

Towards Intelligent Collaborative Decision Support Platforms

Ciprian CANDEA¹, Florin Gheorghe FILIP²

¹ ROPARDO Ltd.;
2A, Reconstructiei Street,
Sibiu, RO-550129, Romania
ciprian.candea@ropardo.ro

² The Romanian Academy (BAR and INCE);
125, Calea Victoriei,
Bucharest RO-910071, Romania
ffilip@acad.ro

Abstract. The paper is about the use of new concepts and information technologies (IT) in the field of computer supported group decision-making. iDS (intelligent DecisionSupport) , a practical IT platform that supports the activities in the group decision-making is used as a vehicle to illustrate the methodology and evolution of the solution adopted under the influence of new information technologies and practical applications. iDS has been designed and evolved to be useful, usable and used, by avoiding the need for complex training and preventing user rejection. It provides a collaborative working environment where group members can create and attend different types of decision sessions, named 'work sessions'. The iDS platform covers collaborative and group decision sessions, as well as individual decision sessions in a way that users on each decision stage are free to collaborate with other users.

Keywords: cloud computing, decision-making model, mobile computing, service oriented solutions, workflow.

1. Introduction

As presented in (Keen 1980; Filip 1995, 2008; Power and Phillips-Wren 2011; Borne et al 2013: p 295), the systems meant to support decisions evolve under the influence of various factors such users' requirements and skills, technology development, usage practice and results and so on. The story of DISPATCHER family of DSS (decision support systems) designed, evolved and used in continuous process industry in 1980's and early 2000's (Figure 1) is described in (Filip, 2011, 2012). Since then the technology and business models

changed at a high pace and new solutions have been designed and deployed.

Groups represent a special case that assumes several people, possibly located in different places, are involved in decision-making activities(Filip, 2008). The modern information technologies such as mobile computing, web technology and service oriented architectures enable the implementation and deployment of effective solutions. The paper presents iDS, a practical IT platform meant to support group decision-making activities, which is used as a vehicle to illustrate the modern technologies

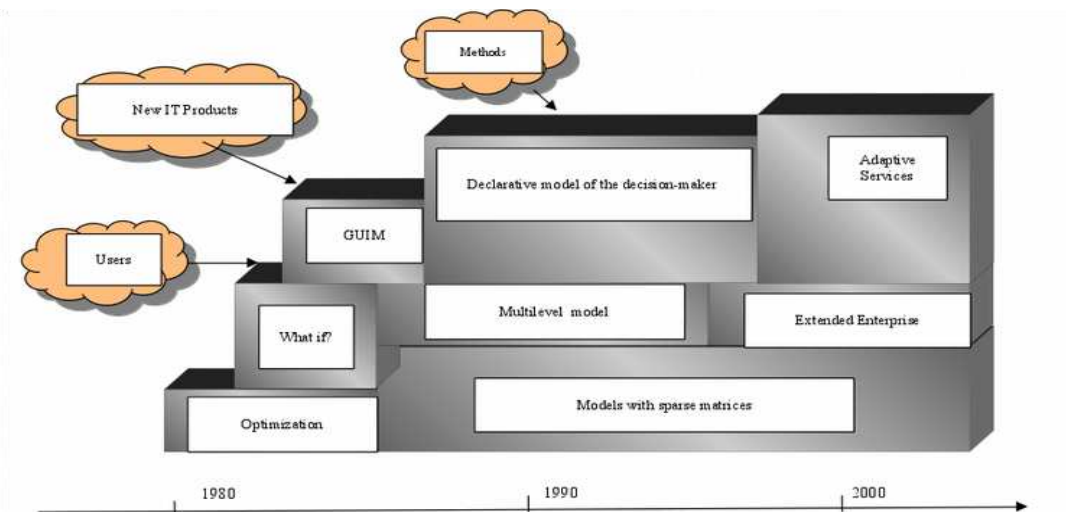


Figure 1. Evolution of DISPATCHER DSS family (from Filip, 2011, 2012)

and the evolutions in the field of computer supported group decision-making.

From a technical point of view, iDecisionSupport has been 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 et al 2007). To actively support user needs, the iDS platform facilitates each step of the decision making process – intelligence, design, choice, and implementation and review. For each of these phases, at least one software tool is provided so that the user could select and use the most appropriate one. To ensure a full flexibility and to respond to various application specific needs, iDS allows integrating third party tools.

The default set of tools proposed are:

- a) the *discussion list* (a forum-like tool for discussions),
- b) *voting* (a tool that allows grading or expressing the agreements over a set of issues),
- c) *electronic brainstorming* based on the *Issue Based Information System* (IBIS) approach (Conklin 2003; Conklin and Begeman, 1988),
- d) *Mind Map* (Buzan 2003), and
- e) *Categorization* (Sebastiani 2002).

