A Fuzzy Ontology-Based Decision Tool for Concept Selection to Maintain Consistency Throughout Design Iterations

Yan Liu
Department of Computer Science, College of Mathematics and Computer Science, Shantou University, Shantou 515063, China
e-mail: yanliu@stu.edu.cn

Xinru Chen
Department of Computer Science, College of Mathematics and Computer Science, Shantou University, Shantou 515063, China
e-mail: 21xrchen@stu.edu.cn

Claudia Eckert
School of Engineering and Innovation, The Open University, Walton Hall, Milton Keynes MK7 6AA, UK
e-mail: claudia.eckert@open.ac.uk

Xin Zhang
School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China
e-mail: zhangxin@cust.edu.cn

Keywords: concept selection, fuzzy ontology, TOPSIS, knowledge reuse, decision-making, design iterations

1 Introduction

At the beginning of a product development process, companies generate multiple product concepts in response to customer needs. How well the chosen concept meets these needs greatly impacts the quality of the final product and its development process [1]. This makes concept selection, as an engineering design decision problem, one of the most important activities in new product development processes [2]. Product development processes are iterative including exploration, concretization, refinement, and incremental completion, but also correction of mistakes and rework as assumptions or requirements change [3]. In response, concept selection is also iterative as concepts are modified or new concepts are generated. New criteria could also be introduced due to changing requirements [4,5] and designers becoming aware of new contextual factors [6]. In each iteration, one or more concepts are selected for further investigation or development [7,8]. It is important to maintain consistency in the criteria and constraints as far as possible and have transparency in the decision process.

The literature on concept selection focuses largely on single decisions by developing mathematical models for scoring concepts with given criteria, for example, the works by Ayag [9], Hayat et al. [10], Gironimo and Grazioso [11], Jing et al. [12], Qi et al. [13], and Song et al. [14]. It neglects the iterative nature of concept selection and misses the potential to reuse decision chunks to reduce the effort in setting up decision-making problems.

This article proposes a fuzzy ontology-based decision tool for concept selection, aiming to answer two questions:

- RQ1: How to support the reuse of decision elements for concept selection in the design process?
- RQ2: How to facilitate the reuse of decision elements across selection iterations in the design process?

The proposed tool allows designers to choose which criteria, concepts, associated values, and calculation method they intend to reuse from previous iterations. It also supports modifications on these elements or introducing new ones. This maintains consistency in decisions between iteration cycles, offers transparency, and reduces the effort in using a decision-making tool. Section 2 explains the principle behind the tool using an example from the literature on designing golfing machines. Section 3 presents the tool and develops the example further. Section 4 discusses the benefits of concept selection in engineering design. This article is concluded in Sec. 5.

2 The Scheme of the Proposed Tool

Figure 1 presents the framework of the tool. The ontology model describes the elements responsible for articulating concept selection decisions. The computing engine implements the algorithms for scoring the concepts. The engine is fully automatic but allows interaction if the user wants to modify the data.

2.1 The Fuzzy Ontology Model. To allow reuse, the knowledge needs to be captured and represented in an understandable and interoperable manner. This can be done using ontologies, which are explicit specifications of a shared conceptualization, describing features, attributes, and restrictions [15,16]. The
ontology models vary depending on the definitions of domain problems. Concept selection is formalized ontologically in engineering design, for example, by Shah et al. [17], Gosnell and Miller [18], Ranjan et al. [19], and Siddharth, et al. [20]. They explicitly define the measurement elements and outcomes such as quality, novelty, and creativity and propose equations to calculate the scores for each concept. These provide a useful starting point for designers and, however, do not reflect the need to reuse these decision elements across different decision cycles and are not presented in a way that lends itself to be used in computer tools.

This article develops an ontology model for concept selection, which defines the decision elements, their relations, and the properties of elements. It uses the formal terms of a generalized ontology and is implemented in the standard Web Ontology Language OWL. This captures information for concept selection at the computational level that can be easily processed by computers and implemented in a tool.

- **Classes**: Collection of instances that share common properties and provide conceptual description of the domain knowledge scope.
- **Individuals**: Specific instances within a class, representing a particular object in the domain.
- **Data properties**: Attributes or characteristics of individuals.
- **Object properties**: Relationships or connections between classes or individuals.

