It cannot be expected that twenty people that simply have to say “yes” or “no” to some issues will be familiar with the whole system in order to be able to login, select the meeting from maybe another ten of the same kind, open it and vote. To avoid this, all the active participants are notified with e-mails that contain access links to the meetings. Following these links the users will gain access directly to the meeting without any knowledge about the whole system. Simple actions like grading an issue or expressing the agreement of one can also be done directly from a form that they receive in the e-mail along with a briefing of the meeting. Overall, the system was designed to deal with group decision making as well as with individual decisions and support synchronous and asynchronous collaboration.

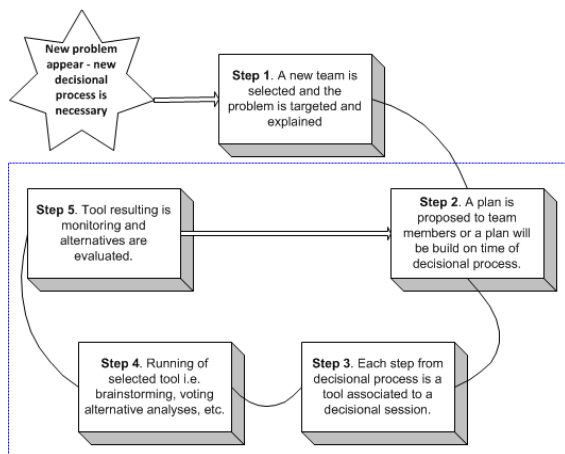


Figure 1- Decision Life Cycle

The main application flow as showed in Figure 1 is designed to be intuitive and easy to be exploited by users that know to work with web applications at normal level.

In our case the facilitator is the initiator when he/she considers that a new problem appeared and needs to consult a team, and then he/she becomes the facilitator of that session. We find it important for the facilitator to be a person that needs a response, and know what team to assemble in order to address it. You should know that most of our practical cases using this approach were successful. In this case meeting the initiator is the facilitator and using shared plans selected group is going to find the right path to solving the problem using tools provided by the system at that precise moment.

Of course iDS system also allows a more complex usage, when a dedicated facilitator is available and takes an active role in the decision making.

### 2.2 Decisional process as a Shared Plan

“Shared Plans theory states that the participants need to have mutual beliefs about their goals and actions to be performed and the capabilities, intentions, and commitments of the participants. Originally inspired from a social type of human collaboration, the model, which formalizes these mental states of collaborative actions, was applied afterwards to model teamwork of agents” (Zamfirescu, Candea, Luca, 2001). Each meeting is configured according to the characteristics of the decision problem and the group (e.g. duration of the meeting, time interval, anonymity settings, number of interventions, voting type, etc.). This configuration can be constantly updated during the meeting and, at the same time, it is possible to be imported from previous meetings or from containing plans.

The system provides a commitment phase for each meeting where all the participants can agree or disagree upon the meeting configuration. (e.g. propose a different time interval, voting time, number of interventions, etc). In this phase all these issues can be discussed until a mutual agreement from the group is reached.

### 2.3 Roles and Security

The system provides four basic roles for the participants: system facilitator, meeting initiator, participant and observer. These are designed to facilitate the functionalities described in the previous subsection.

The system facilitator is responsible for creating resource profiles, classifying available resource, granting user access and other administration issues.

The meeting initiator is the one that has the right to create and configure new meetings and invite the necessary participants. It can be looked at as the meeting leader in real life, the meeting initiator or the chairperson. The user that has this role must have thorough knowledge about the system in order to manage the meetings. “Group facilitators may provide an interface between the group and the technology to assist the group in using computer-based models.” (DeSanctis, Gallupe, 1987).

The participant is the user that can access those meeting plans that directly concerns him/her and perform the functions permitted by the initiator and the facilitator (e.g. grade items, come up with ideas, comments etc). This type of participant can be seen as the usual one that does not need thorough knowledge about the system because he/she is notified by e-mail about all the meetings and attends them only by following the received links.

The observer role is attributed to participants that are only invited to “observe” the meeting and they are not allowed to perform any actions.

Access in the system is controlled by using usernames and passwords. Each user can access a number of projects based on project-membership assigned by the project responsible. If the user is not a member in a project then he cannot see any information from that project. Additionally, the access to each decisional session is controlled through membership. A user can have one of the 4 roles in a session and can see and change information in that session based on the rights defined by its role. If he has no membership to a session then he cannot perform any action in it (not even opening for viewing). Based on the system organization a user can gain access only to specific projects and sessions, as setup by the project responsible, without having access to other parts of the system.