The decision model that is implemented on the iDS platform is based on the *Shared Plans* theory (Grosz and Kraus 2002) and it was tested for the first time as a software prototype in 2001, as described in (Zamfirescu et al 2002).

2. The Concept

2.1 Terms and definitions

iDS platform uses terms as projects and plans, workflows, sessions and tools as there are shown in Figure 2. Reporting functionality is present at all iDS levels and gives users the possibility to search and extract data from application at any level of details, allowing the analysis of each aspect of the decision-making process. At *project and plan level*, definition of members' rights within the project; its goal and duration are defined. The *workflow* is

made up of a succession of sessions, with the possibility of being pre-defined or revised during the decision-making process. The *session* aim is to define the timeframe, members and the tool that is most representative for its purpose. The configuration functionality of the tool will also be provided. The tool is the key element for a session, being used to reach the goal for which the session was created. It can have its own interface and function in iDS context, or it can be a stand-alone application that integrates with the iDS API (Application Programme Interface).

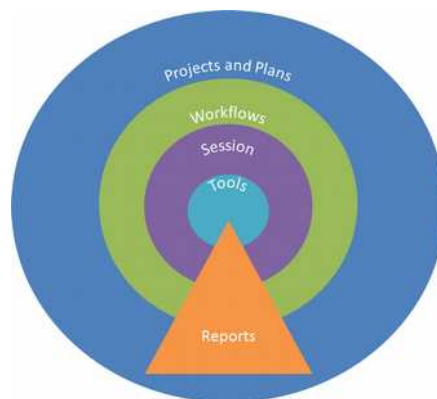


Figure 2. iDS concept

2.2 The decision making process model

Selecting the best solution presupposes in most cases passing through the steps of the decision process, such as intelligence, design, choice, implementation and review. iDS provides users with a hierarchical decision making model where users can organize decision processes as a tree structure. A leaf of the tree represents one decisional session that involves a tool. On top of leaves, any level of structure can be defined by starting with the root that can be represented by a plan or project with any levels of sub-plans.

In Figure 3 it is presented an example of iDS tree structure where *Project A* has two plans: *Plan A* and *Plan B*, each of them showing two key characteristics of this model. Firstly, a sub-plan or a plan may have several sessions that can run, in time, sequentially or in parallel. Secondly, sessions can be further organized using sub-plans. Because the model may evolve in time, it is not necessary for each leaf (sub) plan to have a minimum of one session.

As the time is a valuable resource in any decision process, iDS is representing time in all

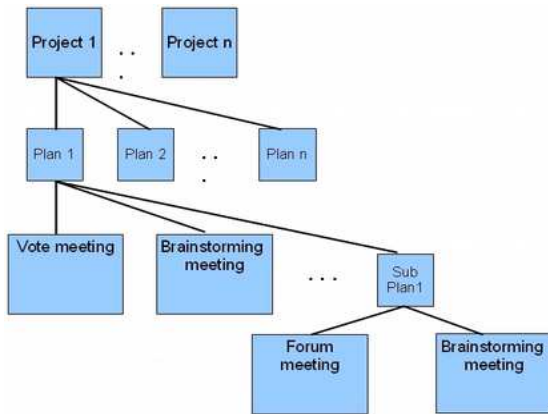


Figure 3. iDS tree structure of the decision making process

decision session phases. A decision session is defined as the period of time allocated to a specific decision making activity.

iDS models the decision session as a sequence of precise activities such as *draft*, *commit*, *work* and *report* (Figure 4). For each phase, a precise time frame must be allocated, except the Report that can be generated any time after *work* phase is closed.

Once the user considers using the iDS platform, he has to create a new decision session, in the *draft* activity. The user, as an author, is the only one that can view/access the session and he/she must configure it, namely describe the problem, decide whether he/she needs a commitment phase or not, and select his/her team members, as well as decide which tool is to be used. As soon as the user defines and configures the session, he/she is publishing the session and makes it known to the team members and, in this moment, the decision session is starting. If the *commitment* phase was selected, team members are informed regarding the time

deadline of the phase. During the time window available, the users discuss/negotiate important aspects for the session, such as exchanging supporting documents, accepting or denying to proceed to the next activity. Commitment automatically ends when time window expires and all users are informed about the conclusions and the team members are invited to the main *work* phase. During *work* activity, the session participants are using a decision support tool. Each session has only one decision support tool bind to it.

Choosing a certain decision support tool (during session configuration) establishes the type of the session (i.e. brainstorming sessions, voting sessions etc.). When time window for *work* activity expires, the system notifies all users with the result of the session and all *reports* are prepared. Workflow engine built in iDS platform is meant to manage the session flow and provide the flexibility needed for such a complex process.

