Software Component Selection: An Optimized Selection Criterion for Component-based Software Engineering (CBSE)

Ahmad Nabot
Department of Software Engineering, Zarqa University, Jordan
anabot@zu.edu.jo

Abstract: *Component-Based Software Engineering (CBSE) is gaining popularity in software development due to time and cost limitations. As software applications have become integral to people's lives, developing high-quality and user-friendly applications within a reasonable timeframe and budget has become increasingly challenging. Software development firms frequently use Commercial-Off-The-Shelf (COTS) components to address this challenge and reduce development costs and time. However, selecting appropriate components that meet customer requirements and integrate seamlessly with the target system is a complex task requiring considering the entire software system's quality. This study investigates the critical factors that software industry practitioners and experts must consider when selecting software components. First, the author asked practitioners to identify the most important quality criteria for an online bookstore from a list using subjective judgment and evaluation grades. Then, the study employed the Evidential Reasoning (ER) approach to tackle the multi-level evaluations and information uncertainty associated with software component selection issues. The ER approach's primary features, such as weight normalization, probability assessment, uncertainty management, and utility intervals, offer several benefits for COTS selection problems, including cost and time reduction, improved software reliability, effectiveness, and efficiency. This study assessed the quality criteria for an online bookstore using the ER approach and provided analysis results based on the approach's computational steps. Finally, the study ranked the four components according to their weights, evaluation grades, and belief degrees for selection.*

Keywords: *Multi-criteria decision making, assessment, evidential reasoning, software component, quality criteria.*

Received November 25, 2023; accepted January 30, 2024
https://doi.org/10.34028/iajit/21/2/4

1. Introduction

In recent years, software-developing organizations have relied more on reusing software components from Commercial-Off-The-Shelf (COTS), in-house developed components, and open-source components due to software applications' tremendous spread and use. Reusability increases the reliability of the reused components, reduces process risk, and ultimately lowers development and maintenance costs, as well as the time and effort required to build software to reach the market within the expected time [29, 78]. Component-Based Software Engineering (CBSE) is an approach to complex software development that relies on reusing existing software components from different environments to provide the functional and non-functional requirements of the targeted system [16]. However, selecting software components that fulfill the end user's needs is challenging because the components' quality criteria must be considered, which increases the difficulty of selecting the proper components for integration. When the available number of components that comply with the functional and non-functional requirements in the repository is large, the selection process will be complex, which might conflict with the reuse objectives, such as minimizing price and maximizing quality. Developers face many challenges when selecting the right components, such as Multi-Criteria Decision Making (MCDM) or multi-objective optimization [29]. Optimizing software requirements and development costs are essential for component selection [16, 40]. Evaluating COTS when selecting them, especially in a complex system, is complicated due to the lack of information, uncertainty, evolving COTS, and changing requirements [13, 17]. Thus, systems developed using existing COTS should consider all stages of software development, such as requirement analysis, software design, integration, and maintenance [58]. Many researchers have conducted research studies to address these challenges to solve the component selection problem. To handle incomplete or uncertain information, these studies used decision-making and integrated Artificial Intelligence (AI) techniques, such as Analytic Hierarchy Process (AHP), ANP, Coefficient of Variation (CV), Preference Ranking Organization Method for Enrichment Evaluation (PROMETHEE), and clustering-based techniques. However, these techniques have many shortages, such as the number of software requirements, the kind of software requirements, uncertain information, and many more problems. Therefore, making the selection decision crucial requires further investigation. The contribution of this study is to deploy a method for selecting software components that fit the user's needs in terms of use and quality. This study utilized the Evidential Reasoning (ER) approach to achieve this aim. ER is a MCDM approach that analyzes problems under uncertainty or incomplete information by representing them in a multi-level or multi-criteria evaluation decision matrix [18]. The ER approach is developed based on Dempster-Shafer's (D-S) theory of evidence to deal with uncertainties such as missing or incomplete information using a belief structure [71]. The hierarchy contains sets of criteria distributed on different levels. Each criterion is evaluated using a set of evaluation grades with varying degrees of belief. The ER approach has more features than other techniques that optimize decisions regarding component selection problems. To illustrate the ER approach, an online bookstore software system containing four components with the same quality criteria and different belief degrees, evaluation grades, and weights was assessed. The study provides the complete assessment process steps, implementation, and final decision results. This paper uses the ER approach to select software components that fit the user's needs in terms of use and quality. The paper's structure includes the research method in section 2, a background of related work in section 3, a description of the software component selection problem in section 4, the methodology of the ER approach in section 5, the case study in section 6, the results in section 7, a discussion of the results and research questions in section 8, limitations and challenges in section 9, and finally, the conclusion in section 10.

2. Research Method

In this study, the author followed the guidelines proposed by Kitchenham *et al*. [34], which involve three stages. The first stage involves developing a research strategy and framework for the review. The second stage involves conducting the review, which includes searching for relevant papers, selecting them, extracting data from them, and synthesizing them. In the final stage, the researchers report their findings and determine how to disseminate the review to the public. Thus, the current paper was developed based on these guidelines.

2.1. Review Planning

The main aim of this review is to offer a summary of the domain and examine the various components of the approach, including artifacts, search criteria, representation of artifacts, evaluation techniques, and outcomes utilized in the experimental process.

2.2. Research Questions

The review of this study aims to answer the research questions presented in Table 1. These questions cover all aspects of software component selection and selection methods/techniques.

Table 1. Research questions (RQs).

RQ#	Question
RQ1	What are the most important criteria for COTS selection?
RQ2	Does the ER approach assess criteria and sub-criteria of software components?
RQ3	What are the constraints of other selection methods?
RQ4	Do current selection methods deal with large-size components and incomplete information?
RQ5	Does ER approach cost and time-effective?

2.3. Search Strategy

The search terms in the search strategy are derived from the study research questions and the related studies. Figure 1 shows the search strategy terms.

("Software") AND ("Component") AND ("Selection" OR "Select") AND ("Method" OR "Technique" OR "Multi-Objective" OR "Fuzzy Logic" OR "Artificial Intelligence" OR "MCDM" OR "Functional Requirements" OR "Non-Functional Requirements" OR "Uncertainty" OR "Incomplete Information" OR "Constraints" OR "Evaluation" OR "Software Engineering" OR "Component-Based Software Engineering")

Figure 1. Search strategy terms.

2.4. Selection Criteria Related Studies

A set of inclusion and exclusion criteria was established and presented in Table 2 to identify relevant studies from various sources. The selection process began by using search terms from primary sources and search engines to narrow the search. The final decision for selection was based on reading the title and abstract. Primary sources and search engines were chosen based on their impact on publishing software engineering publications, including IEEE Xplore, Wiley Online Library, ACM digital library, Springer, Science Direct, Elsevier, and Web of Science. Figure 2 illustrates the number of studies on software component selection for 1999-2022.

Figure 2. Number of published papers in the area of software component selection.

Table 2. Inclusion and exclusion criteria.

Inclusion criteria	Exclusion criteria
All papers that use software component selection methods/techniques.	Out of Scope papers.
Multicriteria decision-making methods.	Papers were not written in English.
	Abstracts, technical reports, thesis, symposiums, books, and patents.

3. Background

3.1. Component Selection Methods/Approaches

In CBSE, selecting software components is critical to developing high-quality, cost-effective, and efficient software systems [54]. The selection process involves several activities, such as qualification, adaptation, and composition, to ensure functional and non-functional requirements are met [30]. However, selecting components can be challenging due to using Specialized Sourcing Engines (SSE) or component repositories [3]. As a result, researchers have proposed various selection methods and techniques to solve such problems. Kontio [35] presented techniques to bridge the gap between actual requirements and decision-making, as Off-The-Shelf Option (OTSO) and Procurement-Oriented Requirements Engineering (PORE) cannot handle incomplete information and criteria weighting. Kunda and Brooks [38] introduced the Social-Technical Approach to COTS Evaluation (STACE) approach to COTS evaluation, which considers non-technical factors such as business issues and reliability. Grau *et al*. [20] developed Description, evaluation, and selection of COTS components (DesCOTS) software that integrates various software component selection and evaluation tools. Maxville *et al*. [43] used AI techniques, including C4.5 and neural network classifiers, to automate component assessment for selection and evaluation, increasing the accuracy level.
Also, Gilke *et al.* [19] proposed an AI decision support system based on a fuzzy logic tool (FLT) to select the appropriate alternative when inputs are qualitative. Carvallo *et al*. [8] extended the International Organization for Standardization/International Electronical Commission (ISO/IEC) 9126-1 catalog for managing software requirements during COTS selection. Bhuta *et al*. [5] proposed an attribute-driven framework for COTS and connector selection based on interoperability and glue code. Haghpanah *et al*. [23] used genetic and greedy algorithms to reduce selection costs and time. Gashi and Popov [17] applied Bayesian assessment methods to select COTS based on probability distributions. Neubauer and Stummer [50] introduced a multi-objective approach to address challenges in COTS selection. Cortellessa *et al*. [10] proposed a framework for COTS selection in the requirement phase based on the Decision support for component-based software (DEER) optimization tool. Vijayalakshmi *et al*. [68] proposed an automated approach for component selection using a Genetic Algorithm (GA) that considers functional and non-functional requirements. Neubauer *et al*. [49] proposed a two-phase decision support for selecting COTS. The first phase determines the feasible set of COTS alternatives that satisfy set of constraints, then identify the pareto-efficient solution from this set based on the decision objectives. The second phase provide the decision makers with a mechanism to explore the solutions determined in phase 1. Finally, decision makers can investigate the different scenarios to find the appropriate set of components that meet their objectives. Jadhav and Sonar [26] compared the AHP, Weighted Scoring Method (WSM), and Hybrid Knowledge Based System (HKBS) approaches for component selection, with HKBS outperforming AHP and WSM regarding efficiency, flexibility, knowledge reuse, and consistency. Şerban *et al*. [60] proposed a new algorithm based on metrics and fuzzy clustering for component selection, while Kwong *et al*. [39] utilized a GA to select the optimal software component for small and medium-sized enterprises. Mancilla *et al*. [42] combined COSTUME and Azimut+ to classify software components based on their non-functional requirements for selection. However, the efficiency of this technique was not proven. Ibrahim *et al*. [24] proposed the Uncertainty Handling in COTS Selection (UnHOS) approach, which uses AHP and Bayesian Belief Network (BBN) for selecting and evaluating COTS. Jha *et al.* [27] proposed a fuzzy multi-objective approach for selecting software components to build a reliable and efficient software system. Rafsanjani and Rakhshan [55] proposed an approach based on the 0/1 Knapsack algorithm to reduce development costs and increase cohesion between component.

