Distributed Information Systems in Group Decision Making Problems

Stanislao Grazioso
Student Member, IEEE
Department of Industrial Engineering
University of Naples Federico II
CREATE Consortium
80125, Napoli, Italy
stanislao.grazioso@unina.it

Mateusz Gospodarczyk
Student Member, IEEE
Department of Computer Science
University of Rome Tor Vergata
00133, Rome, Italy
mateusz.gospodarczyk@uniroma2.it

Giuseppe Di Gironimo
Department of Industrial Engineering
University of Naples Federico II
CREATE Consortium
80125, Napoli, Italy
giuseppe.digironimo@unina.it

***Abstract*—This paper presents the role of distributed information systems in enhancing multiple criteria decision making problems. This goal is achieved showing the architecture and implementation of ELIGERE, a distributed software platform designed to rank a discrete set of alternatives with respect to multiple evaluation criteria. ELIGERE distributed architecture provides several features of interest in group decision making: a web−based interface where experts express their opinion, a remote computational module implementing a multiple criteria decision making method (fuzzy AHP) for ranking the alternatives, a database for collecting both the answer of the experts and the results of the calculations from the computational module. The motivation behind this work is to speed up the concept selection in product design. An illustrative example, the concept selection of a sensored platform for mobile robots, shows how the distributed architecture of ELIGERE results in an enhancement of the concept selection, in terms of both time and experts' interactive experience.**

Keywords*—distributed information systems, web services, group decision making, fuzzy AHP, product design and development, robotics

I. Introduction

Group decision making (GDM) problems consist in finding the best alternative from a set of feasible alternatives $A = \{A_1, A_2, \ldots, A_m\}$ according to preferences provided by a group of experts $E = \{E_1, E_2, \ldots, E_r\}$ [1]. When consisting of different evaluation criteria $C = \{C_1, C_2, \ldots, C_n\}$, these problems go under the name of Multiple Criteria Decision Making (MCDM) [2]. Many complex decision making situations arise from the lifecycle of every system or process, as in the product design and development cycle [3]. Within this process (see Fig. 1), the phase of *concept selection*, consisting in finding the optimal design solution among a set of alternatives and according to the initial specifications, can be treated as a MCDM problem. The advantage of a collaborative approach using a GDM method in design is the enhancement of the overall process, which leads to the development of successful products [4]. Among the MCDM methods the Analytic Hierarchy Process (AHP), introduced by Saaty in [5], is an extremely elegant approach for addressing and analyzing discrete alternative problems with multiple conflicting criteria. It decomposes a complex problem into a hierarchic structure of objectives, criteria, sub-criteria and alternatives and provides a scale of relative importance to represent expert verbal judgment in the form of a pairwise comparison. Since it is difficult to assign a *crisp* value to a comparison judgment, Van Laarhoven and Pedrycz enhanced AHP with *fuzzy* judgments, resulting in the fuzzy AHP method [6]. The inconvenient of GDM in collaborative design resides in the difficulty to physically implement these methods in industries and research centers: such implementation can be possible only by means of IT infrastructures simple to use and manage. Distributed information systems may constitute the framework for accomodating group decision making methods in the design context.

In the past few years distributed information systems (DIS) have been widely employed to enhance industrial processes, with applications focused on: product data management [7]; product lifecycle management [8]; agile manufacturing [9]; exchanging informations for integrated product and process design [10]; integrating product design and process planning within the same environment [11]. Few previous works are related to the application of DIS for group decision making to enhance collaborative and more productive design. A review paper about group decision support systems is presented by Kraemer *et. al* in [12]. Chen in *et. al* [13] presented a distributed engineering knowledge management approach for the practice of collaborative product design. A comprehensive work made by Bellotti and Bly in [14] discusses the importance of collaboration in a product design team. Since mobility may be a key issue, a further discussion about the potentiality of web services in providing support for remote collaboration is presented.

The main purpose of this work is to show how distributed and web services systems may enhance group decision making in different fields of application, such as product design, even remotely. It will be shown through the distributed architecture of ELIGERE [1] that product design can be enhanced using a collaborative approach in a simple and straightforward manner. The main result of using DIS for collaborative design is the reduction in time for the concept selection phase, thus reducing the *time−to−market* in releasing new products. Moreover, since fuzzy AHP has been used successfully in many different scenarios (economics and finance [15], logistic and management [16], engineering [17]), ELIGERE may enhance the selection process in any of these fields.