2.3 Current version

Since the creation of the first version of iDS, technologies, software engineering techniques and computation power have evolved. Current iDS platform takes advantage of web 3.0 technologies to support collaborative work. It also integrates social network models into DSSs. The iDS platform implements a modular architecture that enables integrating third party tools though modern APIs (Application Program Interface). It also facilitates asynchronous decisions accessible through web 2.0 clients or dedicated mobile clients. Based on this system architecture, the iDS can be distributed as: *BaaS* (Business as a Software),

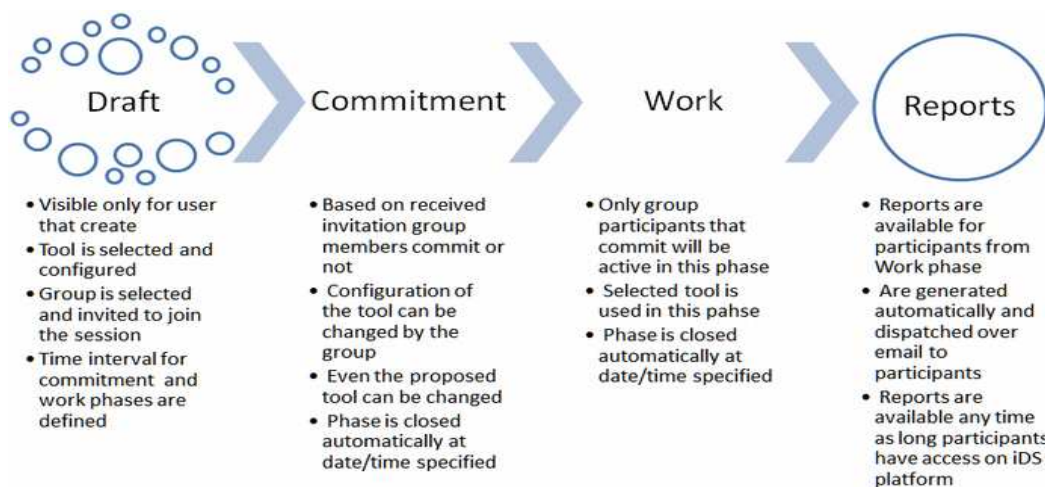


Figure 4. iDS session phases

SaaS (Service as a Software), *PaaS* (Platform as a Software) and *IaaS* (Infrastructure as a Software) (Radu et al 2014).

The functional characteristics of each variant are:

- *SaaS* (Figure 5b): software modules and all decision support tools as services for the customer company;
- *PaaS* (Figure 5c): APIs are accessible, new tools may be plugged in the iDS platform, and new GUIs (Graphic User Interface) or mobile clients can be deployed;
- *IaaS* (Figure 5d): custom deployment models may be established at this level, for each specific customer. Due to platform flexibility, different decision processes can be defined to respond to specific business models, such as: managing meetings, performing analysis regarding customer needs or project’s costs and benefits, risk assessment and so on.

2.4 User roles

The system supports three main user roles: *facilitator*, *active*, and *observer*. The roles are valid for each decisional session; the facilitator role allows a user to perform all operations on each of the above entities. In an implicit manner, the iDS user is a facilitator for his/her decision tree (including all decision sessions on that tree) but he/she plays an implicit active role for any decision session where he/she is invited.

The *active role* enables the user to participate in the decision session. However, the user is not allowed to configure the session. He/she can only perform actions, such as voting,

commenting, adding ideas or any other specific decision session action.

The *observer role* means that the user can only watch what happens in the decision session, without having the right to actively participate.

One special function is the “anonymity” of the user. When this function is activated for a specific decision session, in the work phase, all user inputs (i.e. voting preferences, proposed ideas during brainstorming and so on) are expressed in an anonymous manner and no indication to identify the person is available. iDS does not store this data at all.

3. Platform Description

The iDS server has three pillars (Figure 6): server, clients and tools; any of them can be distributed across the cloud.

The server is the central component that handles the decision sessions, registers tools, provides access rights for users, and assures global security of the system, including *REST* (Representational State Transfer) APIs; communication is encrypted using *SSL* (Secure Socket Layer) protocol.

iDS clients are provided access, via a GUI interface, to the entire system. There are three types of clients grouped into two categories:

- Clients for *administrative* proposes allowed to configure and administrate the entire iDS platform, namely:
 - The web admin client;
- Clients for *end-user* proposes:
 - Web 2.0 Client;