All the operations done in a session (tool usage) can be tracked during its lifetime if the meeting initiator decides this is needed. In other situations the session can be set as anonymous; in this case the operations can be done by the users having the right access (based on roles) but the tracking is not done and there is no way to tell that any given operation was done by a specific user. Votes are counted; opinions are entered and appear in discussion lists or in decisional tool but without author or timestamp. In this way the end-user can be assured there will be no harm upon him due to his opinions. Access to a session and its data is dictated by the user roles and session status (information can be changed only when the session is open, in other cases it is read-only for everybody).

### 3. System Architecture

The most important aspect of iDecisionSupport is its decentralized architecture which allows integrating different third party applications as support tools. These can reside anywhere on the Internet and are seamlessly accessed by the users inside iDecisionSupport without being affected by their location. This distributed environment (illustrated in Figure 2) is created by the main components of the system:

- the iDS Server which is the central component that handles the decisional meetings, registers tools, provides access rights for users
- the iDS Web Client which provides web access/interface to the whole system
- the decision support tools

Each of these components (including each individual tool) can reside anywhere on the internet and they communicate with each other through web service.

After an application is integrated as a support tool (detailed in subsection 4.4), it runs inside a meeting and user context that is provided by the web client. Basically, the client issues a HTTP request on the application’s web entry point and opens the tool inside a browser window/frame. The request contains a meeting token that the tool uses to load meeting related information from the server, information that actually represents the meeting context.

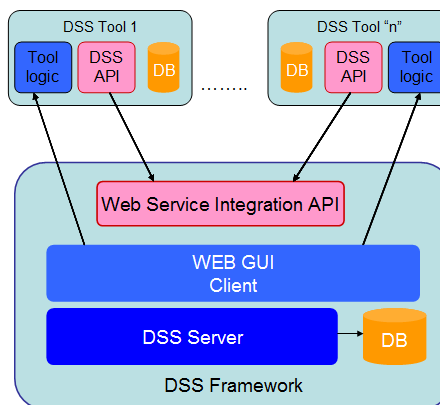


Figure 2 - DSS Framework

This consists of: meeting time span or duration, list of participants along with their assigned rights, meeting configuration and current work. First of all the tool must know what the time period (time span) of each meeting so it can display the correct available actions. If it is passed its end date then all the information must be “read-only” (no more actions can be performed by the participants) or if a meeting has not started yet then no actions can be performed but it is still possible to configure it. Then, the tool must know what users participate at the meeting and what rights has each of them so it will display the correct available actions. The user context represents the actual user that is logged in the system to participate at the current meeting and it is also loaded from the server based on the meeting token.

During the meeting, the tool saves the actual work into the server on each user action or periodically depending on network traffic. This procedure is part of the work synchronization process which ensures that all work that is done during a meeting is also available on the server for further processing. The process starts when the tool configuration is used to create the “blank” work data (e.g. in case of a voting meeting it consists of the items, and data that represents the voting type, anonymity settings, etc). This initial data is updated during the meeting by recording all the actions performed by the participants (e.g. grading an item or adding a reply or idea, etc). Depending on the tool implementation this data is persisted into the tool as well, or just maintained as long as there is at least one user that accesses the meeting. Work results are stored into the server in XML format. There is a common result format that the server can interpret and it is used in the transfer process of the results. This process allows the server to create meeting chains where the output of one meeting is the input of the next. (e.g. the main topics that result out of a brainstorming are automatically inserted as the topics that have to be processed in a voting meeting). It is not necessary for this result format to be at the same time the internal data format of

each tool. If the tool has a different format then it must provide XSL transformation files (XSLTs) from and to the common results format so it can be included in a meeting chain.

An example of an integrated third party application would be FreeMind (<http://freemind.sourceforge.net/>) which is a mind mapping software described in subsection 4.4. FreeMind has its own XML representations of the mind map modelling and is also a desktop application. First, a web access point was developed for it which uses the meeting token to load the meeting and user context from the server and then guides the user to access it through Web Start. Being a desktop application it must be downloaded. The application package is the one that actually handles the work and uploads the results on the server through the exposed web-service.