[1]https://github.com/eligere/

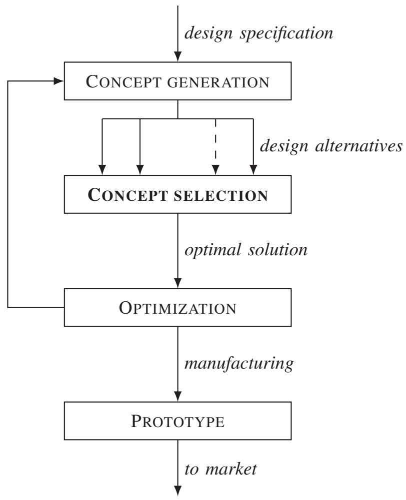

Fig. 1: Product design and development phasis [3].

The rest of this paper is organized as follows. In the second part of this section we provide the related work in distributed information systems architecture suitable for decision support systems. In Section II we present an overview of ELIGERE, whose implementation is explained into details in Section III. Section IV is related to the results that the distributed nature of ELIGERE brings to collaborative design in concept selection. Section V concludes the paper and discusses possible future developments.

A. Related Work

Literature about distributed information systems architecture and paradigm is widely extensive. Here we report only the related works which have led to the development of ELIGERE as a DIS. The central idea in developing distributed applications consists in dividing them in different application modules which belong to various architectural levels and exchange informations by means of messages. The idea of software modules which mediate between applications and database is well discussed in [18]. Papazoglou [19] introduced the concept of service−oriented computing (SOC), the computing paradigm that utilizes services as fundamental elements for developing applications. SOC relies on the service oriented architecture (SOA), discussed in [20]. Guessoum [21] analyzed the modeling procedure of developing distributed systems as a set of organized software agents that interact in a common environment. The importance of designing human−centered distributed systems is discussed in the work of Zhang *et. al* [22]. All those ideas are embodied in Model−View−Controller (MVC) architecture, well explained in [23]. The use of MVC pattern in distributed applications has been proven in different works ([24], [25], [26]). A comprehensive review on web−server systems can be found in [27].

ELIGERE has been built according to MVC architecture. Following the open−source paradigma in software development, it implements an Apache HTTP Server [28] and a MySQL relational database [29]. The strength of these choices is evident in the resulting simplicity of the distributed system, as well as in its modularity and reliability.

II. ELIGERE OVERVIEW

ELIGERE is a distributed, interactive, multi−language, open−source software platform for group decision making in engineering design. It implements fuzzy AHP, a method for decision making problems which allows to rank a discrete set of alternatives with respect to multiple evaluation criteria. Fuzzy AHP is based on the analytical assessment of questionnaires administrated to a panel of experts: thanks to fuzzy set theory their linguistic judgments are translated into quantitative results managed by a proper algorithm to rank first the criteria (preference section) and after the alternatives (suitability section). ELIGERE allows: (1) to show the different alternatives via a web−interface; (2) to collect and store data of the questionnaire results from expert in a database; (3) to instantly rank criteria and alternatives, and thus compute the best alternative with respect to the selected criteria.

ELIGERE workflow is depicted in Fig. 2 and briefly explained as follows.

1 The person who generates the questionnaire (in the following referred as *admin*), by means of an automatic form, inserts criteria and alternatives. Moreover, he can upload images, videos, or 3D files for each alternative. After, the generated questionnaire is available on the server and is ready to be sent to expert for filling. Each questionnaire has its own name and password.

2 The admin sends the questionnaire link to experts.

3 Each expert fills the questionnaire on his own browser. When he submits it, his answers are uploaded into the database.

4 The computational module queries the database every 1 hour and when the data from one questionnaire (answers from all experts) are available, it processes them accordingly to the implemented fuzzy AHP. After the calculations, the final results (the weight of the criteria, the weight of the alternatives with respect to each criterion, the final best alternative) are once again uploaded on the database and available to be consult.

