Advanced Engineering Informatics 59 (2024) 102329

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

Multicriteria requirement ranking based on uncertain knowledge representation and reasoning☆

Yufeng Ma, Yajie Dou *, Xiangqian Xu, Jiang Jiang, Kewei Yang, Yuejin Tan

College of Systems Engineering, National University of Defense Technology, Deya Road No. 109, Changsha, 410073, China

ARTICLE INFO

Keywords:
Fuzzy requirement knowledge graph
Predicate fuzziness
Attribute confidence
Grey relational analysis
Multicriteria ranking

ABSTRACT

The abundance of complex natural language descriptions related to requirements data presents major challenges for requirement analysis. Knowledge graphs (KGs), as the latest achievement in symbolic studies, are widely used in various fields due to their rich semantic expressive abilities. However, existing methods that mine resource description framework (RDF) triples in classical KGs cannot effectively capture the fuzzy semantic information in the product development system. To address this, we propose a multicriteria requirement ranking method based on uncertain knowledge representation and reasoning (KRR). First, we defined a model for representing uncertain knowledge to organize diverse data from multiple sources. This process was accomplished by constructing the requirement ontology, which is based on the function-behaviour-structure (FBS) model and requirements modelling-related documents. Next, a knowledge representation approach named the fuzzy requirement knowledge graph (FRKG) was devised by combining attribute confidence and predicate fuzziness. Then, knowledge reasoning rules were designed to enhance the edges in the FRKG, unveiling potential relationships between nodes. Utilizing the enriched FRKG (EFRKG), we proposed a multicriteria requirement ranking method based on grey relational analysis (GRA). To validate the effectiveness of the proposed approach, we conducted a case study involving unmanned aerial vehicles (UAVs). Furthermore, the semantic extension capability of FRKG was evaluated, and a comparison with traditional multicriteria requirement ranking methods was performed to demonstrate the efficiency of the proposed approach from both perspectives.

1. Introduction

With the increasing diversity of products, the rising complexity of product structures and functionalities, and the accelerating pace of product updates, the role of requirement analysis throughout the product life cycle has become increasingly prominent. Specifically, the quality of requirement analysis directly determines the level of user satisfaction [1]. Traditional requirement analysis can be summarized into two steps: first, using methods such as questionnaire surveys to extensively collect requirements with broad user participation; then, systematically decomposing user requirements through expert brainstorming sessions to generate actionable knowledge for development. However, due to the numerous, complex, and natural language descriptions of requirement data, this method has become costly and inefficient.

Currently, there have been some studies attempting to simplify and automate requirement analysis. Addressing the challenge of "natural language descriptions", some approaches based on natural language processing (NLP) have been developed to transform natural language into standardized requirement models, such as retrieving equivalent requirements [2], extracting glossaries and clustering [3], classifying app reviews [4], mining tweets [5], extracting user stories [6], and analysing requirements-related legal texts [7]. However, despite these advancements, requirement analysis in industrial practice remains predominantly manual.

To address the difficulties of "numerous" and "natural language descriptions", researchers have started to investigate technologies such as deep learning, KGs, and graph embedding. KGs, with their excellent data visualization and traceable reasoning paths, have attracted increased interest. A KG is a graph-based knowledge representation approach, and its core data model is the RDF schema (RDFS). This model can be encoded as triples in the form of subject–predicate–object, known as RDF triples. RDF is a W3C-certified semantic description model and has rapidly become the standard data model for metadata management and KG construction [8]. As RDFS is a logical framework that provides definitions for classes, class hierarchies, and property

☆ This work was supported by the National Natural Science Foundation of China (Grant No. 72231011).
* Corresponding author.
E-mail addresses: mashuang9707@163.com (Y. Ma), yajiedou_nudt@163.com (Y. Dou).

https://doi.org/10.1016/j.aei.2023.102329
Received 26 July 2023; Received in revised form 22 November 2023; Accepted 16 December 2023
Available online 26 December 2023
1474-0346/

hierarchies associated with resources, it enables RDF to support basic semantic reasoning. Therefore, RDFS is well suited for representing requirement-related knowledge within this framework. For instance, based on RDF, knowledge can be extracted from various data sources, represented in a unified manner, and used to construct a KG [9]. Currently, there are numerous studies on KG-based requirement analysis, which primarily focus on precise triple data representation. However, as requirement data are gathered from an open environment with multiple users participating, they inevitably contain errors, outdated information, and inconsistencies. Therefore, a knowledge representation method considering data uncertainty is needed. The sources of uncertainty mainly come from three aspects: (1) The inherently uncertain types of human knowledge in practical scenarios. For instance, in the industrial domain, product design data can be experiential and probabilistic. However, the classic RDF data model assumes that RDF data are reliable and accurate, ignoring their characteristics, and using traditional triples to express such data would be inaccurate or even wrong. (2) Inherent characteristics of triples. Many relationships between entities cannot be measured with numerical or Boolean values. Thus, many relationship predicates in triples have inherent uncertainty. However, the current RDFS and its corresponding reasoning rules cannot accurately represent this factor. (3) Noise generated during the construction process. Early KGs were constructed by domain experts through manual annotation, which introduced subjective noise. Later, large-scale KG construction relied on automation technologies such as machine learning, which often introduces certain noise during the process. Even if the initial KG is clear and complete, as more heterogeneous data sources are integrated, inconsistent or uncertain data often emerge.

To date, there have been many works on modelling uncertain data. Regarding data representation methods, there are extensions and queries for fuzzy concepts data models, fuzzy (relational and object-oriented) database models, and fuzzy XML models [10,11]. Regarding logic-based knowledge representation methods, there are fuzzy extensions and axiom formulations based on description logic, establishment and learning of fuzzy ontologies [12], and design and data queries for RDF fuzzy data models [13,14]. Finally, to address the challenge of "complex scenarios", researchers must consider multiple factors for requirement ranking to adapt to complex scenario requirement analysis, as each requirement item encompasses multiple evaluation metrics.

Taking into account the three challenges mentioned above, we propose a multicriteria requirement ranking method based on uncertainty KRR. Our method addresses data uncertainty in two ways: first, by designing an uncertainty KRR model, and second, by integrating the vital tool GRA into the requirement ranking model. Specifically, this paper first constructs a requirements ontology based on the FBS model and relevant requirements files. To measure the uncertainty of RDF triples, we introduced attribute confidence and predicate fuzziness into the RDF triples, designing a representation approach. To unearth potential relationships between nodes, we developed knowledge reasoning rules to establish more connections among nodes. Finally, based on the enriched FRKG, we propose a multicriteria requirement ranking method using GRA. GRA measures the degree of correlation of indicator sequences by applying the grey correlation coefficient and evaluates things comprehensively from a holistic viewpoint. This method does not require a priori information, does not need to assume that the data obey a specific distribution, is able to deal with fuzzy and uncertain information, and has good interpretability, so it can be used for various types of data. In this study, by applying GRA, we summarize the multicriteria scores for each requirement and transform them into a single score. Finally, the requirements are ranked based on the resulting scores.

Therefore, this paper makes contributions in three main aspects:

(1) An uncertain knowledge representation approach including predicate fuzziness and attribute confidence is proposed. This method effectively measures the fuzziness of the requirement data and transforms traditional qualitative reasoning into quantitative reasoning.

(2) A fuzzy semantic reasoning-based method to extend the FRKG is introduced. This method aims to uncover associations among multiple related predicates, addressing the limitation of rules defined in RDFS, which only support triple reasoning for specific predicates.

(3) A multicriteria requirement ranking method based on GRA, which considers uncertainty during the requirement ranking process from another perspective, is presented. This method will allow researchers to address uncertainty in a multidimensional manner.

The remainder of the paper is structured as follows: related works are described in Section 2; the multicriteria requirement ranking method based on uncertainty KRR is presented in Section 3; the experiments, including a case study and two evaluations, are described in Section 4; and the summary and prospects of this study are presented in Section 5.

2. Related works

2.1. Knowledge representation and reasoning for fuzzy data

Knowledge representation and reasoning (KRR) mimics nonexpert users to address specific problems based on knowledge gathered from domain experts, finding extensive applications in artificial intelligence and decision support systems. The existing methods are numerous, evolving from the initial logical reasoning to the present, where innovations are tailored to different application domains. These methods can be fundamentally categorized into two main types: those fully based on logic and those not solely based on logic (or involving other types of tools simultaneously).

The first category is description logic, which encompasses a first-order logic subset and syntax used for defining individuals, classes, relations, and hierarchical knowledge. Reasoning is carried out through logical reasoning, and new facts can be defined by assessing their consistency. Handling uncertainty has been a challenging task in this type of KRR [15]. There are two approaches to addressing uncertainty in this category: (1) Explicitly defining uncertainty. For instance, specifying the probability of an event occurring as 0.7 indicates a 70% confidence in the event's occurrence. However, due to assumptions and constraints, it may be difficult to accurately estimate probability distributions in certain cases. Additionally, this approach often treats uncertainty as an independent probability value, making it challenging to seamlessly integrate uncertainty with logic. (2) Using nonmonotonic logic to define sometimes-true facts. For example, "birds can fly", but in certain situations, such as when a bird is injured, this proposition may be false. Therefore, the truth value of the proposition can vary in different contexts rather than remaining fixed. This approach allows multiple interpretations or truth values, potentially leading to ambiguity and difficulties in determining the best inference results. Furthermore, formalizing nonmonotonic logic involves flexible handling of context, which may require more advanced KRR methods.

The second category of nonlogic-based KRR methods mainly includes probability theory, dynamic causal relationship graphs [16], logic programming [17], concept graphs [18], fuzzy logic, ontologies, KGs, and more.

(1) Probability theory is the most important method for reasoning with uncertain data. Common methods include Bayesian networks and inference [19], Markov chains and models [20], and combined probabilistic logical methods [21]. However, these methods require accurate probability parameters, which can be challenging to obtain in complex scenarios.