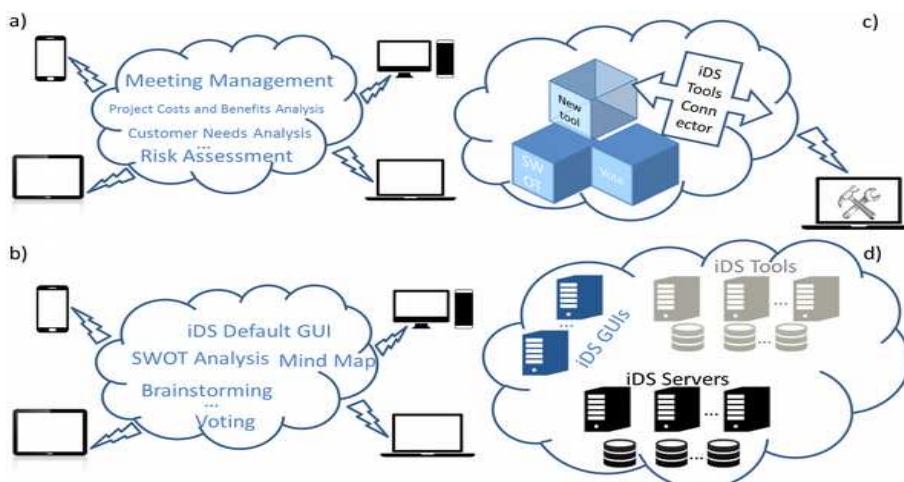


Figure 5. a) iDS BaaS level; b) iDS SaaS level; c) iDS PaaS level; d) iDS IaaS level (from Radu et al, 2014)

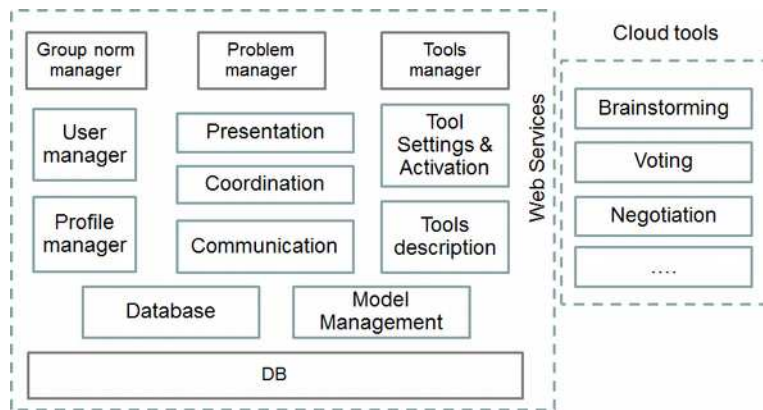


Figure 6. The iDS platform architecture

- Mobile Client, for *mobile operating systems*, such as Android® and iOS®.

Decision support *tools* implement various parts of the decision process. They can be used to model complex decision workflows. The tools can be integrated with the iDS core part through *iDS Tools Connector*.

Each of the above components (including individual tools) can reside anywhere on the Internet. The integration is done through REST API web services of the iDS server.

3.1 The iDS server

The iDS server is the central part of the system and presents a decentralized architecture (Figure 7). The modules are designed with high scalability in mind.

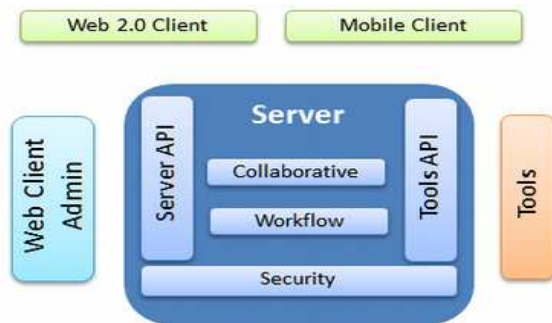


Figure 7. The iDS server architecture

One particular aspect of the architecture is the “plug in” feature of the cloud tools to implement the *tool as a service* concept. iDS provides a platform for decision support system, where different tools can be used, if are accessible on the cloud. Consequently, it is possible to create a “tool marketplace”, where different providers sell their special decision supporting tools.

The dedicated architecture and the definition of interfaces between iDS platform and tools (Figure 8) enable the supporting “plug in” functionality.

The server and tools can exchange information using syntax (message format) and semantics (message meaning). For the moment, the system is not implementing a communication standard, such as *KQML* - Knowledge Query and Manipulation Language (Finin et al 1994) but uses *XML* and *XSLT*, a language for transforming original XML documents into other XML documents (Clark 1999) for encoding and interpreting the syntax and the semantics.

To register a new tool into the iDS platform, it is necessary to follow a manual or automatic registration procedure. In both cases, the tool must provide to the server through the iDS tools with API import data structures serving to: a) configuration parameters, and b) result structuring.

The *key data* for registering a new tool with iDS platform are:

- URL, where is responding – can be any Internet location;
- Help content, the tool help content (as CSS(Cascading Style Sheets)/HTML);
- Tool connections, a list of connections of the tool with other tools. This implies that the registered tool will be able to pass its

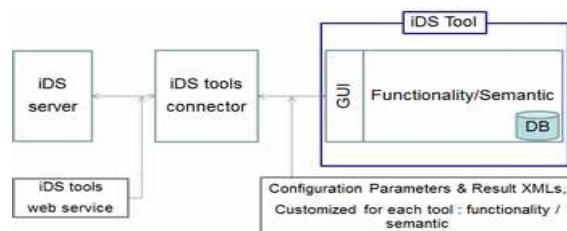


Figure 8. The iDS tool connector architecture

results to other existing tools. For this, an XSLT file (which will handle the XML results' transformations) must be provided;

- Tool configuration, as XML and XSD(XML Schema Definition), the schema for the tool configuration;
- Tool result, as XML and XSD the schema for the tool result;
- Report template, for the tool results, so that the iDS server can properly generate a report.