This conceptual workflow has been translated into the architecture depicted in Fig. 3, following a previous work of Selfa *et. al* [30]. Since the architecture has been developed using MVC pattern, ELIGERE articulates in presentation, business and persistence layer.

a) Presentation layer: its responsibility is to provide a Graphical User Interface (GUI) to enable interaction with end users. This layer interact directly with business layer but not with persistence layer.

b) Business layer: it contains the main logic of the software (the *fuzzy AHP engine*). It also moves and processes data between the two other layers.

c) Persistence layer: its responsibility is to persist business data in permanent storage, as relational databases.

The developed architecture results in the following main components of ELIGERE:

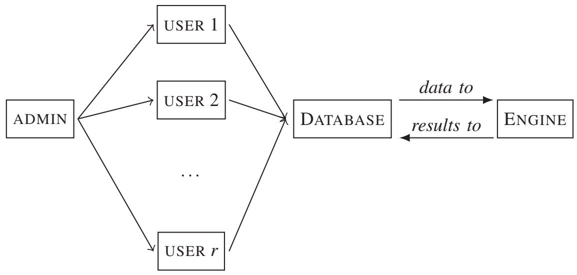

Fig. 2: ELIGERE workflow

.

1) Fuzzy AHP engine: the main component of ELIGERE, all others applications are build around it. Developed in C++ programming language, the engine represents the core of the business layer. It provides the answers' processing of the questionnaires provided to experts. Data input for starting the calculations are provided by a database, with whom the engine is connected thought a mysqli function. Final results about the best alternative is provided and make available in the database. Fuzzy AHP Engine and Engine GUI are built respectively upon Eigen [2] and Q_t [3] libraries.

2) Relational Database: as every database system, it collects, stores and make available data. ELIGERE relational database system has the following tasks:

- provide an history of past surveys;
- collect data from the r questionnaires (one for each expert) related to the same survey;
- provide the input data to the *fuzzy AHP engine*;
- store the output data from *fuzzy AHP engine*.

3) Dynamic web application: ELIGERE web application, based on client−server paradigma and developed with PHP and HTML, is responsible to:

- generate the survey with an automatic form. Each survey is made up of a questionnaire section, criteria section and alternative section;
- administrate the questionnaires to users;
- provide a tool for compiling and submitting the questionnaires from users.

III. Implementation

In this section we describe the implementation of the distributed architecture of ELIGERE presented in the previous section. Fig. 4 shows the four main nodes of ELIGERE: *web clients*, *web server*, *SQL server*, *fuzzy AHP engine*, who are described in detail in the following. They exchange messages through the network via TCP/IP and HTTP protocol.

[2]Eigen is a C++ template library for linear algebra: matrices, vectors, numerical solvers, and related algorithms (eigen.tuxfamily.org)

[3]Qt is a cross−platform library for developing user interfaces. Qt open source is available from http://www.qt.io/download-open-source/

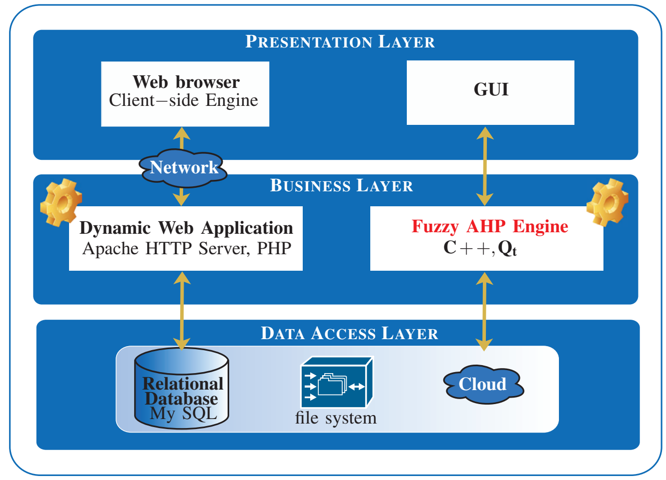

Fig. 3: ELIGERE MVC architecture

.

A. Web clients

Web clients exchange informations with the web server via HTTP Protocol. Clients request and visualize the web−pages of ELIGERE using classical web browser applications. The communication is started by a client program which communicates with a server program for: (1) retrieving a specific survey; (2) creating a new survey if the user is an admin.

B. Web server

Web server is represented by an Apache HTTP Server [28], which enables ELIGERE to be able to:

1) process HTTP request from the users;
2) dynamically generate the surveys through the PHP (hypertext pre−processor) interpreter.
3) retrieve data from database and save them on database by means of mysqli php function.

