A Decision Support System for Cloud Service Provider Selection Problems in Software Producing Organizations

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

Abstract— Cloud computing enables software producing organizations to replace in-house IT infrastructure and provides them with scalable computing and flexible low cost. As cloud vendors and services on offer increase rapidly, cloud service provider selection is becoming a significant challenge for businesses. Cloud service providers and their offered services are characterized using multiple criteria, such as their popularity, geographic location, and deployment model, so it is essential to have a reliable method to select desirable cloud vendors based on decision-makers' requirements. In this study, we present a decision support system that supports decision-makers in choosing the most suitable Infrastructure-as-a-Service cloud providers. The case studies and experts confirm that the approach increases insight into the selection process, provides a richer prioritized option list than if they had done their research independently, and reduces the time and cost of the decision-making process.

Index Terms—multi-criteria decision-making, decision support system, knowledge management, cloud service provider selection, infrastructure-as-a-service

1 Introduction

Nowadays, cloud computing is influencing the IT landscape and becoming a significant economic factor for software producing organizations. Cloud computing is a fast-growing technology in a non-transparent market with diverse vendors, each of them having their specific services and deployment models. Typically, the service portfolios are heterogeneous and combined with complicated service features and pricing models. The challenge consists of evaluating and selecting the most suitable Infrastructure-as-a-Service Cloud Providers for software producing organizations according to their preferences and requirements.

The selection process is complicated because many factors, such as security and cost, have to be considered. In this study, the Infrastructure-as-a-Service Cloud Provider, in short, Cloud Service Provider (CSP) selection process is modeled 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 [24].

In most cases, a unique optimal solution for an MCDM problem does not exist, and it is necessary to use a decision-maker's preferences to differentiate between and prioritize solutions [14]. In recent years researchers introduced a considerable variety of techniques, methods, and tools to address MCDM problems. The majority of MCDM approaches in the literature use pairwise comparison techniques to calculate the weight of each decision criterion based on decision-makers' opinion. Pairwise comparison is a time-consuming process that gets more complex as the number of criteria increases [22]. Moreover, most MCDM methods are not scalable, so in the case of modifying the list of alternatives or criteria, the whole process of evaluation has to be repeated. Traditional methods are costly and applicable for a small number of criteria and alternatives. Hence, a reusable, evolvable, and expandable decision-making approach is needed to make the right decision based on the decision-makers' requirements and preferences.

This study introduces a Decision Support System (DSS) to help decision-makers with MCDM problems, such as CSP 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 [14] to build maintainable and evolvable decision models for MCDM problems, and makes the knowledge acquisition more reliable and trustful. In our previous work, we built a decision model for database technology selection problem [5], then conducted three case studies to evaluate the DSS. The final results showed that the DSS performed well to address the database selection problem for the software-producing organizations. The novelty of the DSS lies in utilizing the MoSCoW prioritization technique (*MoSCoW*) [4] 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 traditional approaches to solving decision-making problems such as ours. Section 4 outlines the details of the proposed decision support system and emphasizes

- *S. Farshidi, S. Jansen, and S. Brinkkemper are with Department of Information and Computer Science, Utrecht University, Utrecht, Netherlands. E-mail: {s.farshidi, slinger.jansen, s.brinkkemper}@uu.nl*
- *Rolf de Jong is with ARTIN Software, Eefde, Netherlands. E-mail: rolf@artinsoftware.nl*

This work is a result of the AMUSE project. (www.amuse-project.org)
Manuscript received April 15, 2018; revised ,.

the usage of novel techniques such as ISO qualities and the *MoSCoW*. Section 5 illustrates an application of the DSS to address the CSP selection problem, using four case studies to evaluate and emphasize the significance of the approach. Afterward, section 6 interprets the results of the case studies according to expert interviews and opinions; we find that the users of the DSS draw conclusions quicker and more reliably than without it. Section 7 highlights and overcomes barriers, such as motivational and cognitive biases, to the knowledge acquisition and decision-making process. Finally, section 8 summarizes the proposed approach, defends its novelty, and offers directions for future studies.

2 Research Method

Software-producing organizations typically are not knowledgeable in the problem domain, which is finding the most suitable CSPs for their businesses based on their requirements and priorities. The knowledge regarding the problem domain does not make any difference in the selection process, because the right selection requires regular studying and tracking available technologies and vendors in the market.

The selection process can be modeled 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.

Knowledge acquisition and keeping the acquired knowledge up-to-date are time-consuming and costly processes for Software-producing organizations. To support these organizations, 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 infused with expert knowledge, which is gathered through three series of interviews. Twelve experts (three DSS experts, six cloud consultants, and three cloud architects) participated in this research to evaluate the DSS in interviews that lasted between 45 and 90 minutes. The domain experts were pragmatically selected according to their expertise and experience that they mentioned in their professional profile. Each of the interview series followed a semi-structured interview protocol. Data collected during one interview, would typically be propagated to the next, to incrementally build and validate the knowledge base. The knowledge base was sent to the interview participants afterward for final confirmation.

Secondly, the efficiency and usefulness of the DSS are evaluated through four exploratory theory-testing case studies. The unit of analysis is a unique CSP selection for a Software Producing Organization. We performed four such case studies at four software producing organizations to evaluate the DSS. The case studies typically lasted one day and consisted of (1) defining the domain feature requirements, (2) prioritizing them, and (3) comparing the DSS feasible solutions with their solutions.

3 Related Work

The proposed DSS applies the six-step decision-making process [14] to build decision models for MCDM problems and distinguishes itself from the currently existing DSSs in the following ways: 1) the DSS utilizes the *MoSCoW* [4] to assess criteria weights and reduce uncertainty, 2) employs assessment models to measure the values of non-boolean criteria, and 3) uses the ISO/IEC quality aspects to indicate the relationship among criteria according to domain experts' knowledge.

