

A decision support system for software technology selection

Siamak Farshidi, Slinger Jansen, Rolf de Jong & Sjaak Brinkkemper

To cite this article: Siamak Farshidi, Slinger Jansen, Rolf de Jong & Sjaak Brinkkemper (2018) A decision support system for software technology selection, *Journal of Decision Systems*, 27:sup1, 98-110, DOI: [10.1080/12460125.2018.1464821](https://doi.org/10.1080/12460125.2018.1464821)

To link to this article: <https://doi.org/10.1080/12460125.2018.1464821>



© 2018 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group



Published online: 08 May 2018.



Submit your article to this journal [↗](#)



Article views: 1652



View Crossmark data [↗](#)

A decision support system for software technology selection*

Siamak Farshidi^a, Slinger Jansen^a, Rolf de Jong^b and Sjaak Brinkkemper^a

^aDepartment of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands; ^bARTIN Software, Utrecht, The Netherlands

ABSTRACT

Software producing organisations face the challenge of including new technology in their products, such as cloud technologies and database management systems. As software architects and senior developers are not experts in this domain, they need to consult external experts or acquire the knowledge themselves. Software production, therefore, is a suitable domain to deploy decision support systems, that intelligently support these decision-makers in selecting the desirable technology for their product. We present a decision support system that supports decision-makers in choosing the most suitable database technology. The case studies and experts confirm that the approach increases insight into the selection process, provides a richer prioritised option list than if they had done their research independently, besides reduces the time and cost of the decision-making process.

ARTICLE HISTORY

Received 15 March 2018

Accepted 4 April 2018

KEYWORDS

Multi-criteria decision-making; decision support system; technology selection; database management system

1. Introduction

Technology selection is the process of assessing the potential value of technologies and their contribution to the competitiveness and profitability of Software Producing Organisations (SPOs). Moreover, technology selection is one of the most significant processes in evaluating innovation, popularity and suitability of technologies for SPOs. Therefore, technology selection is an essential decision-making process for SPOs. The challenge consists of evaluating and selecting the most suitable technologies for SPOs according to their preferences and requirements. The selection process is complex because too many factors, such as suitability and cost, should be considered. Therefore, the technology selection process can be modelled as a multi-criteria decision-making (MCDM) problem that deals with the evaluation of a set of alternatives, and taking into account a set of decision criteria (Triantaphyllou, Shu, Sanchez, & Ray, 1998).

In recent years, researchers introduced a significant variety of techniques, methods and tools to solve different technology selection problems for SPOs. Many variations exist, but all share the vital phases of the decision-making process. The majority of MCDM approaches

CONTACT Siamak Farshidi  s.farshidi@uu.nl

*This work is a result of the AMUSE project. See amuse-project.org for more information.

use pairwise comparison as the weighting method, which typically is not scalable. Thus, in the case of modifying the list of alternatives or criteria, the whole process of evaluation must be repeated. These methods are costly and only applicable for a small number of criteria and alternatives. Technology selection decisions are often made ad hoc, without reference to reliable models or sound methodologies. Furthermore, the results of technology selection solutions in the literature are valid for a specified period, so by technology advances, they should be performed again. Hence, a reusable, evolvable and expandable decision-making approach is needed to make the right decision based on the characteristics of the environment.

This study introduces a Decision Support System (DSS) to help decision-makers with MCDM problems, such as DBMS selection. The DSS is a tool that can be used over the full life-cycle and can co-evolve its advice based on evolving requirements. The DSS applies the six-step decision-making process (Majumder, 2015) to build maintainable and evolvable decision models for MCDM problems, and makes the knowledge acquisition more reliable and trustful. The sets of criteria and alternatives plus the relationship among them for an MCDM problem can be up-to-date and regularly manipulated without having impacts on the validity of its decision model. The novelty of the DSS lies in utilising the MoSCoW prioritisation technique (*MoSCoW*) (Consortium, 2014) to assess criteria weights and reduce uncertainty, in introducing assessment models to measure the values of non-boolean criteria, and in using ISO/IEC quality aspects to indicate the relationship among criteria according to domain experts' knowledge.

