Towards a Cloud-Based Group Decision Support System

Ciprian Radu, Ciprian Cândea, Gabriela Cândea, Zamfirescu B. Constantin

Ropardo S.R.L., Sibiu, Romania
{ciprian.radu, ciprian.candea, gabriela.candea}@ropardo.ro,
zbc@acm.org

Abstract. Decision Support Systems have constantly benefited from the technological advances in Computer Science. Cloud Computing is a technology that could become useful for the Decision Support Systems, too. Moreover, Decision Support Systems may support the "global brain" programming paradigm. This paper presents iDS, a system designed to become a collaborative- and cloud-based Group Decision Support System. This Decision Support System will be made available as a Business as a Service model. After presenting the architecture of the developed iDS, the main design directions for integrating it into the cloud are described. The goal is to obtain a platform that supports collective intelligence, in terms of human-computer networking.

Keywords: Group Decision Support System, Cloud Computing, Collaborative Platform, Collective Intelligence

1 Introduction

The Group Decision Support System (Group DSS, GDSS) is defined as a combination of communication, decision and computer technologies working together to offer support for decision phases like: problem identification, formulation and solution generation, during group meetings [1].

Nowadays, the results of collaboration activities inside a group are based on coordination, decision making and negotiation. Most of these phases rely on knowledge of the organization. In many organizations, decision making situations are naturally recurring and the majority of them are critical.

Current research at Ropardo results in a collaborative decision making support platform, named iDS (iDecisionSupport) and designed to be easy to use by the majority of users (thus avoiding long trainings and preventing user rejection) [2]. The implemented decisional model is based on the Shared Plans theory [3]. It was tested for the first time in a software prototype by our group in 2001 [4].

Cloud Computing is not something new because it essentially refers to computing as a utility [5]. However, the term "cloud" started to become popular in 2006 [6].

According to [7], the essential characteristics of Cloud Computing are:
on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service.

Cloud Computing technologies are mainly classified in terms of service and deployment models [7]. There are three types of service models. With the Software as a Service (SaaS) model, the cloud offers applications that are available to clients through interfaces like web browsers. The user has no control over the cloud infrastructure. With Platform as a Service (PaaS) the consumer cannot control the cloud infrastructure but, custom applications (developed or acquired by the service user) can be deployed into the cloud. In Infrastructure as a Service (IaaS) the service user still cannot control the cloud infrastructure but, access to operating systems, storage and even some limited networking control are allowed. From the service models point of view, everything is a service. Hence, the XaaS term was coined [5], [8].

A Cloud Architecture has several deployment models: private, community, public and hybrid cloud. A private cloud is used solely by a single organization. The community cloud is meant for a certain community of users, from multiple organizations, which have some common interests. The public cloud is available to anybody. A combination of at least two of the above three types of clouds forms a hybrid cloud.

In this paper we focus on the advantages of the Business as a Service (BaaS) model, in which the system is not simply provided as a service to the customer but, it is also managed so that the business goals are met. Business as a Service proposes to the companies a mix between business support, proven methodology, strategy support and hosted technology. The BaaS model is cost-effective and it provides better business results because customers can benefit from the vendor's experience in how to best use the delivered system.

The main contribution is a collaborative- and cloud-based GDSS at BaaS level. The goal is to obtain a platform that supports human-computer networks.

The remainder of this paper is organized as follows. Section 2 briefly presents the related work. Section 3 describes iDS' architecture. Section 4 concludes this paper and outlines some further work directions.

2 Related Work

Besides supporting information access [9], a GDSS can, at the same time, radically change the dynamics of group interactions by improving communication, by structuring and focusing problem solving efforts and by establishing and maintaining an alignment between personal and group goals.

Traditional decision making processes follow an iterative process, having the following phases: analysis and definition of the problem, divergence and then convergence on the set of possible solutions and lastly choosing the final one. Since the first DSSs [1], many improvements were done by the research community, based on technological advances. For example, Web 2.0 and 3.0 technologies are nowadays used to support collaborative-based GDSSs. Focus is put on the social aspects by adopting social network models into DSSs [10].