The server stores the tool intermediate and final results in XML format. Each tool may use the standard iDS XML format or may use its own format, in which case, it must provide a XSLT (EXtensible Stylesheet Language) transformation from and to its own format. One of the server's functionalities is to store these XSLT transformation files as connections between any two different tools. If both of the tools use standard results, then a transformation file is not necessary. It can be provided, if there are any specific conversions to be done. Based on the above technical implementations, any session can take as input the results of one or more previous sessions. In this way, one can easily create a chain of sessions (Figure 9).

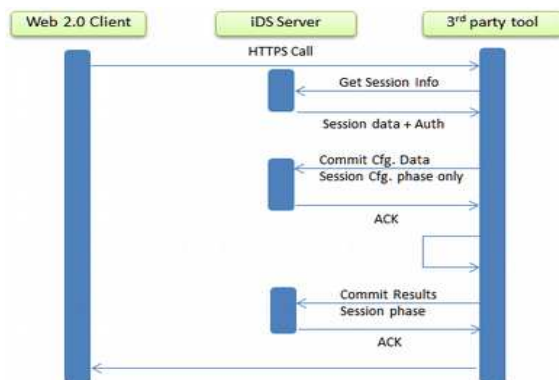


Figure 9. iDS Server – Tool communication protocol sample

All these structures, saved on the server side, will be used in communication process (as seen in Figure 9) with the 3rd party tools. For an easy integration, a library is published as “open source” – iDS tool Connector (ITC) – that allows any provider to quickly learn how to integrate it with the platform.

3.2 The GUI of the Web client

Based on the experience gained during the implementation of iDS platform in different organizations (Zamfirescu et al 2001, Candea et

al 2012) and based on users' interviews / feedback received after 6 month of platform usage, two features were obvious:

- There is a need to have smart clients to interact with the system;
- The User Interface Terminology must be adapted to something that the user is familiar with.

The first version of iDS was created with a Web GUI that was representing a cutting edge approach for such an application domain. At present, Web 2.0 iDS Client application is available for users who can easily access all platform functionalities. The actual Web 2.0 iDS Client is pushing rich functionalities to the users that facilitate them to focus on the decision process and less on the supporting aspects such as discussion forum, to do list, tree like categorization, project definition, discussion list, file management, members management, calendar management, social network, etc.. All of the features above are forming iDS Web 2.0 “*smart client*”.

Based on the feedback, plans were named objectives (sub-plans were named sub-objectives) and sessions were called meetings. The users consider that such terms are easier to be adopted. iDS GUI is coming with a predefined action flow: define projects with objectives; for reaching the objectives, the users attend multiple online meetings that are enriched with decision support tools.

In Figure 10, a part of the functionalities is presented: the project that is called VFF has several plans (objectives) defined; objectives contain sessions (meetings); one session is bound to a particular decision support tool (e.g.: Action Plan, Vote, SWOT analysis).

Other functionalities provided by this GUI may be noticed: the user has a customizable profile, has access to a personal calendar (where iDS meetings are automatically displayed) and may exchange ideas and share information, from key areas of the application, by having access to forums and discussion lists; the iDS system provides functionalities specific to Collaborative Platforms.

3.3 The iDS Tools Connector

By using the iDS Tools Connector (ITC), any decision support tool (web based or client-server) can be connected / communicate with