Pande *et al*. [53] proposed an integer programming-based method to maximize liability for selecting software components. However, this method was limited by not considering all quality attributes and their weights. Tomar and Gill [65] proposed an algorithm for component selection using best-fit and first-fit strategies, but it had shortcomings in terms of cost and time. Shakeel Faridi *et al*. [62] proposed an Idealized Recommendation Off-The-Shelf (IROTS) approach based on the ISO/IEC 25010 model, but the method did not consider the weights of the requirements. Mittal and Bhatia [46] proposed a framework for selecting and evaluating components based on reusability using AHP. However, the process may be expensive and time-consuming for many criteria. Faundes *et al*. [13] proposed a fuzzy decision-making system for comparing and evaluating COTS and their impact on IT organization. Kaur and Singh [32] proposed PROMETHEE as a method for component evaluation and selection, which considered some quality attributes but did not account for uncertain information. Nazir *et al*. [47] proposed an Analytic Network Process (ANP) for component selection based on several criteria; this method's features were insufficient for providing a strong decision. Khan *et al*. [33] proposed CBSE for software reusability, which considers reducing development costs and time. Kaur and Tomar [29] proposed a MOO model using pre-emptive goal programming for software component selection. However, the technique does not provide accurate results for many components. Konys [36] proposed using Ontology Web Language (OWL) and information tools ontology to enhance the component selection process. However, this approach was criticized for its high cost and complexity. Kaur and Tomar [31] presented fuzzy clustering-based algorithms for component selection. This method has limitations in identifying the number of clusters in advance and selecting the correct distance cluster. Sekar and Sethuraman [59] proposed a method for component selection based on fuzzy ranking and rough sets, while Tian *et al*. [63] proposed a method for component selection based on clustering and information entropy weighting. Gupta *et al*. [21] suggested a MOO model utilizing Data Envelopment Analysis (DEA); the model was constrained in handling information uncertainty, which led to increased execution time and reduced reliability. Similarly, Farshidi *et al*. [12] presented a Decision Support System (DSS) based on Moscow and quality models like ISO/IEC 25010 and ISO/IEC 9126 for software component selection; the approach was limited to a specific type, and the number of requirements. Verma *et al*. [67] proposed a nonlinear MOO model that used a Technique for Order Preference by Similarity to the Ideal Solution (TOPSIS) for software component selection; the method did not consider the priorities of the chosen components, leading to conflicting requirements. Kaur and Tomar [30] presented a four-tier architecture using clustering for component selection; the approach required evaluating only one component per cluster, causing longer evaluation time for many components. Nazir *et al*. [48] proposed a fuzzy logic model to evaluate the security of COTS based on ISO/IEC 18028-2; this approach has limitations in dealing with other requirements and large-size components. Rodas-Silva *et al*. [56] developed Recommender System that suggests implementation of components from selected features (RESDEC), a prototype component-based recommender system that selects suitable components based on specific features but does not address information uncertainty and requirement priorities. Padhy *et al*. [52] proposed a reusability matrix-based approach to identify reusable software components, whereas Gusev *et al*. [22] used an artificial bee colony algorithm for functional requirement-based component selection. However, the latter approach does not address information uncertainty. Garg [15] proposed a Fuzzy set theory and Modified Distance-Based Approach (FMDBA) for component selection, but it has limitations in handling large-sized components. Mehlawat *et al*. [44] developed a multi-period multi-objective optimization framework constrained to specific component types and parameters for software component selection, evaluation, and integration. Chatzipetrou *et al*. [9] identified critical factors for practitioners in selecting appropriate components using descriptive surveys and analysis techniques. Bibi *et al*. [6] proposed a hybrid approach that combines natural language processing techniques and domain knowledge for component inquiry based on specific criteria. Jabbarpour *et al*. [25] proposed an AI-based framework for component selection, but it faced issues in utilizing AI techniques due to political, social, and financial concerns. Kalantari *et al*. [28] optimized software component selection using Fuzzy-Intra Coupling Density (Fuzzy-ICD) and functionality as objective functions while considering budget, delivery time, and reliability as constraints. However, their approach does not address information uncertainty and requirement priorities. Banga and Bhatia [4] integrated a short-term memory approach with neural network mechanisms to optimize component selection but faced difficulties in dealing with uncertain information and subjective judgments. Mehta *et al*. [45] used the Fuzzy Analytic Hierarchy Process (FAHP) to calculate criteria weights and Complex Proportional Assessment of alternatives with grey Relations (COPRAS-G) for software component ranking and selection. Table 3 compares the ER approach and other approaches used for software component selection.

3.2. The Evidential Reasoning (ER) Approach

The approach of CBSE is centered around integrating various components to build a software system that satisfies the requirements of software stakeholders. However, selecting the appropriate components to meet stakeholder needs is a complex task that depends on several criteria. The ER approach is a promising method for dealing with uncertain assessments and decision-making in this context [41]. This study utilizes the ER approach for selecting and weighing criteria to rank and select the appropriate component. ER is MCDM method that uses a multi-level assessment hierarchy and an extended decision matrix with a belief structure to assess software components. This approach has been used in different research studies to solve various problems. For example, Zhang and Deng [77] and Dong *et al*. [11] applied the ER approach to analyze fault diagnosis issues in an uncertain environment. Akhoundi and Nazif [2] used ER to assess wastewater reuse alternatives. Ng and Law [51] utilized ER to investigate user preferences regarding affection words in social network websites. Tian *et al*. [64] embedded ER approach with probabilistic linguistics to solve MCDM problem, considering the decision-makers psychological preferences.

4. Problem Description

Selecting software components depending on subjective judgments to differentiate between alternatives based on quality attributes. Therefore, based on belief structure, ER utilizes an extended decision matrix to describe each criterion and its alternative sub-criterion depending on evaluation grades. For instance, the result of the evaluation grades of the quality of the software component could be described as follows:

$$H = \{H1, H2, H3, H4, H5\} = \{(Worst, Poor, Average, Good, Excellent)\} \quad (1)$$

According to Yang *et al*. [72], evaluation grades can help capture uncertainties, including subjectivity, and provide a precise structure of degrees of belief. However, it is challenging to evaluate software components directly as they are general terms. Therefore, software components should be broken down into lower-level concepts such as criteria and sub-criteria, as suggested by Fu *et al*. [14] and Xu and Yang [69]. For example, the sub-component "order item" can be evaluated in an online bookshop store using usability, performance, reliability, security, maintainability, portability, and flexibility criteria. If necessary, these criteria can be further subdivided into more detailed concepts. The lowest level of the hierarchy involves aggregating sub-criteria to be directly assessed, which refers to the sub-criteria of the system [8, 69]. According to Yang and Sen [73], assessment criteria have a multi-level structure that enables the evaluation of higher-level criteria through lower-level sub-criteria. For example, usability (y) can be measured using several evaluation factors, such as learnability, ease of use, and satisfaction, denoted by ($e_1, e_2, ..., e_n$), where (e_1) denotes learnability, (e_2) for ease of use, and (e_3) for satisfaction. Figure 3 illustrates this structure of criteria evaluation [74].