This paper focuses on the integration of a GDSS solution with Cloud Computing technology. According to [6], the current Cloud Computing state-of-the-art implementations have the following architectural characteristics: (1) uniform high capacity (maximum communication bandwidth between any two computers from the data center), (2) free Virtual Machine migration (a Virtual Machine can migrate from a physical machine to another one rapidly and easily), (3) fault-tolerance, (4) scalability and (5) backward compatibility.

DSSs using collaborative- and cloud-based platforms represent a rather new research field. GRUPO-MOD [11] is a collaborative decision making system based on analytical hierarchy processes. It is implemented using a client – server web architecture, which facilitates asynchronous decisions. Its authors stress out that modern collaborative decision making systems are transiting to Cloud Computing enabled solutions. To this end, they intend to bring their system at the SaaS level.

The iDS system presented in this paper is a collaborative decision making system based on a client – server web architecture, similar with GRUPO-MOD. Both systems adopt a plugin-based approach. The main difference is that we intend to bring iDS to an upper level: BaaS. While at SaaS level the vendor is responsible for the software applications provided, at BaaS level, the vendor is responsible for the entire business of a company, i.e. how and in what processes are the software applications used. This way, a company may remain focused on how to develop its business rather than how to implement specific actions.

Moreover, we intend to obtain with iDS a platform that supports the "global brain" concept, i.e. the network of all people and computers from our planet [12].

3 The iDS Architecture

This section presents the architecture of the developed iDS system. We start by presenting its decision making process model. Then we focus on its main subsystems. Finally, we present further architectural aspects intended for making iDS a computing utility (by integration with Cloud Computing technologies).

iDS is designed as a generic framework for decision support. To better detail its architecture, we will give an example of how it may be used for meeting management.

3.1 The Decision Making Process Model

Decision making can be defined as the process of choosing the best solution, from two or more possible ones, in order to solve a particular problem. It is especially characteristic to groups of people or to organizations.

The decision making model adopted in iDS is hierarchical. The decisional activities are associated with projects. A project has one or more plans. Any plan can have sub-plans and both are time-boxed. Otherwise, it can have one or more sessions. **Fig. 1** presents an example of a decision making process, which can be obtained with such a model. Project A, has two plans (i.e. A and B), each of them showing two key characteristics of this model. Firstly, a (sub-)plan may have several sessions that can run, in time, sequentially or in parallel. Secondly, sessions can be further organized using sub-plans. It is not necessary that each leaf (sub-)plan (leaf because the model is a tree) has at least a session. This is because the model may evolve in time. A session is defined as the period of time allocated to a specific decision making activity.

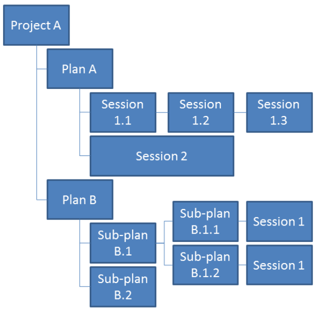

Fig. 1. iDS decision making process example

The session is modeled in iDS as a sequence of well-defined phases: *draft*, *commit*, *work* and *report*. While in *draft*, only the session's author has access to it. This phase is used for configuration purposes, which may also be done during *commit*. The difference between *draft* and *commit* is that once *draft* ends, the session becomes public to anyone invited to participate. Each person can accept or deny the participation to the session until the end of the *commit* phase. After the *commit* phase, the session enters in the *work* phase. There can be a break between these two phases, which allows for the work to be scheduled at a later time. During *work*, the session participants (which accepted to attend the session), use a decision support tool in order to make some decisions. The results of their work will be available after this phase ends, i.e. in the *report* phase.

This decision making process model may be enhanced with collective experience. For example, collaborative deliberation [12] is a "global brain" programming metaphor that involves both computers and humans in obtaining, analyzing and selecting solutions. Decision processes are part of the "global brain" programming language.

For the meeting management case study, hierarchical plans may be used to define a project's objectives. Each objective may have several sub-plans, which may represent different meetings (required for reaching the objective). Each sub-plan may have several sessions, which form the meeting's agenda.

