Multiple Criteria Decision Support in Requirements Negotiation

Siamak Farshidi[1], Slinger Jansen[1], Rolf de Jong[2], and Sjaak Brinkkemper[1]

[1] Department of Information and Computing Sciences, Utrecht University
{s.farshidi, slinger.jansen, s.brinkkemper}@uu.nl
[2] AFAS Software
r.dejong@afas.nl

Abstract. Software producing organizations regularly make complex technology decisions as part of the software production process, even when there is low availability of expertise in the domain within the organization. Software production is consequently a suitable domain to deploy decision support systems that intelligently assist decision-makers in selecting desirable technologies according to their requirements and priorities. In this tool paper, we introduce a decision support system that supports decision-makers in group and individual decision-making problems to select the most suitable technologies. Technology experts from different domains, such as database management systems and cloud service providers, confirm that the approach provides more knowledge than they could have collected independently, increases insight into the selection process, and reduces the time and cost of the decision-making process.

Keywords: multi-criteria decision-making; decision support system; technology selection; requirement priorities; group decision-making

1 Introduction

Thanks to rapidly evolving technologies and an abundance of technology vendors with homogeneous offerings in the marketplace, selecting suitable technologies for software producing organizations is a challenging process. Software architects and senior developers, who are not experts in each decision domain, need trusted partners who can effectively understand their requirements and priorities as well as have deep technical expertise in software technologies in the market. Technology selection is the process of assessing the potential value of technologies and their contribution to the competitiveness and profitability of Software Producing Organizations. The selection process is complicated because there are many factors, such as suitability and cost, that have to be considered. The technology

---

selection process can be modeled as a multi-criteria decision-making (MCDM) problem that deals with the assessing of a set of alternatives, and considering a set of decision criteria.

This study introduces a Decision Support System Tool (DSST) that helps decision-makers with MCDM problems, such as Database Management System and Cloud Service Provider (CSP) selection problems.

2   Related Work

In recent years researchers introduced a considerable variety of techniques, methods, and tools to solve different technology selection problems for Software Producing Organizations. Many variations exist, but all share the crucial phases of the decision-making process. The majority of the techniques in the literature use pairwise comparison as the main method to assess the weight of criteria. The pairwise comparison is a time-consuming process, and gets more complicated as the number of criteria increases. Some of the methods, such as *Analytic Hierarchy Process*, are not scalable, so in the case of modifying the list of alternatives or criteria, the whole process of evaluation should be conducted repeatedly. Therefore, these methods are costly and applicable for only a small number of criteria and alternatives. The majority of the MCMD techniques in literature define domain-specific quality attributes to evaluate the alternatives. Such studies are mainly appropriate for specific case studies. Furthermore, the results of these MCDM approaches are valid for a specified period, so by technology advances and new service offering they will be out-of-date.

The Commercial-Off-The-Shelf (COTS) selection problem is a subclass of MCDM problems. Numerous approaches have been proposed for the general problem of software component evaluation and selection. Becker et al. [1] present a multi-criteria decision support system (MCDSS) for the COTS selection problem. The MCDSS evaluates a total of 51 COTS components against a total of 631 decision criteria. The authors specified metrics, such as the *key decision factors* and *efficient criteria sets*, for the quantitative evaluation of decision criteria and sets of criteria, and illustrated their application to a set of real-world decision cases. The DSST and MCDSS both provide a substantial number of criteria to support decision-makers in the technology selection problem. Furthermore, they use ISO SQUARE as a standard set of quality attributes. The main difference between the DSST and MCDSS is their weighting methods. The DSST and MCDSS both provide a substantial number set of criteria to support decision-makers. Furthermore, they use the ISO SQUARE as a standard set of quality attributes.

The DSST is unique in that it utilizes standard concepts from software production: the MoSCoW prioritization technique (*MoSCoW*) [2] to assess criteria weights and reduce uncertainty, and ISO/IEC quality aspects to indicate the relationship among criteria according to domain experts' knowledge. The DSST can be used over the full life-cycle and can co-evolve its advice based on evolving

requirements. Moreover, the DSST empowers Software Producing Organizations with its consensus platform for group decision making.

3   Decision Support System

The fundamental components of a typical DSS [3] are the DataBase management system, the Model-Base management system, and the Dialog Generation management system.

The *DataBase management system* is a set of domain features facts related to the MCDM problem. Furthermore, it contains previous solutions, best practices, and lesson learned from prior made decisions. The *Model-Base management system* is a collection of rules, heuristics, and knowledge related to the MCDM problem. The *Dialog Generation management system* is a user interface which is used to acquire decision-makers' requirements and preferences.