For concept selection, concepts are scored by combining the weights of criteria and the performance values. The one with the top score is considered the best choice. To represent the selection process, six key decision elements are extracted as classes, and the ontology model is designed as shown in Fig. 2.

1. **Concept**: The concept of an alternative solution that meets the problem definition and fulfills the specifications.
2. **Criterion**: A specification or requirement that a concept should meet.
3. **Performance**: How well a concept satisfies the evaluation criterion under consideration.
4. **Preference**: The judgment from an expert on the importance of a criterion.
5. **Ranking**: The calculated results including the score and the rank of a concept.
6. **Weight**: The calculated priorities of a criterion.

In the ontology model, DecisionKnowledge links all the classes, while Result gathers the calculated results. The semantic relationships between classes are captured by object properties. The classes are connected to data via data properties. The definitions of the properties are presented in the Appendix. In each iteration, designers can select existing instances under these classes or create new ones and then retrigger the computing engine.

During the selection, there are two types of data: objective values, such as "price" and subjective judgments, such as "easy to manufacture." Linguistic terms (e.g., high, medium, and low) describe how concepts perform under subjective criteria. To address this uncertainty, the ontology model incorporates triangular fuzzy numbers (TFNs), a special type of fuzzy sets. A classical fuzzy set consists of two components, a set of elements x, and an associated membership function that assigns to each element a value between 0 and 1 as the degree to which it belongs to the set. For a TFN, membership function is defined as shown in Fig. 3(a), which is usually expressed as a triple \((l, m, h)\). Introducing fuzzy sets into ontology gives rises

![Fig. 1 The framework of the proposed tool](https://example.com/fig1.png)

![Fig. 2 The ontology model for concept selection](https://example.com/fig2.png)

![Fig. 3 TFN: (a) membership function and (b) an example expression in fuzzy OWL 2](https://example.com/fig3.png)
to fuzzy ontology [21]. Although there is not a standard for fuzzy ontology representation, fuzzy OWL 2 proposed by Bobillo and Straccia [22] is a convenient extension to standard Web Ontology Language OWL. Figure 3(b) shows an example of expressing a TFN. In this way, the proposed tool can well support the use of OWL editors such as Protégé.2

Wrapping up decision elements into classes has been suggested by Liu and Stewart [23] to support the reuse of knowledge in decision analysis. However, their object-oriented methodology only considers aggregation and inheritance, which are not enough to capture the relations between the decision elements in concept selection, for example, the relation between the class of design concept and the class of performance.

2.2 The Computing Engine. To fill the values of weight and ranking instances in the ontologies, the computing engine parses the data from the associated Concept, Criteria, Performance and Preference instances, calculates the criteria weights, and scores the concept. Weighted fuzzy TOPSIS is used to provide a practical and straightforward way of calculation. TOPSIS is a compromise decision method proposed by Hwang and Yoon [24]. Its principle is that the chosen alternative should have the shortest distance to the positive ideal solution (PIS) and the farthest distance to the negative ideal solution (NIS). The working process is shown in Fig. 4 and explained with an example of designing remote-controlled golfing machine from Kosky et al. [25]. This example is further developed in Sec. 3. Three concepts (Original, Cannon, and Robogolfer) are assessed by four criteria (Drives well, Putts well, Loader is robust, and Easy to transport).

Phase I: Calculating the weights of the selection criteria of a DecisionKnowledge instance as follows:

- **Extract criteria and preference values.** All the Criterion instances are extracted by tracing the object property hasCriterion from a DecisionKnowledge instance. Their importance judgments are stored under the data property preJudgement of Preference, which are linked to Criterion via object property owningPreference. Let \( C = \{c_1, c_2, \ldots, c_i\} \) be the set of Criterion instances and \( P = \{p_1, p_2, \ldots, p_t\} \) be the set of their corresponding preferences. \( i \) is the number of the Criterion instances. Each \( p_k = (l_i, m_i, h_i) \) stands for a linguistic term expressed by a TFN. Table 1 lists the mappings between linguistic terms and TFNs for importance used in this work.