Snowballing was employed as the principal method to investigate the existing literature related to the techniques which address MCDM problems for Software-producing organizations. Some recent methods can be listed as follows: The *Analytic Hierarchy Process (AHP)* is a structured method for organizing and analyzing 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 *Fuzzy Delphi Method (FDM)* is a more advanced version of the Delphi Method in that it utilizes triangulation statistics to determine the distance between the levels of consensus within the expert panel. The *FAHP* and *FTOPSIS* are the combinations of Fuzzy logic with the AHP and TOPSIS methods. The *FMCDM* assesses the ratings of alternatives versus criteria and the importance weights of criteria based on semantic values represented by fuzzy numbers.

Table 1 illustrates selected MCDM approaches from literature. The majority of the techniques in literature use pairwise comparison as the main method to assess the weight of criteria. For a problem with n number of criteria $\frac{n(n-1)}{2}$ number of comparison is needed [21]. It means that the 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, 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 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.

Franch et al. [7] introduced a six-step method to solve the Commercial Off-The-Shelf selection problem. The six-step method considers the ISO/IEC 9126-1 standard as for quality attributes and decomposes it into the domain features of the Commercial Off-The-Shelf packages. Moreover, decision-makers should define specific metrics for each domain feature to assign a value to it. Finally, the results of considered Commercial Off-The-Shelf packages will be compared. Becker et al. [1] present a multi-criteria decision support system (MCDSS) for software component selection. The MCDSS evaluates a total of 51 Commercial Off-The-Shelf components against a total of 631 decision criteria.

TABLE 1
This table compares selected MCDM methods from literature to address technology selection problems. The second column (Problem domain) points out the problem domain. The third column (MCDM) denotes the MCDM approach. The fourth column (Pairwise Comparison) indicates whether the approach applies pairwise comparison as a weight calculation method or not. The fourth column (Quality Attributes) determines the type of quality attributes. The seventh and eighth columns (Criteria and Alternatives) signify the number of criteria and alternatives that were considered in the problem domain.

Author(s)	Problem domain	MCDM	Pairwise Comparison	Quality Attributes	Criteria	Alternatives
[13]	Cloud vendor selection	TOPSIS FDM	NO	Domain specific	4	4
[18]	Software-as-a-Service product selection	AHP	YES	Domain specific	57	3
[8]	Cloud service ranking	AHP	YES	ISO/IEC SMI	29	3
[10]	Cloud security service selection	AHP	YES	Domain specific	16	5
[12]	Cloud service selection	FAHP FDM	YES	Domain specific	18	5
[9]	Software-as-a-Service product seletion	AHP	YES	Domain specific	21	4
[20]	ERP software	FMCDM	YES	Domain specific	23	4
[19]	Risk management pproach	FAHP	YES	Domain specific	5	5
[2]	Product development partner	FAHP FTOPSIS	YES	Domain specific	16	6
[17]	Content Management System selection	AHP TOPSIS	YES	Domain specific	7	4
[1]	Commercial Off-The-Shelf selection	DSS	NO	ISO/IEC 25010 Domain specific	631	51
[5]	Database technology selection	DSS	NO	ISO/IEC 25010 EX. ISO/IEC 9126	307	73
This paper	Cloud service provider selection	DSS	NO	ISO/IEC 25010 EX. ISO/IEC 9126	300	40

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.

One of the weaknesses of the six-step method [7] is, when the number of alternatives and domain features is high, measuring the qualities of domain features for each alternative is not possible, or is a very time-consuming process. Furthermore, the assigned values for the domain features will be changed by technologies advances. The proposed DSS is superior to the six-step method because it is an evolvable and expandable model-based approach that splits down the decision-making process into four maintainable phases (Section 4). The DSS and MCDSS both provide a substantial number set of criteria to support decision-makers. Furthermore, they use the ISO/IEC 25010 as a standard set of quality attributes. The main difference between the DSS and MCDSS is their weighting methods. We built a decision model for database technology selection problem [5], then conducted three case studies to evaluate the DSS. The final results showed that the DSS performed well to address the database selection problem for the software-producing organizations. The DSS utilizes the *MoSCoW* to assess the importance of criteria and reduce the uncertainty, moreover it introduces assessment models to measure the values of non-boolean criteria, such as the cost and popularity of the alternatives.

4 Multi-Criteria Decision-Making

The fundamental components of a typical DSS [23] 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 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 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 DSS[1] comprises of the standard DSS components. The proposed DSS [6] applies the six-step decision-making process [14] to build decision models for MCDM problems. Furthermore, it makes the knowledge acquisition more reliable and trustful. A decision model defines a decision structure to solve a specific MCDM problem. Figure 1 depicts the structure of the DSS.

4.1 Decision Model

Knowledge acquisition is the process of extracting, structuring and organizing knowledge from different sources of knowledge, including human experts, documentation, and literature. This process applies to define knowledge base facts and rules. This section elaborates the knowledge acquisition process, the main sources of knowledge, and constituent parts of a decision model based on the six-step decision-making process for building a decision model to address an MCDM problem.

1. We implemented an online Decision Model Studio (http://dss.amuse-project.org) to build decision models for MCDM problems in Software-producing organizations.