(2) The dynamic causal relationship diagram is a graphic technique for representing the causal relationships between entities or events. Building and maintaining these diagrams require a substantial amount of domain-specific knowledge, and in certain complex situations, causal relationships may become ambiguous or difficult to determine. Based

on this, methods have been developed to describe fuzzy causal knowledge graphs [22], as well as to enhance decision-making approaches based on causal knowledge graphs [23,24].

(3) Logic programming typically employs formal logic languages such as Prolog to represent knowledge and rules, followed by querying and reasoning operations performed by an inference engine. Logic programming is relatively limited in handling uncertainty and fuzziness, as it is more suitable for precise logical reasoning and may lack the flexibility required for uncertain real-world problems.

(4) Concept graphs are a graphical technique used to visualize relationships between knowledge and concepts, consisting of nodes (representing concepts or entities) and edges (representing relationships). However, constructing and maintaining concept graphs can be complex and time-consuming.

(5) Fuzzy logic provides a robust framework for uncertain knowledge reasoning [25] and is widely used in information analysis and control tasks. Among these, FPR [26] leverages the framework of fuzzy logic to represent and manage knowledge. FPR extracts and manages knowledge from multiple experts, where the database can be directly input by humans or generated by automated machine learning algorithms. However, the rules generated by this method often tend to be redundant or suboptimal or have weak generalization capabilities, requiring further processing. The combination of fuzzy logic with Petri nets results in FPN, where transitions are defined as fuzzy rules [27]. In response to the difficulty of accurately simulating expert experiences or cognition with FPN, methods such as linguistic Petri nets [28], simplified setrosophic Petri sets [29], and double hierarchy hesitant linguistic Petri nets [30] have been proposed. While graphical representation provides a clearer and more expressive understanding, the efficiency of its inference algorithms is relatively low, and there are some restrictive requirements.

(6) Web ontology language (OWL) is used to describe modelling ontologies for various domains, representing classes, individual properties, and relationships through predefined axioms. To design more complex inference rules, RDFS introduces a subset of OWL integrated into RDFS++ that is used for reasoning over RDF triples [31].

(7) KGs are entity-relationship graphs, and reasoning is the process of filling knowledge gaps and automatically establishing new connections [32]. In this context, knowledge is represented as text related to the vocabulary, and the purpose of reasoning is to analyse and understand the text, with results including sentiment analysis, automatic text summarization, or automated knowledge extraction. Semantic reasoning is a key feature of KGs, distinguishing them from other knowledge representation frameworks.

However, the above KRR methods lack generalization and scalability, making it challenging to model complex relationships between entities. These methods require experts to redefine rules for similar reasoning problems in different domains. Moreover, the application of these highly expert-dependent methods in industrial environments is also challenging. Most importantly, these methods are unable to handle binary relations with fuzzy semantics. Therefore, it is necessary to design a scalable KRR method capable of embedding fuzzy semantics.

2.2. Knowledge graph-based requirement analysis

Inspired by the crucial role of natural language in requirement analysis, researchers have been making efforts to develop NLP tools and methods for requirements identification [33]. However, in industrial practice, requirement analysis is still predominantly carried out manually. The NaPiRE survey revealed that only 16% of companies are utilizing automated techniques for requirement analysis [34]. KGs have attracted growing attention due to their ability to capture, integrate, process, and leverage vast entities and relationships inherent in products. On the one hand, KGs have been introduced as a novel knowledge management tool in the product life cycle, enabling the description and classification of data in a more user-friendly manner. On the other hand, owing to their distinctive knowledge representation capabilities, KGs can enhance task applicability across domains and facilitate the implementation of a knowledge-as-a-service model. Specific studies are presented in Table 1.

As shown in Table 1, existing studies mainly consist of specific case studies tailored to particular application scenarios. Different researchers have explored various intelligent assistance tasks, such as defect prediction [32], price prediction [38,39], and requirement extraction [40,41]. For each task, distinct datasets have to be collected, and separate intelligent tools have to be developed, making it challenging to effectively reuse knowledge in product development. Moreover, these methods exclusively employ the traditional RDF triple format for constructing KGs, overlooking the crucial aspect of knowledge uncertainty. Nevertheless, in the context of product manufacturing, uncertainty is an important characteristic that demands careful consideration.

2.3. Multicriteria requirement ranking

Requirement ranking, also known as determining the priority of requirements, plays a crucial role in deciding how requirements are selected and handled. Given that requirements often involve multiple stakeholders, potential conflicts between different requirements, a lack of clear ranking criteria and guidance, and dynamic environmental changes, experts need to carefully assess the importance and value of various requirements to establish priorities.

As a result, in the development process, multicriteria decision-making methods are commonly used for ranking, such as TOPSIS, ORESTE, AHP, VIKOR, MULTIMOORA and COPRAS, with AHP being one of the most popular and widely cited methods [42]. AHP determines the priority of each requirement through pairwise comparison matrices. Due to the straightforward application and structure of AHP, several requirement ranking methods based on AHP have been developed, including the power analytic hierarchy process [43], case-based ranking [44], and preferences and dependencies-based ranking [45]. However, the aforementioned methods are time-consuming and challenging to scale for large sets of requirements [46]. Additionally, most methods utilize pairwise comparisons for individual evaluations, which do not provide a holistic view of the project. Another approach is context-based priority assessment [47], which employs a relationship matrix to prioritize requirements from multiple stakeholders, considering the relationships between viewpoints in the requirements. The Kano method [48] also allows prioritizing requirements based on stakeholders' preferences; however, it focuses on product differentiation features. Similarly, the Lancaster theory method [49] employs a quantitative model to compare requirement priorities, whereas the Wiegers method [50] estimates requirement priorities using a scheme based on the quality function deployment (QFD) customer value concept. In general, the aforementioned requirement ranking methods capture different elements for ranking requirements from the project's entirety (business objectives, customer expectations, product definition, stakeholder requirements). Nonetheless, these methods are mostly suitable for small-scale requirement sets.

We have observed that existing requirement ranking methods lack scalability to handle large-scale and dynamically managed requirements. Additionally, in most cases, the results obtained from priority-based methods may not fully align with the desired requirement ranking expected by stakeholders. This finding is because many priority sorting methods only consider certain types of quantitative information, which may lead to conflicts with stakeholders' viewpoints [51].

3. Definition and modelling

Our method mainly consists of two parts: knowledge representation and knowledge reasoning, which integrate ontology, logical reasoning, KGs, and probability theory. In the knowledge representation part, we first construct the ontology of requirement knowledge, serving as

Table 1
Comparison of knowledge graph applications in various studies.

Reference	Application scenarios	Functions of KG	Method of study
Wang et al. [32]	Spacecraft fault diagnosis	Human–machine interface	Information integration, FBS model
Nanduri et al. [35]	Fraud prevention in e-commerce	Describe procurement transactions	Machine learning
Muñoz et al. [36]	Prediction of adverse drug reactions	Generate different feature sets	Multilabel learning
Sun et al. [37]	Automated attack and defence for 5G security	Solutions, hidden relationship discovery	Multilayer framework
Liu et al. [38]	Forecasting stock prices	Feature mining	Deep learning
Long et al. [39]	Forecasting stock prices	Correlation mining	Convolutional neural networks, deep learning
Wang et al. [40]	Requirement extraction for bicycle design	Description of contextual scene relationships	Random wandering algorithm
Wang et al. [41]	Smart bike design	Knowledge management	Random wandering algorithm

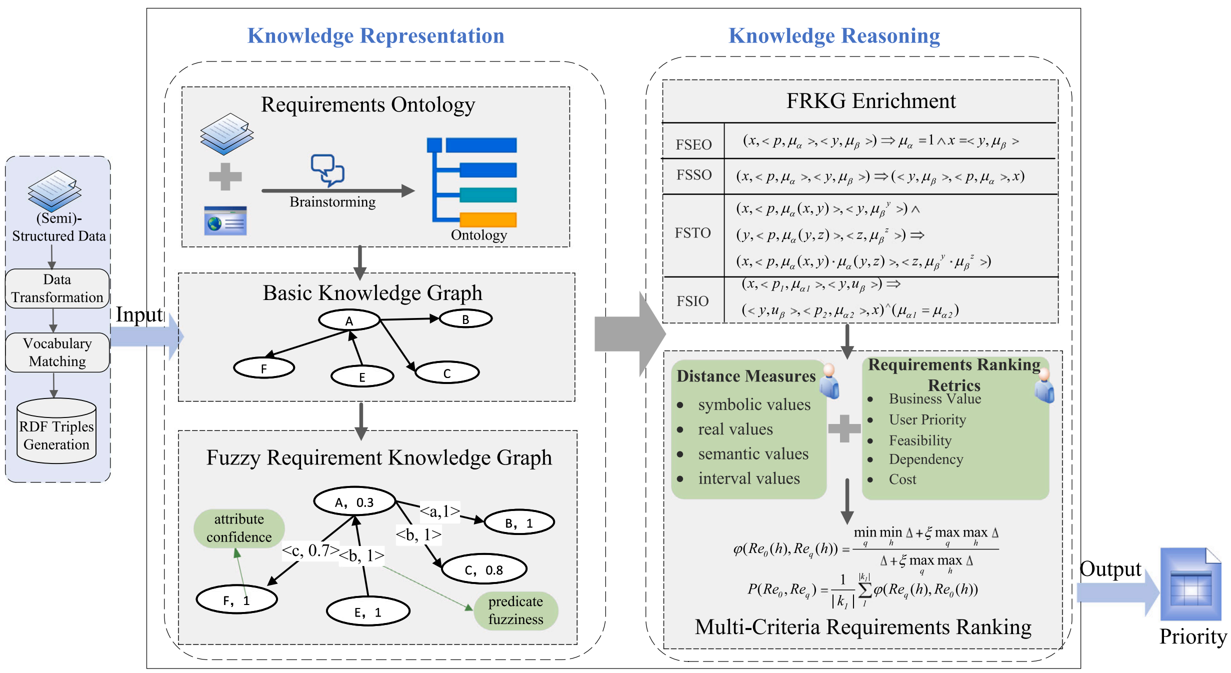

Fig. 1. Overall framework of our method.

the logical foundation for requirement analysis. Based on this, we define an FRKG, which incorporates predicate fuzziness and attribute confidence into the basic knowledge graph (BKG) to measure uncertain requirement knowledge. Moving on to the knowledge reasoning, we initially define the logical reasoning rules considering predicate fuzziness and attribute confidence. Based on these rules, we then introduce the FRKG enrichment method, enabling the exploration of potential relationships between nodes. Finally, we define multiple requirement ranking metrics and employ the GRA multicriteria requirement ranking method for requirement prioritization. The technical framework of this paper is illustrated in Fig. 1.