3.2 The Developed iDS System

iDS operates as a software system accessible through the Internet. As such, both synchronous and asynchronous decision processes are supported. It provides a range of services that stimulate the collaboration between individuals. Using a common Application Programming Interface (API), iDS allows for pluggable decision support tools. This system was already successfully integrated in a virtual factory environment [13].

The system's core is the decision support engine. It provides various iDS services over a default Graphical User Interface (GUI). The decisional support is available through different decision support tools. The tools can be integrated with the iDS core through the tools plugin support module that acts as an iDS Tools Connector. The following subsections provide more details about the architecture of the developed iDS system. As it will be shown throughout the paper, iDS is designed to become a computing utility, by integrating it into the cloud.

The iDS Server.

The iDS server is the core of the system. **Fig. 2 a)** presents a top view on the server's architecture by showing its main subsystems and modules.

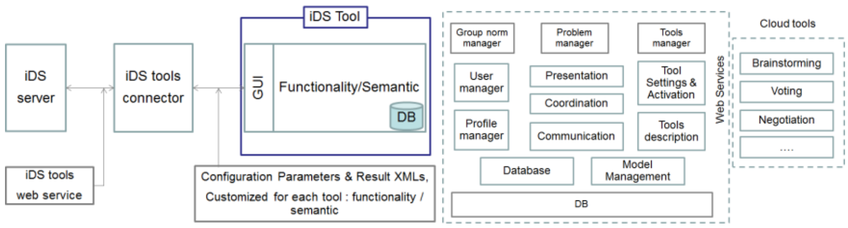

Fig. 2. a) The iDS server architecture b) iDS server components and cloud tools

iDS uses a Relational Database Management System for data persistence but, we are also investigating the benefits of using a semantic data model [13].

To enable the cloud paradigm, the system implements a decentralized architecture. It presents the concept of server and independent tools as a service. The system's communication mechanism is based on syntax (message format) and semantics (message meaning). For the moment, the system is not implementing yet a communication standard (like KQML) but, uses XML and XSLT for encoding and interpreting the messages. iDS considers the fact that, for a successful decision support, different tools must be used and nowadays many of them can be accessible from the cloud. For this reason, the iDS server is as in **Fig. 2 b)**, where different layers that support the users and their decisional process are presented.

iDS exposes its functionalities through web services. Data access is obviously secured. The entire iDS server can be administered remotely using a web application.

The iDS server has three pillars: groups, decisions/problem and supporting tools, and adopts the collective intelligence principles (openness, peering, sharing and acting globally).

The Group pillar refers to the iDS users, their roles and profiles within the system. Its main objective is to offer collaborative functionalities.

The Decision/problem pillar essentially refers to the decision making process model. This subsystem has the objective of managing the lifecycle of each session automatically, while structuring sessions hierarchically. It also communicates with the decision support tools by configuring them, commanding them and gathering their output. A reporting subsystem facilitates the aggregation of data received from the tools.

The decision support tools form another important pillar, which will be described in a later section of the paper.

To follow-up on the meeting management example, groups of people may be defined for the considered project (using the administration system). They may be involved in the decisions of different objectives, having different responsibilities (roles). Some may coordinate the project at a global level, setting its objectives. Others may act as decision making facilitators [15].

The Default Graphical User Interface.

The default iDS GUI was developed as a Web 2.0 application. Users can experiment with the decision making process model available in iDS. For convenience, plans are named objectives (sub-plans are named sub-objectives) and sessions are named meetings. We reached the conclusion that such terms are easier to adopt by users.

Fig. 3 illustrates a part of this GUI. A project called VFF has several plans (objectives) defined. An objective may contain sessions (meetings). A session is bound to a particular decision support tool (e.g.: Action Plan, Vote, SWOT).