This example shows the flexibility of our framework as well as its decentralization, tools that can be

- web based hosted on a different server that DSS server runs,
- standard application that runs on local computer and process complex data but still can be connected with the framework

#### 4. COMPONENTS

The iDecisionSupport framework is made up of two main components: the DSS server and the DSS web client (the interface). In addition, there are of course the tools (described in Chapter 5) that can reside anywhere and integrate directly with the server using the server API and that can be accessed through the web-client.

##### 4.1 Server and Web Client

The server handles all the business logic by separating meetings in projects and plans (subplans) and provides a web-service API for the tools. It uses a relational database (PostgreSQL, MySql or other) to store the entire information of the meeting (metadata, duration, participants, etc.) as well as the conclusions of the meeting that are transferred from the actual tools. It also manages the data transfer from one tool to another.

The iDecisionSupport web-client is the access point to all the described features. It has a tree view of all the meetings (grouped by projects and plans) and manages the access to the meeting tools. Moreover it has different entry points and views for accessing the system from the notification links.

Both components are built using the Java language and Java related technologies (spring, struts, axis for web-services, JSP/JSTL, JADE (Java Agents Development Framework), jBPM for the workflow engine).

##### 4.2 Workflow engine

The iDecisionSupport system has its own internal workflow engine which allows the creation of meeting chains of different types where the output data of one is the input data for another. This allows having multiple meetings meant to resolve more complex issues. For example, if some ideas need

to be developed and the most convenient one has to be selected, a workflow is used that has a brainstorming meeting followed by a voting one. In this case the ideas that come up in the brainstorming meeting are considered separate issues and automatically imported in the vote meeting that follows, graded and the one that gathers more votes is then selected for further processing. Once the first meeting has ended the workflow system automatically starts the second one and imports the items and details into it. The workflows respect the WfMC/Wf-XML standard and can be developed with any compliant workflow editor.

##### 4.3 Agents

„Internet-based, distributed systems have become essential in modern organizations. When combined with artificial intelligence (AI) techniques such as intelligent agents, such systems can become powerful aids to decision makers”. (McDermott, Mikulak, Beauregard, 2008) The utilization of software agents in Decision Support Systems was proven to be a very efficient approach to this kind of systems as described in studies like Yen, Fan, Sun, Hanratty, Dumer (2007). Ropardo S.R.L. started researching software agents in 1999 and in 2001 when the first prototype of iDecisionSupport was developed and included several agent features. The current version of iDecisionSupport has a mature internal agent environment for software agents that perform very specific actions which cannot be generalized.

##### 4.4 Reports

iDecisionSupport allows generating reports for the work that was performed in a meeting. This can be done from the tool side (if it is implemented) or from the server's side. Each tool may generate reports in its own way and provide it to the user for download. However, regardless the tool reporting procedure which may or may not exist, the server has its own reporting system that can generate reports using the work results that are saved in the common results format. Reports are generated following defined templates.

#### 5. SUPPORTING TOOLS

iDecisionSupport provides a default set of tools that can be combined to efficiently process of a certain set of items. Other custom tools can be developed and integrated in the system if necessary. Most of them are web-based but desktop applications are/can be also integrated.

Next we show few of the tools that are available, each of them are detailed on the literature and but it is not our scope to explain each of them in detail.

##### 5.1 Discussion List

The Discussion List is the tool that provides forum like discussion threads for different aspects that need to be discussed. It inherits all the forum features for thread management and also adds new ones like maximum replies per user, maximum replies per topic or anonymous replying.

## 5.2 Vote

Vote is a decision support tool for voting (pooling) activities. It handles different items or issues (described in free text) that can be graded or voted by the users using several methods. These include: 10 point scale grading (Figure 3, where the items receive grades from 1 to 10), yes/no voting (where users can vote yes or no for each item), agree/disagree voting (where the users can express their agreement level over the items) or multiple choice (where users select a subset of the initial items).

## 5.3 Brainstorming

Brainstorming is another decision support tool that allows the development of electronic brainstorming meetings (sessions). The activity starts with an initial aspect (problem or question) that is the subject for debate and ideas coming from the users. Each of these ideas is commented with positive or negative arguments which are going to be the source of new questions that continue the cycle. Throughout this procedure lots of ideas are generated and their positive and negative aspects are emphasized.