3.1. Knowledge representation

3.1.1. Requirement ontology construction

The main purpose of this section is to clarify the top-level schema and define the domain concepts as references for the later requirement ranking. Considering the requirement-related textual resources (e.g., specifications, manuals, technical reports, maintenance reports) and the structure of mature requirement analysis methods, we define the ontology based on *requirement specification documents*, *requirement analysis methods* and *FBS-based models*.

Requirement specification documents contain historical requirement lists that are made by consulting customers and can also be integrated by user stories. A requirement representation method is proposed based on a well-known requirement boilerplate [41]. In this boilerplate, a requirement can be represented as “Under what condition, a system should do what process”. Therefore, we extract four types of entities including *goal*, *requirement* and *actor*.

Regarding *requirements analysis methods*, graphical representation methods such as requirement diagrams, use case diagrams, and swimlane diagrams are widely used for visualizing and expressing requirements in the requirements analysis process. Although they are not strictly knowledge representation methods, these methods can assist in expressing and organizing relevant knowledge and information during requirements analysis. These graphical methods use symbols, relationships, and structures to describe system requirements, stakeholders, functions, and interactions. Therefore, when constructing an ontology, we can use existing graphical representation methods as a reference.

The *FBS-based model* injects certain professional domain knowledge into the KG, the initial structure of which is built based on the internal relationships among function (goals of the design object), behaviour (attributes derived from the structure) and structure (components and their relationships). Because of its ability to fully characterize the internal relationships of design knowledge, this model is therefore widely used for knowledge representation of product design in the schematic design phase. Since the *FBS-based model* is usually established before the usage stage and does not contain contextual information, scenario- and user-related data extracted from *requirement specification documents* should be connected with it. Finally, the ontology is constructed based on the abovementioned data, as shown in Fig. 2.

3.1.2. Fuzzy requirement knowledge graph construction

This section begins by defining the BKG and then considers the uncertainty of requirement analysis-related data, proposing the concept

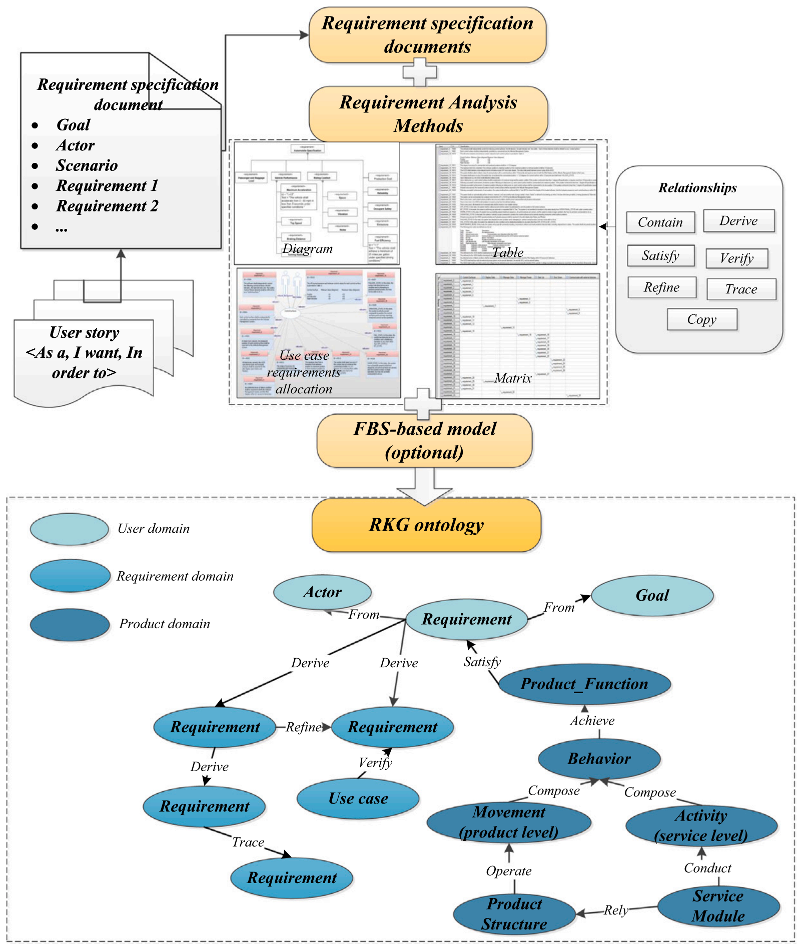

Fig. 2. Requirements ontology construction.

of an FRKG. This graph takes into account the semantic predicates that cannot be quantitatively computed, such as “satisfy”, “influence”, and “like”, and attributes that can be quantitatively measured, such as “length” and “time”. On the one hand, traditional knowledge representation methods cannot accurately capture the associations between fuzzy concepts and requirement descriptions. By introducing predicate fuzziness, we can better understand the relationships and similarities between fuzzy concepts, thereby improving the accuracy of requirement expression and matching. On the other hand, requirement analysis often relies on expert experience or existing knowledge to obtain attribute information, but this information may contain uncertainty or errors. By introducing confidence, we can quantify the reliability of attributes, reducing the impact of erroneous information on requirement analysis results and enhancing the reliability of the analysis results. Therefore, the introduction of fuzziness and confidence provides major advantages for requirement analysis problems. By constructing an FRKG and assigning predicate fuzziness and attribute confidence, we can better handle semantic fuzziness and attribute uncertainty, thereby improving the accuracy and reliability of requirement analysis. This knowledge representation method can accurately measure fuzzy semantic relationships and enhance the accuracy of reasoning through confidence levels.

Assuming that V is a finite set of vertices, E is a set of edge-connected vertices, and L is a set of mapping relations and their corresponding labels between V and E. Σ is a set of all labels involved in L. Then, we define the BKG as follows.

Definition 1. Let $G = (V, E, \Sigma, L)$ be a BKG. For an RDF triple (s, p, o), $v^s, v^o \in V$ are two vertices that correspond to s and o. Then, $L(v^s)$ is equal to s, and $L(v^o)$ is equal to o. In addition, E is a directed edge from v^s to v^o. An example of the BKG is illustrated in Fig. 3.

To supplement the BKG above, we propose a formal definition of an FRKG considering the fuzziness of predicates and the confidence of attributes.

Definition 2. Let $G^F = (V, E, \Sigma, L^V, L^E, \mu^\alpha, \mu^\beta)$ be an FRKG with finite RDF triples, where L^V and L^E specify the label corresponding to each vertex or edge, and $L^V = C^i \cup C^b \cup C^r \cup C^s$, which represent the interval value, symbolic value, real number value and semantic value, respectively. Additionally, $\mu^\alpha, \mu^\beta \in [0, 1]$, μ^α is the fuzziness of the predicates, and μ^β is the confidence of the attributes. An example of the FRKG is illustrated in Fig. 4.

In the above definition, each triple has a label for its vertices and edges. Confidence is assigned to vertices representing attributes, which is highly valuable for expressing the credibility of quantifiable properties in product design parameters and the like. Fuzziness is assigned to edges to measure the fuzzy semantics of the relationships. In most cases, FRKGs contain fuzzy terms for fuzzy vertices or fuzzy relations, with their confidence or fuzziness calculated based on actual circumstances. It can also be observed from the above definition that BKGs are a special case of FRKGs, where both the confidence of attribute vertices and the fuzziness of edges are set to 1.

3.2. Knowledge reasoning

3.2.1. FRKG enrichment based on fuzzy semantic reasoning

By annotating the fuzzy semantics of RDF triples, we obtain the FRKG based on BKG. The essence of FRKG enrichment in this section is

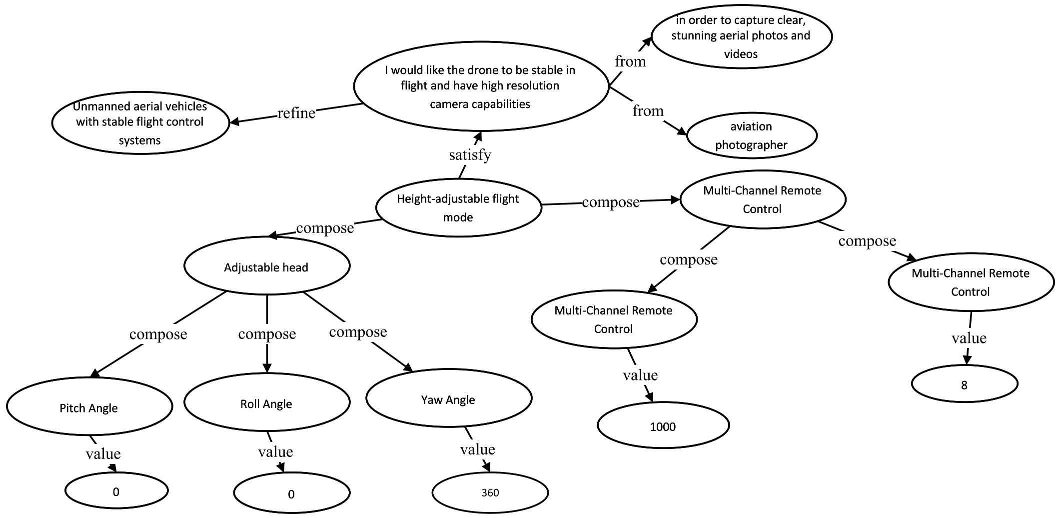

Fig. 3. Example of the BKG.