The PHP functions involved in this node can be gathered in: administrator php functions (APF), client php functions (CPF) and utility php functions (UPF).

APF functions:

- `mainAdmin.php`: it implements a user−friendly panel for the admin of ELIGERE. The admin can create new surveys, see the data processed and the results for a specific survey;
- `insertQuestionnarie.php`: it is used to create a new survey and to show the list of surveys associated to an specific admin.
- `insertAlternativeAndCriteria.php`: it is used by the admin to insert the discrete alternatives and criteria by means of an automatic form.
- `results.php`: it is used to show the elaborated information for a specific survey by Fuzzy AHP engine;
- `showDataBase.php`: it is used to show the data for a specific survey that must be elaborated by Fuzzy AHP engine.

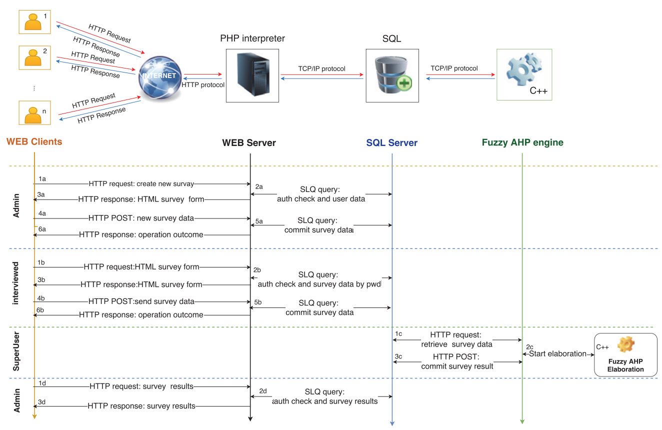

Fig. 4: ELIGERE network setup: the diagram displays a typical data flow events between client and remote hosts (server, database and fuzzy engine) involved.

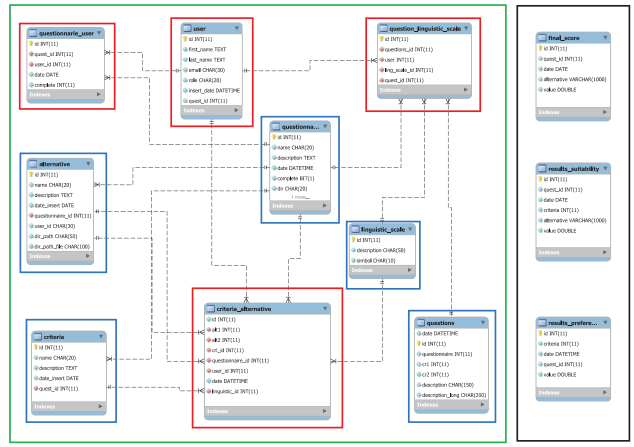

Fig. 5: ELIGERE entity−relationship diagram. Green and black boxes show respectively the relationships between the tables managed in writing/reading mode by the web application and by the fuzzy engine nodes. The tables inside the blue and the red boxes are managed respectively by the admin and the users.

CPF functions:

- `main.php`: it presents a user-friendly panel for managing all surveys associated to a specific user.
- `getPreferences.php`: it is designated to ask the expert to express a preference about the given evaluation criteria. Each question in this section asks the expert to compare two criteria at time (*i.e.* pairwise comparison);
- `getSuitability.php`: it is designated to compare the candidate alternatives with respect to each criterion separately.
- `success.php`: it is invoked once that getSuitability.php was completed, if and only if all survey data were processed without errors.

UPF functions:

- `getDataFromDataBase.php`: it is used to retrieve data from database by different *get* functions that return php object. In ELIGERE, a php object is an instance of a php class representing a specific entity (as user, criterion or alternative);
- `connection.php`: it returns, by means of a mysqli PHP function, an object which represents the connection to a MySQL Server. The connection class object is used by getDataFromDataBase.php.

Once that all the users have successfully completed the survey, the "complete" filed in the questionnaire table is set to one. At this point the data can be elaborated by the fuzzy ahp engine node as described later on.