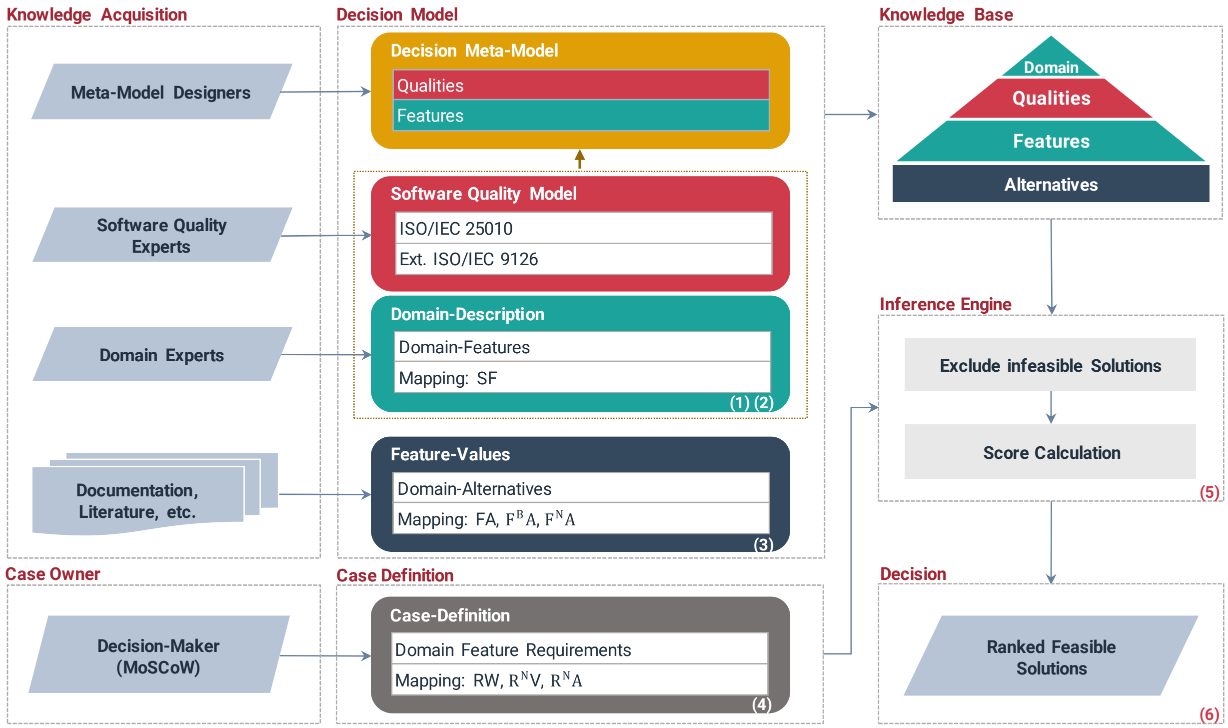

Fig. 1. A model-based decision support system for MCDM problems.

4.1.1 Decision Meta-Model

The *Decision Meta-Model* defines the base structure (abstraction) of a decision model in the knowledge base. The *Decision Meta-Model* includes two primary sets (*Qualities* and *Features*). The set *Qualities*, denoted by Q, is a set that keeps software quality attributes, and the set *Features*, denoted by F, is a set that retains domain features of an MCDM problem.

4.1.2 Software Quality-Model

The *Software Quality-Model* determines the software quality attributes (Q), and defines relationships, based on hierarchical structure, among elements of the set Q, thus, the Q is a nested set of quality attributes. The DSS utilizes the *ISO/IEC 25010* standard [11] and *extended ISO/IEC 9126* standard [3] in order to define the set Q. These quality standards are domain-independent software quality models and provide reference points by defining a top-down standard quality model for software systems. The ISO/IEC standard quality models have two hierarchical levels, being *Characteristics* and *Sub-characteristics*. The mappings between these levels are defined as follows. Suppose C and S are the sets of the *Characteristics* and *Sub-characteristics* of the ISO/IEC quality models. Then, the mapping between these two sets, $CS : C \times S \to \{0, 1\}$, are defined according to the ISO/IEC quality models. So that, $CS(c, s)$, where $c \in C$ and $s \in S$, is equal to one when c is connected to s, otherwise it is equal to zero. The elements of the *Software Quality-Model* apply to classify domain features (F) of an MCDM problem based on their impact on *Sub-characteristics* of the ISO/IEC quality models (SF); Moreover, they use to calculate the impact factor of domain feature requirements based on decision-makers' preferences and domain experts' knowledge (equation 3).

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. In other words, the *Domain Description* specifies the domain features (F) of an MCDM problem. Each domain feature has a data type, which could be *Boolean*, denoted by F^B, or *Numeric*, denoted by F^N, where $F^B \cap F^N = \emptyset$ and $F = F^B \cup F^N$. For example, the data type of a domain feature like the *popularity* of alternatives is *Numeric*. The mapping between sets Q and F is based on the domain experts' knowledge. As mentioned prior, the DSS uses the ISO/IEC quality models to define the set Q, which is a nested set. The last level of the hierarchal structure in the set Q is the *Sub-characteristics* of the ISO/IEC quality models, denoted by S. Therefore, the mapping, $SF : S \times F \to \{0, 1\}$, defines the relationship between the *Sub-characteristics* (S) and domain features (F). So that, $SF(s, f)$, where $s \in S$ and $f \in F$, is equal to one when s is connected to f, otherwise it is equal to zero. Domain features could organize into conceptual hierarchical structures. So that, generic domain features split down into more specific domain features (sub-features). For instance, in the CSP selection problem, *Automation and orchestration* is a generic domain feature, moreover *Kubernetes*, *Docker Swarm*, and *Ansible* are considered as its sub-features. When an alternative supports a generic domain feature, means that it supports at least one of the sub-features of the generic feature. The domain features were identified through interviews with the domain experts. The main aim of the interviews was to establish the prominent domain features and identify domain features that could be left out. Note that, conceptual hierarchical structures of domain features assist decision-makers to prioritize the domain features based on their expertise and knowledge of the project requirements. Again, these hierarchies were established through domain expert interviews. Moreover, no difference exists between generic domain features and sub-features from decision model perspective. Thus, the set F contains generic domain features and sub-features, and SF maps them to the set S.