This paper is structured as follows. Section 2 describes the design science method followed and the exploratory theory testing case studies that have been performed. Section 3 gives a window into the literature of software technology selection and the multiple approaches to solving decision-making problems, such as ours. Section 4 formulates the technology selection problem in SPOs and describes the proposed DSS. Then, Section 5 illustrates an application of the DSS to address the DataBase Management System (DBMS) selection problem, using multiple case studies to evaluate and emphasise the significance of the approach. Afterward, Section 6 interprets the results of the case studies according to expert interviews and opinions. Next, Section 7 highlights and overcomes barriers to the knowledge acquisition and decision-making process. Finally, Section 8 summarises the proposed approach and offers directions for future studies.

2. Research method

The problem we are trying to solve is that software-producing organisations typically are not knowledgeable in the domains in which they need to make technology selections for integration into their products. The technology selection process can be modelled as an MCDM problem that deals with structuring, planning, and solving the problem concerning a set of criteria: (1) Identifying the objective, (2) Selection of the features, (3) Selection of the alternatives, (4) Selection of the weighing method, (5) Applying the method of aggregation (6) Decision-making based on the aggregation results.

To support these organisations, we propose a DSS, created using design science, based on the six-step decision-making process. The DSS has the goal of finding suitable alternatives that support a set of domain feature requirements. The traditional design science cycle is followed, and the DSS is inspired by expert knowledge, which is gathered through

Table 1. This table compares selected MCDM methods from literature to address technology selection problems.

Author(s)	Domain	MCDM	PC	QA	#F	#A
This paper	DBMS	DSS	No	ISO/IEC 25010 EX. ISO/IEC 9126	307	73
Jusoh, Chamili, Che Pa, and Yahaya (2014)	DBMS	AHP	Yes	Domain specific	12	3
Brahimi, Bellatreche, and Ouhammou (2016)	DBMS	ML	No	Domain specific	20	3
Garg, Sharma, and Sharma (2017)	DBMS	FMCDM	Yes	Domain specific	14	5
Lin, Hsu, and Sheen (2007)	Data warehouse system	FAHP	Yes	Domain specific	16	6
Onut and Efendigil (2010)	ERP software	FAHP	Yes	ISO/IEC 9126	13	3
Kohli and Sehra (2014)	Software Quality Model	FMCDM	Yes	Domain specific	3	3
Rodriguez et al. (2017)	Risk management approach	FAHP	Yes	Domain specific	5	5
Fu, Shi, Yang, and Yu (2010)	Project management software	FAHP FMCDM	Yes	Domain specific	14	4
Büyüközkan and Güleriyüz (2016)	Product development partner	FAHP FTOPSIS	Yes	Domain specific	16	6
Becker, Kraxner, Plangg, and Rauber (2013)	COTS	DSS	No	ISO/IEC 25010 Domain specific	631	51

Notes: The first column (Domain) points out the problem domain. The second column (MCDM) denotes the MCDM approach. The third column (PC) indicates whether the approach applies pairwise comparison(PC) as a weight calculation method or not. The fourth column (QA) determines the type of quality attributes. The seventh and eighth columns (#F and #A) signify the number of criteria and alternatives that were considered in the problem domain.

three series of interviews. Fourteen experts (three DSS experts, two academics, five Software Developers and four Software Architects) participated in this research to evaluate the DSS in interviews that lasted between 45 and 90 min. The domain experts were pragmatically selected according to their expertise and experience that they mentioned in their professional profile.

Secondly, the efficiency and usefulness of the DSS is evaluated through three exploratory theory-testing case studies. The unit of analysis is a unique technology selection decision in a software product. We performed three such case studies at two SPOs to evaluate the DSS. The case studies typically lasted one day and consisted of (1) defining the domain feature requirements, (2) prioritising them and (3) comparing the DSS feasible solutions with their own solutions.

3. Related work

In recent years, researchers introduced a variety of MCDM methods to address technology selection problems for SPOs. The *Analytic Hierarchy Process(AHP)* is a structured method for organising and analysing MCDM problems. This method has been extensively applied and combined with other techniques to solve MCDM problems. The *Technique for Order Preference by Similarity to Ideal Solution(TOPSIS)* suggests that the selected alternative should have the shortest distance from an ideal solution and the farthest distance from the negative-ideal solution. The *FAHP* and *FTOPSIS* are the combinations of Fuzzy logic with the *AHP* and *TOPSIS* methods. The *Fuzzy MCDM(FMCDM)* assesses the ratings of alternatives vs. criteria and the importance weights of criteria based on semantic values represented by fuzzy numbers. The *Machine Learning(ML)* explores the study and construction of algorithms that can learn from and make predictions on data.