C. MySQL server

ELIGERE database uses MySQL[4] server for its simplicity, security and interoperability. Fig. 5 shows the schema of the ELIGERE database, composed by two groups of tables. Green box groups the tables accessible in writing/reading mode by the web application nodes (web clients and web server) and only in reading mode by fuzzy engine node. Black box groups the tables accessible in writing/reading mode by the fuzzy engine node and only in reading mode from the web application nodes. This logic was adopted for improving the security of the data, allowing the two groups to be deployed on a different host machine with different security privileges. Moreover, the tables inside the green box could be divided again in admin managed (AM) tables inside the blue boxes, and in users managed (UM) tables inside the red boxes.

AM tables:

- `linguistic_scale`: since fuzzy AHP transforms linguistic judgment into fuzzy numbers, this table contains a static map of *Linguistic scale for importance* and *Simbol*. This table is in one-to-many relationship with criteria_alternative table and in many-to-many relationship with questions;
- `questionnaire`: this table stores the information about the questionnaire This table is related by one-to-many relationship with criteria_alternative, question_linguistic_scale, alternative and criteria tables and with user table by a many-to-many relationship;
- `alternative`: this table contains the information about the discrete alternatives provided by the admin of the survey by the insertAlternativeAndCriteria.php function;
- `criteria`: this table contains the information about the criteria, inserted by the admin through the insertAlternativeAndCriteria.php function;
- `questions`: this table contains the questions that will be used to ask the expert to express a preference about a given evaluation criterion in getPreferences.php script;

[4]MySQL is an open-source relational database management system (RDBMS) available on https://www.mysql.com/

UM tables:

- `user`: this table stores the user-data informations. This table is related by one-to-many relationship with questionnaire_user, criteria_alternative and question_linguistic_scale tables;
- `question_linguistic_scale`: this table is a joint table between linguistic_scale and question and contains the preferences about a given evaluation criterion. It is related to the questionnaire and user table through a one−to−many relationship;
- `criteria_alternative`: the core table of the whole database. It stores the user data which are processed through getSuitability.php function, *i.e.* the comparison of two alternatives at time under a specific criterion.
- `questionnaire_user`: it is a joint table between user and questionnaire tables. This table carry out the role of a controller through the *complete* filed: if all data from the *i-th* user are correct, then the *complete* filed is set to 1 and the data could be elaborated, otherwise not.

The second group (black box), managed by fuzzy AHP application, contains:

- `results_preferences`: this table stores the results of the preference section;
- `results_suitability`: this table contains the results of the suitability section;
- `final_score`: this table stores the final result about the best alternative for a specific survey;

D. Fuzzy AHP engine

The main node implements fuzzy AHP. It is responsible of ranking the alternatives with respect to the criteria: the computation procedure starts when all data of a survey are stored in the database. The connection between the engine and the SQL server database is made possible by a `QSqlQueryModel` function which creates and opens the connection with the database within the engine. Once that the connection is established, the fuzzy AHP engine can retrieve all the data from database, through TCP/IP protocol, for elaborating the computation. After the computation, the results (ranking of criteria, ranking of alternatives with respect to each criterion and final ranking of alternatives) are stored again into the database. For the fuzzy AHP engine source code, refers to `htttp://github.com/eligere/fuzzyAHPcore/`.

IV. Results

ELIGERE has been tested with an heterogeneous team composed by 7 experts in a concept selection review session where six design alternatives (Fig. 6) have been ranked with respect to three evaluation criteria (TABLE I). The case study is the optimum selection of a sensored platform used in mobile robots for localization tasks. Results of the review sessions show that Alt. 1 has been considered the optimal design solution (TABLE IV). TABLE II is the result of the ranking for criteria, while TABLE III the ranking of alternatives with respect to each criterion.

Criteria	
C1	Simplicity (mechanical and of operations)
C2	Aesthetic design
C3	Integrability with sensors and electronics

TABLE I: Case study: criteria

C_1	C_2	C_3
0.37	0.24	0.39

TABLE II: Ranking of criteria

.

	C_1	C_2	C_3
A_1	0.33	0.20	0.21
A_2	0.26	0.26	0.19
A_3	0.07	0.12	0.15
A_4	0.14	0.25	0.19
A_5	0.13	0.07	0.14
A_6	0.07	0.10	0.12

TABLE III: Ranking of alternatives with respect to each criterion

.