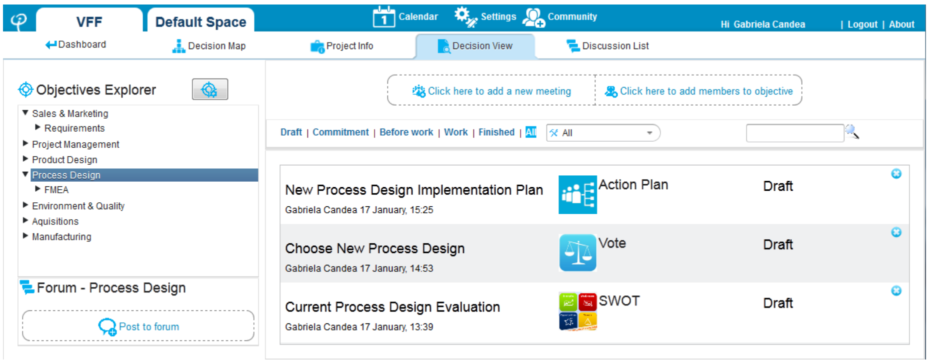

Fig. 3. The iDS default GUI

Other functionalities provided by this GUI may be observed. The user has a profile, access to a personal calendar and may exchange ideas and share information by having access to forums and discussion lists in key areas of the application. Overall, the iDS system provides functionalities specific to Collaborative Platforms.

This default GUI presents an implicit and generic view of the iDS system in general. It is intended however to provide particular views as well. A particular view will correspond to a specific business model. Such approach will allow iDS to be customized for different businesses. As it will be showed later in the paper, Cloud Computing provides advantages for a BaaS approach.

Meeting management will have a particular GUI, focused on objectives, meetings and their agenda. The GUI will access the iDS system's web services in a secure manner, starting from user authentication and continuing with defining objectives, meetings and configuring decision support tools.

Tools in Cloud.

The iDS Tools Connector (ITC) allows any software tool to be connected and communicate with the iDS system. It essentially establishes a communication protocol between iDS and the software tools, based on an API for the iDS system and an API for the tools. Potentially, any software can be adaptable to iDS.

The iDS API is a web service that allows the iDS server to: instantiate, disembody and command a tool. The tool is allowed to be in three sequential phases: configuration, run and report. A tool is initially configured manually by the user, or automatically by the system (as in **Fig. 2 a)**). Finally, the output it produces is made available to the iDS system. An ontology was defined for the interaction and communication between any tool and the iDS server. Based on the ontology that is represented in **Fig. 4**, we ensure that all tools available the cloud will refer the terms semantics of messages in a similar way [16].

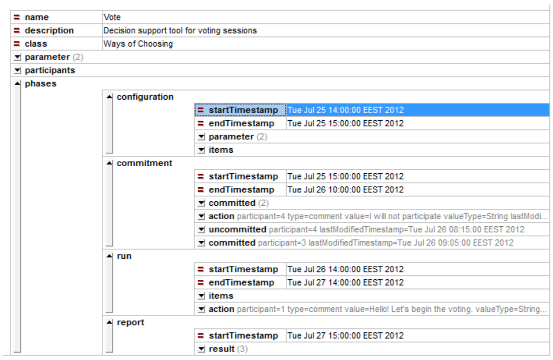

Fig. 4. Tool message structure

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 delegated via this interface. Secondly, there is a web service through which the tool can send data to the iDS server, asynchronously.

We successfully integrated so far decision support tools like: vote, brainstorming, SWOT, action plan, Mind Map and categorizer. As it will be showed in a subsequent section of the paper, Cloud Computing can increase the advantages of our approach.

For meeting management, facilitators will request iDS different support tools. As such, tools will be instantiated and then the users may configure and use them. When a tool becomes unneeded, iDS will remove it and free the resources allocated for it. In case a customer needs a decision support tool which is not available in iDS, such tool could be developed and easily integrated in the iDS.

3.3 Further Architectural Considerations

The architectural design of the iDS system includes its integration with Cloud Computing technologies. The objective is to create from iDS a GDSS which benefits from some of the characteristics of Cloud Computing. The iDS computing capabilities are foreseen to be provided without human interaction, through an on-demand self-service, and also to be available over the Internet, on different platforms like mobile phones, tablets, desktop and laptop computers (broad network access). Each iDS user will benefit from rapid elasticity of the computing capabilities. This will allow each user to quickly scale those iDS system parts which are mostly needed. The iDS resources will be made available (through virtualization) to multiple clients, without knowing their express (physical) location. Resource pooling will be thus another Cloud Computing characteristic that will be inherited by iDS.