- **Normalize the fuzzy numbers.** To limit the values within the unit interval, the fuzzy numbers are normalized to obtain the fuzzy weights for each Criterion instance by Eq. (1) [26,27].

\[
f_{wi} = (l_wi, m_wi, h_wi)
\]

\[
f_{wi} = \left( \frac{l_i}{l_i + \frac{m_i}{\sum_{k=1}^{n} h_k}} \right.
\]

For example, let the TFNs for criteria Drives well, Putts well, Loader is robust and Easy to transport be (4,5,6), (3,4,5), (4,5,6) and (1,1,2), respectively. Using Eq. (1), the fuzzy weights of Drives well is calculated as follows:

\[
f_{w_{Drives\ well}} = \left( \frac{4}{4 + 5 + 6 + 2} \right) \approx (0.2353, 0.3333, 0.4286)
\]

- **Defuzzify the normalized fuzzy numbers.** The fuzzy weights are first defuzzified into crisp values \( w_i \) and then normalized for crisp weights \( w_c \), allowing for an intuitive comparison on the
importance ranking of the criteria. The crisp weights are obtained based on the centroid method in Eq. (2).

\[
\begin{align*}
    c_{wi} &= \frac{l_{wi} + 2m_{wi} + h_{wi}}{4} \\
    w_i &= \frac{c_{wi}}{\sum_{m=1}^{n} c_{m}}
\end{align*}
\]

(2)

With the example, the defuzzified weight of Drives well is calculated as \( c_{Drives\_well} = 0.2353 + 2 \times 0.3333 + 0.4286 \times 4 = 0.3326 \). Its crisp weight \( w_{Drives\_well} = 0.3326(0.3326 + 0.2667 + 0.3326 + 0.0857) = 0.3269 \), where 0.2667, 0.3326, and 0.0857 are the defuzzified weights of the other three criteria.

- The fuzzy and the crisp weights correspond to the data properties wgtFuzzyValue and wgtCrispValue of a Weight instance. They are attached to a Criterion instance via the object property belongingToCriWgt between Criterion and Weight.

**Phase II: Scoring the concepts.** To deal with the fuzzy datatype in the ontologies, fuzzy TOPSIS is utilized, which replaces the crisp values in traditional TOPSIS with fuzzy numbers for calculation [28]. The steps are as follows:

- **Extract concepts and their performance.** All the concepts are extracted following the object property hasConcepts from a DecisionKnowledge instance. Their performance under each criterion can be looked up according to the object properties ownerPerformance and pfsUnderCri. Let \( A = \{a_1, a_2, \ldots, a_n\} \) be the set of Concept instances, where \( n \) is the number of the Criterion instances. Their performance set is organized as a matrix in the next step.

- **Establish the performance matrix.** The matrix is denoted by \( E = [e_{ij}]_{n \times n} \), where \( e_{ij} \) is the Performance instance of Concept instance \( i \) under Criterion instance \( j \). Each \( e_{ij}= (l_{ij}, \bar{m}_{ij}, u_{ij}) \) stands for a linguistic term expressed by a TFN, Table 2 shows the mappings between linguistic terms and TFNs for performance.

- **Normalize the matrix.** The performance matrix \( E \) is normalized by Eq. (3), where \( r_{ij} \) is the normalized fuzzy performance value. \( B \) stands for the benefit criteria that a higher value is expected like “easy to manufacture,” whereas \( C \) stands for the cost criteria that a lower is expected such as price. This attribute of a criterion is defined as data property criExpectation of the ontology.

\[
r_{ij} = \begin{cases} 
    \left( \frac{l_{ij}}{u_{ij}}, \frac{u_{ij}}{l_{ij}}, \frac{u_{ij}}{u_{ij}}, \frac{u_{ij}}{u_{ij}}, \frac{u_{ij}}{u_{ij}} \right) & j \in B \\
    \left( l_{ij}, \frac{l_{ij}}{u_{ij}}, \frac{l_{ij}}{u_{ij}}, \frac{l_{ij}}{u_{ij}}, \frac{l_{ij}}{u_{ij}} \right) & j \in C 
\end{cases}
\]

(3)

where \( u_{ij} = \max u_j \) or a predefined maximum boundary value if \( j \in B \), \( l_{ij} = \min l_j \) or a predefined minimum boundary value if \( j \in C \). Take the three concepts Original, Cannon, and Robogolfer, for example. Let their performance under Drives well be (6, 7, 8), (7, 8, 8), and (6, 7, 8) respectively. The normalization of the entry of concept Original against criterion Drives well, \( r_{Original\_Drives\_well} = (6/8, 7/8, 8/8) \) because this is a benefit criterion and 8 is the maximum value.