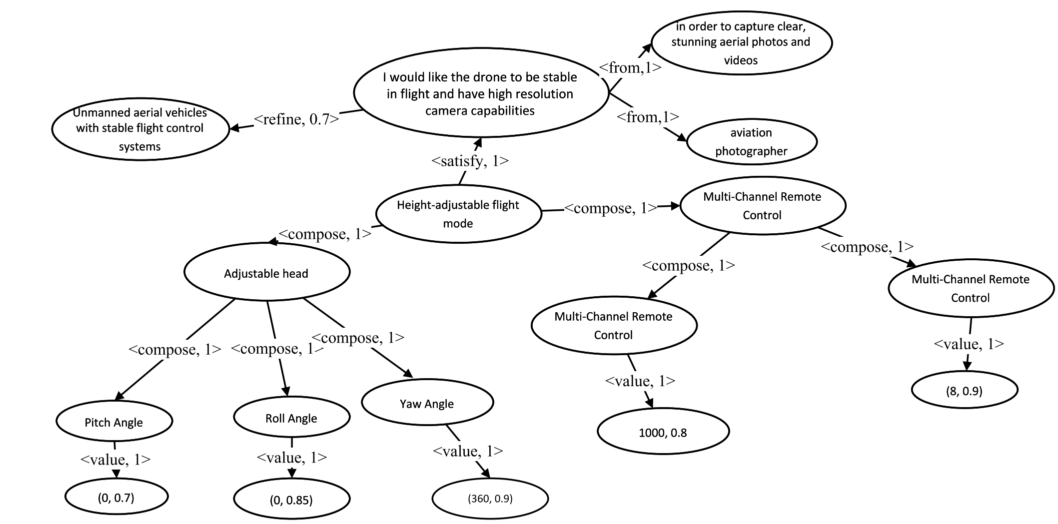

Fig. 4. Example of the FRKG.

to reason based on known edges and rules to obtain unknown edges to establish more potential relations among nodes. One study [52] analysed the semantic properties of binary predicates in RDF triples and constructed the inference rules for BKG shown in Table A.1, but the rules did not consider the fuzzy semantics and were not directly applicable to FRKG; thus, the further proposed inference rules are shown in Table A.2 [53], which can describe the semantic vagueness and the semantic changes between fuzzy RDF triples in the reasoning process. However, these rules do not consider the rules for computing attribute confidence. Therefore, in this section, we will design a new fuzzy reasoning mode for FRKG and apply it to perform reasoning in FRKG to explore relationships between nondirectly connected entities with higher accuracy.

Table 2 provides some simple fuzzy reasoning rules extended from Table A.1. From Table 2, when the fuzzy semantic operation is FSEO, FSSO, and FSIO, these three fuzzy operations only represent relationships between two entities, and the fuzziness μ^{α} and confidence μ^{β} are not calculated during the inference process. However, the predicate operation labelled FSTO involves the transitivity of multiple RDF triples. Therefore, both the predicate fuzziness and attribute confidence will be accumulated as the transitive path increases. As the path length increases, the relationship represented by the predicate p changes from direct to indirect, and the association between entities decreases as the distance increases, resulting in an increase in the fuzziness of semantics and a decrease in the confidence of attributes. This conclusion aligns with real-world situations. However, the reasoning rules in Table 2 only express the fuzzy semantic attributes between two specific entities, without demonstrating the semantic associations between different fuzzy RDF triples; hence, they cannot discover more implicit knowledge within FRKG. Therefore, reasoning rules between different RDF triples need to be defined.

The above rules with FSTO are only applicable to operations with the same predicate. To establish more relationships between entities in FRKG, we expand the FSTO rule for different fuzzy predicates. Based on its directionality and asymmetry, we divide FSTO into four categories, as shown in Table 3. When using this model, we must first determine the relationship between the predicates r and p. For example, if "prefer" is FSTO^{oin} of "influence", as shown in Fig. 5, compared to the initial triple $(Consumer_I, Interested, Product_A)$, the recommended result triple $(Consumer_I, Interested, Requirement_2)$ involves the replacement of the $Object$, and $Product_A$ is inside the inference path, therefore represented as FSTO^{oin}.

Based on the above inference rules, the edge enrichment algorithm for FRKG in this section is as follows:

Table 2
Simple reasoning rules considering predicate fuzziness and attribute confidence.

Fuzzy semantic operation	Symbol	Logical explanation
Fuzzy semantic equal operation	FSEO	$(x, \langle p, \mu^{\alpha} \rangle, \langle y, \mu^{\beta} \rangle) \Rightarrow \mu^{\alpha} = 1 \wedge x = \langle y, \mu^{\beta} \rangle$
Fuzzy semantic symmetry operation	FSSO	$(x, \langle p, \mu^{\alpha} \rangle, \langle y, \mu^{\beta} \rangle) \Rightarrow (\langle y, \mu^{\beta} \rangle, \langle p, \mu^{\alpha} \rangle, x)$
Fuzzy semantic transmission operation	FSTO	$(x, \langle p, \mu^{\alpha}(x, y) \rangle, \langle y, \mu^{\beta\, y} \rangle) \wedge$ $(y, \langle p, \mu^{\alpha}(y, z) \rangle, \langle z, \mu^{\beta\, z} \rangle) \Rightarrow$ $(x, \langle p, \mu^{\alpha}(x, y) \cdot \mu^{\alpha}(y, z) \rangle, \langle z, \mu^{\beta\, y} \cdot \mu^{\beta\, z} \rangle)$
Fuzzy Semantic Inverse Operation	FSIO	$(x, \langle p^{1}, \mu^{\alpha 1} \rangle, \langle y, u^{\beta} \rangle) \Rightarrow$ $(\langle y, u^{\beta} \rangle, \langle p^{2}, \mu^{\alpha 2} \rangle, x)^{\wedge}(\mu^{\alpha 1} = \mu^{\alpha 2})$

Table 3
Extension of FSTO rules for different fuzzy predicates.

Symbol	Logical explanation
FSTO^{sin}	$(x, \langle r, \mu_{\alpha}(x, y) \rangle, \langle y, \mu_{\beta}{}^{y} \rangle) \wedge (z, \langle p, \mu_{\alpha}(z, x) \rangle, \langle x, \mu_{\beta}{}^{x} \rangle) \Rightarrow$ $(z, \langle r, \mu_{\alpha}(x, y) \cdot \mu_{\alpha}(z, x) \rangle, \langle y, \mu_{\beta}{}^{y} \cdot \mu_{\beta}{}^{x} \rangle)$
FSTO^{sout}	$(x, \langle r, \mu_{\alpha}(x, y) \rangle, \langle y, \mu_{\beta}{}^{y} \rangle) \wedge (x, \langle p, \mu_{\alpha}(x, z) \rangle, \langle z, \mu_{\beta}{}^{z} \rangle) \Rightarrow$ $(z, \langle r, \mu_{\alpha}(x, y) \cdot \mu_{\alpha}(x, z) \rangle, \langle y, \mu_{\beta}{}^{y} \cdot \mu_{\beta}{}^{z} \rangle)$
FSTO^{oin}	$(x, \langle r, \mu_{\alpha}(x, y) \rangle, \langle y, \mu_{\beta}{}^{y}(x, y) \rangle) \wedge (z, \langle p, \mu_{\alpha}(z, y) \rangle, \langle y, \mu_{\beta}{}^{y}(z, y) \rangle) \Rightarrow$ $(x, \langle r, \mu_{\alpha}(x, y) \cdot \mu_{\alpha}(z, y) \rangle, \langle z, \mu_{\beta}{}^{y}(x, y) \cdot \mu_{\beta}{}^{y}(z, y) \rangle)$
FSTO^{oout}	$(x, \langle r, \mu_{\alpha}(x, y) \rangle, \langle y, \mu_{\beta}{}^{y} \rangle) \wedge (y, \langle p, \mu_{\alpha}(y, z) \rangle, \langle z, \mu_{\beta}{}^{z} \rangle) \Rightarrow$ $(x, \langle r, \mu_{\alpha}(x, y) \cdot \mu_{\alpha}(y, z) \rangle, \langle z, \mu_{\beta}{}^{y} \cdot \mu_{\beta}{}^{z} \rangle)$

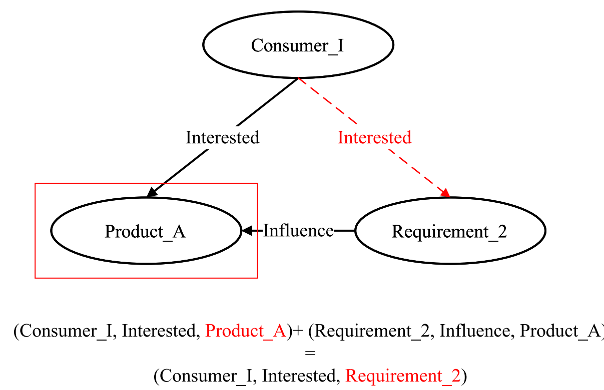

Fig. 5. Illustration of FSTO^{oin}.