4.1.4 Feature-Values

The *Feature-Values* defines the third step, indicated by *Selection of the alternatives*, of the decision-making process. The *Feature-Values* determines a set of alternatives, denoted by A, and maps them to the domain features set (F). The main source of knowledge in this phase is documentation of alternatives, literature studies, social networks, domain experts, etc. As mentioned in the *Domain Description* phase, the data type of domain features could be *Boolean* or *Numeric*. The mapping, $F^B A : F^B \times A \rightarrow \{0, 1\}$, maps the boolean domain features (F^B), and the mapping, $F^N A : F^N \times A \rightarrow \mathbb{R}_{\geq 0}$, maps the numeric domain features (F^N) to alternatives (A). So that, $F^B A(f, a)$, where $f \in F^B$ and $a \in A$, is equal to one when f is connected to a (boolean domain feature f is supported by alternative a), otherwise it is equal to zero. Moreover, $F^N A(f, a)$, where $f \in F^N$ and $a \in A$, specifies the value of domain feature f regarding alternative a. In other words, the mapping $F^N A$ assigns the values of assessment models to numeric domain features, such as *cost* and *popularity*.

4.2 Case Definition

The *Case Definition* defines the fourth step, denoted by *Selection of the weighing methods to indicate the importance of the features*, of the decision-making process. The DSS utilizes *MoSCoW* to define decision-makers' domain feature requirements and assess the importance of required domain features. Note that domain feature requirements set (R) is a subset of domain features, where $R \subseteq F$ and $R^B = R \cap F^B$ and $R^N = R \cap F^N$. 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* [4], where $\forall w \in W_{MoSCoW}; 1 \geq w \geq 0$. In other words, a case definition, based on a decision-maker's preferences (*MoSCoW*), is a way to select domain feature requirements and assign priorities to them, $RW : R \rightarrow W_{MoSCoW}$.

The importance of a domain feature with *Must Have* priority must be greater than all domain features with *Should Have* priority, where $\sum_{\forall r \in R; RW(r) = w_{Should}} RW(r) < w_{Must}$. Furthermore, the importance of a domain feature with *Should Have* priority must be greater than all domain features with *Could Have* priority, where $\sum_{\forall r \in R; RW(r) = w_{Could}} RW(r) < w_{Should}$.

Decision-makers specify desirable values for numeric domain feature requirements, $R^N V : R^N \rightarrow \mathbb{R}_{\geq 0}$. For example, a decision-maker could be interested in prioritizing the CSPs with Total Cost of Ownership (TCO) less than 500 USD as more important than others. Therefore, the TCO less than 500 USD could be considered as a *should have* domain feature. Consequently, a mapping, $R^N A : R^N \times A \times \mathbb{R}_{\geq 0} \rightarrow \{0, 1\}$, is considered to define these types of numerical criteria by decision-makers. The *Case Definition* receives mappings RW and $R^N V$ as its input from the user interface of the DSS. Indeed, a decision-maker is the main source of the knowledge in this phase. Domain feature requirements with *Must Have* or *Won't Have* priorities act as hard constraints (H) and domain feature requirements with *Should Have* and *Could Have* priorities act as soft constraints.

4.3 Inference Engine

Each decision model defines a decision structure for an MCDM problem systematically. Moreover, the mappings define rules and facts. Therefore, the Knowledge Base is a collection of decision-models, which are groups of rules and facts. The *Inference Engine* defines the fifth step, indicated by *Applying the method of aggregation*, of the decision-making process. The *Inference Engine* ranks the alternatives based on their calculated scores. The score calculation process begins with computing the weight of each aspect of a decision model. As mentioned prior the relationship between aspects are defined based on the mappings (CS, SF, $F^B A$, $F^N A$, RW, $R^N V$, and $R^N A$). The summary of the sets and mapping of a decision model and a case definition for an MCDM problem is shown in table 2. Note that the weight of domain feature requirements ($RW(r)$, where $r \in R$) assign by the decision-maker via the *MoSCoW*. Moreover, the weights *Sub-characteristics* (W_s) and *Characteristics* (W_c) of the ISO/IEC quality models in the set Q are the sum of the weights of their children.

$$W_{s \in S} = \sum_{\forall r \in R; r \notin H} SF(s, r).RW(r) \quad (1)$$

$$W_{c \in C} = \sum_{\forall s \in S} CS(c, s).\bar{W}_s \quad (2)$$

$\bar{W}_s$ is the normalized to unity weights of the *Sub-characteristics*. Next, the impact factor, denotes by I_r, of each domain feature requirement ($r \in R$) is equal to the sum of products of its parents' weights plus the weight of the domain feature requirement. The reason behind this impact factor calculation is finding the importance of domain feature requirements based on the decision-maker preferences and the relationship among domain feature requirements according to domain experts' knowledge. Moreover, it assures that the *MoSCoW* priorities of the domain feature requirements never change. In other words, if the decision-maker assigned the *Could Have* priority to a domain feature requirement, its importance would not become greater than a domain feature requirement with the *Should Have* priority.

$$I_{r \in R} = RW(r) + \sum_{\forall s \in S} SF(s, r).\bar{W}_s \sum_{\forall c \in C} CS(c, s).\bar{W}_c \quad (3)$$

$\bar{W}_c$ is the normalized to unity weights of the *Characteristics*. A feasible alternative 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. Equation 4 through mappings $F^B A$, $R^N A$, and $R^N V$ indicate whether all boolean domain feature requirements ($H \cap R^B$) and numeric domain feature requirements ($H \cap R^N$) with *Must Have*and *Won't Have* priorities (hard constraints) are supported by the alternative a or

TABLE 2
The summary of the sets and mapping of a decision model and a case definition for an MCDM problem.