A_1	A_2	A_3	A_4	A_5	A_6
0.25	0.23	0.11	0.19	0.12	0.10

TABLE IV: Final alternative ranking

.

ELIGERE has been implemented on a standard machine with Intel® Core™ i7 CPU 2.67 GHz and 8 Gb RAM, running under Windows 7 and Ubuntu 14, 64 bits. Since only simple matrix algotithm are involved for the fuzzy AHP procedure, once that data are available on the database, the computation of the optimal solution requires less than 1s (phase 4 of the workflow explained in Sec. II). The other phases are operator−dipendent, and ELIGERE acts as software tool for the *admin* in setting up, managing and saving the interactive design session.

V. Conclusions

In this paper we emphasized the importance of distributed information systems for group decision making in product design. We testified it by means of ELIGERE, a software platform developed by the authors. The distributed nature of ELIGERE allows the following advantages for group decision making problems: (i) experts can partecipate to the concept selection review session even remotely (the questionnaire is filled out online); (ii) nobody has to take care about collect the answers of experts (the database takes care of it); (iii) nobody has to insert the collected data into the fuzzy AHP engine for elaborating the results (fuzzy AHP engine is linked directly with the database). The results and the organized

(a) Alt. 1 (b) Alt. 2 (c) Alt. 3 (d) Alt. 4 (e) Alt. 5 (f) Alt. 6

Fig. 6: Case study: alternatives

design review session for validating ELIGERE demostrated that this software platform allows instant computation of the optimal design solution, reducing significantly the concept selection phase in product design and development, and a more interactive experience for the experts as well. The simplicity of ELIGERE architecture, as well as its reliability, may enhance group decision problems in any application field (engineering, finance, economics, social science, ...).

The authors wanted to release ELIGERE under GNU GPL licence, since they believe that open−source could represent a way to widespread this software in academia and research centers, who hopefully can contribute for the future in integrating new functionalities.

ACKNOWLEDGMENT

The authors would like to thank the colleagues of the Fraunhofer Institute for Computer Graphics and the Department of Industrial Engineering, University of Naples Federico II, for their support in testing and validating ELIGERE.

REFERENCES

[1] F. Herrera, E. Herrera-Viedma, and J. verdegay, "A model of consensus in group decision making under linguistic assessments," *Fuzzy Sets and Systems*, vol. 78, no. 1, pp. 73 – 87, 1996.

[2] R. E. Steuer and P. Na, "Multiple criteria decision making combined with finance: A categorized bibliographic study," *European Journal of operational research*, vol. 150, no. 3, pp. 496–515, 2003.

[3] K. T. Ulrich, *Product design and development*. Tata McGraw-Hill Education, 2003.

[4] C. Renzi, F. Leali, M. Pellicciari, A. Andrisano, and G. Berselli, "Selecting alternatives in the conceptual design phase: an application of fuzzy-ahp and pugh's controlled convergence," *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 9, no. 1, pp. 1–17, 2015.

[5] T. L. Saaty, "A scaling method for priorities in hierarchical structures," *Journal of Mathematical Psychology*, vol. 15, no. 3, pp. 234 – 281, 1977.

[6] P. van Laarhoven and W. Pedrycz, "A fuzzy extension of saaty's priority theory," *Fuzzy Sets and Systems*, vol. 11, no. 1, pp. 199 – 227, 1983.

[7] Y. Kim, S.-H. Kang, S.-H. Lee, and S. B. Yoo, "A distributed, open, intelligent product data management system," *International Journal of Computer Integrated Manufacturing*, vol. 14, no. 2, pp. 224–235, 2001.

[8] R. Sudarsan, S. Fenves, R. Sriram, and F. Wang, "A product information modeling framework for product lifecycle management," *Computer-Aided Design*, vol. 37, no. 13, pp. 1399 – 1411, 2005.

[9] L. Zhou and R. Nagi, "Design of distributed information systems for agile manufacturing virtual enterprises using corba and step standards," *Journal of manufacturing systems*, vol. 21, no. 1, p. 14, 2002.

[10] F. Mervyn, A. S. Kumar, S. Bok, and A. Nee, "Developing distributed applications for integrated product and process design," *Computer-Aided Design*, vol. 36, no. 8, pp. 679 – 689, 2004.