Input: A fuzzy requirement knowledge graph G_F
Output: The closure of G_F, denoted as $G_F{}'$
Initialization: $G_F{}' = G_F$

While $|G_F| \geq |G_F{}'|$
  $G_F = G_F{}'$ % Apply rules for the same predicate
  for $i = 0$ **to** $m - 1$
    if $\mathrm{P_i}$= FSEO then apply rule (FSEO) to $G_F{}'$
    if $\mathrm{P_i}$= FSSO then apply rule (FSSO) to $G_F{}'$
    if $\mathrm{P_i}$= FSTO then apply rule (FSTO) to $G_F{}'$
  end for
  % Apply reasoning rules for different predicates
  for $i = 0$ **to** $n - 1$
    for $j = 0$ **to** $m - 1$
      if $\mathrm{R_i \& P_j}$ = FSIO
      then apply rule (FSIO) to $G_F{}'$
      if $\mathrm{R_i \& P_j} = \mathrm{FSTO_s^{in}}$
      then apply rule ($\mathrm{FSTO_s^{in}}$) to $G_F{}'$
      if $\mathrm{R_i \& P_j} = \mathrm{FSTO_s^{out}}$
      then apply rule ($\mathrm{FSTO_s^{out}}$) to $G_F{}'$
      if $\mathrm{R_i \& P_j} = \mathrm{FSTO_o^{in}}$
      then apply rule ($\mathrm{FSTO_o^{in}}$) to $G_F{}'$
      if $\mathrm{R_i \& P_j} = \mathrm{FSTO_o^{out}}$
      then apply rule ($\mathrm{FSTO_o^{out}}$) to $G_F{}'$
    end for
  end for
  if $G_F = G_F{}'$ **then** break
end while
return $G_F{}'$

3.2.2. Multicriteria requirement ranking based on GRA

The operation in Section 3.2.1 significantly enriches the number of edges in FRKG, establishing more relationships between requirement nodes and facilitating the exploration of potential deep-level knowledge. Obviously, the diverse data types of nodes essentially form a heterogeneous information system. Therefore, this section abstracts the requirement analysis problem as a multicriteria requirement ranking problem for a heterogeneous information network. To calculate the scores of different requirements for each evaluation criterion, we selected five requirement indicators [53]: business value, user priority, feasibility, dependency and cost. The calculation method for various indicators based on the enriched FRKG is as follows:

(1) ***Business Value***: Assess the contribution of requirements to goals. This is measured through connections between "requirement" nodes and "goal" category nodes. For example, the "goal" node connected to the "achieve autonomous flight" has an interval value of "increase flight time by 15%–25%", and a semantic value of "medium" to describe the urgency or priority of the goal.

(2) ***User Priority***: Understand the priority of different requirements from the perspective of users. Prioritizing the needs of key user groups can enhance user satisfaction, which can be measured by the sum of distances between the "requirement" nodes and "user" nodes. For example, the "user" node contains symbolic value "professional photographer", a verbal value "minimum flight altitude between 80 m to 120 m", indicating a semantic value of "minor" for the importance level of the user's requirement.

(3) ***Feasibility***: Evaluate the implementation difficulty of the requirements. Requirements that are relatively easy to implement may be prioritized for quick delivery and meeting user needs, which can be measured by the sum of distances between the "requirement" nodes and "product_structure" nodes. In the "product_structure" node named "high performance camera", there is a real value "camera resolution is 20 megapixels", and a semantic value of "essential" for the importance level of the product feature.

(4) ***Dependency***: If a requirement is a prerequisite for other requirements, it may need to be prioritized for fulfilment, indicating its greater importance and need for early implementation. This can be measured by the sum of distances between the target "requirement" node and other "requirement" nodes. These nodes also include various data formats such as real values, semantic values, and interval values.

(5) ***Cost***: Evaluate the balance between the cost of meeting the requirements and the expected benefits. Choosing requirements with good cost-effectiveness can maximize resource utilization efficiency, which can be measured by the sum of distances between the "requirement" nodes and the "cost" attribute of "Product_structure" nodes. For instance, it may contain an interval value such as "development team labor cost between $9000 and $11,000", along with a semantic value describing the importance of the cost as "high".

Table 4
Heterogeneity-based multicriteria scoring table.

	F_1			…	F_p			…	F_m		
	E_1	…	$E_{\|k_1\|}$		E_1	…	$E_{\|k_p\|}$		E_1	…	$E_{\|k_m\|}$
Re_1	S_0	…	9		6	…	[3, 5]		3	…	F
Re_2	S_1	…	3		7	…	[1, 9]		7	…	M
…	…	…	…		…	…	…		…	…	…
Re_n	S_5	…	6		7	…	[6, 9]		6	…	F

Table 5
Single-criteria scoring table for requirements-criteria.

	Factor				
	F_1	…	F_p	…	F_m
Re_1	P_{11}		P_{1p}		P_{1m}
Re_2	P_{21}		P_{2p}		P_{2m}
…	…		…		…
Re_n	P_{n1}		P_{np}		P_{nm}

Finally, the multicriteria scoring matrix is shown in Table 4. Re_n represents the n requirement, F_m represents the m criterion for requirements, and $E_{|k_1|}$ represents different data types related to requirement information under different criteria.

Considering the "grey nature" of requirements analysis and the application scenario of GRA, it is introduced into the multicriteria requirements ranking. Next, the GRA method is used to calculate the correlation between the actual evaluation sequence of requirements and the optimal value (where all criteria scores are set to the maximum value) to determine their proximity to the optimal value. The smaller the correlation, the lower the requirement's ranking. Multicriteria ranking considers both the total score of requirements for the indicators and the scores of requirements for different indicators. The formula for calculating the correlation between the target node and candidate node is as follows:

$$\varphi(Re_0(h), Re_q(h)) = \frac{\min_q \min_h \Delta + \xi \max_q \max_h \Delta}{\Delta + \xi \max_q \max_h \Delta} \tag{1}$$

$\varphi(Re^0(h), Re^q(h))$ is the correlation coefficient between the evaluation sequence Re^q and the reference sequence Re^0 for the h criterion; $\xi(\xi \in [0,1])$ is the discrimination coefficient, and ξ is generally less than or equal to 0.5, mainly to reduce the impact of extreme values on calculations; $\Delta = d(Re^0(h), Re^q(h))$ is the distance between the evaluation sequence Re^q and the reference sequence Re^0 for the h criterion, and the calculation of distance can refer to Eqs. (2), (3), (4), and (5).

Due to the presence of different types of nodes in FRKG, such as symbolic values, real values, semantic values, and interval values, denoted as C^b, C^r, C^s, and C^i, respectively, distance measures for two entities in different attribute sets are defined in FRKG to handle complex product data.

(1) Assuming that $L^v \in C^b$, μ^α is the fuzzy predicate degree between two nodes $x, y \in V$, $\mu^{\beta x}$ and $\mu^{\beta y}$ are the confidence degrees of the two nodes, the distance between $v(x)$ and $v(y)$ is defined as follows:

$$d(v(x), v(y)) = \begin{cases} 0, v(x) = v(y) \\ 1 \cdot \mu^\alpha \cdot \mu^{\beta x} \cdot \mu^{\beta y}, v(x) \neq v(y) \end{cases} \tag{2}$$

(2) Assuming that $L^v \in C^r$, μ^α is the fuzzy predicate degree between two nodes $x, y \in V$, $\mu^{\beta x}$ and $\mu^{\beta y}$ are the confidence degrees of the two nodes, and the distance between $v(x)$ and $v(y)$ is defined as follows:

$$d(v(x), v(y)) = \mu^\alpha \cdot \mu^{\beta x} \cdot \mu^{\beta y} \cdot \frac{|v(x) - v(y)|}{\max\{v(i)\} - \min\{v(i)\}} \tag{3}$$

(3) Assuming that $L^v \in C^s$, the set of semantic values is defined as $S = \{s_t | t = -\tau, \dots, -1, 0, 1, \dots, \tau\}$. μ^α is the fuzzy predicate degree between two nodes $x, y \in V$, $\mu^{\beta x}$ and $\mu^{\beta y}$ are the confidence degrees of the two nodes. The distance between $v(x)$ and $v(y)$ is defined as follows:

$$d(v(x), v(y)) = d(s_\alpha, s_\beta) = \mu^\alpha \cdot \mu^{\beta x} \cdot \mu^{\beta y} \cdot \frac{|\alpha - \beta|}{2\tau + 1} \tag{4}$$

(4) Assuming that $L^v \in C^i$, μ^α is the fuzzy predicate degree between two nodes $x, y \in V$, $\mu^{\beta x}$ and $\mu^{\beta y}$ are the confidence degrees of the two nodes, and the distance between $v(x)$ and $v(y)$ is defined as follows:

$$\begin{aligned} d(v(x), v(y)) = & \mu^\alpha \cdot \mu^{\beta x} \cdot \mu^{\beta y} \cdot \frac{\sqrt{2}}{2} \\ & \times \sqrt{(v(x)^u - v(y)^u)^2 + (v(x)^l - v(y)^l)^2} \end{aligned} \tag{5}$$

where $v(x)^u$ and $v(x)^l$ are the upper and lower bounds of the interval values $v(x)$, respectively, and $v(y)^u$ and $v(y)^l$ are the upper and lower bounds of the interval values $v(y)$, respectively.

In summary, after retrieval of G^F as a target node and its corresponding attribute list A, the hybrid distance between nodes x and y is calculated as follows:

$$d_A(x, y) = \sqrt{\frac{1}{|A|} \sum_{a \in A} d^2(v(x), v(y))} \tag{6}$$

For the purpose of comparison, the correlation coefficients under different criteria are integrated into the GRA ($h = 1, 2, \dots, |k^i|$), which is defined as Eq. (7).

$$P(Re^0, Re^q) = \frac{1}{|k_1|} \sum_{1}^{|k_1|} \varphi(Re^q(h), Re^0(h)) \tag{7}$$

By applying the GRA method to calculate the scoring values of different requirements for all criteria, the heterogeneity-based multicriteria ranking can be transformed into a single-score ranking. The recommendation scores in Table 4 are transformed into Table 5. Through the use of the GRA method to aggregate the different criteria of the requirements, it is possible to overcome the limitations of traditional methods applied to diverse multicriteria rankings. Additionally, the introduction of predicate fuzziness and attribute confidence in the distance calculation formula enhances the accuracy and interpretability of the results.

4. Experiments

As the proposed requirement ranking method in this paper integrates multiple techniques, there is currently a lack of similar requirement ranking methods for comparison. Therefore, this section aims to validate the method through three aspects of verification. First, taking UAVs as an example, we will gradually construct a UAV requirement ontology, build an FRKG, enrich the FRKG, and perform requirement ranking. The results obtained will be analysed to validate the usability and effectiveness of the proposed method. Next, the evaluation of the FRKG semantic extension results will ensure the accuracy of the requirement ranking results. Finally, the requirement ranking results proposed in this paper will be compared with other ranking methods to verify its efficiency.