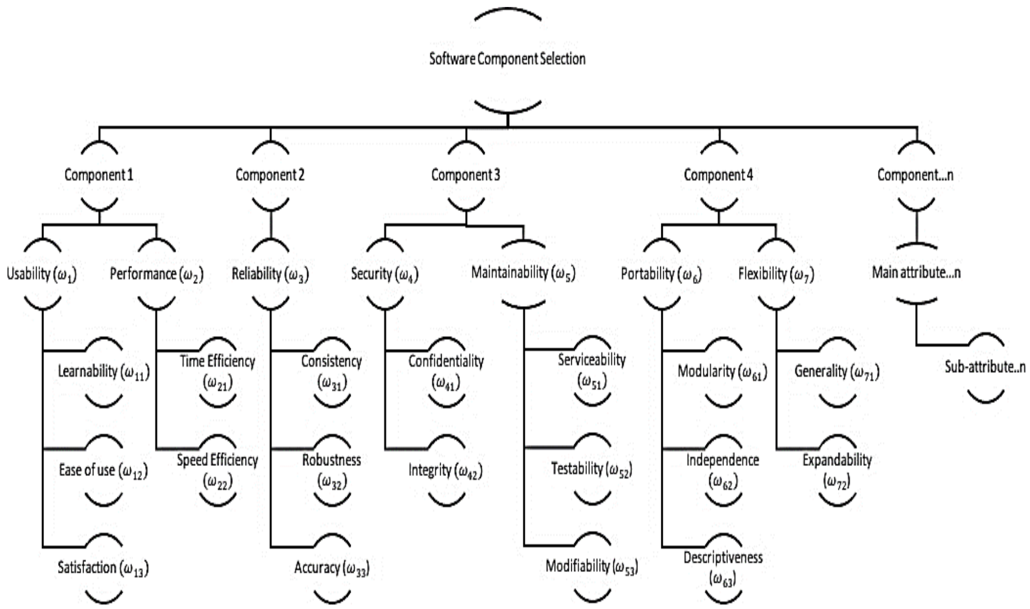

Figure 3. Evaluation hierarchy for software components.

Table 3 compares the ER approach with other approaches such as AHP, ANP, TOPSIS, CV, PROMETHEE, and clustering-based techniques. The comparison table highlights ER relative strengths and weaknesses over other techniques using factors such as uncertainty handling, computational complexity, consistency, reusability, scalability, ease of modeling, and subjectivity.

Table 3. Comparison between ER and other approaches.

Method/approach	ER	AHP	ANP	TOPSIS	CV	PROMETHEE	Clustering-based
Handling uncertainty	High	Low	Low	Low	Low	Low	Medium
Computational complexity	Medium	Low	Medium	Low	Medium	Medium	High
Consistency	High	Medium	Medium	Medium	Medium	Medium	Low
Reusability	High	Medium	Medium	Medium	Low	Medium	Low
Scalability	Medium	High	Medium	High	Medium	Medium	Low
Ease of modeling	Medium	Easy	Difficult	Easy	Difficult	Medium	Difficult
Subjectivity	Medium	High	High	High	High	High	Medium

5. Methodology

The motivation for using the ER approach is the different way of developing DSS than other MCDM approaches. In addition, it deals with the problem of having qualitative and quantitative information with subjectivity and uncertainty [79]. ER approach was developed based on several science disciplines, including statistical analysis, AI, and information technology [1, 70, 71, 73, 74, 75, 80]. Moreover, the difference between such an approach and other MCDM approaches is employing the ER algorithm to aggregate the belief degrees based on the D-S theory [1, 7, 69, 71, 73, 74, 76]. This framework is flexible in describing an MCDM problem and preventing information loss by converting two-dimensional values into one--dimensional ones through modeling. Suppose we have L basic criteria at the lower level of the hierarchy $A_i(i=1, .., L)$ associated with the general component concept y, K alternatives $O_j(j=1, ..., K)$ and N for evaluation grades $H_n(n=1, ..., N)$ for each criterion, where $SA(O)$ is given as follows:

$$S\left(A_i(O_j)\right) = \{(H_n, \beta_n, i(O_i))\}, n = (1, \ldots, N), i = (1, \ldots, L), and\ j = (1, \ldots, K) \quad (2)$$

where (β_n,$i(O\_j)$) represents the degree of belief of the alternative O_j, which is assessed by the nth grade of the *ith* criterion. Each criterion might have its evaluation grade, which could differ from other criteria in the hierarchy [73]. ER algorithm can be described by transforming the belief degree into masses where $m_{(n,i)}$ and $m_{(H,i)}$ are calculated as follows:

$$m_{n,i} = \omega_i \beta_{n,i} \quad (3)$$

$$m_{H,i} = 1 - \sum_{n-1}^{N} m_{n,i} = 1 - \omega_i \sum_{n=1}^{N} \beta_{n,i} \quad (4)$$

Suppose the weight of the *ith* criterion $m_{n,i}$ is given by $\omega=\{\omega_1, \omega_2, \omega_3, \ldots, \omega_i\}$. So, the probability mass represents the nth evaluation grade H_n of the *ith* criterion. The residual probability mass $m_{(H,i)}$ unassigned to any individual grade after assessing the *ith* criterion.

$$m_{H,i} = \bar{m}_{H,i} + \tilde{m}_{H,i}$$

for $i=1,...,L$ and $\sum_{i=1}^{L} \omega_i = 1$. Assigns the evaluation grades $H=\{H_1, H_2, ..., H_N\}$ to the probability mass and L criteria are aggregated to generate the combined belief degree for each evaluation grade H_n. The unassigned evaluation grades H_n can be calculated as follows:

$$\bar{m}_{H,i} = 1 - \omega_i \text{ and } \tilde{m}_{H,i} = \omega_i \left(1 - \sum_{n=1}^{N} \beta_{n,i}\right) \quad (5)$$

$\bar{m}_{H,i}$for calculating the relative importance of the *ith* criterion and $\tilde{m}_{H,i}$ is used for the incomplete information of the *ith* criterion.

Therefore, $m_{H,I(L)}=\bar{m}_{H,I(L)}+\tilde{m}_{(H,I(L)}$, $n=(1,...,N)$ are combined probability assignments by aggregating all original probability masses using the aggregation of the following ER algorithm [71]:

$$\{\{H_n\}: m_{n,I(i+1)} = K_{I(i+1)}\left[m_{n,I(i)} m_{n,i+1} + m_{H,I(i)} m_{n,i+1} + m_{n,I(i)} m_{H,i+1}\right] \quad (6)$$

where $n=(1,...,N)$,

$$\{H\}: m_{H,I(i)} = \bar{m}_{H,I(i)} + \tilde{m}_{H,I(i)} \quad (7)$$

$$\tilde{m}_{H,I(i+1)} = K_{I(i+1)}\left[\tilde{m}_{H,I(i)} \tilde{m}_{H,i+1} + \bar{m}_{H,I(i)} \tilde{m}_{H,i+1} + \bar{m}_{H,i+1} \tilde{m}_{H,I(i)}\right) \quad (8)$$

$$\bar{m}_{(H,I(i+1))} = K_{(I(i+1))}\left[\bar{m}_{H,I(i)} \bar{m}_{H,i+1}\right) \quad (9)$$

$$K_{I(i+1)} = \left[1 - \sum_{t=1}^{N} \sum_{\substack{i=1 \\ j \neq t}}^{N} m_{t,I(i)} m_{j,i+1}\right]^{-1} \text{Where} i = (1, 2, \ldots, L-1) \quad (10)$$

$$\{H\}: \beta_H = \frac{\tilde{m}_{H,I(L)}}{1 - \bar{m}_{H,I(L)}} \quad (11)$$

$$\{H_n\}: \beta_n = \frac{m_{n,I(L)}}{1 - \bar{m}_{H,I(L)}} n = 1, \ldots, N \quad (12)$$

where β_n is the belief degree to the L criteria which can be assessed by H_n and β_H is the residual belief degrees unassigned to any H_n, which proves that $\sum_{n=1}^{N} \beta_n + \beta_H = 1$ [71]. Therefore, the final distribution evaluation for O_j can be produced by aggregating L criteria as follows:

$$S(O_j) = \left\{\left(H_n, \beta_n(O_j)\right), n1, \ldots, N\right\} \quad (13)$$

To compute the average of $S(O_j)$for an individual output of H_n, suppose that H_n is denoted by $u(H_n)$ as follows:

$$u(O_j) = \sum_{n=1}^{N} \beta_n (O_j) u(H_n) \quad (14)$$

β_n denotes the lower bound of the probability of the evaluated alternative O_j to H_n and the upper bound can be computed by (β_n+β_H) [37, 61, 70, 72, 77, 80]. Also, in case there are uncertainties such as missing or incomplete information, they can be characterized by the maximum, minimum, and average score of S(A*) as follows:

$$u_{\max(O_j)} = \sum_{n=1}^{N-1} \beta_n (O_j) u(H_n) + \left(\beta_N(O_j) + \beta_H(O_j)\right) u \quad (15)$$

$$u_{\min(O_j)} = \left(\beta_1(O_j) + \beta_H(O_j)\right) u(H_1) + \sum_{n=2}^{N} \beta_n (O_j) u(H_n) \quad (16)$$

$$u_{avg}(O_j) = \frac{u_{\max(O_j)} + u_{\min(O_j)}}{2} \quad (17)$$

where $(H_n+1) \geq u(H_N)$.

When the assessments of $S(A_i(O_j))$ in the belief decision matrix are complete, the result of βH (O_j)=0 and $u(S(O_j))=u_{max(Oj)}=u_{min(Oj)}=u_{avg}(Oj)$. These mathematical equations are used for assessment characterization rather than criteria aggregation.

The critical distinction between the ER approach and other MCDM techniques is that ER transforms various evaluation information types to assess functional and non-functional criteria [71]. In addition, information uncertainties of the assessed criteria are treated based on D-S theory of evidence [80].