The iDS system will be easily and broadly accessible through the network. It will be distributed across four levels: BaaS, SaaS, PaaS and IaaS.

Different customers will be able to manage their businesses' decision making processes by interacting with the iDS BaaS level, which is detailed in **Fig. 5 a).**

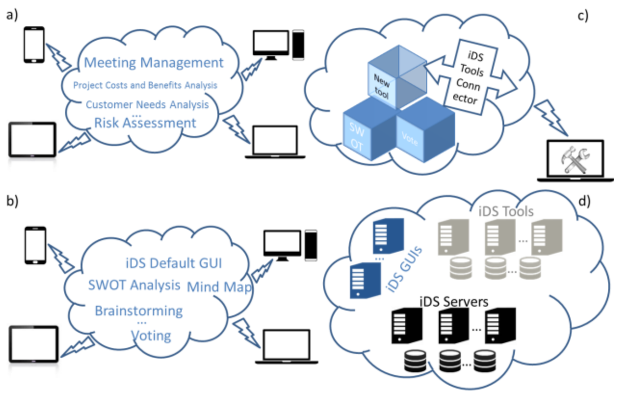

Fig. 5. a) iDS BaaS level; b) iDS SaaS level; c) iDS PaaS level; d) iDS IaaS level

At BaaS level, iDS will allow users to manage different types of decision processes that are encountered in their businesses. The generic decision process model developed in iDS can be used for example by a company to: manage its meetings, or to perform analysis regarding customer needs or project's costs and benefits, or even for risk assessment. Other scenarios can obviously be considered. In order to meet the particular needs of a certain business, different views of the iDS default GUI will be developed. They will then be integrated in a web portal. Therefore, each business will take advantage only from the iDS decision making functionalities that are of interest for it, presented in a customized manner.

As it is presented in **Fig. 5 b)**, at the SaaS level, iDS will provide all its software modules (default GUI, decision support tools) that can offer services to the end user.

The main difference between the iDS BaaS and SaaS levels is that, at business level, iDS is presented in a customized form, which can vary from customer to customer.

At the SaaS level, the user may be interested in experimenting with the iDS decision process model, by using the iDS default GUI. Also, the available decision support tools may be tested. To this end, the current iDS implementation already provides a public (default) space that can be used by users for trying out iDS.

At PaaS level (**Fig. 5 c)**), new tools may be plugged in the iDS system by exposing to the developer well defined APIs. A thorough presentation of the iDS Tools Connector is considered beyond the scope of this paper.

Finally, the iDS IaaS level is presented in **Fig. 5 d).** This level will only be accessible to the administrators of the iDS Cloud architecture.

An important problem will be addressed at this level: what deployment models of the iDS system (with all its components) must be used in order to properly meet the customers' needs and expectations. Since customers will most likely manifest particular interests regarding the iDS system, which are driven by their businesses, resource allocation will have to differ from one case to another.

One important objective for Cloud Computing is to require the customer to pay only for the needed resources, when they are needed. To this end, for example, decision processes should occupy only the computational resources (tools) that they need, when they need them. Moreover, each client will require computational resources only for the decision support tools that he or she needs.

At BaaS level, meeting management is just one example (see **Fig. 5 a)**) of how the iDS services may be aggregated into a business level service. In the cloud, a middle layer will be responsible with provisioning iDS services for different organizations and their business. For example, the relevant decision support tools will be made available from the most suitable geographic locations. Another example is that only the GUIs required for an organization's business will have resources allocated. Such custom, business-oriented configurations will be made in a semi-automatic way, without concerning the customers with the implementation details.

4 Conclusions and Further Work

This paper presented the developed iDS solution, a collaborative group decision support system. Its main architectural aspects were described where one important novelty was presented: the ontology for data exchange between tools. iDS provides a flexible decision making process model and it has pluggable decision support tools. Based on tools that now can run in the cloud (to support different business scenarios), meeting management is a wide spread scenario that is met in any company, and that can be easily supported by iDS.