## 5.4 MindMap

MindMap is a tool that serves to creating mind map diagrams that are meant to generate, classify and visualize ideas. This is actually the Free Mind (<http://freemind.sourceforge.net>) application integrated in the system and it is also an example of a desktop application that can be integrated. It communicates with the other components in the same way the web-based tools do, the only difference is that it has to be downloaded locally in order to be run.

## 5.5 Custom tools - integrating a new application

The custom tools represent third party applications that are integrated into the system or tools that are built from scratch to serve some custom purposes. Regardless the type, they all need to follow the same guidelines so they can be integrated in iDecisionSupport. The most important aspect is the need for a web access point so they can respond the HTTP requests that are issued by the web client, so they can be opened inside a browser window or frame. In case of a standalone application it still needs a web-access point that reads meeting and user context from the server (using the HTTP request token) and then lets the user download run it locally.

Upon registration, the tool (or tool developers) must provide the web access point URL and other information regarding the tools behaviour like: type of results (follows the common results format or has its own internal representations) and XSLT transformations in case it does not follow the common results format.

# 6. IMPLEMENTATIONS

## 6.1 Workflow for approving research project proposals

The system was implemented at "Lucian Blaga" University of Sibiu and it was designed to support the process of approving different projects proposals. When a new project is submitted for approval it is first reviewed by the approving committee

from different point of views and different amendments may come out. If there are any amendments then the project is sent back to the authors and resubmitted until there are no more amendments. The second step consists of the project proposal endorsement vote by all the approving committee members. So that it can be approved it needs a 50+1 % vote (figure 3) result, otherwise it must be revised by the authors.

This process was implemented in iDecisionSupport through a workflow which had a discussion meeting in the amendments generating step and a vote meeting for approving the project. In the discussion meeting all the amendments appeared as new topics of discussion (that were of course commented with suggestions) and in the vote meeting the project proposal was approved or rejected. All the meetings were created and configured automatically by the workflow system and the necessary participants were notified via e-mails to attend the meetings.

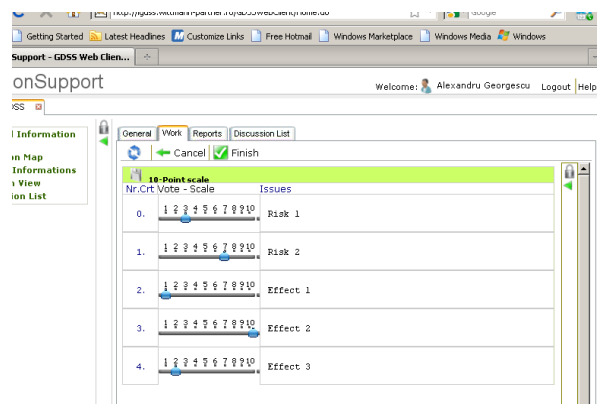


Figure 3 - Voting tool

Once one project is approved the iDecisionSupport system is invoked in the validation activities step through a group decision from the approving committee; here the validation/invalidation of the continuation of the project is decided by usage of voting tool to project completion.

At the end of the project the phase of analysis of the documents step is followed by a group decision regarding the originality of the work.

For the ULBS project, the implementation objectives of the iDecisionSupport system were: collaborative work regarding the technical documentation of the projects, notifications for all team members in order to be up-to-date with the status of all projects, all the collaboration work must be achieved even if the team are dispersed for a limited period of time, or unavailable from on-line point of view at one moment in time.

## 6.2 Support for the FMEA process

Failure Mode and Effect Analysis (FMEA) is the procedure of preventing process and production problems before they occur. FMEA processes are based on worksheets that contain important information about the system, such as the revision date or the names of the components. On these worksheets all the items or functions of the subject should be listed in a logical manner, based on the block diagram. For each item or function, the possible failure modes, effect and causes are listed and each of them are graded for their severity (S),



frequency of occurrence (F), and detection rating (D). Afterwards, the Risk Priority Number (RPN) is calculated by multiplying S, F and D. Once this is done it is easy to determine the areas of greatest concern.

The main steps of FMEA process described above can be sketched in one picture like in the following:

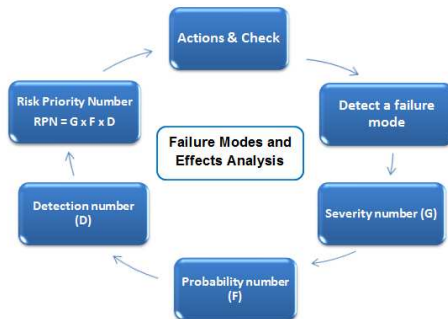


Figure 4 - Main steps of FMEA process

Decisions are necessary at each step from FMEA process, starting with identification of a potential failure mode, through grading steps, to the recommended actions.