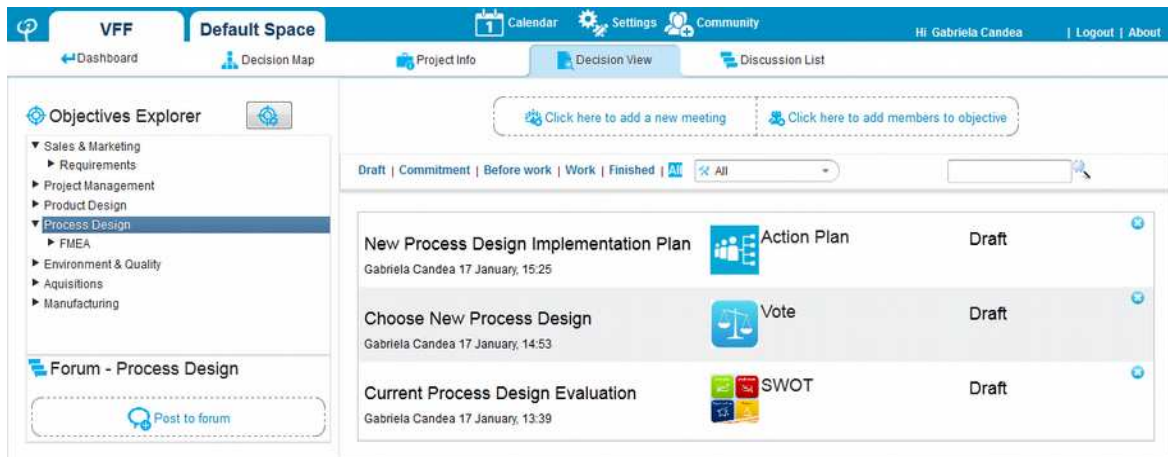


Figure 10 The iDS Web Client

the iDS platform. It essentially defines a communication protocol between iDS and the tools, based on two APIs: one for the iDS platform and another for the tools (Figure 11).

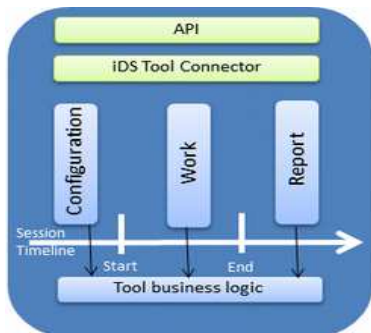


Figure 11 ITC architecture

The iDS API allows the iDS server to enroll – command - unregister a tool (Figure 11). ITC implement all the sequential phases - configuration, run and report - needed for a

decisional session. There are three separate behaviors (Figure 11) that the tool must show, depending on the decision session status. Each of the three phases: Work, Configuration and Report have their own GUI and their own functionalities.

Business logic is producing output results, once that are available are communicated to the iDS server. Ontology (Figure 12) was defined for the interaction and communication between any tool and the iDS server. This approach allows all iDS tools to have common semantics.

The Tool API has two parts. Firstly, there is a programming interface that contains a list of operations through which the ITC can request the tool to perform actions. The requests coming from iDS are forwarded, from the server, to the tool, via this interface. Secondly,

name	Vote
description	Decision support tool for voting sessions
class	Ways of Choosing
parameter (2)	
participants	
phases	
configuration	
startTimestamp	Tue Jul 25 14:00:00 EEST 2012
endTimestamp	Tue Jul 25 15:00:00 EEST 2012
parameter (2)	
items	
commitment	
startTimestamp	Tue Jul 25 15:00:00 EEST 2012
endTimestamp	Tue Jul 26 10:00:00 EEST 2012
committed (2)	
action	participant=4 type=comment value=I will not participate valueType=String lastModi...
uncommitted	participant=4 lastModifiedTimestamp=Tue Jul 26 08:15:00 EEST 2012
committed	participant=3 lastModifiedTimestamp=Tue Jul 26 09:05:00 EEST 2012
run	
startTimestamp	Tue Jul 26 14:00:00 EEST 2012
endTimestamp	Tue Jul 27 14:00:00 EEST 2012
items	
action	participant=1 type=comment value=Hello! Let's begin the voting. valueType=String...
report	
startTimestamp	Tue Jul 27 15:00:00 EEST 2012
result (3)	

Figure 12. Message structure



Figure 13. Tool message structure

there is a web service that allows the tool to send data to the iDS server, asynchronously.

We successfully have integrated so far decision support tools like: Vote, Brainstorming, SWOT (Figure 13), Action Plan, Mind Map and Categorizer.

4. Evolution

Since 1999, when Ropardo Research and Innovation (www.ropardo.ro) started investigation on how to create a practical decision support system, up to the present times of cloud computing, mobile wireless networks and service orientation, iDS was implemented in different organizations, also served as a research instrument to test new information technologies and methods.

In 2001, an integrated agent-based model for GDSS supporting explicit representation of the decision-makers role, the procedural and contextual settings along with the group commitment to share a plan of actions as a way to achieve a common goal, results in improved capabilities, range and flexibility of GDSS was presented in (Zamfirescu et al 2002).

During the time period 2006 – 2007, the system was adapted to academic environment by using “several intelligent software tools that assist the process of quality assurance and management, such as: students’ performance indicators extractor, electronic voting for the selection of grants proposals, quality evaluation questionnaires manager (questionnaires generator, distributor and analyzer (Oprean et al 2009). The GDSS system was integrated with *eUNIV* (Candea et al 2008), a project that transferred an e-business solution of knowledge management to the academic environment, and on the university information system.

Latter, in 2007, the iGDSS software framework for decision support systems focused on developing a conceptual tool where any third party can contribute with creative ideas for modeling the decision process (Georgescu et al 2007).