- **Construct weighted normalized matrix.** The fuzzy weights of the Criterion instances are incorporated into the normalized performance matrix. Let \( W = (w_{f1}, w_{f2}, \ldots, w_{fi}) \) be the weight vector. \( v_{ij} \) is the weighted normalized value obtained by Eq. (4).

\[
v_{ij} = w_{j} \times v_{ij}
\]

(4)

The weighted normalized entry of Original against Drives well, \( v_{Original\_Drives\_well} = (0.2353, 0.3333, 0.4286) \times (6/8, 7/8, 8/8) = (0.1765, 0.2917, 0.4286) \).

- **Determine the positive and negative ideal solutions (PIS and NIS).** PIS and NIS describe the best and the worst concepts, working as references to judge the candidate concepts. They are represented by the two vectors that contain, respectively, the best and the worst performance under the criteria. They correspond to the largest and the smallest ideal values in the weighted normalized matrix as shown in Eq. (5) or are the predefined desired level and worst level, respectively.

\[
\begin{align*}
    \text{PIS} &= (v_1^+, v_2^+, \ldots, v_n^+) = \left\{ \max v_{ij} | i = 1, 2, \ldots, n, j = 1, 2, \ldots, t \right\} \\
    \text{NIS} &= (v_1^-, v_2^-, \ldots, v_n^-) = \left\{ \max v_{ij} | i = 1, 2, \ldots, n, j = 1, 2, \ldots, t \right\}
\end{align*}
\]

(5)

In the example, after normalization, all the values are bounded by (0, 0, 0) and (1, 1, 1), and thus, the two boundaries combined with the weights of Drives well, Puts well, Loader is robust, and Easy to transport can be used as PIS and NIS, i.e., \( \text{PIS} = ((0.2353, 0.3333, 0.4286), (0.1765, 0.2667, 0.3571), (0.2353, 0.3333, 0.4286), (0.0556, 0.0667, 0.1538)) \), and \( \text{NIS} = ((0, 0, 0), (0, 0, 0), (0, 0, 0)) \).

- **Calculate the distances to the PIS and NIS.** For each Concept instance, Eq. (6) calculates the distances to the PIS and NIS, denoted by \( d_i^+ \) and \( d_i^- \), respectively.

\[
\begin{align*}
    d_i^+ &= \sum_{j=1}^{m} d(v_{ij}, v_{ij}^+) \quad i = 1, 2, \ldots, n \\
    d_i^- &= \sum_{j=1}^{m} d(v_{ij}, v_{ij}^-) \quad i = 1, 2, \ldots, n \\
    d(v_1, v_2) &= \frac{1}{3} \left[ (l_{v_1} - l_{v_2})^2 + (m_{v_1} - m_{v_2})^2 + (h_{v_1} - h_{v_2})^2 \right]
\end{align*}
\]

(6)

The distance to the PIS of Original is calculated as follows:
Compute the final scores. The two distances $d_i^+$ and $d_i^-$ are aggregated to generate a final value by Eq. (7). The Concept instances are ranked in a descending order according to the scores.

$$s_i = \frac{d_i^-}{d_i^+ + d_i^-}$$ (7)

The final score of Original $S_{\text{Original}} = 0.7283/(0.3622 + 0.7283)$, where 0.7283 is its distance to NIS.

The score and the rank correspond to the data properties score and rank of a Ranking instance. They are further connected to a Concept instance via the object property belongingToCptRnk between Concept and Ranking.