6. Real-life Case Study

The ER approach was utilized in this section to analyze the quality attributes of online bookstore software and select the best component for customer needs, with four components selected (component 1, component 2, component 3, and component 4) as depicted in Figure 3 [70, 73, 74]. While the ER approach can be employed for selecting and evaluating numerous components, this study only considers qualitative software attributes to illustrate the ER algorithm, with quantitative attributes to be included in a later study.

To validate the proposed method, a group of practitioners and experts from various software companies and academia were gathered based on their knowledge of software engineering. Due to the importance of data in COTS selection and evaluation and the limited available data, a field experts' opinion approach was chosen to collect relevant data for the COTS selection and evaluation problem. A questionnaire was used to collect primary data, including the estimated weights of COTS criteria, which was distributed into three parts. The first part contained a cover letter outlining the study's purpose. In contrast, the second part gathered demographic data for the practitioners and experts, such as organization name, expertise field, length of experience, qualification, and role. The third part consisted of weights assigned to the selection criteria/quality criteria by rating them using five evaluation grades (worst, poor, average, good, and excellent) and providing a rank for evaluation grade from 0 to 1. The weight of the selection criteria estimated using practitioners' and experts' opinions are presented in Table 4. Assessing the main software quality attributes at higher levels, such as usability, performance, reliability, security, maintainability, probability, and flexibility, can be challenging as they are considered general attributes that are difficult to evaluate directly. To overcome this challenge, lower-level attributes can be utilized to evaluate the higher-level attributes. For instance, component 1's quality attributes can be evaluated using lower-level attributes such as learnability, ease of use, and satisfaction. The attribute hierarchy for the four software components is shown in Figure 3, where ω_i, $\omega_i j$ and $\omega_i jk$ represents the relative weights of all attributes.

Thus, degrees of belief, evaluation grades, and attributes defined for this sub-criterion can be represented as follows:

- Learnability=(Average, 0.4), (Good, 0.5).
- Ease of use=(Good, 1.0).
- Satisfaction=(Good, 0.4), (Excellent, 0.6).

The assessment of software quality attributes can be complete or incomplete based on the value of the belief degree of the sub-criteria. If the belief degree value equals one, the assessment is complete. On the other hand, if the belief degree value is less than one, the assessment is incomplete. For instance, the assessment of learnability may result in a value of 0.9, which is incomplete, while the ease of use and satisfaction assessment is complete. It is worth noting that only grades with degrees of belief greater than zero are included in the distributions [73].

Table 4 shows the subjective assessment for usability sub-criteria, including their degrees of belief. In the table, the letters W, P, A, G, and E represent Worst, Poor, Average, Good, and Excellent, respectively. The number in brackets indicates the degree of belief to which an attribute is evaluated. For instance, G(0.5) denotes "good to a degree of 0.5 (50%)." The overall usability assessment is generated by aggregating several sub-criteria using the ER framework, as depicted in Figure 3. Additionally, the ER framework provides a method for dealing with the aggregation problem.

Table 4. Subjective judgments for evaluating software system usability.

Degree of belief (β)		Evaluation grade				
		Worst	Poor	Average	Good	Excellent
Weight	Learnability			0.4	0.5	
	Ease of use				1.0	
	Satisfaction				0.4	0.6

The original judgments in Table 4 need to be aggregated to select the component with the highest usability quality. This aggregation requires assigning the relevant importance of the other three components to generate an accurate assessment. There are various methods for weight assessment, as documented in the literature [66]. This study will use the ER approach to address the assessment problem.

Table 4 summarizes the subjective evaluations for the qualitative attributes using degrees of belief for each component in the hierarchy. Experienced professionals in software development assessed these sub-criteria based on their relevance to the software.

7. Results

Hypothetical weights of the ER are used to aggregate assessments for components' quality attributes. The aggregated sub-criteria weights are considered to generate the assessment for all criteria, as shown in Figure 3. From Equation (3), we have the belief degrees for usability sub-criteria on component 1 as follows:

$\beta_{1,1}=0$, $\beta_{2,1}=0$, $\beta_{3,1}=0.4$, $\beta_{4,1}=0.5$, $\beta_{5,1}=0$
$\beta_{1,2}=0$, $\beta_{2,2}=0$, $\beta_{3,2}=0$, $\beta_{4,2}=1.0$, $\beta_{5,2}=0$
$\beta_{1,3}=0$, $\beta_{2,3}=0$, $\beta_{3,3}=0$, $\beta_{4,3}=0.4$, $\beta_{5,3}=0.6$

To calculate the basic probability masses $m_{n,i}$ when the three quality criteria are equally important $\omega_1=\omega_2=\omega_3=3$, we use Equations (2) and (3) as follows:

$m_{1,1}=0$, $m_{2,1}=0$, $m_{3,1}=0.4/3$, $m_{4,1}=0.5/3$
$m_{5,1} = 0,\ \bar{m}_{H,1} = 2/3, \tilde{m}_{H,1} = 0.1/3$
$m_{1,2}=0$, $m_{2,2}=0$, $m_{3,2}=0$, $m_{4,2}=1/3$
$m_{5,2} = \bar{m}_{H,2} = \frac{2}{3}, = \bar{m}_{H,2} = 0$
$m_{1,3}=0$, $m_{2,3}=0$, $m_{3,3}=0$, $m_{4,3}=0.4/3$
$m_{5,3} = 0.6/3, \bar{m}_{H,3} = 2/3, \tilde{m}_{H,3} = 0$

To calculate the combined probability masses, we can use Equations (6), (7), (8), (9), and (10). Thus, suppose $m_{n,I(1)} = m_{n,1}$ for n=1,…,5. The results are calculated using the Intelligent Decision System (IDS) software developed [57, 73] to implement the ER approach and generate the necessary calculations graphically for component selection, as shown in Figure 4.

Figure 4 presents the assessment outcomes produced by the IDS software, demonstrating the combined evaluations of the upper-level criteria for the four components based on their evaluation grades and belief degrees based on Equations (11) and (12). This evaluation method considers the weight of each criterion and sub-criterion and ranks them accordingly. Subsequently, the discrepancies between the four components can be identified and ranked to facilitate the selection process. For instance, the following is an example of the aggregated assessments for the upper-level criteria of component 1 based on Equation (13):

S(Usability=S(Learnability$\oplus$Ease of use $\oplus$Satisfaction)=(average, 0.03), (good, 0.73), (Excellent, 0.23).

S(Performance)=(average, 0.36),(good, 0.55), (excellent, 0.09).

S(Reliability)=(average,0.2), (good, 0.52), (excellent, 0.2).

S(Security)=(average, 0.33), (good, 0.67).

S(Maintainability)=(good, 0.38), (excellent, 0.58)

S(Probability)=(worst, 0.27), (poor, 0.18), (average,0.13), (good, 0.34), (excellent, 0.04).

S(Flexibility)=(average,0.34), (good, 0.63).

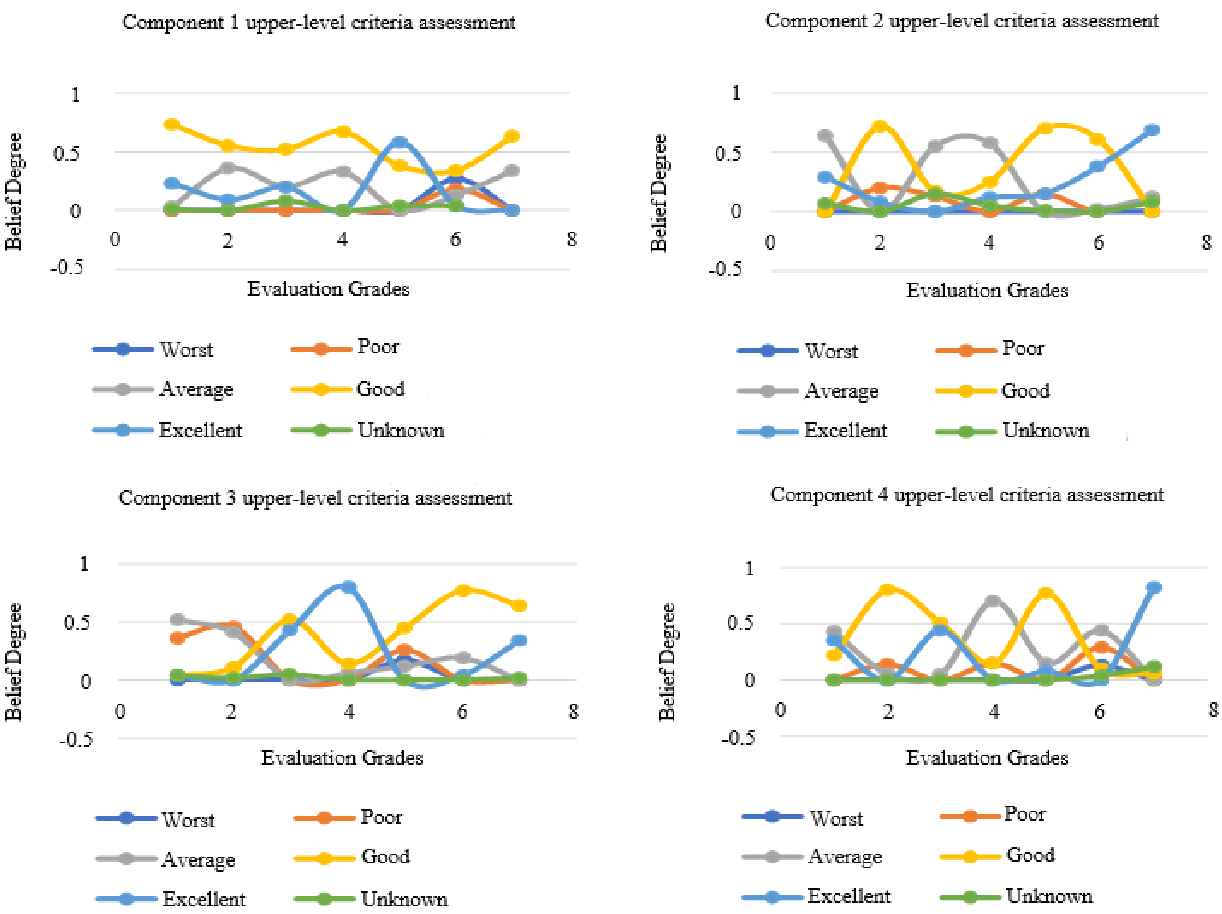

Figure 4. Component's upper-level criteria assessment.