Table 1 illustrates selected MCDM approaches from literature. The majority of the MCDM techniques use pairwise comparison to assess the weight of criteria. For a problem with n

number of criteria, $\frac{n(n-1)}{2}$ comparisons are needed (Saaty, 1990). Pairwise comparison is a time-consuming process and gets more complicated as the number of criteria increases. Some of the methods, such as *AHP* and *FAHP*, are not scalable. For instance, when the list of alternatives or criteria is modified, the whole process of evaluation should be conducted again. These methods are costly and applicable for a small number of criteria and alternatives. The MCDM techniques in literature mainly define domain-specific quality attributes to evaluate alternatives. Such studies are typically appropriate for specific case studies. Furthermore, the results of these MCDM approaches are valid for a specified period, so by technology advances, new updates and releases, they will be out-of-date.

The DBMS selection problem is a subclass of the COTS selection problem, and both problems are a subclass of MCDM problems. Becker et al. present a multi-criteria decision support system (MCDSS) for software component selection. 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 proposed DSS and MCDSS both provide a substantial number of criteria to support decision-makers in the technology selection problem. Furthermore, they use the ISO/IEC 25010 (ISO, 2011) as a standard set of quality attributes. The main difference between our and the MCDSS is their weighting methods. Our DSS utilises the *MoSCoW* to assess the significance of criteria. Moreover, it introduces assessment models to measure the values of non-boolean criteria, such as the cost of alternatives.

4. Multi-criteria decision-making

This study introduces a DSS that applies the six-step decision-making process (Majumder, 2015) to build maintainable and evolvable decision models for MCDM problems, and makes the knowledge acquisition more reliable and trustful. Let *Alternatives* = $\{a_1, a_2, \dots, a_{|Alternatives|}\}$ be a set of alternatives (technologies) in the market. Moreover, *Features* = $\{f_1, f_2, \dots, f_{|Features|}\}$ be a set of domain features, which includes the most prominent technical and non-technical domain features of the alternatives, so each $a \in Alternatives$ supports a subset of the set *Features*. The goal is finding the suitable alternative a which supports a set of required domain features (set *Requirements*), where $Requirements \subseteq Features$. In other words, an alternative a is the suitable one that supports domain feature requirements and satisfies the preferences of the decision-maker. Typically, a unique optimal solution for an MCDM problem does not exist, and it is necessary to use a decision-maker preference to differentiate between solutions (Majumder, 2015).

The fundamental components of a typical DSS (Sage, 1991) 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 an MCDM problem. 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 to interact with decision-makers.

The Inference Engine of a standard DSS infers solutions and does not relay on knowledge base facts and rules, so it works independently from the other components. The Inference Engine 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

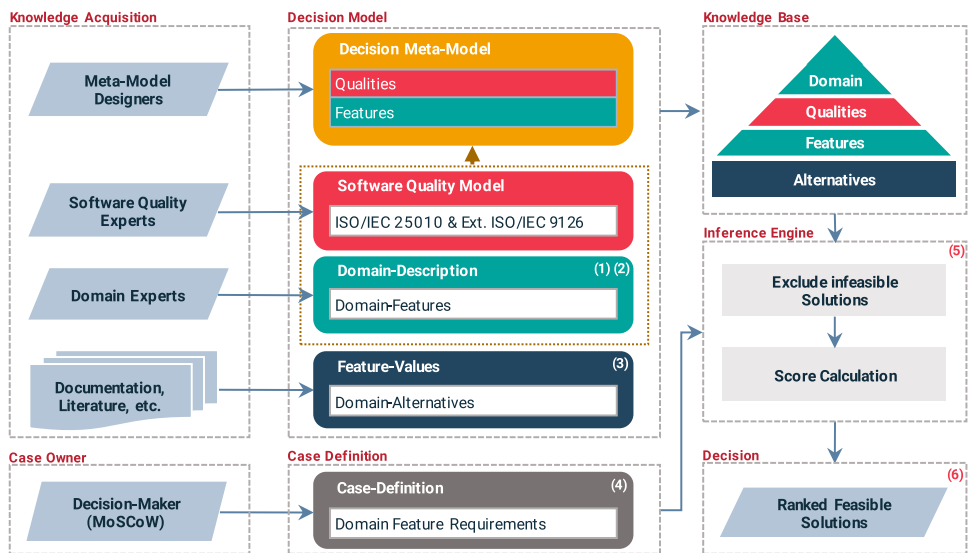


Figure 1. A model-based DSS for technology selection problems.

rules from a collection of models in the *Model-Base management system*. Then, the Inference Engine, by using facts about the *DataBase management system*, deduces decisions. Eventually, it sends ranked feasible solutions to the *Dialog Generation management system*. The DSS¹ composed of the standard DSS components and is illustrated in Figure 1.