The calculation so far reflects one decision maker. Multiple decision makers can be considered by carrying out the aforementioned calculation independently and then aggregating their scores. Alternatively, their judgments can be aggregated first before a single calculation process is carried out. Arithmetic or geometric mean can be used for the aggregation, as Eq. (8), where $q$ is the number of decision makers, $s_i$ and $(l_t, m_t, h_t)$ are the crisp and fuzzy judgment values, respectively, of decision maker $t$.

Arithmetic mean for crisp value:

$$\frac{1}{q} \sum_{t \in T} s_t$$

For fuzzy value:

$$\left( \frac{1}{q} \sum_{t \in T} l_t \right)^\frac{1}{q}, \left( \frac{1}{q} \sum_{t \in T} m_t \right)^\frac{1}{q}, \left( \frac{1}{q} \sum_{t \in T} h_t \right)^\frac{1}{q}$$ (8)

Geometric mean for crisp value:

$$\prod_{t \in T} s_t^{\frac{1}{q}}$$

For fuzzy value:

$$\left( \prod_{t \in T} l_t^{\frac{1}{q}} \right)^\frac{1}{q}, \left( \prod_{t \in T} m_t^{\frac{1}{q}} \right)^\frac{1}{q}, \left( \prod_{t \in T} h_t^{\frac{1}{q}} \right)^\frac{1}{q}$$

2.3 Reuse Steps With the Tool. The tool supports the reuse of a particular concept selection problem in three scenarios as illustrated in Fig. 5.

- **Brand new selection problem.** The designers start with creating a new DecisionKnowledge instance and then Concept instances. When defining the criteria, the designers could first look up in the instance repository to see whether existing Criterion instances meet their requirements, which could be linked to this new problem via object property hasCriteria. Otherwise, new Criterion instances need to be populated. Preference and Performance instances are then created by recording the linguistic judgments. All the instances are linked via the properties as defined in the ontology (see Fig. 2). To compare the concepts, the designers can invoke the engine, which automatically generates the weights of criteria and the scores and rankings of the concepts (i.e., Weights and Ranking instances).

- **Changes in a particular problem.** When there are changes to the concepts, the criteria, or the judgments on the performance or

<table>
<thead>
<tr>
<th>Criteria and Weight</th>
<th>Drives well</th>
<th>Puts well</th>
<th>Ball loader is robust</th>
<th>Easy to transport</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept</td>
<td>Important</td>
<td>Medium low important</td>
<td>Important</td>
<td>Extremely low important</td>
</tr>
<tr>
<td>Cannon</td>
<td>Extremely high</td>
<td>Medium low</td>
<td>Medium high</td>
<td>Extremely high</td>
</tr>
<tr>
<td>Original</td>
<td>Very high</td>
<td>Medium low</td>
<td>Medium high</td>
<td>Medium</td>
</tr>
<tr>
<td>Robogolfer</td>
<td>Very high</td>
<td>Extremely high</td>
<td>Extremely high</td>
<td>Medium</td>
</tr>
</tbody>
</table>

Table 3 Data for concept selection, adapted from Kosky et al. [25]

Fig. 5 Reuse scenarios and steps

Fig. 6 Concept drawing of (a) the Cannon, (b) the Original, and (c) Robogolfer from Kosky et al. [25] (Figures 25.6—25.8).
the weights in a problem, the designers first look for the Decision-Knowledge instance corresponding to the problem and edit them by adding or deleting instances or modifying the values. Once the problem is revised, the engine is invoked for comparison.

New iteration on a problem. Most parts of a selection problem could stay the same in the following iteration. The designers create a new DecisionKnowledge instance, find, and link the instances involved in the previous iterations and fills in the gaps.

Both changes and new iteration reuse the existing solution chunks; however, the changes reuse the entire instance set of a selection, whereas new iteration creates a new DecisionKnowledge
Fig. 10  Results of introducing a new criterion

Fig. 11  Results of introducing a new concept
instance, which helps track the changes and compare the results among iterations.

3 Application of the Tool

This section illustrates reusability of the decision chunks using the example published by Kosky et al. [25]. The example is adapted to cover the three scenarios.