The Inference Engine (IE) of the decision support system infers solutions. The IE receives domain feature requirements and their priorities according to *MoSCoW* from the *Dialog Generation management system* as its input. Next, it finds the most relevant rules from a collection of models in the *Model-Base management system*. Then, the IE by using facts from the *DataBase management system* deduces decisions. Eventually, it sends ranked feasible solutions to the *Dialog Generation management system*. The IE is a protocol for navigating through the rules and facts in a DSS to solve the problem. The IE infers solutions of MCDM problems based on the interaction of domain features facts and rules. In a standard DSS, the IE does not rely on knowledge base facts and rules, so it works independently from the other components.

The DSST[3] comprises all of the fundamental components of a standard DSS. Figure 1 shows the fundamental components of the DSST and their relationships.
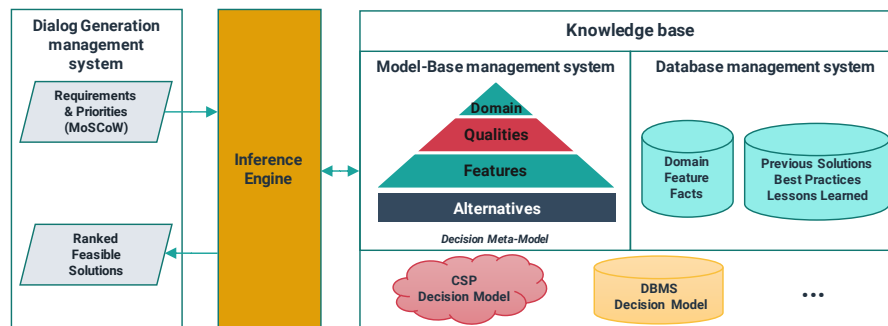

Fig. 1. The fundamental components of the DSST.

---

[3] We have invested roughly 960 hours into the implementation of the online Decision Model Studio (http://dss.amuse-project.org) to build decision models for MCDM problems in Software Producing Organizations.

3.1 Research Method

The DSST is a culmination of several research projects combined. The DSST is central in four research projects with the DSST: one concerning DBMSs, one concerning CSPs, one concerning blockchain platforms, and one concerning Software Architecture decisions. For each of the projects, several experts were approached to evaluate the quality model, the COTS features, and the effectiveness of the tool. In total, twenty-three experts (three DSS experts, two academics, five Software Developers, six cloud consultants, three cloud architects, and four software architects) participated in this research to evaluate the effectiveness and efficiency of the DSST. We selected the experts according to their expertise and experience that they mentioned in their professional profile.

3.2 Decision Model

A decision model in the knowledge base of the DSST contains all facts and rules of an MCDM problem. In other words, a decision model defines a decision graph to solve a specific MCDM problem. The Decision Meta-Model is the base structure of a decision model in the knowledge base. It includes two primary sets (*Qualities* and *Features*). The set *Qualities* is a set that keeps software quality attributes, and the set *Features* is a set that retains domain features of an MCDM problem.

The DSST uses the *ISO/IEC 25010* standard [4] and *extended ISO/IEC 9126* standard [5], which are domain-independent software quality models and provide reference points by defining a top-down standard quality model for software systems, to define the set *Qualities*. Moreover, domain features of an MCDM problem are specified based on the domain experts' knowledge. For instance, *Kubernetes*, *Docker Swarm*, and *Ansible* could be considered as domain features of the CSP selection problem. The mapping between the standard quality models and domain features are defined based on domain experts' opinions, where $Qualities \times Features \rightarrow Boolean$. Figure 2 illustrates part of the mapping between quality models and domain features in CSP selection problem.

	Accountability	Adaptability	Analyzability	Co-existence	Confidentiality	
.Net framework	☐	☑	☐	☑	☐	
API Manager	☐	☑	☐	☑	☐	
Application Performance Monitoring	☐	☐	☑	☐	☑	

Fig. 2. A part of the mapping between the ISO/IEC 25010 and CSP features.

Technology alternatives of an MCDM problem and the mapping between them and domain features are determined based on documentation of alternatives, literature studies, social networks, alternative experts, etc. As mentioned earlier, the DSST utilizes *MoSCoW* to define decision-makers' domain feature requirements and assess the importance of required domain features. Suppose $W_{MoSCoW} = \{w_{Must}, w_{Should}, w_{Could}, w_{Won't}\}$ is the set of priority weights according to the definition of the *MoSCoW* [2]. Domain feature requirements with *Must Have* or *Won't Have* priorities act as hard constraints and domain feature requirements with *Should Have* and *Could Have* priorities act as soft constraints. Figure 3 shows part of the domain feature requirements and priorities for a Software Producing Organization[4].

Fig. 3. Top-10 solutions and part of the domain feature requirements of AFAS CSP selection problem.