Table 5. Quality attributes' degrees of belief distribution for each component.

Criteria	Sub-criteria	Component			
		Component 1	Component 2	Component 3	Component 4
Usability (ω_1)	Learnability (ω_{11})	G (0.4), A (0.5)	A (1.0)	G (0.5), E (0.5)	E (1.0)
	Flexibility (ω_{12})	G (1.0)	A (0.3), E (0.7)	A (0.4), P (0.6)	G (0.5), E (0.5)
	Ease of use (ω_{13})	G (0.4), E (0.6)	A (0.8)	P (0.4), A (0.5)	A (1.0)
Efficiency (ω_2)	Time efficiency (ω_{21})	G (0.5), E (0.5)	P (1.0)	A (0.3), G (0.6)	P (0.7), A (0.3)
	Speed efficiency (ω_{22})	A (0.5), G (0.5)	G (0.9), E (0.1)	P (0.6), A (0.4)	G (1.0)
Satisfaction (ω_3)	Trust (ω_{31})	E (0,3), G (0.7)	A (0.8), G (0.2)	G (0.5), E (0.4)	A (0.6), G (0.4)
	Pleasure (ω_{32})	G (0.5), A (0.4)	P (0.3), A (6)	G (1.0)	G (0.5), E (0.5)
Security (ω_4)	Risk management(ω_{41})	G (1.0)	E (0.7)	A (0.3), G (0.7)	A (0.2), G (0.8)
	Economic risk (ω_{42})	A (0.5), G (0.5)	A (0.7), G (0.3)	E (1.0)	P (0.2), A (0.8)
Compatibility (ω_5)	Compatibility (ω_{51})	G (0.4), E (0.6)	G (0.9)	A (0.3), G (0.7)	E (1.0)
Portability (ω_6)	Portability(ω_{61})	G(0.6), E(0.4)	A (0.2), G (0.8)	G (0.5), E (0.5)	G (1.0)
Efficiency (ω_7)	Effectiveness(ω_{71})	A (0.2), G (0.6)	P (0.5), A (0.5)	G (0.4), E (0.5)	E (1.0)

As shown in Figure 3 and Table 5, the assessment problem for the four components based on the criteria and sub-criteria assessment information arises. Therefore, the relative weights of criteria at a specific level are associated with the same upper-level criteria and defined by ω_i, ω_i j, and ω_i jk the criteria at the other levels in the hierarchy. Thus, to demonstrate the ER algorithm without generality loss, all criteria are considered equally important to ensure the assessment reliability, where ω_i is the relative weight of the ith criterion (A_i) with 0≤Ai≤1. Suppose we have L essential criteria at the lower level of the hierarchy A_i (i=1, …, L) associated with the general component concept Y. Weights are given by $\omega=\omega_1, \omega_2, \dots, \omega_i$ as follows:

$$\omega_1=\omega_2=\omega_3=\omega_4=\omega_5=\omega_6=\omega_7=0.142$$
$$\omega_{11}=\omega_{12}=\omega_{13}=0.3333$$
$$\omega_{12}=\omega_{22}=0.5$$
$$\omega_{31}=\omega_{32}=0.5$$
$$\omega_{41}=\omega_{42}=0.5$$
$$\omega_{51}=1.0$$
$$\omega_{61}=1.0$$
$$\omega_{71}=1.0$$

After generating the aggregated assessment for the criteria of component 1, the final assessment for component 1 is generated as follows:

(Componet1)=(worst, 0.037), (poor, 0.0249), (average, 0.209), (good, 0.567), (excellent, 0.135), (H, 0.0258).

The degree of incompleteness *H* in the evaluation of component 1 is 0.0258 due to incomplete assessment for the sub-criteria for component1 as shown in Table 5; the incompleteness degree is 10%, which is reduced due to the large number of complete assessments in other sub-criteria.

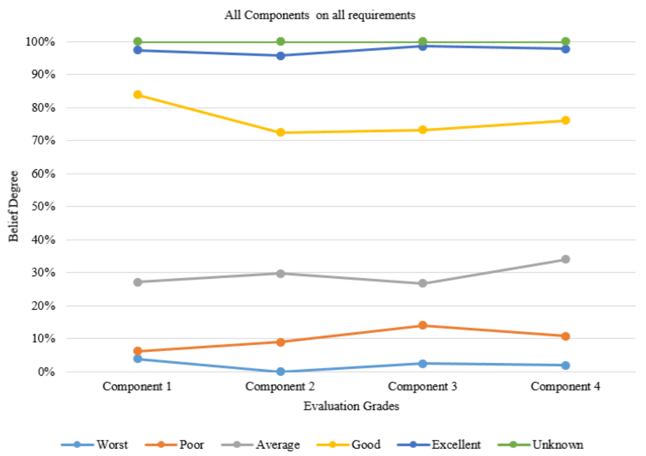

Figure 5. Components' final ranking.

Figure 5 displays the final assessment results for the four components, allowing for partial ranking and selection. The components differ in weight, evaluation grades, and belief degrees across criteria and sub-criteria. For example, the average assessments or the four components are 0.684, 0.701, 0.702, and 0.681, respectively. Based on these results, the four components can be ranked using larger and smaller assessment degrees, with component 3 preferred over component 2 and component 1 preferred over component 4. However, to achieve a precise ranking of the four components, it is necessary to estimate the utilities of their five evaluation grades. Normalization of the utility grades is required in this case. Assuming a utility value of 0 for the worst grade and 1 for the best grade, the following can be derived:

- $u(H_1)=u(poor)=0$, $u(H_5)=u(excellent)=1$

Depending on the probability method, the utilities of the grades can be estimated by offering two hypothetical options to the decision-maker to select the suitable component. The first option offers a component with an average grade assessment. In contrast, the second option offers one component with a poor grade assessment with a probability of $1-p$ and another with an excellent

Grade assessment with a probability of p. The probability ($0\leq p\leq 1$) is regulated until the decision-maker cannot distinguish between both options. Thus, suppose the decision-maker is tardy for both options when considering the value of $p=0.55$; the utility for the first option is calculated by

- $u(H_3)=(average)=(1-p)\times u(poor)+p\times u(excellent)$ $=0.45\times 0+0.55\times 1=0.55$.

Similarly, the utilities of the worst and good grades might be estimated by supposing that $u(H_2)=u(worst)=0.35$ and $(H_4)=(good)=0.85$. Then, belief degrees for component 1 are given as follows:

- $\beta_1=0.037$, $\beta_2=0.0249$, $\beta_3=0.209$, $\beta_4=0.567$, $\beta_5=0.135$, $\beta_H=0.0258$, where β_H denotes the degree of incompleteness.

Since $\beta_H\neq 0$, component 1 assessment is not complete and should be characterized by the utility interval $[um(component1), umax(component1)]$ depending on Equations (14), (15), (16), and (17) which are implemented graphically as shown in Figure 6 uses IDS software to show the final ranking results for the four components using their utility intervals.

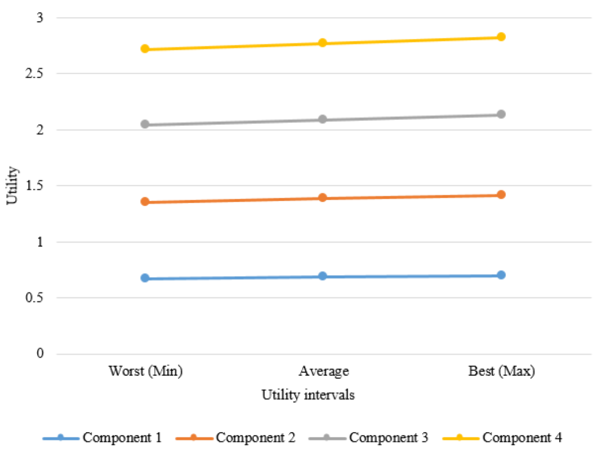

Figure 6. Components' final ranking using utility intervals.

Figure 6 shows the minimum and maximum values for the four components to provide more precise assessment results and select the component with the highest-ranking quality criteria. Thus, the four components' assessment results indicate the ranking for the four Components are as follows:

Component 3 ≻ Component 2 ≻ Component 1 ≻ Component 4

Where ≻ "better than" indicates the best component to be selected. This ranking is generated based on the identical weights for all criteria in the hierarchy, as shown in Table 5.