4.1. Decision model

A decision model for an MCDM problem contains criteria, alternatives and relationships among them (facts and rules). The knowledge acquisition process for building a decision model determines the usefulness and efficiency of the outcome. This section introduces the main sources of knowledge and constituent parts of a decision model based on the six-step decision-making process.

4.1.1. Decision Meta-Model

The *Decision Meta-Model* defines 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 contains software quality attributes, and the set *Features* is a set that consists of domain features of an MCDM problem.

4.1.2. Software Quality Model

The *Software Quality Model* defines the software quality attributes and relationships among elements of the set *Qualities*. The DSS utilises the *ISO/IEC 25010* standard (ISO, 2011) and *extended ISO/IEC 9126* standard (Carvalho & Franch, 2006) in order to define the set *Qualities*. They are domain-independent software quality models and provide reference points by defining a top-down standard quality model for software systems. The elements of the *Software Quality Model* apply to classify domain features of an MCDM problem based on their impact on quality attributes of software technology alternatives.

4.1.3. Domain Description

The *Domain Description* defines the first and second steps, denoted by *Identifying the objective* and *Selection of the features*, of the decision-making process. It specifies the domain features of an MCDM problem and maps the set *Qualities* to the set *Features*, where $Qualities \times Features \rightarrow Boolean$, based on domain experts' knowledge. Each domain feature has a data type, such as *Boolean* and *Numeric*. For example, the data types of domain features like the *popularity* and *Firewall* of a DBMS could be considered as *Numeric* and *Boolean*, respectively.

4.1.4. Feature-Values

The *Feature-Values* defines the third step, indicated by *Selection of the alternatives*, of the decision-making process. It determines a set of alternatives and maps them to the domain features set, where $Alternatives \times Features \rightarrow Boolean$. The main source of knowledge in this phase could be documentation of alternatives, literature studies, social networks, alternative experts, etc.

4.2. Case definition

The *Case Definition* defines the fourth step, denoted by *Selection of the weighing method*, of the decision-making process. The DSS employs *MoSCoW* to define decision-makers' domain feature requirements and assess the importance of required domain features. 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. In other words, a case definition, based on a decision-maker preferences (*MoSCoW*), is a way to select domain feature requirements and assign priorities to them. Decision-makers specify desirable values for numeric domain feature requirements. For example, a decision-maker could be interested in prioritising the DBMSs with TCOs lower than \$5000 USD as more important than others. Therefore, the TCO lower than \$5000 USD could be considered as a *should have* domain feature.

4.3. Inference Engine

The Knowledge Base is a collection of decision models, which are groups of rules and facts. The *Inference Engine* defines the fifth and Sixth steps, indicated by *Applying the method of aggregation* and *Decision-making based on the aggregation results*, of the decision-making process. A feasible solution must support all domain feature requirements with *Must Have* priorities, and must not support all domain feature requirements with *Won't Have* priorities. The *Inference Engine* ranks the feasible alternatives based on their calculated scores. The score calculation process is based on the well-known Weighted Sum Model. Thus, by sorting the feasible solutions in descending order of their scores, the final ranked feasible solutions will be given as the result of the DSS.

5. DBMS selection

The selection of efficient and cost-effective database technology is a crucial challenge for SPOs. A number of decision factors come into play such as database model (relational,

graph, etc.), required functionality (transaction, backup, etc.), cost (license, support, etc.). Decision-makers have to follow a trustworthy and iterative process to choose the DBMS which best fulfills their requirements. Thus, SPOs are faced with an MCDM problem to find their suitable DBMS(s), because a large number of decisions of a similar kind has to make. Besides, the number of potential solutions and decision factors are significantly large.

As mentioned in Section 4.1, Constituent parts of a decision model are *Decision Meta-Model*, *Software Quality Model*, *Domain Description*, and *Feature-Values*. The *Decision Meta-Model* defines the base structure of a decision model in the knowledge base, and it has two sets namely *Qualities* and *Features*. A decision model utilises the *ISO/IEC 25010* standard and *extended ISO/IEC 9126* standard in order to define the set *Qualities*. The *Decision Meta-Model* and *Software Quality Model* are immutable for decision models based on the DSS approach. However, the *Domain Description* and *Feature-Values* should be define to structure a decision model for an MCDM problem.