Notation		Definition	Description	Source of Knowledge
C		-	The set of *Characteristics*.	ISO/IEC standards
S		-	The set of *Sub-characteristics*.	ISO/IEC standards
A		-	The set of alternatives.	Documentation
F	=	$F^B \cup F^N$	The set of domain features.	Domain Experts
F^B		-	The set of boolean domain features.	Domain Experts
F^N		-	The set of numeric domain features.	Domain Experts
R	=	$R \subseteq F$	The set of domain feature requirements.	Decision-Makers
R^B	=	$R \cap F^B$	The set of boolean domain feature requirements.	Decision-Makers
R^N	=	$R \cap F^N$	The set of numeric domain feature requirements.	Decision-Makers
W_{MoSCoW}	=	$\{w_{Must}, w_{Should}, w_{Could}, w_{Won't}\}$	The set of priority weights.	MoSCoW Priorities
CS	:	$C \times S \rightarrow \{0,1\}$	The mapping between the sets C and S.	ISO/IEC standards
SF	:	$S \times F \rightarrow \{0,1\}$	The mapping between the sets S and F.	Domain Experts
F^BA	:	$F^B \times A \rightarrow \{0,1\}$	The mapping between the sets F^B and A.	Documentation
F^NA	:	$F^N \times A \rightarrow \mathbb{R}_{\geq 0}$	The mapping between the sets F^N and A.	Documentation
RW	:	$R \rightarrow W_{MoSCoW}$	The mapping between the sets R and W_{MoSCoW}.	Decision-Makers
R^NV	:	$R^N \rightarrow \mathbb{R}_{\geq 0}$	Desirable values for the set R^N.	Decision-Makers
R^NA	:	$R^N \times A \times \mathbb{R}_{\geq 0} \rightarrow \{0,1\}$	Numerical criteria.	Decision-Makers
H	=	$\bigcup_{\substack{\forall r \in R \\ RW(r)=w_{Must} \vee RW(r)=w_{Won't}}} r$	Hard constraints	Decision-Makers

not. Note, hard constraint numeric domain features contain numerical criteria which indicate by decision-makers. For example, a decision-maker could be interested in considering only CSPs which their TCO values are less than 500 USD, so $TCO < 500$ is a numeric domain feature with *Must Have* priority.

$$Sum = \sum_{\forall r \in (H \cap R^B)} F^BA(r,a) + \sum_{\forall r \in (H \cap R^N)} R^NA(r,a,R^NV(r))$$

$$Feasible_{a \in A} = \begin{cases} 1, & \text{if } Sum = |H| \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The score calculation process (equation 5) involves the sum of products of impact factors of domain feature requirements with *Should Have* and *Could Have* priorities.

$$Score_{a \in A} = Feasible_a . \left(1 + \sum_{\forall r \in (R^B \setminus H)} \bar{I}_r . F^BA(r,a) + \sum_{\forall r \in (R^N \setminus H)} \bar{I}_r . R^NA(r,a,R^NV(r))\right) \tag{5}$$

If in the score calculation process $RW(r)$ is equal to w_{Could} then $\bar{I}_r$ is normalized to $[w_{Could}, w_{Should})$, otherwise, $\bar{I}_r$ is normalized to $[w_{Should}, 1)$.

Equation 5 and equation 6 define the sixth step, denoted by *Decision making based on the aggregation results*, of the decision-making process. Note, the scores of feasible solutions are more than zero.

$$Solutions = A \setminus \bigcup_{\substack{\forall a \in A \\ Score_a = 0}} a \tag{6}$$

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 Cloud Service Provider Selection

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 utilizes 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 based on the DSS approach to address the CSP selection problem. Moreover, four case studies have been conducted to evaluate the efficiency and effectiveness of the DSS to address CSP selection problem for software producing organizations.

5.1 Domain Description for CSP selection

As mentioned in the section 4.1.3, a list of domain features (F) of the domain of interest should be specified. Domain experts are the main source of knowledge to identify domain features. In order to define the domain of CSP selection problem more than 250 features[2] (such as Automation and orchestration, Application Server, Certifications/Attestations, and Cost) have been collected according to domain experts' suggestions. The sub-characteristics of the *Software Quality-Model* provides an abstract view of the software quality model. The decision model decomposes

2. The entire list of the domain features and supportability of considered cloud service providers are available and accessible on the "Cloud Service Provider Selection" website (http://dss.amuse-project.org)

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 sub-characteristics. Some domain features are related to more than one sub-characteristic. For example, *Automation and orchestration* as a CSP feature might include in Availability, Reusability, and Installability. The DD does not enforce a domain feature to present in a single sub-characteristic; Domain features can be part of many of quality aspects. As mentioned earlier, the relationship between sets S and F are defined by the mapping SF according to domain experts' opinion. In this study, CSP features and the mapping SF defined by nine domain experts, including six cloud consultants and three cloud architects in the Netherlands.

TABLE 3
The reference configurations for calculating the Total Cost of Ownership of CSPs. Each reference configuration is indicated by its number of CPU cores, amount of RAM (GB), and SDD capacity (GB).

Server Configurations	CPU(Cores)	Memory	SSD
Basic Server	4	8 GB	100 GB
Intermediate Server	8	32 GB	100 GB
Memory-intensive Server	16	512 GB	200 GB
CPU-intensive Server	24	64 GB	500 GB

5.2 Feature-Values for CSP selection

As mentioned in the section 4.1.4, a list of alternatives of the domain of interest should be defined. Well-known CSPs, websites, related forum, and domain experts are the primary source of knowledge to specify the alternatives. In this study 40 infrastructure-as-a-service CSPs (Leaseweb, Google Cloud, etc.) as the alternatives have been considered. The list of CSP alternatives collected from recent reports of the *Gartner*, *Glassdoor*, and *Forrester* websites.