The iGDSS system was implemented for public administration based on “e Collaborative Decision solution developed for the academic and public administration” (years 2006 – 2008). A successful integration with the m-Business solution that address Small and Medium Enterprises (SME) was realized in order to adapt the solution for a new market (Oprean et al 2002; Radu et al 2014).

Adapting the iGDSS for deployment in manufacturing industry was a task that started in 2006 with *Digital Factory* (DiFac) project (Câdea and Câdea 2012; Câdea et al 2014). DiFac aims at the development of an innovative *Collaborative Manufacturing Environment* (CME) for next generation digital manufacturing (Sacco et al 2007). Latter 2012 iDS platform was successfully integrated in a complex Virtual Factory Framework (Jain 1995) environment and get implemented in different manufacturing factories in Europe (Sacco et al 2012).

In 2014 iDS is presented in an article published in Neurocomputing (Zamfirescu et al 2014) exposing the latest development in the iDS platform related to *Group Decision Process Design* (GDPP) as “a human-computer interaction engineering approach to design a software prototype that provides personalized, contextual and actionable recommendations for the GDPD”.

In time, software solutions evolve, starting with a simple client server implementation, passing

to the first web version and now taking advantages of cloud computing and becoming a complex software platform as complex as an *ERP* (Enterprise Resource Planning). The name of the software also evolved from GDSS (Group Decision Support System) passing to iGDSS (intelligent Group Decision Support System) to iDSS (intelligent Decision Support System) and ending to iDS platform (intelligent Decision Support).

Future research will include Cloud Computing for GDSS as a research domain, and in particular, how iDS platform can benefit from these technologies that are already on the market. Quick scale of computing capabilities, resource pooling will allow more sophisticated decisional tools to be deployed over Internet on different platforms: mobile devices (phones, tablets), desktop and laptop computers, as well as industrial devices.

REFERENCES

1. BORNE, P., D. POPESCU, F. G. FILIP, D. STEFANOIU, **Optimization in Engineering Sciences. Exact Methods**, J. Wiley & Sons, London, 2010.
2. BUZAN, T., **The Ultimate Book of Mind Maps: Unlock Your C, Boost Your Memory, Change Your Life**, Harper Collins, 2005.
3. CÂNDEA, C., G. CÂNDEA, F. G. FILIP, **iDecisionSupport – Web-based Framework for Decision Support Systems**, in Proceedings of the 14th IFAC Symposium on Information Control Problems in Manufacturing, INCOM 2012, pp. 1117-1122.
4. CÂNDEA, C., G. CÂNDEA, **i-Portal and Factory Hub**. In: Caneta, L., Flores, M., Redaelli, C., eds, *Digital Factory for Human Centred Production Systems*, Springer, London, 2011: pp 271-282.
5. CÂNDEA C., A. GEORGESCU, G. CÂNDEA, **iPortal-Management Framework for Mobile Business**, 2008; In: Proceedings of the International Conference on Manufacturing Science and Education MSE 2009, Sibiu, Romania, ISSN 1843 - 2522, 4-6 June.
6. CÂNDEA, C., G. CÂNDEA, C. RADU, W. TERKAJ, M. SACCO, O. SUCIU, **A Practical Use of the Virtual Factory Framework**. In: 14th International Conference on Modern Information Technology in the Innovation Process of the Industrial Enterprises, Budapest, Hungary, 2012.: p 256-266). (available at https://igor.xen.emi.sztaki.hu/mitip/media/MITIP2012_proceedings.pdf#page=256%20 (literal accessed on 26.01.2016).
7. CLARK, J., **Xsl Transformations (xslt)**, World Wide Web Consortium (W3C), 1999. (available at <http://www.w3.org/TR/xslt> accessed on 25.01.2015).
8. CONKLIN, J., **The IBIS Manual: A Short Course in IBIS Methodology**, Touchstone, 2003.
9. CONKLIN, J., M. L. BEGEMAN, **gIBIS: A Hypertext Tool for Exploratory Policy Discussion**. ACM Transactions on Information Systems (TOIS), 1998, vol. 6(4), pp. 303-331.
10. FILIP, F. G., **Towards More Humanized Real-time Decision Support Systems**, in: L. Camarinha-Matos, H. Afsarmanesh, eds, *Balanced Automation Systems: Architectures and Design Methods*, Chapman & Hall, 1995: pp. 230-240.
11. FILIP, F. G., **Decision Support and Control for Large-scale Complex Systems**, Annual Reviews in Control, 2008, vol. 32(1), pp. 61-70.
12. FILIP, F. G., **Designing and Building Modern Information Systems: a Series of Decisions to Be Made**, Computer Science Journal of Moldova, 2011, pp. 119-129.
13. FILIP, F. G., **A Decision-making Perspective for Designing and Building Information Systems**. International Journal of Computers Communications and Control, 2012, vol. 7(2), pp. 264-272.
14. FININ, T., R. FRITZSON, D. MCKAY, R. MCENTIRE, **KQML as An Agent Communication Language**. In: Proceedings of the Third International Conference on Information and Knowledge Management, 1994, ACM: pp. 456-463.
15. GEORGESCU, V., C. CÂNDEA, C. B. ZAMFIRESCU, **iGDSS – Software Framework for Group Decision Support**