3.1 Concept Selection for Remote-Controlled Golfing Machine. The RC portable device must play nine holes of golf at a local golf course with the fewest possible number of strokes. Its functions include driving, chipping, putting, loading balls, and transporting balls. Three concepts as shown in Fig. 6 have been generated that fulfill the design requirements. A comparison will be carried out to select the best concept based on the data in Table 3. Four subjective criteria translated from a customer requirement list are used for evaluation: drives well, puts well, ball loader is robust, and is easy to transport.

Phase 0—Create knowledge instance. The knowledge instances are created according to the fuzzy ontology model. The three concepts, four criteria, and their subjective judgments are modeled first as outlined in Fig. 7 and saved as an OWL file. The designers can either use Protégé or our software tool—CSelector.3

Phase I: Calculating the weights. The tool extracts the instances from the OWL file, classifies the values as shown in Fig. 8, and calculates the weights using Eqs. (1) and (2). The results are highlighted in the red box.

The properties of the criteria and the concepts can be viewed and edited using the tool. Any changes will be written back to the OWL file for further iteration.

Phase II: Scoring the concepts. Using Eqs. (3)–(7), the final results are generated as shown in Fig. 9. Robogolfer is of the highest score (i.e., 0.8786) followed by Cannon (0.7255) and Original (0.6679).

3.2 Introducing New Criteria. A new criterion “easy to manufacture” is introduced, which affects development time. The three concepts perform differently against this criterion. Because the selection in the previous iteration has been recorded in the form of ontology instances, in this iteration, the designers only need to add a new Criterion instance easy_to_manufacture with its two data properties criDescription and criExpectation and then link it.

Fig. 12 Checking reusable instances

3The software tool can be obtained upon reasonable request.
to a Preference instance. In the previous example, a Preference instance \( M \) that describes “important” has been created for criteria \( \text{drives} \_\text{well} \) and \( \text{loader} \_\text{is} \_\text{robust} \). Thus, the designers can reuse it for the object property owningPreference from \( \text{easy} \_\text{to} \_\text{manufacture} \). Adding a new criterion requires new judgments on the performance of the concepts, so three new Performance instances, \( \text{pfsC} \_\text{etm} \), \( \text{pfsO} \_\text{etm} \), and \( \text{pfsR} \_\text{etm} \), and their properties are created according to the ontology model.

With the added data, the tool updates the results. Figure 10 shows that the criteria weights have been redistributed and the ranking has moved Original to the second place. This makes sense because of the good performance of Original under this newly introduced important criterion.

### 3.3 Introducing New Alternative Concept

In the previous selection, Robogolfer has the highest score; however, manufacturing the robot arm of the Robogolfer is very complicated. A new concept OriginalUpdate is introduced by replacing the putter of Original with Robogolfer’s linear spring putter. This leads to a new iteration of the selection. All the five criteria and the three concepts (and their judgments as well) are passed to this iteration. The designers only need to create a new Concept instance, add associated Performance instances, and link them to the DecisionKnowledge instance. The tool receives the updated ontology file and generates the ranking results, where OriginalUpdate is ranked top (see Fig. 11). The criteria weights and the order of Robogolfer, Original, and Cannon, still stay the same because no changes have been made on their judgments.

### 4 Discussion

#### 4.1 Identification of Reusable Decision Instances

The proposed ontology model describes the key decision elements in concept selection illustrating that criteria, concepts, their judgments, and results generated for one iteration can be reused in another iteration. The designers can further look up particular reusable instances via the instance of class DecisionKnowledge, check the properties, and then determine if reusing them or not. This allows transparency between iteration loops. As illustrated in the golfing machine example (see Fig. 12), the designers trace the four criteria and the three concepts at the beginning of the iteration of introducing new criteria, which have been further reused in the iteration of introducing new concept. After, the designers apply the calculation to compare concepts using the existing and modified criteria and weights.