To introduce decisional processes into the cloud, several actions, which need to be taken at infrastructure, platform, service and business level, were presented. However, there are still many research challenges in the field of Cloud Computing [5], [6], which will have to be considered when integrating iDS into the cloud.

Different advantages were presented for each level that is available in the cloud paradigm, and the ultimate goal, BaaS, was evaluated. At the BaaS level any company can benefit from the best in class tools as well best in class business models to manage their problems.

We will also study the integration of iDS with the CowdLang platform [15] towards collective intelligence and programming the "global brain". Our idea is to use iDS for the decision operators from the CrowdLang programming language [17].

References

1. DeSanctis, G., Galluoe, B.: A Foundation for the study of Group Decision Support Systems, Management Science, pp. 589-609 (1987)
2. Georgescu, V., Candea, C., Zamfirescu, C. B.: iGDSS - Software Framework For Group Decision Support Systems, In Procedings of The Good, The Bad and The Unexpected Conference (2007)
3. Grosz, B., Kraus, S.: Collaborative plans for complex group action. Artificial Intelligence, 86, 269-357 (1996)
4. Zamfirescu, C.B., Candea, C., Luca, S.I.: On Integrating Agents Into GDSS. In F.G. Filip, I. Dumitrache and S.S. Iliescu (Eds.) Preprints of the 9th IFAC / IFORS / IMACS / IFIP/ Symposium on Large Scale Systems: Theory and Applications. Bucharest, Romania, p. 231-236. ICI Press, ISBN 973-98407-8-7 (2001)
5. Armbrust, M., et al.: Above the Clouds: A Berkeley View of Cloud Computing. EECS Department. University of California, Berkeley (2009)
6. Zhang, Q., Cheng, L., Boutaba, R.: Cloud computing: state-of-the-art and research challenges. Journal of Internet Services and Applications 1.1, 7-18 (2010)
7. Mell, P., Grance, T.: The NIST definition of cloud computing. NIST special publication 800: 145 (2011)
8. Rimal, B. P., Choi, E., Lumb, I.: A taxonomy and survey of cloud computing systems. Fifth International Joint Conference on INC, IMS and IDC. NCM'09, 44-51 (2009)
9. DeLone, W.H., McLean, E.R.: Information systems success: The quest for the dependent variable. Information Systems Research, 3(1), 60-95 (1992)
10. Antunes, F., Costa, J. P.: Integrating Decision Support and Social Networks, *Advances in Human-Computer Interaction*, (2012)
11. Thimm, H.: Cloud-Based Collaborative Decision Making: Design Considerations and Architecture of the GRUPO-MOD System. IJDSST 4.4, 39-59 (2012)
12. Bernstein, A., Klein, M., Malone, T. W.: Programming the global brain. *Commun. ACM* 55, 5, 41-43 (2012)
13. Candea, G., Candea, C., Radu, C., Terkaj, W., Sacco, M., Suciu, O.: A practical use of the Virtual Factory Framework. 14th International Conference on Modern Information Technology in the Innovation Process of the Industrial Enterprises, Budapest, Hungary (2012)
14. Dickson, G., Poole, S., DeSanctis, G.: An overview of the GDSS research project and the SAMM system, In Bostrom, Watson, Kinney (Eds.), Computer Augmented Teamwork: A guided tour. Van Nostrand Reinhold (1992)
15. Bernstein, A., Klein, M., Malone, T. W.: Programming the global brain. *Commun. ACM* 55, 5, 41-43 (2012)
16. Jasper, R., Uschold, M.: A Framework for Understanding and Classifying Ontology Applications, In Twelfth Workshop on Knowledge Acquisition Modeling and Management KAW'99 (1999)
17. Minder, P., Bernstein, A.: Crowdlang-first steps towards programmable human computers for general computation. In Proceedings of the 3rd Human Computation Workshop (HCOMP 2011). AAAI Press (2011)