Next, supportability of boolean domain features (F^B) by the CSP alternatives (A) should be investigated. The relationship between sets F^B and A defined by the mapping F^BA based on the documentation and websites of the considered CSPs. One of the principal problems is the lack of standard terminology among documentation of CSPs. Different CSPs refer to the same concept (cloud service) by different names, or even worse, the same name might stand for different concepts in different CSPs. Discovering conflicts in the *Feature-Values* is essential to prevent semantic mismatches throughout the CSP selection process.

CSPs tend to provide a partial view of their cloud services. They emphasize their services' benefits, without mentioning weaknesses, or they provide only part of the truth. Some non-commercial articles compare CSPs and features but are often based on the evaluators' limited knowledge and their particular tastes [7]. The next step in building a decision model for the CSP selection problem is defining assessment models for each numeric domain features, such as cost and popularity. After defining suitable assessment models for numeric domain features (F^N), the mapping F^NA maps them to the corresponding CSP alternatives (A). For example, *Total Cost of Ownership*, *Popularity in the market*, *Company Maturity*, and and *Innovation* are non-boolean domain features in the decision model of the CSP selection problem.

Non-boolean domain features could be grouped into a number of categories (ranges) based on their values. Categories facilitate the usage of relational criteria. For example, a decision-maker could be interested in prioritizing the CSPs with Total Cost of Ownership values less than $500 USD as more important than others. Therefore, the Total Cost of Ownership values less than $500 USD could be considered as a *should have* domain feature.

5.2.1 Total Cost of Ownership

The cost of CSPs varies widely, and many factors and options should be considered. The Total Cost of Ownership (TCO) sometimes appear confusing, especially when it comes to well-known service providers (such as Oracle, Microsoft, Google), where a large variety of parameters (such as Operating System Licenses, Storage per GB/TB prices) for calculating the CSP costs are available. Thus, to get a rough estimate of the TCO of CSPs, four reference configurations for three cloud deployment models and server types (including Physical private cloud, Virtual private cloud, and Virtual public cloud) are provided. Table 3 demonstrates the considered reference configurations.

The TCO value of each CSP alternative was asked directly from the CSP or calculated via the offered TCO calculator on the website of the CSP. Note that TCO values should be computed in the same currency (e.g., USD) and time span (e.g., monthly) to provide a correct comparison. Many options, offers, and add-ons were not included in the TCO calculations because they were CSP specific. The TCO as a domain feature of CSP selection attempts to clear the fog somewhat regarding CSP prices. However, estimation of TCO values cannot possibly provide a full and precise insight into the complex pricing models that CSPs use.

5.2.2 Popularity in the market

This non-boolean feature is one of the assessment models in the CSP selection problem. It ranks CSPs based on their popularity in the market by using the following parameters: a. The number of mentions of CSPs on websites, b. The frequency of technical discussions about CSPs on websites, c. The number of job offers on the leading job search engines, and d. Relevance in social networks.

5.2.3 Company Maturity

This assessment model measures the company maturity of CSPs based on three main factors, including company size (number of employees), company revenue, and date of establishment. In other words, a mature CSP company is well-established in the market, with well-known services and loyal customer following with average growth. Mature companies are categorized according to the business stage it is currently in. We considered a three-stage maturity level (high, middle, and low) for the CSPs.

5.2.4 Innovation

Innovation is often viewed as the application of better solutions that meet new requirements, unarticulated needs, or existing market needs [15]. This is accomplished through more effective products, processes, services, technologies, or business models that are readily available to the market. This assessment model measures the innovation of CSPs based on supportability of following factors: a. Internet of Things Cloud, b. Big Data Analytics, c. Business Intelligence, d. Enterprise reporting, e. Dynamically scale to meet capacity demands, and f. Multiple data centers.

5.3 Empirical Evidence: The Case Studies

Four case studies in the context of four software producing organizations have been conducted to evaluate and signify the usefulness and efficiency of the DSS. The case study companies considered a number of feasible CSPs for their organizations through multiple internal expert meetings and extensive investigation into CSP alternations before participating in this research.

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 CSP for the new version of their product.

KPMG - KPMG is a professional service company with more than 189,000 employees and located in the Netherlands. KPMG has three lines of services: financial audit, tax, and advisory. KPMG participated in this research to select a well qualified CSP for one of its customers.

Health Diaries - Health Diaries is a small Software Producing Organization with ten employees and located in the Netherlands. Health Diaries is developing digital healthcare diaries based on expertise from healthcare professionals and medical science, which makes them useful for healthcare institutions. Health Diaries experts are interested in evaluating different CSPs in the market and selecting the suitable one that fulfills their requirements and priorities.

Negometrix - Negometrix produces procurement software. Its customers are one-third government, one-third non-profits, and one-third commercial organizations. Presently, the Negometrix product is being renewed and rebuilt using new Microsoft platforms, and this is a suitable time to rethink the CSP for the new version of the Negometrix product.

Table 4 demonstrates parts of the domain feature requirements of the case studies based on the MoSCoW priorities.