8. Discussion

The primary objective of this study was to address the COTS selection problem for an online bookstore. The study utilized MCDM to model the problem of component evaluation and selection for the case study. The alternative COTS were evaluated and ranked based on criteria and sub-criteria weights, using belief structure and an extended decision matrix to show their evaluation grades. The study applied the main features of the ER approach, such as weight normalization, information uncertainty, probability assessment, and utility intervals, to assist decision-makers in selecting the appropriate set of COTS. To reduce the complexity of the selection process, criteria and sub-criteria of the quality attributes for the online bookstore were aggregated for assessment. Belief degrees and evaluation grades for the sub-criteria were defined by field practitioners based on their subjective assessment and the importance of criteria for such systems. Sub-criteria weights were then set to ensure the reliability of the assessment, which addressed RQ1. Uncertain information was assessed to calculate the incompleteness of all sub-criteria for each component, which was then used for partial ranking and selection. The primary and combined probability masses were calculated using an IDS based on mathematical equations. The results were based on the belief degrees and evaluation grades for upper-level criteria. Partially, COTS was ranked based on the generated assessment results, which addressed RQ2. After weight normalization, utility intervals were applied for a precise assessment to enable decision-makers to make the final selection decision. The ER approach's features improve COTS selection based on features, reducing development costs and delivery time, and dealing with large-size COTS. This increases the developed software's reliability, functionality, and reusability. Various researchers have proposed COTS selection methods, such as AHP, CV, PROMETHEE, fuzzy methods, and MOO models. However, most of these methods have limitations in the selection process, as stated in section 3. In contrast, the utilized approach in this study assessed functional and non-functional criteria and dealt with information uncertainty using the D-S theory of evidence. The ER approach used an extended decision matrix to aggregate all component criteria, which reduced the selection process complexity, addressing RQ3, RQ4, and RQ5 of this study. The proposed approach had not been previously used for COTS selection and could serve as a benchmark method to rapidly solve such component selection problems. Compared with other proposed approaches for solving the COTS selection problem, the ER approach was better in weight assigning, computational efficiency, problem-solving efficiency, reusability, and consistency and presentation of COTS selection results.

9. Limitations and Challenges

As with any method to solve a recurring problem, the ER approach has some limitations in practical settings. These limitations include the complexity of defining criteria weights that could impact the results and dealing with evolving requirements, which can be difficult as the criteria models in ER are relatively static. Furthermore, scalability is a big issue for substantial and complex software components, as many criteria and components lead to substantial computational requirements. Finally, subjectivity in assessments and weights leaves room for bias where more expert inputs are required.

10. Conclusions

Selecting the best software component that fits customer needs reduces the cost and time of software development and improves software quality, which are essential factors for the software development industry. However, the success of the selection process of software components based on its quality criteria requires a reliable method. In this study, the ER approach is applied to solve the problem of selecting a software component based on its quality criteria, including both functional and non-functional quality criteria. ER approach is one of the MCDM methods for decision-making. It has an extended decision matrix and belief structure for the criteria of each component and its alternatives after aggregating these criteria. The ER framework establishes a nonlinear relationship between the aggregated and essential measures in the hierarchy. Can incomplete information be handled to increase the quality of the data analysis? The ER approach helps improve subjective judgments and provides precise degrees of belief structure, improving consistency and reliability in the analysis process and enhancing decisions6. The decision matrices are used to lead to transparent and high-quality choices. The assessment of software components could be done depending on the criteria and original evaluation information related to the sub-criteria of these components. The criteria weights play an essential role in the reliability indication of the assessments. The case study dealt with software component selection problems for an online bookstore to show that the ER approach can solve such issues. Moreover, the ER approach could be applied to different domains such as enterprise systems (i.e., Customer Relationship Management (CRM) and Enterprise Resource Planning (ERP)), embedded systems (i.e., IOT and industrial controls), mobile applications (i.e., maps and payment services), and web-based applications (i.e., APIs and frameworks). However, the criteria and weights would be tailored to each domain's requirements, but the overall ER approach remains applicable due to its hierarchy flexibility. The proposed approach in this study significantly contributes to the body of knowledge in assisting decision-makers in the software COTS evaluation and selection process, which can improve software reliability and effectiveness, minimize development cost and delivery time, and maximize software quality.

Acknowledgment

This research is funded (fully/partially) by Zarqa University- Jordan.

References

[1] Ahmadzadeh F., "Multi Criteria Decision Making with Evidential Reasoning under Uncertainty," *in Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management*, Bali, pp. 1534-1538, 2016. DOI:10.1109/IEEM.2016.7798134

[2] Akhoundi A. and Nazif S., "Sustainability Assessment of Wastewater Reuse Alternatives Using the Evidential Reasoning Approach," *Journal of Cleaner Production*, vol. 195, pp. 1350-1376, 2018. https://doi.org/10.1016/j.jclepro.2018.05.220

[3] Ayala C., Hauge O., Conradi R., Franch X., and Li J., "Selection of Third Party Software in Off-The-Shelf-Based Software Development-An Interview Study with Industrial Practitioners," *Journal of Systems and Software*, vol. 84, no. 4, pp. 620-637, 2011. https://doi.org/10.1016/j.jss.2010.10.019

[4] Banga A. and Bhatia P., "Optimized Component Based Selection Using LSTM Model by Integrating Hybrid MVO-PSO Soft Computing Technique," *Advances in Science, Technology and Engineering Systems Journal*, vol. 6, no. 4, pp. 62-71, 2021. DOI:10.25046/aj060408

[5] Bhuta J., Mattmann C., Medvidovic N., and Boehm B., "A Framework for the Assessment and Selection of Software Components and Connectors in COTS-Based Architectures," *in Proceedings of the Working IEEE/IFIP Conference on Software Architecture*, Mumbai, pp. 1-6, 2007. DOI:10.1109/WICSA.2007.2

[6] Bibi N., Rana T., Qurat-Ul-Ain., and Naseer A., "Conceptual Model for Component Selection: A Research Review on Existing Techniques," *in Proceedings of the 3rd International Conference on Communication Technologies*, Rawalpindi, pp. 22-27, 2021. DOI:10.1109/ComTech52583.2021.9616711

[7] Buchanan B. and Shortliffe E., *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley, 1984. https://www.shortliffe.net/Buchanan-Shortliffe-1984/Chapter-13.pdf

[8] Carvallo J., Franch X., and Quer C., "Managing Non-Technical Requirements in COTS Components Selection," *in Proceedings of the 14th IEEE International Requirements Engineering Conference*, Minneapolis-Saint Paul, pp. 323-326, 2006. https://ieeexplore.ieee.org/abstract/document/1704087

[9] Chatzipetrou P., Papatheocharous E., Wnuk K., Borg M., Alégroth E., and Gorschek T., "Component Attributes and their Importance in Decisions and Component Selection," *Software Quality Journal*, vol. 28, no. 2, pp. 567-593, 2020. https://link.springer.com/article/10.1007/s11219-019-09465-2

[10] Cortellessa V., Crnkovic I., Marinelli F., and Potena P., "Experimenting the Automated Selection of COTS Components Based on Cost and System Requirements," *Journal of Universal Computer Science*, vol. 14, no. 8, pp. 1228-1255, 2008. DOI:10.3217/jucs-014-08-1228

[11] Dong Y., Zhang J., Li Z., Hu Y., and Deng Y., "Combination of Evidential Sensor Reports with Distance Function and Belief Entropy in Fault Diagnosis," *International Journal of Computers, Communications and Control*, vol. 14, no. 3, pp. 329-343, 2019. DOI:10.15837/ijccc.2019.3.3589

[12] Farshidi S., Jansen S., De Jong R., and Brinkkemper S., "A Decision Support System for Software Technology Selection," *Journal of Decision Systems*, vol. 27, no. S1, pp. 98-110, 2018. https://doi.org/10.1080/12460125.2018.146482

[13] Faundes M., Astudillo H., and Hitpass B., "Process-Based Evaluation and Comparison of OTS Software Alternatives," *in Proceedings of the Federated Conference on Computer Science and Information Systems*, Krakow, pp. 1093-1100, 2013. https://ieeexplore.ieee.org/abstract/document/6644151

[14] Fu C., Xue M., Chang W., Xu D., and Yang S., "An Evidential Reasoning Approach Based on Risk Attitude and Criterion Reliability,"

Knowledge-Based Systems, vol. 199, pp. 105947, 2020. https://doi.org/10.1016/j.knosys.2020.105947

[15] Garg R., "A Ranking Model for the Selection and Ranking of Commercial Off-The-Shelf Components," *IEEE Transactions on Engineering Management*, vol. 69, no. 5, pp. 2196-2204, 2022. DOI: 10.1109/TEM.2020.3001943

[16] Garg R., Sharma R., and Sharma K., "Ranking and Selection of Commercial Off-The-Shelf Using Fuzzy Distance Based Approach," *Decision Science Letters*, vol. 5, no. 2, pp. 201-210, 2016. DOI: 10.5267/j.dsl.2015.12.004

[17] Gashi I. and Popov P., "Uncertainty Explicit Assessment of Off-The-Shelf Software: Selection of an Optimal Diverse Pair," *in Proceedings of the 6th International IEEE Conference on Commercial-Off-The-Shelf-Based Software Systems*, Banff, pp. 93-102, 2007. DOI:10.1109/ICCBSS.2007.44