- Systems**, In: Sapio, B., Fortunati, L., Haddon, L., Kommonen, K-H., Mante-Meijer, E., Turk, T., eds, Proceedings, The Bad and The Unexpected Conference, COST Action 298 "Participation in the Broadband Society", European Science Foundation, Moscow, Russian Federation, 2007, ISBN: 5-901907-17-5 (available at <http://www.iis.ru/en/content/view/328/91/>, accessed on 25.01.2016).
16. GROSZ, B., S. KRAUS, **Collaborative Plans for Complex Group Action**. AI, 1996, vol. 86, pp. 269-357.
 17. JAIN, S., **Virtual Factory Framework: a Key Enabler for Agile Manufacturing**, In Emerging Technologies and Factory Automation, 1995. ETFA'95, Proceedings, 1995 INRIA/IEEE Symposium on, Vol. 1: pp. 247-258, IEEE.
 18. KEEN, P., **Adaptive Design for Decision Support Systems**, Data Base, 1980, vol. 12(1), pp. 15-25.
 19. OPREAN, C., C. V. KIFOR, S. C. NEGULESCU, C. CÂNDEA, C. OPREAN, **eCollaborativeDecisions – a DSS for Academic Environment**. In: Proceedings of the World Congress on Science, Engineering and Technology & International Conference on Computer, Electrical, and Systems Science, and Engineering, 2009, pp. 173-179.
 20. OPREAN, C., I. MOISIL, C. CÂNDEA, **eUniv: An e-Business Solution for a University Academic Environment**, In: Proceedings of 3rd Global Congress on Engineering Education, Glasgow, Scotland, United Kingdom, ISBN 0732622018, 2002, pp. 363-366,
 21. POWER, D. J., G. PHILLIPS-WREN, **Impact of Social Media and Web 2.0 on Decision-making**. Journal of decision systems, 2003, vol. 20(3), pp. 249-261.
 22. RADU, C., C. CÂNDEA, G. CÂNDEA, C. B. ZAMFIRESCU, **Towards a Cloud-Based Group Decision Support System**, In A. Bădică et al. (Eds.) Recent Developments in Computational Collective Intelligence, Studies in Computational Intelligence, Springer, vol. 513, 2014, pp. 187-196.
 23. SACCO, M., C. RADAELLI, C. CÂNDEA, V. GEORGESCU, **DiFac: An Integrated Scenario for the Digital Factory**, In: Proceedings of the 15th International Conference on Concurrent Enterprising, ICE 2009, Leiden, The Netherlands, ISBN 978-0-85358-259-5 (available at: https://www.researchgate.net/profile/Marco_Sacco/publication/228361391_DiFac_an_integrated_scenario_for_the_Digital_Factory/links/0deec51ac2f1e18058000000.pdf accessed on 25.01.2016)
 24. SACCO, M., C. REDAELLI, C. CONSTANTINESCU, G. LAWSON, M. D'CRUZ, M. PAPPAS, **DiFac: Digital Factory for Human Oriented Production System**, In Human-Computer Interaction. HCI Applications and Services, Part IV, Springer Berlin Heidelberg, 2007, LNCS 4552, pp. 1140-1149.
 25. SEBASTIANI, F., **Machine Learning in Automated Text Categorization**, ACM Computing Surveys (CSUR), vol. 34(1), 2002, pp. 1-47.
 26. ZAMFIRESCU, C. B., C. CÂNDEA, M. LUCA, **On Integrating Agents into GDSS** In: F. G. Filip, I. Dumitrache and S.S. Iliescu (Eds.) Large Scale Systems: Theory and Applications. A Proceedings volume Proceedings of the 9th IFAC / IFORS / IMACS / IFIP/ Symposium, Bucharest, Romania, 2001, Pergamon Press, an Imprint of Elsevier Science, Oxford: pp. 433-438.
 27. ZAMFIRESCU, C. B., C. CÂNDEA, **On Interacting with Collective Knowledge of Group Facilitation**, In: 5th International Conference on Computational Collective Intelligence Technologies and Applications – ICCCI 2013, LNAI 8083 September 11-13, Craiova, Romania, 2013, pp. 255-265.
 28. ZAMFIRESCU, B. C., C. CÂNDEA, C. RADU, **A Stigmergic Approach for Social Interaction Design in Collaboration Engineering**, In: Kwong, S. and Zhang, Q., eds, Neurocomputing - Bridging Machine learning and Evolutionary Computation (BMLEC) Computational Collective Intelligence, vol. 146, pp. 151-163.