6 Results and Analysis

The feasible solutions of the DSS for the case studies are shown in Table 5. The KPMG domain feature requirements are mainly generic domain features, such as *Automation and orchestration* and *Auditing/Logging*, or standard features, which are supported by most of the CSP alternatives. Therefore, the DSS deduced 8 feasible solutions for KPMG in spite of 22 domain feature requirements with *Must Have* priority (Hard constraints). Health Diaries domain feature requirements target specific CSPs, which support health care companies. Thus, the DSS suggested only 4 feasible solutions for Health Diaries. Negometrix domain feature requirements are mostly technical domain features, moreover Geo-locations of the data-centers is one of the feature requirements with *Must Have*. consequently, the DSS inferred 4 feasible solutions for Negometrix. AFAS domain feature requirements are generic domain features. Obviously, the number of hard constraints in the AFAS requirements are lower than the other case studies. As a result, the DSS recommended 10 feasible solutions for AFAS.

The annual TCO was a *Should Have* domain feature requirement for AFAS and KPMG. Hence, the DSS did not exclude any alternatives based on their TCO values. Some of the feasible solutions proposed by the DSS were not on the shortlist of case participant because they found that the annual TCO of these CSPs, including extra options, end up being much higher than the other feasible solutions. Also, the case participants stated that lack of experience with the performance and Service-Level-Agreement of such CSPs is another reason for ignoring them. Columns *CP Rank* and *DSS score* of table 5 show the score calculation results of the DSS and the short ranked list of the feasible solutions based on the case participants' opinions respectively.

The case study participants confirm that the DSS provides effective solutions to help software producing organizations in their initial decisions for selecting CSPs. 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 software producing organizations should perform further investigations, such as performance testing and actual TCO calculation, to find the optimum CSPs for their software products. Twelve experts (three DSS experts, six cloud consultants, and three cloud architects) participated in this research to evaluate the DSS.

The consulted experts confirm that the DSS contains the main components of a standard DSS. Moreover, they asserted that the score calculation process in the *Inference Engine* of the DSS is not dependent on the knowledge-base facts and rules (i.e., the decision model). Therefore, if a decision model for an MCDM problem is replaced by another one, the *Inference Engine* does not generate invalid solutions.

The experts believe that experience in using a technology provides invaluable knowledge when selecting suitable technology. Consequently, we recommend that our DSS is used in combination with benchmarks where applicable. Furthermore, the experts indicate that supported domain features by CSPs play a significant role in the CSP selection process. Some domain features are supported by specific CSPs, for instance, *NEN 7510* is the standard for information security in health care. Also, the supported domain features are going to change due to technological advances. As such, the knowledge-base must be updated regularly. The experts state that their companies continuously improve and reevaluate their technologies, including the used CSPs.

The case study participants enter 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. 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 software produc-

TABLE 4
A part of AFAS, Negometrix, KPMG, and Health Diaries domain feature requirements based on the MoSCoW. Note that the numbers in the table indicate the number of domain feature requirements in a particular MoSCoW priority for each case study. For example, AFAS has nine domain feature requirements with *Could Have* priority.

MoSCoW	AFAS		Negometrix		KPMG		Health Diaries	
Must Have	Service Fabric, Disaster recovery, etc.	7	.Net, ISO 27001, etc.	21	Node.js, SOC2, etc.	22	Java, MySQL, etc.	13
Should Have	High Company Maturity, Memory-intensive server, etc.	9	Encryption, Packet Filtering, etc.	11	DevOps, GitLab, etc.	12	HL7, Auto Scaling, etc.	4
Could Have	Kubernetes, Windows Server Container, etc.	9	Free private transfer, Network IDS, etc.	17	Automation and orchestration	1	Big data analytics, Database Backup-as-a-service	2

TABLE 5
The feasible solutions of the DSS for AFAS, Negometrix, KPMG, and Health Diaries based on their domain feature requirements and MoSCOW priorities. The column *CP (Case Participant) Shortlist* demonstrates which DSS feasible solutions already considered in the shortlist 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 ranked shortlist of the feasible solutions based on the case study participants' opinions respectively.

Case Study	DSS Feasible solutions	CP Shortlist	DSS Score	CP Rank
AFAS	Microsoft Azure	✓	93.34	3
	IBM Cloud		92.41	-
	OVH		91.67	-
	DataPipe		86.12	-
	KPN (iS)	✓	84.81	4
	Google Cloud	✓	84.59	2
	Leaseweb	✓	83.05	1
	Interoute		77.22	-
	Amazon (AWS)		76.23	-
	1and1		75.15	-
Negometrix	Microsoft Azure	✓	99.86	1
	Leaseweb		99.69	-
	Google Cloud	✓	99.57	2
	KPN (iS)	✓	99.54	3
KPMG	Google Cloud	✓	100.00	3
	Rackspace		100.00	-
	Amazon (AWS)	✓	94.42	2
	Microsoft Azure	✓	94.42	1
	Fujitsu	✓	76.16	5
	Oracle Cloud		51.00	-
	IBM Cloud	✓	51.00	4
	Alibaba Cloud		32.75	-
Health Diaries	Microsoft Azure	✓	100.00	1
	Amazon (AWS)	✓	100.00	2
	Leaseweb	✓	82.24	3
	Fujitsu	✓	82.24	4

ing organizations and MCDM problems. More importantly, the case 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.

7 Discussion

Software producing organizations have different perspectives on their domain feature requirements in different phases of the Software Development Life-Cycle. Decision-makers might want to consider generic domain features in the early phases of the life-cycle, whereas they are interested in more technical domain features as their development process matures. For instance, *Automation and orchestration* could be prioritized as a *Should Have* domain feature in the design phase, but in the implementation phase, one of its sub-features (more technical domain feature), e.g., *Service Fabric*, might be selected instead. Furthermore, domain features' priorities could be changed in different phases. Therefore, the DSS might come up with various solutions for a software producing organization in different phases of its software development life-cycle. 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.