[11] J. Sun, Y. Zhang, and A. Nee, "A distributed multi-agent environment for product design and manufacturing planning," *International Journal of Production Research*, vol. 39, no. 4, pp. 625–645, 2001.

[12] K. L. Kraemer and J. L. King, "Computer-based systems for cooperative work and group decision making," *ACM Comput. Surv.*, vol. 20, no. 2, pp. 115–146, Jul. 1988.

[13] Y.-J. Chen, Y.-M. Chen, and H.-C. Chu, "Enabling collaborative product design through distributed engineering knowledge management," *Computers in Industry*, vol. 59, no. 4, pp. 395 – 409, 2008.

[14] V. Bellotti and S. Bly, "Walking away from the desktop computer: Distributed collaboration and mobility in a product design team," in *Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work*. New York, NY, USA: ACM, 1996, pp. 209–218.

[15] T. T. Nguyen and L. Gordon-Brown, "Constrained fuzzy hierarchical analysis for portfolio selection under higher moments," *Fuzzy Systems, IEEE Transactions on*, vol. 20, no. 4, pp. 666–682, 2012.

[16] C. Kahraman, U. Cebeci, and Z. Ulukan, "Multi-criteria supplier selection using fuzzy ahp," *Logistics Information Management*, vol. 16, no. 6, pp. 382–394, 2003.

[17] P. Dehghanian, M. Fotuhi-Firuzabad, S. Bagheri-Shouraki, and A. A. R. Kazemi, "Critical component identification in reliability centered asset management of power distribution systems via fuzzy ahp," *IEEE Systems Journal*, vol. 6, no. 4, pp. 593–602, Dec 2012.

[18] G. Wiederhold, "Mediators in the architecture of future information systems," *Computer*, vol. 25, no. 3, pp. 38–49, March 1992.

[19] M. P. Papazoglou, "Service-oriented computing: Concepts, characteristics and directions," in *Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on*. IEEE, 2003, pp. 3–12.

[20] T. Erl, *Service-oriented architecture: a field guide to integrating XML and web services*. Prentice Hall PTR, 2004.

[21] Z. Guessoum, "Adaptive agents and multiagent systems," *IEEE Distributed Systems Online*, vol. 5, no. 7, 2004.

[22] J. Zhang, V. L. Patel, K. A. Johnson, and J. W. Smith, "Designing human-centered distributed information systems," *IEEE Intelligent Systems*, vol. 17, no. 5, pp. 42–47, Sep 2002.

[23] G. E. Krasner, S. T. Pope *et al.*, "A description of the model-view-controller user interface paradigm in the smalltalk-80 system," *Journal of object oriented programming*, vol. 1, no. 3, pp. 26–49, 1988.

[24] D. M. Selfa, M. Carrillo, and M. D. R. Boone, "A database and web application based on mvc architecture," in *16th International Conference on Electronics, Communications and Computers (CONIELECOMP'06)*. IEEE, 2006, pp. 48–48.

[25] A. Leff and J. T. Rayfield, "Web-application development using the model/view/controller design pattern," in *Enterprise Distributed Object Computing Conference, 2001. EDOC'01. Proceedings. Fifth IEEE International*. IEEE, 2001, pp. 118–127.

[26] H. J. La and S. D. Kim, "Balanced mvc architecture for developing service-based mobile applications," in *e-Business Engineering (ICEBE), 2010 IEEE 7th International Conference on*, Nov 2010, pp. 292–299.

[27] V. Cardellini, E. Casalicchio, M. Colajanni, and P. S. Yu, "The state of the art in locally distributed web-server systems," *ACM Computing Surveys (CSUR)*, vol. 34, no. 2, pp. 263–311, 2002.

[28] R. T. Fielding and G. Kaiser, "The apache http server project," *IEEE Internet Computing*, vol. 1, no. 4, pp. 88–90, Jul 1997.

[29] H. E. Williams and D. Lane, *Web database applications with PHP and MySQL*. " O'Reilly Media, Inc.", 2004.

[30] D. M. Selfa, M. Carrillo, and M. D. R. Boone, "A database and web application based on mvc architecture," in *16th International Conference on Electronics, Communications and Computers (CONIELECOMP'06)*, Feb 2006, pp. 48–48.