4.1. Case study

The purpose of the case study is to demonstrate how uncertain KRR techniques are used for requirement ranking in real complex systems. UAVs have been rapidly developed in recent years and have been widely applied in various fields, including the military, aerial photography, logistics, agriculture, and more. This rapid development has brought about new user demands that are difficult to precisely grasp. Therefore, this paper focuses on UAVs and conducts a case study.

Through induction and classification, the functional and attribute aspects that users are concerned about in UAVs were extracted and divided into eight categories: flight performance, payload capacity, autonomous navigation capability, remote control ability, camera functionality, lifespan and endurance, safety, and device reliability, totalling 215 functions and attributes. Subsequently, 278 relatively comprehensive user stories were collected from UAV research institutions.

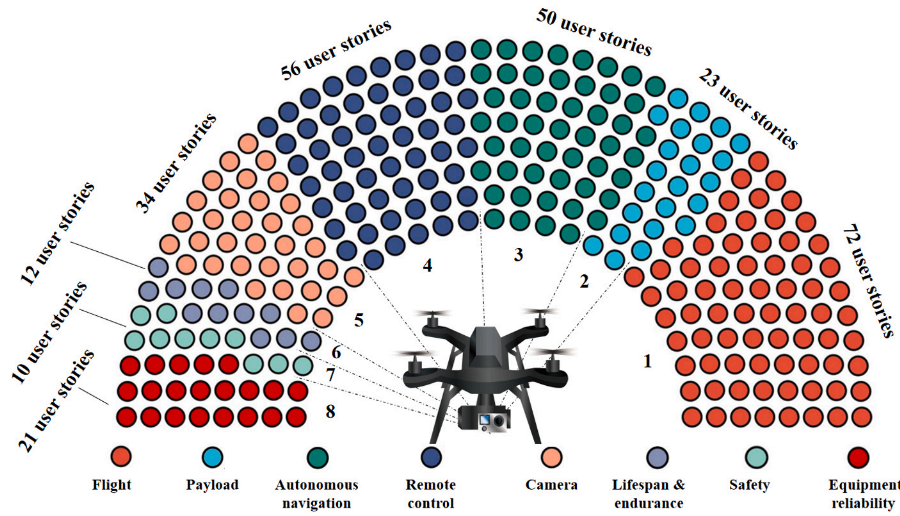

Fig. 6. The 278 user stories in eight categories.

Table 6
The fuzzy semantic attributes of predicates in BKG.

	$\langle satisfy, u\rangle$	$\langle refine, u\rangle$	$\langle derive, u\rangle$	$\langle trace, u\rangle$	$\langle achieve, u\rangle$
FSEO					
FSSO			✓	✓	
FSTO		✓	✓	✓	
FSIO			✓	✓	
			$\langle trace, u\rangle$	$\langle derive, u\rangle$	

Table 7
The semantic relations of different fuzzy predicates.

	$\langle satisfy, u\rangle$	$\langle refine, u\rangle$	$\langle derive, u\rangle$	$\langle trace, u\rangle$	$\langle achieve, u\rangle$
$\langle satisfy, u\rangle$					
$\langle refine, u\rangle$		FSTO	OP_{FSSO}		
$\langle derive, u\rangle$			FSSO FSTO	FSIO	
$\langle trace, u\rangle$		$OP^{(in)}_{FSTO}$	FSIO SP_{FSSO}	FSSO FSTO	
$\langle achieve, u\rangle$					

Fig. 6 displays the distribution of user stories across different functional categories.

Based on the ontology framework in Section 3.1.1 and the collected product attributes and user requirements data in this section, we can construct a BKG. The construction of the BKG is a well-established process and has been implemented in many industrial sectors. However, since this paper aims to achieve uncertain KRR, we first identified and labelled the fuzzy predicates for the relationships in the BKG. The specific results are shown in Table 6. The predicate "From" represents the source of the requirement, and it has accurate annotations in the collected user stories. The predicate "Compose" indicates the product components needed to complete a specific function, which is composed of professional data from within the company and validated by experts. Therefore, these two predicates are defined as deterministic predicates. As the attribute confidence is only involved in multiplication operations in FSTO, there is no need to define additional features for confidence.

Table 6 indicates that FRKG possesses the capability of single-step reasoning and multistep reasoning for the same predicate. To perform multistep reasoning, we need to define the relationships between different fuzzy predicates, as shown in Table 7.

Based on the defined fuzzy predicate attributes and the BKG enrichment method introduced in Section 3.2.1, the resulting FRKG and EFRKG are depicted in Fig. 7. From the graph, it can be observed that FRKG enrichment establishes more relationships between potential edges. The accuracy evaluation of the enrichment results will be further discussed in Section 4.2.

Next, we abstracted the EFRKG into a heterogeneous information network and calculated the scores based on the multicriteria decision-making method introduced in Section 3.2.2. The heatmap of the 278 collected user stories in our case, evaluated based on the five metrics defined in this article, is shown in Fig. 8. Then, the final requirement ranking result is shown as follows. Notably, only the top ten ranked requirement entries are provided here: $Re_{201} \succ Re_{11} \succ Re_{93} \succ Re_{46} \succ Re_{9} \succ Re_{208} \succ Re_{117} \succ Re_{57} \succ Re_{1} \succ Re_{53}$

4.2. Evaluation of FRKG semantic enrichment results

In the previous section, Fig. 7 only displayed the EFRKG compared to FRKG. To more intuitively observe the results of semantic expansion, we first conducted a visualization experiment on our fuzzy semantic representation and reasoning model by extracting a subgraph $FRKG_{child}$ from FRKG, as shown in Fig. 9. Given the semantic equivalence of FSEO, after fuzzy reasoning, entities linked by a given FSEO will have the same topological structure. To ensure visual clarity, we removed some results extended by FSEP.

To analyse the experimental results more clearly, we summarized and analysed the fuzzy semantic expansion results shown in Fig. 9. First, this section defines some formulas to evaluate the rationality and feasibility of our fuzzy semantic representation and reasoning model.

$$Novelty = \frac{|\text{overall\_edge} - \text{original\_edge}|}{|\text{overall\_edge}|} \tag{8}$$

$$ExtraCoverage = \frac{|\text{overall\_size} - \text{original\_size}|}{|\text{original\_size}|} \tag{9}$$

$$Soundness = \frac{|\text{valid\_implicit\_edge}|}{|\text{overall\_edge} - \text{original\_edge}|} \tag{10}$$

Table 8 provides all the statistical data of the experimental measurements. In Table 8, "Novelty" and "ExtraCoverage" indicate that compared to BKG, FRKG can uncover more implicit fuzzy knowledge. Additionally, by judging the semantics of the derived implicit fuzzy knowledge through the fuzzy semantic enrichment method, "Soundness" demonstrates that Algorithm 1 can effectively describe the semantic fuzziness between entities and the changes in fuzzy semantics during the reasoning process. Notably, ensuring the soundness of fuzzy semantic reasoning results requires capturing the fuzziness of the annotated predicates themselves and the relationships between different predicates.

Next, we used a knowledge retrieval approach to evaluate the usability of EFRKG. This evaluation was conducted by analysing the

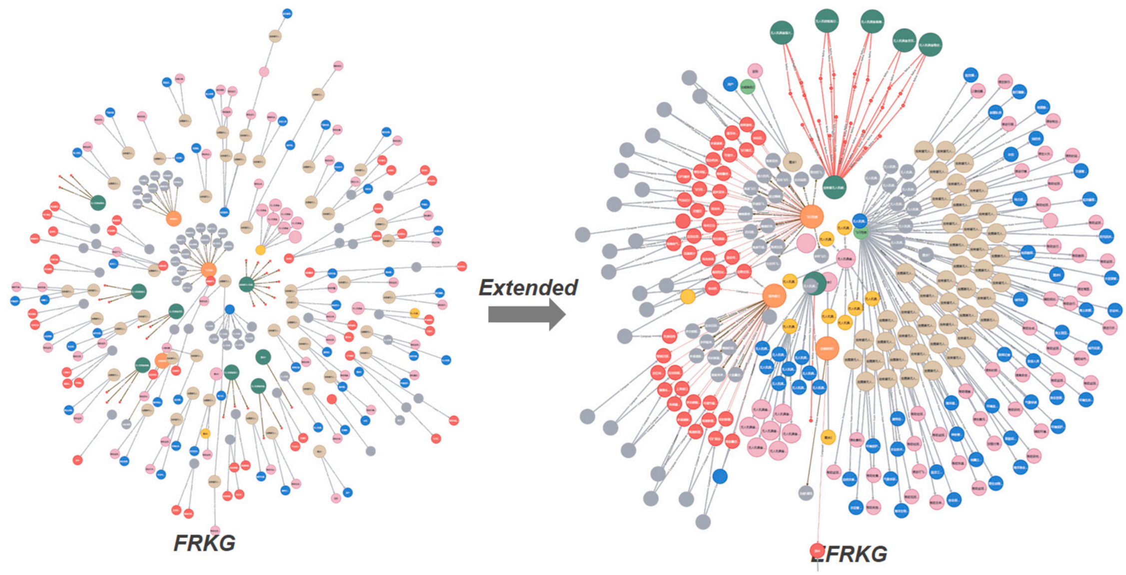

Fig. 7. Visualization of FRKG and EFRKG.

Table 8
The fuzzy semantic enrichment result of Fig. 9.