At the beginning of the FMEA process the main tool from iDecisionSupport that is feasible to use is the list of discussion because the identification of the potential failure modes isn't fix or following guidelines. After the Discussion List tool synthesize the result, the chosen failure modes are graded based on the guidelines and based on each team member's experience. For this activity the most suitable is the Voting Tool from iDecisionSupport system. At the end of the FMEA process the recommended actions can be identified using the Brainstorming tool.

Although one person is typically responsible for coordinating the FMEA process, all FMEAs are team based. The scope of a FMEA team is to bring a range of perspectives and experiences in the project. Because each FMEA is unique in dealing with different aspects of the product or process, FMEA teams are hard to meet due to the unavailability (de-localized team, overlap of membership between the teams) of team members.

Based on the asynchronous and de-localization capability of the iDecisionSupport there was the obvious choice to integrate the iDecisionSupport in the FMEA process.

Thus, iDecisionSupport is under implementation at a local automotive manufacturer to facilitate this process. Mainly, it is used in the grading process where different meetings are created and participants are invited to grade the severity, frequency of occurrence and detection for items. The results of the grading are stored back into the main system as described in McDermott, Mikulak, Beauregard, 2007. The FMEA methodology has for each industry specific guidelines in order to facilitate the grading process, but this grading phase is influenced by their own knowledge of each member of the FMEA team.

However, before the process arrives at the grading phase there is the need of identifying the potential failure phase, and here

the iDecisionSupport system can facilitate the process. For example, some items must be discussed before they can be graded or they simply need some details collected from each participant. In this case, a workflow is created where the first step is performed in a discussion list or brainstorming meeting that is followed automatically by the grading (vote) meeting.

## REFERENCES

- A. Dolgui, O. Guschinskaya, N. Guschinsky, G. Levin, Decision Making and Support Tools for Design of Machining Systems, *Encyclopedia of Decision Making and Decision Support Technologies*, Vol I, 2008
- A. V. Georgescu, C. Candea, C. B. Zamfirescu, iGDSS - Software Framework For Group Decision Support Systems, In *Proceedings of The Good, The Bad and The Unexpected Conference* (2007).
- B. Grosz, S. Kraus, Collaborative plans for complex group action. *Artificial Intelligence*, 86, 269-357, (1996)
- C.B. Zamfirescu, C. Candea, S.I. Luca, On Integrating Agents Into GDSS, In *Preprints of the 9th IFAC / IFORS / IMACS / IFIP/ Symposium on Large Scale Systems: Theory and Applications*. Bucharest (2001), 231-236.
- C. Candea , G Candea . I portal and factory hub. In *Digital Factory for Human –Centred Production Systems* ( Luca Caneta, Myrna Flores, Claudia Redaelli, eds) , pp. 271-282, Springer, London ( 2011).
- D.J. Power, Decision Support Systems Concept, *Encyclopedia of Decision Making and Decision Support Technologies*, Vol II, 2008
- F.G. Filip, *Sisteme suport pentru decizii (Decision Support Systems )*. 2nd edition , Ed. Tehnica, Bucuresti, (2007) ( in Romanian)
- F. G. Filip . d.A. Donciulescu, Cr. I. Filip.Towards intelligent real- time decision support systems for industrial milieu. *Studies in Informatics and Control*, vol 11, no 4 , 303-311 (2002)
- G. DeSanctis, R.B. Gallupe, A foundation for the study of group decision support systems, *Management Science*, Vol. 33, No5, USA (1987)
- G.E. Philips-Wren, Intelligent Agents in Decision Support Systems, *Encyclopedia of Decision Making and Decision Support Technologies*, Vol II, 2008
- J Yen, X. Fan, S. Sun, T. Hanratty, J. Dumer, Agents with Shared Mental Models for Enhancing Team Decision-Makings, *Decision Support Systems*, (2006)
- P.R. Cohen, H.J. Levesque Teamwork. *Cognitive Science and Artificial Intelligence*, 25(4), 487-512. (1991)
- R.E. McDermott, R.J. Mikulak, M.R. Beauregard, The basics of FMEA. *Productivity Press*, USA, 2008.