Each decision model defines a decision structure for an MCDM problem systematically. The Knowledge Base is a collection of decision models, which are groups of rules and facts. The $InferenceEngine$ ranks the alternatives based on their calculated scores. The score calculation process begins with computing the weight of each aspect of the decision model. The score calculation process is based on the well-known *Weighted Sum Model*. The scores of feasible solutions are more than zero. Moreover, by sorting the feasible solutions in descending order of their scores, the final ranked feasible solutions will be given as the result of the DSST. Figure 3 and Figure 4 demonstrate a part of the decision graph and top-10 solutions for the Software Producing Organization respectively.

---

[4] AFAS Software is an ERP vendor in the Netherlands with approximately 350 employees. One of AFAS' current challenges is validating whether they have chosen the right CSP for the new version of their product.
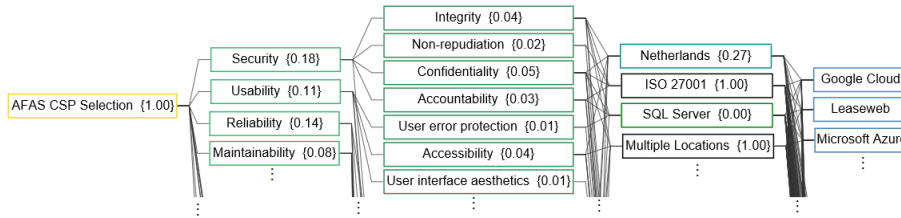
Fig. 4. A part of the Decision Graph for AFAS CSP selection problem.

3.3 Group Decision Making

The DSST provides a discussion and negotiation platform to enable Software Producing Organizations to make group decisions. It detects and highlights the conflicts in the assigned priorities to the domain feature requirements by decision-makers, asks them to resolve disagreements. The DSST asks decision-makers to define individual domain feature requirements based on *MoSCoW*. Next, it collects the individual prioritized domain feature requirements of decision-makers and considers the maximum *MoSCoW* priority for each feature requirement. For example, if three decision-makers assign two *Should Have* and one *Must Have* priority to a domain feature requirement, the collective priority becomes *Must Have*. Moreover, The DSST detects inconsistent values, such as two *Must Have* and one *Could have*, and asks decision-makers to revise the considered priories. Figure 5 illustrates an example of the group decision-making process and how the tool supports it.

Fig. 5. An example of group decision-making. Note, the DM columns show the prioritized domain feature requirements by decision-makers based on *MoSCoW*. Moreover, the collective decision column indicates the maximum priority for each domain feature requirement.

4 Discussion

Figure 3 illustrates top-10 feasible technology solutions for a case participant (AFAS). The DSST recommended the same solutions as the case participant experts suggested to the organization after extensive analysis and discussions. However, the DSST offers a short ranked list of feasible solutions; therefore Software Producing Organizations should perform further investigations, such as performance testing and actual TCO calculation, to find the optimum CSPs for their software products.

The technology experts confirm that the DSST contains the main components of a standard DSS. Furthermore, they find the DSST useful and provides more knowledge than they could have collected independently. The experts believe that experience in using a technology provides invaluable knowledge when selecting suitable technology. We, therefore, recommend that the DSST be used in combination with benchmarks where applicable.

5 Conclusion

This tool paper introduces a decision support system (DSST) to accelerate the process of technology selection. The DSST empowers Software Producing Organizations with its discussion platform to perform group decision making and reaching a collective decision. The DSST utilizes the MoSCoW prioritization technique (*MoSCoW*) to assess criteria weights and reduce uncertainty, and ISO/IEC quality aspects to indicate the relationship among criteria according to domain experts' knowledge. The experts asserted that the approach increases insight into the selection process and that it reduces the time and cost of the decision-making process. We plan to create a community around the DSST that regularly will update the curated knowledge base with new technology alternatives and features. Additionally, we are looking at methods to automatically extract domain features from manuals and documentation, using text mining techniques.

References

1. C. Becker, M. Kraxner, M. Plangg, and A. Rauber, "Improving decision support for software component selection through systematic cross-referencing and analysis of multiple decision criteria," in *System Sciences (HICSS), 2013 46th Hawaii International Conference on.* IEEE, 2013, pp. 1193–1202.
2. Atern, "The handbook," *DSDM Consortium*, 2008.
3. A. Sage, *Decision support systems engineering*, ser. Wiley series in systems engineering. J. Wiley, 1991.
4. ISO, "Iec25010: 2011 systems and software engineering–systems and software quality requirements and evaluation (square)–system and software quality models," *International Organization for Standardization*, vol. 34, p. 2910, 2011.
5. J. P. Carvallo and X. Franch, "Extending the iso/iec 9126-1 quality model with non-technical factors for cots components selection," in *Proceedings of the 2006 international workshop on Software quality.* ACM, 2006, pp. 9–14.