This section presents a decision model according to the DSS approach to address the DBMS technology selection problem. Moreover, three case studies have been conducted to evaluate the efficiency and effectiveness of the DSS to solve DBMS selection problem for SPOs.

5.1. Domain description for DBMS selection

As mention in the Section 4.1.3, a list of domain features of technology alternatives within the domain of interest should be specified. Domain experts are the main source of knowledge to identify the right set of domain features, although documentation and literature study regarding technology alternative could be utilise to pinpoint an initial list of domain features. In order to define the domain of DBMS selection problem more than 250 features² (such as Auditing, Backup) have collected according to domain experts' suggestions. The *Software Quality Model* provides an abstract view of the software quality model. The decision model decomposes abstract concepts into more concrete ones, the domain features. Domain features have to define precisely to clarify the underlying quality concepts that they represent and to link them with the appropriate quality aspects in the set *Qualities*. The *Domain Description* does not enforce a domain feature to present in a single quality aspect; Domain features can be part of many of quality aspects. For example, *Immediate Consistency* as a DBMS feature might connect to multiple quality aspects such as *Recoverability* and *User error protection*.

In this study, domain features and the mapping between the sets *Qualities* and *Features* for the DBMS selection problem defined by nine domain experts, including two university professors, five Software Developers, and two Software Architects in the Netherlands. The domain features identified by six semi-structured interviews, then three experts participated in the research to map the considered domain features to the set *Qualities* based on a boolean adjacency matrix ($Qualities \times Features \rightarrow Boolean$).

5.2. Feature-values for DBMS selection

As mentioned in the Section 4.1.4, a list of technology alternatives of the domain of interest should be defined. Well-known technology solutions, websites, related forum and domain experts are the primary source of knowledge to specify the list of technology alternatives.

In this study, 73 DBMS technologies (Oracle Enterprise Edition 12.1, MongoDB Enterprise Server 3.4.3, etc.) from 10 data storage models (Relational, Document, etc.) have been considered. Next, supportability of boolean domain features by the DBMS technologies investigated. The relationship between the sets *Features* and *Alternatives* defined based on the documentation and websites of the considered DBMS technologies. One of the principal challenges is the lack of standard terminology among documentation of DBMS technologies. Different vendors refer to the same concept by different names, or even worse, the same name might stand for different concepts in different DBMS technologies. Discovering conflicts in the *Feature-Values* is essential to prevent semantic mismatches throughout the DBMS selection process. Manufacturers tend to provide a partial view of their products. They emphasise their product's benefits, without mentioning weaknesses, or they provide only part of the truth. Some non-commercial articles compare DBMS technologies and features but are often based on the evaluators' limited knowledge of the technologies and their particular tastes (Franch & Carvallo, 2003). The next step in building a decision model for the DBMS selection problem is defining assessment models for each non-boolean domain features, such as *Popularity in the market* and *Total Cost of Ownership*.

5.2.1. Popularity in the market

In this study, the results of DB-Engines Ranking³ is used to provide a metric on the popularity of DBMS technologies in the market. DB-Engines measures the popularity of a database system by using some parameters, such as the number of mentions of the system on websites and general interest in the system. *Popularity in the market* is a numeric domain feature of the DBMS selection problem that finds the most popular technologies in the market based on decision-makers' domain feature requirements.

5.2.2. Total cost of ownership

The cost of DBMS technologies varies widely, from entirely free to staggeringly expensive, and many factors and options should be considered. Database licensing can sometimes appear confusing, especially when it comes to well-known vendors, such as Oracle and Microsoft. Additionally, a considerable variety of pricing methods and models, such as per core and server, for calculating the database licensing costs are available.

We defined four reference configurations,⁴ including the PC (1, 1, 4, 16, 25, 256, N, 5), Basic server (1, 1, 8, 64, 2×256 , N, 25), Intermediate server (2, 2, 2×6 , 2×256 , 8×960 , A/P, 15000), and Advanced server (2, 2, 2×24 , 3000, 24×960 , A/P, ∞), to get a rough estimate of the Total Cost of Ownership (TCO) of DBMS technologies.