[18] Gholamshahi S. and Hasheminejad S., "Software Component Identification and Selection: A Research Review," *Software Practice Exprience*, vol. 49, no. 1, pp. 40-69, 2019. https://doi.org/10.1002/spe.2656

[19] Gilke N., Mantha S., and Thampi G., "Fuzzy Logic based Decision Support System for Mass Customization," *International Journal of Computer Applications*, vol. 62, no. 7, pp. 31-37, 2013. https://research.ijcaonline.org/volume62/number7/pxc3884738.pdf

[20] Grau G., Carvallo J., Franch X., and Quer C., "DesCOTS: A Software System for Selecting COTS Components," *in Proceedings of the 30th EUROMICRO Conference*, Rennes, pp. 118-126, 2004. DOI:10.1109/EURMIC.2004.1333363

[21] Gupta P., Mehlawat M., and Mahajan D., "Data Envelopment Analysis Based Multi-Objective Optimization Model for Evaluation and Selection of Software Components under Optimal Redundancy," *Annals of Operations Research*, vol. 312, no. 1, pp. 193-216, 2022. DOI:10.1007/s10479-018-2842-y

[22] Gusev A., Ilin D., and Nikulchev E., "The Dataset of the Experimental Evaluation of Software Components for Application Design Selection Directed by the Artificial Bee Colony Algorithm," *Data*, vol. 5, no. 3, pp. 1-11, 2020. DOI:10.3390/data5030059

[23] Haghpanah N., Moaven S., Habibi J., Kargar M., and Yeganeh S., "Approximation Algorithms for Software Component Selection Problem," *in Proceedings of the Asia-Pacific Software Engineering Conference*, pp. 159-166, Nagoya, 2007. DOI:10.1109/ASPEC.2007.38

[24] Ibrahim H., Elamy A., Far B., and Eberlein A., "UnHOS : A Method for Uncertainty Handling in Commercial Off-The-Shelf (COTS) Selection," *International Journal of Energy, Information and Communications*, vol. 2, no. 3, pp. 21-48, 2011. https://gvpress.com/journals/IJEIC/vol2_no3/3.pdf

[25] Jabbarpour M., Saghiri A., and Sookhak M., "A Framework for Component Selection Considering Dark Sides of Artificial Intelligence: A Case Study on Autonomous Vehicle," *Electronics*, vol. 10, no. 4, pp. 1-24, 2021. https://doi.org/10.3390/electronics10040384

[26] Jadhav A. and Sonar R., "Analytic Hierarchy Process (AHP), Weighted Scoring Method (WSM), and Hybrid Knowledge Based System (HKBS) for Software Selection: A Comparative Study," *in Proceedings of the 2nd International Conference on Emerging Trends in Engineering and Technology*, Nagpur, pp. 991-997, 2009. DOI:10.1109/ICETET.2009.33

[27] Jha P., Bali S., and Kumar U., "A Fuzzy Approach for Optimal Selection of COTS Components for Modular Software System under Consensus Recovery Block Scheme Incorporating Execution Time," *Turkish Journal of Fuzzy Systems*, vol. 2, no. 1, pp. 45-63, 2011. https://scholar.google.com/citations?user=P-exLsMAAAAJ&hl=en

[28] Kalantari S., Motameni H., Akbari E., and Rabbani M., "Optimal Components Selection Based on Fuzzy-Intra Coupling Density for Component-Based Software Systems under Build-or-Buy Scheme," *Complex and Intelligent Systems*, vol. 7, no. 6, pp. 3111-3134, 2021. https://link.springer.com/article/10.1007/s40747-021-00449-z

[29] Kaur J. and Tomar P., "Multi Objective Optimization Model using Preemptive Goal Programming for Software Component Selection," *International Journal of Information Technology and Computer Science*, vol. 7, no. 9, pp. 31-37, 2015. https://www.mecs-press.org/ijitcs/ijitcs-v7-n9/v7n9-5.html

[30] Kaur J. and Tomar P., "Clustering Based Architecture for Software Component Selection," *International Journal of Modern Education and Computer Science*, vol. 10, no. 8, pp. 33-40, 2018. https://www.mecs-press.org/ijmecs/ijmecs-v10-n8/v10n8-4.html

[31] Kaur J. and Tomar P., "A Software Component Selection Technique Based on Fuzzy Clustering," *in Proceedings of the 1st India International Conference on Information Proceeding*, Delhi, pp. 1-5, 2017. DOI:10.1109/IICIP.2016.7975350

[32] Kaur K. and Singh H., "Quantifying COTS Components Selection Using Multi Criteria Decision Analysis Method PROMETHEE," *Global Journal of Computer Science and* Technology, vol. 14, no. 2, pp. 53-58, 2014.

https://globaljournals.org/item/3185-quantifying-cots-components-selection-using-multi-criteria-decision-analysis-method-promethee

[33] Khan A., Khan K., Amir M., and Khan M., "A Component-Based Framework for Software Reusability," *International Journal of Software Engineering and its Applications*, vol. 8, no. 10, pp. 13-24, 2014. DOI:10.14257/ijseia.2014.8.10.0

[34] Kitchenham B., Budgen D., and Brereton P., *Evidence-Based Software Engineering and Systematic Reviews*, Chapman and Hall/CRC, 2015. DOI: https://doi.org/10.1201/b19467

[35] Kontio J., OTSO: A Systematic Process for Reusable Software Component Selection, Technical Reports, University of Maryland, 1995. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=379cf62668b3b182570196871c10afe20d53fa2c

[36] Konys A., *Advances in Intelligent Systems and Computing*, Springer, 2015. https://doi.org/10.1007/978-3-319-15147-2_17

[37] Krause P., "Approximate Reasoning Models by Ramon Lopez de Mántaras, Ellis Horwood, Chichester, 1990, pp 109.-Search, inference and Dependencies in Artificial Intelligence by Murray Shanahan and Richard Southwick, Ellis Horwood, Chichester, 1989, pp 140.," *The Knowledge Engineering Review*, vol. 6, no. 3, pp. 239-242, 1991. https://doi.org/10.1017/S0269888900005841

[38] Kunda D. and Brooks L., "Applying Social-Technical Approach for Cots Selection," *in Proceedings of the 4th UKAIS Conference*, York, pp. 552-565, 1999. https://www.researchgate.net/publication/2441283

[39] Kwong C., Mu L., Tang J., and Luo X., "Optimization of Software Components Selection for Component-Based Software System Development," *Computers and Industrial Engineering*, vol. 58, no. 4, pp. 618-624, 2010. https://doi.org/10.1016/j.cie.2010.01.003

[40] Li J., Asif Ullah., Li J., Nazir S., Khan H., Rehman H., and Amin Ul Haq., "Attributes-Based Decision Making for Selection of Requirement Elicitation Techniques Using the Analytic Network Process," *Mathematical Problems in Engineering*, vol. 2020, pp. 1-13, 2020. https://doi.org/10.1155/2020/2156023

[41] Liao H., Ren Z., and Fang R., "A Deng-Entropy-Based Evidential Reasoning Approach for Multi-Expert Multi-Criterion Decision-Making with Uncertainty," *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 1281-1294, 2020. DOI:10.2991/ijcis.d.200814.001

[42] Mancilla F., Astudillo H., and Visconti M., "Combining COSTUME and Azimut+ to Address Functional and Non-Functional Requirements in Software Component Selection," *in Proceedings of the 29th International Conference of the Chilean Computer Science Society, SCCC*, Antofagasta, pp. 102-109, 2010. DOI:10.1109/SCCC.2010.47

[43] Maxville V., Armarego J., and Lam C., "Intelligent Component Selection," *in Proceedings of the 28th Annual International Computer Software and Applications Conference*, Hong Kong, pp. 244-249, 2004. DOI:10.1109/CMPSAC.2004.1342839

[44] Mehlawat M., Gupta P., and Mahajan D., "A Multi-Period Multi-Objective Optimization Framework for Software Enhancement and Component Evaluation, Selection and Integration," *Information Sciences*, vol. 523, pp. 91-110, 2020. https://doi.org/10.1016/j.ins.2020.02.076

[45] Mehta P., Tandon A., and Sharma H., *Springer Series in Reliability Engineering*, Springer, 2022. https://doi.org/10.1007/978-3-030-78919-0_12

[46] Mittal S. and Bhatia P., "Framework for Evaluating and Ranking the Reusability of COTS Components Based upon Analytical Hierarchy Process," *International Journal of Innovations in Engineering and Technology*, vol. 2, no. 4, pp. 352-360, 2013.

[47] Nazir S., Anwar S., Khan S., Shahzad S., Ali M., "Software Component Selection Based on Quality Criteria Using the Analytic Network Process," *Abstract and Applied Analysis*, vol. 2014, pp. 1-14, 2014. https://doi.org/10.1155/2014/535970

[48] Nazir S., Shahzad S., Mahfooz1 S., and Nazir M., "Fuzzy Logic Based Decision Support System for Component Security Evaluation," *The International Arab Journal of Information Technology*, vol. 15, no. 2, pp. 224-231, 2018. https://www.iajit.org/PDF/March%202018,%20No.%202/10649.pdf

[49] Neubauer T., Pichler J., and Stummer C., "A Case Study on the Multicriteria Selection of Software Components," *in Proceedings of the 3rd IEEE Asia-Pacific Services Computing* Conference, Yilan, pp. 1005-1012, 2008. DOI:10.1109/APSCC.2008.138

[50] Neubauer T. and Stummer C., "Interactive Decision Support for Multiobjective COTS Selection," *in Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, Hawaii, pp. 1-8, 2007. DOI:10.1109/HICSS.2007.283.