BKG size	Fuzzy semantic extension results					
	Implicit fuzzy knowledge	Valid implicit fuzzy knowledge	Overall	Novelty	Extra coverage	Soundness
14	22	20	36	55.56%	157%	90.91%

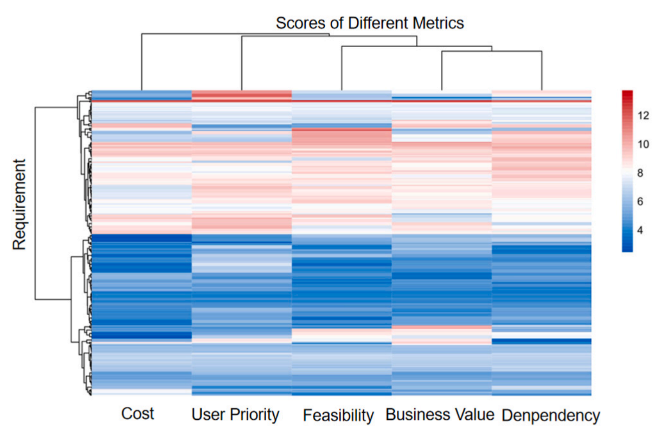

Fig. 8. Heatmap plot of requirement scores.

query results obtained from both BKG and its fuzzy extended version, FRKG and EFRKG. SPARQL is essentially a triple pattern query language containing one or more variables marked with "?" or "$". Unlike traditional keyword-based queries, there are few standard benchmarks available to measure SPARQL-based queries. Therefore, in this section, we provided some queries as new benchmarks, centred around the FRKG shown in Fig. 9, and designed these queries using natural language to further evaluate the fuzzy semantic enrichment model. The complete list of queries is shown in Table 9.

From Table 10, we can observe that if only considering the fuzzy attributes of predicates without EFRKG based on fuzzy reasoning rules, the relationships between potential nodes still cannot be obtained. FRKG incorporates predicate fuzziness and attribute confidence information on the basis of the basic triple representation, which verifies the richness of the EFRKG results.

To verify the accuracy of the semantic enrichment method, we conducted SPARQL queries on the entire FRKG, and precision P, recall R, and F_1 scores were measured. The calculation methods are shown in Eqs. (11), (12), and (13). F_1 combines precision P and recall R to provide a more comprehensive evaluation of the model's performance. The calculation method is shown in (14), where TP represents the number of cases that were evaluated as "qualified" by the model and manually labelled as "qualified" in the experiment; FP represents the number of cases that were evaluated as "qualified" by the model but manually labelled as "unqualified"; FN represents the number of cases that were evaluated as "unqualified" by the model but manually labelled as "qualified"; and TN represents the number of cases that were evaluated as "unqualified" by the model and manually labelled as "unqualified". All results are shown in Fig. 10.

$$P = \frac{TP}{TP + FP} \tag{11}$$

$$R = \frac{TP}{TP + FN} \tag{12}$$

$$R = \frac{TP}{TP + FN} \tag{13}$$

$$F_1 = \frac{2PR}{P + R} \tag{14}$$

4.3. Evaluation of multicriteria requirements ranking

To validate the effectiveness of the multicriteria requirement ranking method proposed in this paper, we plan to design comparative experiments from two aspects. The first perspective involves comparing our method with various excellent multicriteria ranking methods. Many multicriteria ranking methods are not suitable for scenarios involving a large amount of data. For example, ORESTE is commonly used to handle decision matrices containing a small amount of information because the relationships between solutions depend on subjective judgements, and a large number of pairwise comparisons may lead to inconsistent results. In the case of AHP, the table used for the average random index (RI) has a maximum value of 15, and when n is large, there may be significant differences between the judgement matrix and the

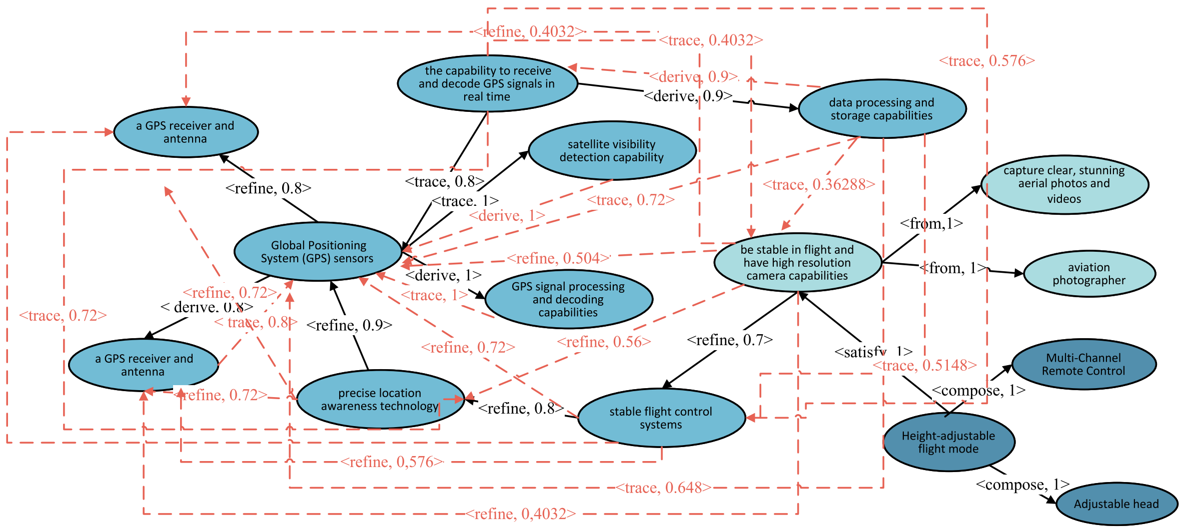

Fig. 9. $\text{FRKG}_{\text{child}}$ visualization.

Table 9
Experimental query list.

Query ID	Query requirements	The corresponding triple
Q1	Is there any precondition for "UAV equipped with GPS receiver and antenna?"	(UAV equipped with GPS receiver and antenna, trace, **?who**)
Q2	Which high-level requirements can be decomposed into the subrequirement "UAV equipped with GPS receiver and antenna?"	(**?who**, refine, UAV equipped with GPS receiver and antenna)
Q3	Does "UAV equipped with GPS receiver and antenna" refine "UAV equipped with real-time reception and decoding of GPS signals?"	(UAV equipped with GPS receiver and antenna, refine,**?who**) (̂**?who** = "UAV equipped with real-time reception and decoding of GPS signals")
Q4	Who is refined by "UAV equipped with precise location sensing technology?"	(UAV equipped with precise location sensing technology, refine, **?who**)
Q9	Does "UAV equipped with data processing and storage functionality" trace "unmanned aerial vehicle equipped with precise location sensing technology?"	(Clark Glymour, trace,**?who**) (̂**?who** = "UAV equipped with precise location sensing technology")

Table 10
Retrieval results in different knowledge graphs.

Query	BKG	FRKG	Extended FRKG
Q1	Null	Null	(UAV equipped with GPS receiver and antenna, ⟨*trace, 0.8*⟩, UAV equipped with GPS sensor)
Q2	Null	Null	(UAV equipped with precise location sensing technology, ⟨*refine, 0.72*⟩, UAV equipped with GPS receiver and antenna) (UAV equipped with stable flight control system, ⟨*refine, 0.576*⟩, UAV equipped with GPS receiver and antenna) (I hope the UAV can fly steadily and have high-resolution camera functionality, ⟨*refine, 0.4032*⟩, UAV equipped with GPS receiver and antenna)
Q3	Null	Null	Null
Q4	(UAV equipped with precise location sensing technology, *refine*, UAV equipped with GPS sensor)	(UAV equipped with precise location sensing technology, ⟨ *refine, 0.9*⟩, UAV equipped with GPS sensor)	(UAV equipped with precise location sensing technology, ⟨*refine, 0.9*⟩, UAV equipped with GPS sensor) (UAV equipped with precise location sensing technology, ⟨*refine, 0.72*⟩, UAV equipped with real-time reception and decoding of GPS signals) (UAV equipped with precise location sensing technology, ⟨*refine, 0.72*⟩, UAV equipped with GPS receiver and antenna)
Q9	Null	Null	(UAV equipped with data processing and storage functionality, ⟨*trace, 0.648*⟩, UAV equipped with precise location sensing technology)

consistency matrix. Therefore, AHP is not suitable for the application of requirement analysis. Consequently, the comparative methods chosen in this paper are TOPSIS, VIKOR, MULTIMOORA, and COPRAS. These methods involve fewer subjective judgements and do not require frequent code parameter modifications due to changes in the data source. The second perspective includes comparing our method with methods that disregard fuzzy semantics and do not perform graph enrichment.

We constructed a standard dataset containing 15 requirements based on the case data in Section 4.1. After analysis by four experts, the ranking results for these 15 requirements are as follows: $R_1 \succ R_6 \succ$

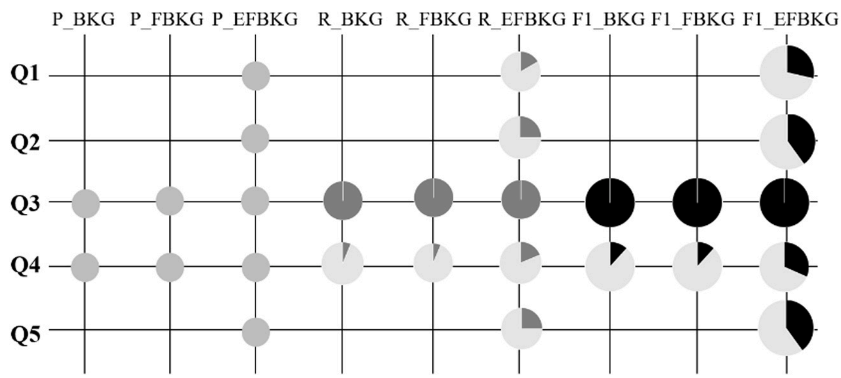

Fig. 10. P, R and F_1 of BKG, FBKG and EFRKG (Grid nodes represent the percentage of that evaluation value, and the absence of a pie chart indicates a KG query result of 0.).