The TCO of each alternative was asked directly from its vendor/maintainer or calculated via offered TCO calculators on websites of DBMS vendors. Many options, offers and add-ons were not included in the TCO calculations because they were vendor specific. The TCO as a domain feature of the DBMS selection problem attempts to clear the fog somewhat regarding database licensing. However, estimated values of the TCO cannot possibly provide a full and precise insight into the complex pricing and licensing schemes that DBMS providers use.

Table 2. The number of domain feature requirements of the case studies based on the MoSCoW priorities.

MoSCoW	AFAS QS	AFAS SS	NX1
Must have	7	6	50
Should have	8	4	5
Could have	7	2	17

Table 3. The feasible solutions of the DSS for AFAS software and ProcureComp based on their domain feature requirements and MoSCOW priorities.

Case study	Feasible solutions	Desirable suggestion	Undesirable suggestion	DSS score (%)	CP rank
AFAS QS	MySQL	✓		100	2
	DB2		✓	100	–
	Oracle Database		✓	100	–
	Postgres	✓		99.80	1
	SQL Server	✓		99.77	3
AFAS SS	Postgres	✓		100	1
	MySQL	✓		100	2
	MongoDB	✓		100	4
	DB2		✓	100	–
	Oracle Database		✓	100	–
	SQL Server	✓		99.45	3
	SQL Server	✓		99.74	1

Notes: The columns *Desirable* and *Undesirable* suggestions demonstrate which DSS feasible solutions already considered in the short list of case study participants based on their internal meetings and investigations. Moreover, the Columns *CP Rank* and *DSS score* of the table show the score calculation results of the DSS and the short ranked list of the feasible solutions based on the case study participants' opinions respectively.

5.3. Empirical evidence: the case studies

Three case studies have been conducted in the context of two SPOs to evaluate and signify the usefulness and efficiency of the DSS to address MCDM problems, specifically DBMS selection problem in this study. The case study companies considered a number of feasible DBMS technologies for their organisations through multiple internal expert meetings and extensive investigation into DBMS alternations before participating in this research.

5.3.1. AFAS Software

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 DBMS(s) for the new version of their main product. The new product requires two primary data storage, namely AFAS QS and AFAS SS.

5.3.2. ProcureComp

ProcureComp is an SPO that produces procurement software. ProcureComp's product is based on Microsoft technology. Presently, the ProcureComp product is being renewed and rebuilt using new Microsoft platforms, and this is a suitable time to rethink the data storage strategy for the new version of the ProcureComp product (NX1).

Table 2 demonstrates the number of domain feature requirements, which indicated by case study participants, of the AFAS QS, AFAS SS and NX1 based on *MoSCoW*.

6. Results and analysis

Table 3 illustrates the feasible solutions of the DSS for AFAS QS, AFAS SS and NX1. The DSS deduced just one feasible solution for NX1 because ProcureComp experts restricted the search space by assigning 50 domain features as *Must Haves*, i.e. hard constraints. The reason is that the software architecture of NX1 depends heavily on the relational data storage model and on Microsoft technology. Thus, ProcureComp experts were primarily interested in finding the edition of SQL Server (Enterprise edition 2016) that best covers their requirements and priorities. For AFAS, the software architecture is not dependent on a specific data storage model or vendor. Moreover, most of the domain feature requirements of AFAS QS and AFAS SS do not require specific DBMS technology.

The amount of annual TCO was a *Should Have* domain feature for AFAS. Hence, the DSS did not exclude any alternatives based on their TCO values. Table 3 shows that *Oracle* and *IBM DB2* database technologies are not desirable suggestions for AFAS. Because the case participants find that the annual TCO of these DBMS technologies, including extra options, end up being much higher than the other feasible solutions. In other words, they perceive that *MySQL*, *SQL Server* and *Postgres* DBMS technologies are interesting suggestions because of their relatively low TCO for an intermediate server configuration, including extra options. Moreover, AFAS experts mentioned *IBM DB2* is undesirable, because they do not have enough experience with its performance, support and licensing.

The case study participants at both companies confirm that the DSS provides effective solutions to help SPOs in their initial decisions for selecting DBMS technologies. In other words, the DSS recommended the same solutions as the case participants suggested to their companies after extensive analysis and discussions. However, the DSS offers a short ranked list of feasible solutions; therefore SPOs should perform further investigations, such as performance testing and actual TCO calculation, to find the optimum DBMS technology for their software products. The case study participants state that their companies continuously improve and reevaluate their technologies, including the used DBMS technologies.