#### 4.2 Reuse Across Decision Problems

The decision models can also be applied to support different problems with overlapping criteria, as all the instances in different concept selection problems are connected via the semantic relationships defined in the ontology model. This helps designers to setup a new concept selection problem through following the semantic relations between the instances. For example, when designing a golf ball collector where three criteria \( \text{stand} \_\text{well} \), \( \text{light} \_\text{weight} \), and \( \text{easy} \_\text{to} \_\text{transport} \) are considered for three concepts \( \text{HandPusher} \), \( \text{HandPicker} \), and \( \text{RoboPicker} \), other criteria might also be suitable. To explore more available criteria, a new selection problem is created in Protegé according to the proposed template and stored in the ontology instance repository, named \( \text{SltBallCollector} \). All the ontology instances form a knowledge graph for selection as shown in Fig. 13. With the graph, the designers will be guided to the RC golfing machine selection via the criterion \( \text{easy} \_\text{to} \_\text{manufacture} \), which is common in both selection problems. By looking at the criteria also connected to the same concept, the designers could be prompted to include \( \text{easy} \_\text{to} \_\text{manufacture} \). This example also reflects the reusability of the proposed selection template and a single element of a decision (i.e., Criterion). As more selection
problem instances are added, more reusable information becomes available.

4.3 Adaptation of the Proposed Ontology. The proposed ontology was designed for concept selection so that the data properties of a concept record the target product, function, and design principle. The ontology includes common-sense decision elements relevant to selection decision problems. It can be modified or extended by changing the definitions of the elements and their properties, adding new elements or removing existing ones. For example, if a design requirement "probability of jamming" is identified, it can be considered as a new criterion for assessing the concepts for the RC golfing machine. The value of a concept against this criterion is percentage rather than subjective judgment. To adapt the ontology, the data type "fuzzy number" of Performance via data property pfsJudgement is changed to "float."

People may also use different terms in their problems. For example, when evaluating the novelty of designs, Shah et al. [17] use the terms "idea" and "attribute." In this case, Concept and Criterion can be replaced by Idea and Attribute as illustrated in Fig. 14(a). Preference is ignored because the weights of attributes are predefined instead of being calculated. Instances can then be created according to the ontology. Figure 14(b) illustrates part of instances related to one idea Entry #1 (highlighted in purple), including the instance NoveltyCalculation for class DecisionKnowledge, four attribute instances, their weights, and performance.

5 Conclusion and Future Work

Existing concept selection methods are mainly based on multiple-criteria decision-making (MCDM) methods like analytic hierarchy process (AHP) or analytic network process (ANP). Although new ideas can be updated with these methods, only the calculation methods are reused. The designers have to reset the problem and reorganize the criteria and the concepts. In this work, the proposed ontology wraps up six decision elements into classes, enabling the tool to model concept selection in a reusable manner and allowing the designers to identify the reusable decision instances. This answers RQ1: how to support the reuse of decision elements for concept selection in the design process. Across iterative selection cycles, the tool supports designers to select from the existing criteria, concepts, and values according to their particular scenarios. It maintains consistency and enhances decision transparency. This answers RQ2: how to facilitate the reuse of decision elements across selection iterations in the design process. This reusability feature is particularly beneficial when dealing with similar decision-making scenarios or when modifications are required to existing decision criteria. With the implemented calculation methods, the tool further allows for easy modification and expansion of decision criteria and concepts. The flexibility in reusing and customizing the decision elements and the transparency of the ranking process could also encourage the designers to use the tool in that a more detailed and straightforward selection process could gain more trust from the decision makers [29].

The illustrative example shows the applicability of the proposed tool. To further verify the effectiveness of the tool, experiments in real-world decision-making scenarios will be conducted. Feedback from designers will be gathered in terms of usefulness, ease of use, and the tool’s impact on decision-making outcomes. We intend to recruit 15 students in year 3 or 4 from mechanical design. They will first be given the proposed tool to evaluate their designs in the three scenarios in Sec. 4 (stage 1) and then be provided an MCDM method for the same evaluation (stage 2). Their user experience and consumed time in both stages will be recorded and compared. For a better validation, we would like to conduct a case study in companies for expert designers’ opinions toward the tool.