Biases, such as motivational and cognitive [16], arise because of shortcuts or heuristics that decision-makers use to solve problems and perform tasks. The Hawthorne effect, which is the tendency for decision-makers to change their behavior when they are being observed, is a form of cognitive bias. The case study participants 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 have been conducted.

We define DSS success when it in part aligns with the CP's shortlist and when it provides new suggestions that are identified as being of interest to the CP. Using the CP experts' opinion as a measurement instrument is risky, as the CP may not have sufficient knowledge to make a valid judgment. We counter this risk by conducting more than one case study, by assuming that the CP expert is handling in its interest, and by applying the DSS to other problem domains, where we find similar results [5], [6].

8 Conclusion and Future Work

Finding a feasible solution for the Infrastructure-as-a-Service Cloud Provider selection problem based on decision-makers' priorities and requirements requires deep investigation into the documentation of cloud vendors and extensive expert analysis. This study introduces a Decision Support System (DSS) to accelerate the process of finding the right Infrastructure-as-a-Service Cloud Provider for software producing organizations. The DSS comprises all of the fundamental components of a standard DSS. A decision model in the knowledge base of the DSS contains all facts and rules of an MCDM problem. In other words, a decision model defines a decision structure to solve a specific MCDM problem.

The novelty of the proposed DSS lies in utilizing the *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. Our website[3] is up and running to keep the knowledge base of the DSS up-to-date and valid. We plan to create a community around the platform that will regularly update the curated knowledge base with new Infrastructure-as-a-Service Cloud Provider features.

Probing deeper, the decision model presented in this paper also provides a foundation for future work in MCDM problems. We intend to build trustworthy decision models to address *software architecture pattern* and *blockchain solution* selection problems as our (near) future work.

3. http://dss.amuse-project.org

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*, pages 1193–1202. IEEE, 2013.

[2] G. Büyüközkan and S. Güleryüz. A new integrated intuitionistic fuzzy group decision making approach for product development partner selection. *Computers & Industrial Engineering*, 102:383–395, 2016.

[3] 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*, pages 9–14. ACM, 2006.

[4] D. Consortium et al. The dsdm agile project framework. *http://www.dsdm.org/dig-deeper*, 2014.

[5] S. Farshidi, S. Jansen, R. de Jong, and S. Brinkkemper. A decision support system for software technology selection. *Journal of Decision Systems*, 2018.

[6] S. Farshidi, S. Jansen, R. de Jong, and S. Brinkkemper. Multiple criteria decision support in requirements negotiation. *International Conference on Requirements Engineering: Foundation for Software Quality*, 2018.

[7] X. Franch and J. P. Carvallo. Using quality models in software package selection. *IEEE software*, 20(1):34–41, 2003.

[8] S. K. Garg, S. Versteeg, and R. Buyya. Smicloud: A framework for comparing and ranking cloud services. In *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*, pages 210–218. IEEE, 2011.

[9] M. Godse and S. Mulik. An approach for selecting software-as-a-service (saas) product. In *Cloud Computing, 2009. CLOUD'09. IEEE International Conference on*, pages 155–158. IEEE, 2009.

[10] T. Halabi and M. Bellaiche. Evaluation and selection of cloud security services based on multi-criteria analysis mca. In *Computing, Networking and Communications (ICNC), 2017 International Conference on*, pages 706–710. IEEE, 2017.

[11] ISO. Iec25010: 2011 systems and software engineering–systems and software quality requirements and evaluation (square)–system and software quality models. *International Organization for Standardization*, 34:2910, 2011.

[12] S. Lee and K.-K. Seo. A hybrid multi-criteria decision-making model for a cloud service selection problem using bsc, fuzzy delphi method and fuzzy ahp. *Wireless Personal Communications*, 86(1):57–75, 2016.

[13] S. Liu, F. T. Chan, and W. Ran. Decision making for the selection of cloud vendor: An improved approach under group decision-making with integrated weights and objective/subjective attributes. *Expert Systems with Applications*, 55:37–47, 2016.

[14] M. Majumder. *Multi Criteria Decision Making*, pages 35–47. Springer Singapore, Singapore, 2015.

[15] S. Maranville. Entrepreneurship in the business curriculum. *Journal of Education for Business*, 68(1):27–31, 1992.

[16] G. Montibeller and D. Winterfeldt. Cognitive and motivational biases in decision and risk analysis. *Risk Analysis*, 35(7):1230–1251, 2015.

[17] B. Oztaysi. A decision model for information technology selection using ahp integrated topsis-grey: The case of content management systems. *Knowledge-Based Systems*, 70:44–54, 2014.

[18] J. Repschlaeger, T. Proehl, and R. Zarnekow. Cloud service management decision support: An application of ahp for provider selection of a cloud-based it service management system. *Intelligent Decision Technologies*, 8(2):95–110, 2014.

[19] A. Rodriguez, F. Ortega, and R. Concepción. An intuitionistic method for the selection of a risk management approach to information technology projects. *IS*, 375:202–218, 2017.

[20] S. Rouhani and A. Z. Ravasan. Multi-objective model for intelligence evaluation and selection of enterprise systems. *International Journal of Business Information Systems*, 20(4):397–426, 2015.

[21] T. L. Saaty. How to make a decision: the analytic hierarchy process. *European journal of operational research*, 48(1):9–26, 1990.

[22] T. L. Saaty and L. G. Vargas. *Decision making with the analytic network process*. Springer, 2006.

[23] A. Sage. *Decision support systems engineering*. Wiley series in systems engineering. J. Wiley, 1991.

[24] E. Triantaphyllou, B. Shu, S. N. Sanchez, and T. Ray. Multi-criteria decision making: an operations research approach. *Encyclopedia of electrical and electronics engineering*, 15(1998):175–186, 1998.