The case study participants entered a limited set of domain feature requirements. We were surprised to find that the experts have a limited view of what the domain feature requirements of the technology are. Furthermore, the case participants themselves were surprised to find what their primary concerns seem to be, especially when the opinions of different experts are combined. The fact that the DSS has led to discussions that determine decision-making for the technology illustrates that the DSS is a useful tool for SPOs and COTS decision-making. Furthermore, the DSS enables decision-makers to meet more obscure requirements that they might have. More importantly, the case study participants confirm that the updated and validated version of the DSS is useful and valuable in finding the shortlist of feasible solutions. Finally, it reduces the time and cost of the decision-making process.

The consulted experts confirm that the DSS contains the main components of a standard DSS. Furthermore, they state that the DSS is a useful tool, which 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 DSS should use in combination with benchmarks where applicable.

7. Discussion

SPOs have different perspectives on their domain feature requirements in different phases of the Software Development Life-Cycle. Decision-makers typically consider generic domain features in the early phases of the life-cycle, whereas they are interested in more detailed and specific domain features as their development process matures. For instance, *Access Control* could be prioritised as a *Should Have* domain feature in the design phase, but in the implementation phase, one of its sub-features, e.g. *Label Based Access Control*, might be selected instead. Furthermore, domain features' priorities could be changed in different phases. Therefore, the DSS might come up with various solutions for an SPO in different phases of its software development life-cycle. The proposed DSS is a tool that can be used over the full life-cycle and can co-evolve its advice based on evolving requirements. As the choices of the participants are stored in the DSS, it does not cost a significant amount of time to rerun the decision-making process. Presently, we are designing solutions that enable "the crowd" to participate in contributing knowledge, without letting anyone commercial party influence the knowledge base to its advantage. Furthermore, we are looking at methods to automatically extract domain features from manuals and documentation, using text mining techniques.

Decision-makers could bias determination of domain feature requirements and their priorities. Biases, such as motivational and cognitive (Montibeller & Winterfeldt, 2015), arise because of shortcuts or heuristics that decision-makers use to solve problems and perform tasks. The Hawthorne effect, which is the tendency of decision-makers to change their behavior when they are being observed, is a form of cognitive bias. The case study participants (AFAS and NX1 decision-makers) might have been more careful in the experimental setting than they would be in the real setting because they are being observed by scientists judging their selected domain feature requirements and priorities. Moreover, the Bandwagon effect, which is the tendency to do or believe things because many other decision-makers do or believe the same, is another form of cognitive bias. The Bandwagon effect typically shows up in group decisions. To mitigate the Hawthorne and Bandwagon effects, individual and group interviews conducted to collect the domain feature requirements for each case study.

8. Conclusion and future work

SPOs are faced with an MCDM problem when finding suitable COTS. The number of potential solutions (alternatives) and decision factors are significantly high. This paper is the first attempt at supporting architects in making complex decisions, where we ventured into the domain of DBMS technologies.

In recent years a variety of studies has been conducted to benchmark, compare and evaluate database technologies. However, according to expert analysis, selecting a suitable DBMS technology for a software product is not utterly subjective. Finding a feasible solution for this problem based on decision-makers' priorities and requirements requires deep investigation into the documentation of database technologies and extensive expert analysis. This study introduces a DSS to accelerate the process of finding the right DBMS technologies and suitable data storage models for SPOs. The novelty of the proposed DSS lies in utilising *MoSCoW* to assess criteria weights and reduce uncertainty, in introducing assessment models to measure the values of non-boolean criteria, and in using ISO/IEC

quality aspects to indicate the relationship among criteria according to domain experts' knowledge.

To keep the knowledge base of the DSS up-to-date and valid, a website⁵ has created. We plan to create a community around the platform that will regularly update the curated knowledge base with new DBMS technologies and features. It could be imagined that the DSS implementation is used as a discussion platform, that highlights conflicts and priorities, to emphasise these and lead the decision process. Probing deeper, the decision model presented in this paper also provides a foundation for future work in technology selection problems. We intend to build trustworthy decision models to address *software architectural pattern*, *cloud service provider*, and *blockchain platform selection* as our (near) future work.