This work takes TOPSIS as the fundamental scoring method. However, the introduction or removal of alternatives could change the alternative’s order of preference, which is known as rank reversal in decision-making. This exists not only in TOPSIS but also other MCDM methods like AHP and ELECTRE. Using extreme values for normalization and to represent PIS and NIS is an effective solution to the rank reversal problem in TOPSIS [30,31]. This particularly suits the scenario where all the criteria have the same range of values. For instance, the same scale is used for the judgments against the criteria in our example, and thus, the lower and upper boundaries can be used. Otherwise, the decision makers need to redefine appropriate boundaries based on additional information [32] or some information expansion algorithm [33]. An interesting piece of work regarding this problem is to introduce a loop during the calculation procedure, where the user is asked whether they want to change the PIS and NIS. This would make the decision less comparable, but it might be possible to add visualization to complement the results that show the differentiation in priorities.

Many MCDM methods can be used to build various concept selection models, such as AHP, simple additive weighting (SAW), TOPSIS, and ELECTRE. AHP stands out, because of its capability of calculating both criteria weights and alternative
scores. While pairwise comparison forces users to reflect about criteria in detail, comparing every two factors/alternatives is effort intensive. Its computational complexity is $O(n^2)$, where $n$ is the number of the factors, while that of most others is $O(n)$. SAW, which is most intuitive and easy, is more appropriate for the scenarios where values under different criteria are comparable. ELECTRE more suits eliminating unqualified alternatives. TOPSIS can incorporate both real data and judgment into the calculation but cannot derive criteria weights. It is challenging to determine a decision method. Researchers suggest examining the characteristics of the methods from nine characteristics including preferences, uncertainty, and desired outcome. Cinelli et al. [35] guide the choice among methods. For example, W

method. Researchers suggest examining the characteristics of this work provides insights for selecting a method for concept selection. This work assumes that the criteria and their values have been well formulated. However, customer requirements as a main source of the selection criteria are complex and interconnected. It would be important and interesting to elicit and gather customer requirements as well as their preferences from public data such as online product comments, and then to translate them into applicable criteria. Integrating this will help evolve the tool into a more powerful platform.

**Acknowledgment**

The authors would like to thank the anonymous reviewers and the editors for the valuable comments that helped them in improving the quality of this paper.

**Funding Data**

- The National Natural Science Foundation of China (Grant No. 62002031).
- Scientific Research Start-up Fund of Shantou University (Grant No. NTF21042).
- National Key Research and Development Program of China (Grant No. 2021YFB1714400).

**Conflict of Interest**

There are no conflicts of interest.

**Data Availability Statement**

The authors attest that all data for this study are included in the paper.

**Appendix**

<table>
<thead>
<tr>
<th>Table 4 Object properties of the ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>hasConcepts</td>
</tr>
<tr>
<td>hasCriteria</td>
</tr>
<tr>
<td>hasPerformance</td>
</tr>
<tr>
<td>hasPreference</td>
</tr>
<tr>
<td>hasResults</td>
</tr>
<tr>
<td>hasWeights</td>
</tr>
<tr>
<td>hasRankings</td>
</tr>
<tr>
<td>owningPerformance</td>
</tr>
<tr>
<td>pfUnderCri</td>
</tr>
<tr>
<td>owningPreference</td>
</tr>
<tr>
<td>belongingToCptRnk</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 5 Data properties of the ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>cptTargetProduct</td>
</tr>
<tr>
<td>cptFunction</td>
</tr>
<tr>
<td>criDescription</td>
</tr>
<tr>
<td>criExpectation</td>
</tr>
<tr>
<td>criUnit</td>
</tr>
<tr>
<td>preJudgement</td>
</tr>
<tr>
<td>pfJudgement</td>
</tr>
<tr>
<td>rankRank</td>
</tr>
<tr>
<td>rankScore</td>
</tr>
<tr>
<td>wgtFuzzyValue</td>
</tr>
<tr>
<td>wgtCrispValue</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 6 Criteria properties of the ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>hasPerformance</td>
</tr>
<tr>
<td>hasPreference</td>
</tr>
<tr>
<td>hasResults</td>
</tr>
<tr>
<td>hasWeights</td>
</tr>
<tr>
<td>hasRankings</td>
</tr>
<tr>
<td>owningPerformance</td>
</tr>
<tr>
<td>pfUnderCri</td>
</tr>
<tr>
<td>owning Preference</td>
</tr>
<tr>
<td>belongingToCptRnk</td>
</tr>
</tbody>
</table>

**References**