[51] Ng C. and Law K., "Investigating Consumer Preferences on Product Designs by Analyzing Opinions from Social Networks Using Evidential Reasoning," *Computers and Industrial Engineering*, vol. 139, pp. 106180, 2020. https://doi.org/10.1016/j.cie.2019.106180

[52] Padhy N., Panigrahi R., and Satapathy S.,

"Identifying the Reusable Components from Component-Based System: Proposed Metrics and Model," *in Proceedings of the 5th International Conference on Information Systems Design and Intelligent Applications India*, Mauritius, pp. 89-99, 2018. https://library.villanova.edu/Find/Record/2176583/Description?sid=144216912#tabnav

[53] Pande J., Garcia C., and Pant D., "Optimal Component Selection for Component Based Software Development Using Pliability Metric," *ACM SIGSOFT Software Engineering Notes*, vol. 38, no. 1, pp. 1-6, 2013. https://doi.org/10.1145/2413038.2413044

[54] Petersen K., Badampudi D., Shah S., Wnuk K., Gorschek T., and Papatheocharous E., "Choosing Component Origins for Software Intensive Systems: In-House, COTS, OSS or Outsourcing?-A Case Survey," *IEEE Transactions on Software Engineering*, vol. 44, no. 3, pp. 237-261, 2018. DOI:10.1109/TSE.2017.2677909

[55] Rafsanjani M. and Rakhshan N., "Using a 0/1 Knapsack Algorithm for Software Components Selection in Component-based Software System Development," *Journal of American Science*, vol. 7, no. 12, pp. 641- 648, 2011.

[56] Rodas-Silva J., Galindo J., García-Gutiérrez J., and Benavides D., "RESDEC: Online Management Tool for Implementation Components Selection in Software Product Lines Using Recommender Systems," *in Proceedings of the 23rd International Systems and Software Product Line Conference*, Paris, pp. 33-36, 2019. https://doi.org/10.1145/3307630.3342390

[57] Ruhe G., "Intelligent Support for Selection of COTS Products," *in Proceedings of the Web, Web-Services, and Database Systems NODe-Web and Database-Related Workshops*, Erfurt, pp. 34-45, 2002. https://doi.org/10.1007/3-540-36560-5_3

[58] Saaty T., *Group Decision Making and the AHP*, Springer, 1989. https://doi.org/10.1007/978-3-642-50244-6_4

[59] Sekar K. and Sethuraman J., "Optimal Component Selection for Rich Internet Applications in Web Engineering," *in Proceedings of the International Conference on Networks and Advances in Computational Technologies*, Thiruvananthapuram, pp. 420-425, 2017. DOI:10.1109/NETACT.2017.8076808

[60] Şerban C., Vescan A., and Pop H., "A New Component Selection Algorithm Based on Metrics and Fuzzy Clustering Analysis," *in Proceedings of the 4th International Conference on Hybrid Artificial Intelligence Systems*, Salamanca, pp. 621-628, 2009. https://doi.org/10.1007/978-3-642-02319-4_75

[61] Shafer G., *A Mathematical Theory of Evidence*, Princeton University Press, 2020. https://www.jstor.org/stable/j.ctv10vm1qb

[62] Shakeel Faridi N., Javed Z., Haris Abid M., Ahmed M., and Bin Ngadi M., "IROTS: A Proposed COTS Evaluation a Selection Methodology for Component Based Software Engineering in Under-Development Countries," *in Proceedings of the 2nd International Conference on Advances in Computer Science and Engineering*, Los Angeles, pp. 20-23, 2013. DOI:10.2991/cse.2013.6

[63] Tian S., Wang H., Jiang Y., and Chen X., "A New Software Component Selection Method Based on Clustering and Information Entropy Weighting," *Computer and Digital Engineering*, vol. 45, no. 12, pp. 2437-2441, 2017. https://caod.oriprobe.com/issues/1774056/toc.htm

[64] Tian Z., Nie R., and Wang J., "Probabilistic Linguistic Multi-Criteria Decision Making Based on Evidential Reasoning and Combined Ranking Methods Considering Decision-Makers' Psychological Preferences," *Journal of the Operational Research Society*, vol. 71, no. 5, pp. 700-717, 2020. https://doi.org/10.1080/01605682.2019.1632752

[65] Tomar P. and Gill N., "New Algorithm for Component Selection to Develop Component-Based Software with X Model," *Lecture Notes on Software Engineering*, vol. 1, no. 3, pp. 298-302, 2013. DOI: 10.7763/lnse.2013.v1.65

[66] Tzeng G. and Huang J., *Multiple Attribute Decision Making, Methods and Applications*, CRC press, 2011. https://doi.org/10.1201/b11032

[67] Verma S., Mehlawat M., and Mahajan D., "Software Component Evaluation and Selection Using TOPSIS and Fuzzy Interactive Approach Under Multiple Applications Development," *Annals of Operations Research*, vol. 312, no. 1, pp. 441-471, 2022. DOI:10.1007/s10479-018-3022-9

[68] Vijayalakshmi K., Ramaraj N., and Amuthakkannan R., "Improvement of Component Selection Process Using Genetic Algorithm for Component-Based Software Development," *International Journal of Information Systems and Change Management*, vol. 3, no. 1, pp. 63-80, 2008. DOI: 10.1504/IJISCM.2008.019289

[69] Xu L. and Yang J., Introduction to Multi-Criteria Decision Making and the Evidential Reasoning Approach, Working Paper, Manchester School of Management, 2001. https://personalpages.manchester.ac.uk/staff/jian-bo.yang/JB%20Yang%20Book_Chapters/XuYang_MSM_WorkingPaperFinal.pdf

[70] Yang J., "Rule and Utility Based Evidential Reasoning Approach for Multiattribute Decision Analysis under Uncertainties," *European Journal of Operational Research*, vol. 131, no. 1, pp. 31-61, 2001. https://doi.org/10.1016/S0377-

2217(99)00441-5

[71] Yang J. and Xu D., "On the Evidential Reasoning Algorithm for Multiple Attribute Decision Analysis under Uncertainty," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 32, no. 3, pp. 289-304, 2002. DOI:10.1109/TSMCA.2002.802746

[72] Yang J., Liu J., Wang J., Sii H., and Wang H., "Belief Rule-Base Inference Methodology Using the Evidential Reasoning Approach-RIMER," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 36, no. 2, pp. 266-285, 2006. DOI:10.1109/TSMCA.2005.851270

[73] Yang J. and Sen P., "A General Multi-Level Evaluation Process for Hybrid MADM with Uncertainty," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 10, pp. 1458-1473, 1994. DOI:10.1109/21.310529

[74] Yang J. and Singh M., "An Evidential Reasoning Approach for Multiple-Attribute Decision Making with Uncertainty," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 1, pp. 1-18, 1994. DOI:10.1109/21.259681

[75] Yang J. and Xu D., "Nonlinear Information Aggregation via Evidential Reasoning in Multiattribute Decision Analysis under Uncertainty," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 32, no. 3, pp. 376-393, 2002. https://personalpages.manchester.ac.uk/staff/jian-bo.yang/JB%20Yang%20Journal_Papers/ER-Aggregation-IEEE.pdf

[76] Yen J., *Studies in Fuzziness and Soft Computing*, Springer, 2008. https://doi.org/10.1007/978-3-540-44792-4_21

[77] Zhang H. and Deng Y., "Engine Fault Diagnosis Based on Sensor Data Fusion Considering Information Quality and Evidence Theory," *Advances in Mechanical Engineering*, vol. 10, no. 11, pp. 1-10, 2018. https://doi.org/10.1177/16878140188091

[78] Zhang J., Nazir S., Huang A., and Alharbi A., "Multicriteria Decision and Machine Learning Algorithms for Component Security Evaluation: Library-Based Overview," *Security and Communication Networks*, vol. 2020, 2020. https://doi.org/10.1155/2020/8886877

[79] Zhang X., Wang Y., Chen S., and Chen L., "An Evidential Reasoning Based Approach for GDM with Uncertain Preference Ordinals," *Journal of Intelligent and Fuzzy Systems*, vol. 37, no. 6, pp. 8357-8369, 2019. DOI:10.3233/JIFS-190915

[80] Zhang Z., Yang J., and Xu D., "A Hierarchical Analysis Model for Multiobjective Decisionmaking," *IFAC Proceedings Volumes*, vol. 22, no. 12, pp. 13-18, 1989. https://doi.org/10.1016/S1474-6670(17)53048-7

Ahmad Nabot earned his BSc in Computer Information Systems (CIS) from Al-zaytoonah University, Amman, Jordan, in 2007. Subsequently, he pursued an MSc in Information Systems at DePaul University, Chicago, USA, completing it in 2009. He attained his PhD from Brunel University, London, UK 2014. He is an assistant professor in the Software Engineering Department at Zarqa University, Zarqa, Jordan.