Notes

1. We implemented an online Decision Model Studio (<http://dss.amuse-project.org>) to build decision models for technology selection problems in SPOs.
2. The entire list of the domain features and supportability of considered database technologies are available and accessible on the "DBMS Selection Model" website (<http://dss.amuse-project.org>).
3. The db-engines.com ranks database management systems according to their popularity. The ranking is updated monthly.
4. Each reference configuration is indicated by a 7-tuple (CPU, Socket, Core, RAM, SSD, Failover, Max.DB), consisting of the number of CPUs, number of sockets, number of cores, amount of RAM (GB), SSD capacity (GB), failover type (None and Active/Passive), and maximum database file size (GB).
5. <http://dss.amuse-project.org>.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by NWO AMUSE [project number 628.006.001].

References

- Becker, C., Kraxner, M., Plangg, M., & Rauber, A. (2013). Improving decision support for software component selection through systematic cross-referencing and analysis of multiple decision criteria. *2013 46th Hawaii international conference on system sciences (HICSS)* (pp. 1193–1202). Wailea, Maui, HI: IEEE.
- Brahimi, L., Bellatreche, L., & Ouhammou, Y. (2016). A recommender system for dbms selection based on a test data repository. In J. Pokorný, M. Ivanović, B. Thalheim, & P. Saloun (Eds.) *Advances in databases and information systems* (pp. 166–180). Cham: Springer: International Publishing.
- Büyükožkan, G., & Güleriyüz, S. (2016). A new integrated intuitionistic fuzzy group decision making approach for product development partner selection. *Computers & Industrial Engineering*, *102*, 383–395.
- Carvalho, J. P., & Franch, X. (2006). Extending the iso/iec 9126–1 quality model with non-technical factors for cots components selection. *Proceedings of the 2006 international workshop on software quality* (pp. 9–14). New York, NY: ACM.

- Consortium, D.. (2014). *The dsdm agile project framework handbook*. Ashford, Kent: DSDM Consortium.
- Franch, X., & Carvallo, J. P. (2003). Using quality models in software package selection. *IEEE Software*, 20(1), 34–41.
- Fu, L., Shi, L., Yang, Y., & Yu, B. (2010). The selection of project management software by fahp and fmcdm in automobile r&d process. *2010 2nd International conference on networking and digital society (ICNDS)* (Vol. 1, pp. 66–69). Wenzhou, China: IEEE.
- Garg, R., Sharma, R., & Sharma, K. (2017). Mcdm based evaluation and ranking of commercial off-the-shelf using fuzzy based matrix method. *Decision Science Letters*, 6(2), 117–136.
- International Organization for Standardization. (2011). *Iso/iec 25010: 2011 systems and software engineering-systems and software quality requirements and evaluation-system and software quality models*. ISO.
- Jusoh, Y. Y., Chamili, K., Che Pa, N., & Yahaya, J. (2014). Open source software selection using an analytical hierarchy process (ahp). *American Journal of Software Engineering and Applications*, 3(6), 83–89.
- Kohli, R., & Sehra, S. K. (2014). Fuzzy multi criteria approach for selecting software quality model. *International Journal of Computer Applications*, 98(11), 11–15.
- Lin, H.-Y., Hsu, P.-Y., & Sheen, G.-J. (2007). A fuzzy-based decision-making procedure for data warehouse system selection. *Expert Systems with Applications*, 32(3), 939–953.
- Majumder, M. (2015). *Multi criteria decision making* (pp. 35–47). Singapore: Springer.
- Montibeller, G., & Winterfeldt, D. (2015). Cognitive and motivational biases in decision and risk analysis. *Risk Analysis*, 35(7), 1230–1251.
- Onut, S., & Efendigil, T. (2010). A theoretical model design for erp software selection process under the constraints of cost and quality: A fuzzy approach. *Journal of Intelligent & Fuzzy Systems*, 21(6), 365–378.
- Rodriguez, A., Ortega, F., & Concepción, R. (2017). An intuitionistic method for the selection of a risk management approach to information technology projects. *Information Sciences*, 375, 202–218.
- Saaty, T. L. (1990). How to make a decision: the analytic hierarchy process. *European Journal of Operational Research*, 48(1), 9–26.
- Sage, A. (1991). *Decision support systems engineering. Wiley series in systems engineering*. Hoboken, NJ: Wiley.
- Triantaphyllou, E., Shu, B., Sanchez, S. N., & Ray, T. (1998). Multi-criteria decision making: an operations research approach. *Encyclopedia of Electrical and Electronics Engineering*, 15(1998), 175–186.