$R_2 \succ R_5 \succ R_{12} \succ R_{13} \succ R_{14} \succ R_{11} \succ R_4 \succ R_8 \succ R_{10} \succ R_5 \succ R_3 \succ R_9 \succ R_7$

Moreover, based on AHP, the weights of the five requirement evaluation criteria were preset as follows: business value: user priority: feasibility: dependency: cost = 0.1: 0.1: 0.3: 0.3: 0.2. Next, the standard dataset containing 15 requirements was divided into multiple experimental sets, each containing 5 requirements, totalling 3003 sets. TOPSIS, VIKOR, MULTIMOORA, and COPRAS were used for separately ranking each experimental set. Notably, the original dataset, as shown in Table A.3, contains four different data types: real values, interval values, semantic values, and symbolic values. Classical multicriteria ranking methods do not propose methods to handle semantic and symbolic data, so the input data for classic multicriteria ranking methods are shown in Table A.4, which retains the real values, interval values (calculated using the interval mean), and symbolic values (converted into real-form ranks) from Table A.3. Next, to verify the effectiveness of FRKG and its extension methods, based on Table A.4, we will make comparisons from three dimensions, BKG, FRKG, and EFRKG, which involve extracting nodes and relationships mined in BKG, FRKG, and EFRKG, constructing judgement matrices, and using the GRA method for ranking. The evaluation experiment takes as input sets 5 requirements along with their corresponding evaluation criteria. Different requirement ranking methods are applied, and the output is the sorted sequence of requirements. The experiment reveals that the probability of classical multicriteria ranking methods producing results that are completely consistent with expert recommendations is low. Therefore, we only compare the best and worst values of the output sequence with the expert ranking results. If the two rankings are correct, it is considered a positive outcome; otherwise, it is a negative outcome. Finally, the accuracy of the prediction results for each method is summarized in Fig. 11:

As shown in Fig. 11, the evaluation results of the four classic multicriteria ranking methods are similar. Among them, COPRAS uses fuzzy relationships to determine attribute weights, thus achieving the best performance. Although VIKOR can also determine attribute weights through uncertainty measures, it preset the attribute weights during the experiments, resulting in performance similar to TOPSIS and MULTIMOORA. The gap between the classic methods and the method proposed in this paper is partly due to the introduction of more non-numeric attributes for calculation, especially the interdependencies between requirements. For example, if more related requirements are discovered for a particular requirement, it is more likely to be defined as an important requirement, aligning more with expert judgement. This factor is crucial in requirement analysis but is not considered in classic multicriteria sorting methods.

Furthermore, the accuracy of the methods applied to BKG+GRA, FRKG+GRA, and EFRKG+GRA steadily improves. This improvement is attributed to the discovery of more relationships between nodes, enriching the information in the requirement-criterion table and subsequently increasing the accuracy of the results. Notably, the improvement from FRKG+GRA to EFRKG+GRA is greater than the improvement from BKG+GRA to FRKG+GRA. This finding is because, compared to the introduction of fuzzy values, establishing relationships between more potential edges can significantly enhance sorting performance. Therefore, future research will delve into methods for mining relationships between nodes in greater depth.

5. Conclusions

Considering the complexity of requirement analysis work and the ambiguity of data sources, this paper integrates relevant knowledge from the field of knowledge management to design a multicriteria requirement ranking method based on uncertain KRR. Specifically, we start from the context of requirement analysis, construct an ontology, define an uncertain knowledge representation method that adapts to data fuzziness, and explicitly define logical reasoning rules to reveal potential relationships among requirement analysis objects. Additionally, we establish requirement evaluation metrics to achieve multicriteria ranking. Next, we emphasize the advancement of our approach from three aspects:

(1) High Confidence: Our method in this paper integrates various KRR methods, including ontology construction, logical reasoning, KGs, and probability theory, to overcome the limitations of traditional methods in handling fuzzy semantics. Furthermore, the introduction of confidence or probability information for triples provides reliable confidence assessment for reasoning results, thereby enhancing result credibility.

(2) Interpretability: Compared to classic multicriteria ranking methods, our approach excels in explaining sorting results. In addition to a greater variety of relationship types, we have also explicitly defined logical rules that enable us to trace inference paths, understanding the basis for each ranking decision. This makes the decision-making process more transparent and rational.

(3) Scalability: Our method in this paper demonstrates outstanding scalability. On the one hand, when introducing new requirements, we can rely on the ontology to establish new relationships in the KG without the need for predefined standards. On the other hand, our method takes into account the grey nature of requirement analysis systems, requiring no prior knowledge and enabling rapid startup, greatly simplifying the requirement sorting calculation process. Therefore, this method efficiently handles large-scale datasets without compromising performance.

In summary, uncertain KRR provides richer reference information for requirement ranking. Its flexibility, accuracy, and efficiency make it a powerful tool for addressing practical requirement sorting problems, providing robust support and guidance to decision-makers. To augment the effectiveness of the proposed method, future research can focus on two key aspects. First, we will explore the automatic generation of personalized logical reasoning rules tailored to the application context. This approach aims to unveil additional potential relationships,

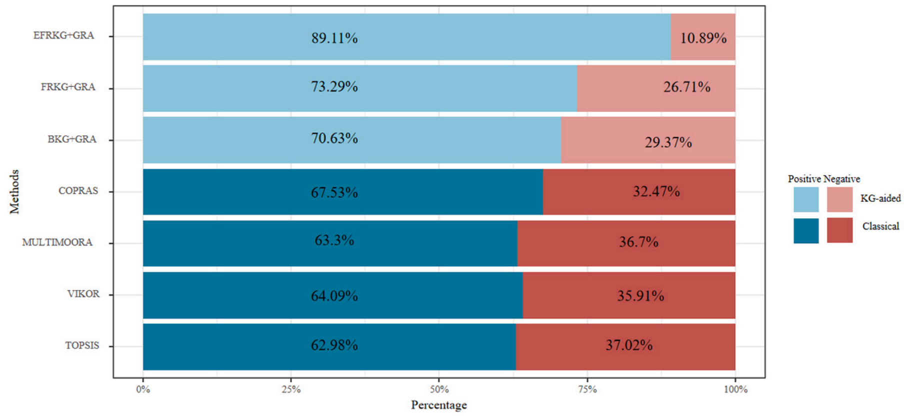

Fig. 11. Comparison of seven multicriteria ranking methods.

Table A.1
The common semantic attributes for the predicate.

Semantic attribute for predicate p	Symbol	Logical explanation
Semantic equal predicate (p)	SEP	$(x, p, y) \Rightarrow x = y$
Semantic symmetry predicate (p)	SSP	$(x, p, y) \Rightarrow (y, p, x)$
Semantic transmission predicate (p)	STP	$(x, p, y) \wedge (y, p, z) \Rightarrow (x, p, z)$
Semantic inversion predicates ($p_1 \neq p_2$)	SIP	$(x, p_1, y) \Rightarrow (y, p_2, x)$

Table A.2
The simple reasoning rules for fuzzy predicate.

Fuzzy semantic operation	Symbol	Logical explanation
Fuzzy semantic equal operation	FSEO	$(x, \langle p, \mu_p \rangle, y) \Rightarrow \mu_p = 1 \wedge x = y$
Fuzzy semantic symmetry operation	FSSO	$(x, \langle p, \mu_p \rangle, y) \Rightarrow (y, \langle p, \mu_p \rangle, x)$
Fuzzy semantic transmission operation	FSTO	$(x, \langle p, \mu_p(x, y) \rangle,) \wedge (y, \langle p, \mu_p(y, z) \rangle, z) \Rightarrow$ $(x, \langle p, \mu_p(x, y) \cdot \mu_p(y, z) \rangle, z)$
Fuzzy semantic inverse operation	FSIO	$(x, \langle p_1, \mu_{p1} \rangle, y) \Rightarrow (y, \langle p_2, \mu_{p2} \rangle, x)$^$(\mu_{p1} = \mu_{p2})$

Table A.3
Example of input data format for our proposed method.

	Business value			User priority		Feasibility				Dep.	Cost	
	G.	Effici.	Import.	Job.	Import.	Inter.	Compo.	V.	Import.	Re.	Value	Import.
R1	G1	80%–87%	Low	Farmer...	High	RS-232…	Battery ...	5	High	R2\…	820	High
R2	G11	75%–82%	Medium	Aerial Photo grapher...	High	I2C…	Gyroscope..	4	High	R5…	1350	Medium
R3	G3	70%–77%	Very high	Fire fighter...	Very high	UART…	Motor...	1	Medium	N/A	670	Very high
R4	G2	85%–92%	High	Power inspector...	Very high.	PWM…	Obstacle Avoidance sensor...	1	Very high	R9\…	1300	High
……												
R15	G7	74%–81%	Low	Realtor...	Low	N/A	HDTV camera…	4	High	R4\…	1500	High

thereby enhancing result accuracy. Second, integration with advanced uncertain KRR and multicriteria decision-making methods could be pursued. This may involve automatically adjusting indicator weights based on the context, fostering a more comprehensive approach to decision-making.

CRediT authorship contribution statement

Yufeng Ma: Writing – original draft, Writing – review & editing. **Yajie Dou:** Funding acquisition. **Xiangqian Xu:** Data curation. **Jiang Jiang:** Investigation. **Kewei Yang:** Resources. **Yuejin Tan:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix

See Tables A.1–A.4.

Table A.4
Example of input data format for classical multicriteria ranking.

	Business value		User priority	Feasibility		Cost	
	Efficiency	Importance	Importance	Vendor	Importance	Value	Importance
R1	0.835	1	3	5	3	820	4
R2	0.785	2	3	4	3	1350	2
R3	0.735	4	4	1	2	670	4
R4	0.885	3	4	1	4	1300	3
……							
R15	0.775	1	1	4	3	1500	3

References

[1] M. Cai, Y. Tan, B. Ge, Y. Dou, G. Huang, Y. Du, PURA: A product-and-user oriented approach for requirement analysis from online reviews, IEEE Syst. J. 16 (1) (2022) 566–577, http://dx.doi.org/10.1109/JSYST.2021.3067334.

[2] D. Falessi, G. Cantone, G. Canfora, Empirical principles and an industrial case study in retrieving equivalent requirements via natural language processing techniques, IEEE Trans. Softw. Eng. 39 (1) (2013) 18–44, http://dx.doi.org/10.1109/TSE.2011.122.

[3] C. Arora, M. Sabetzadeh, L. Briand, F. Zimmer, Automated extraction and clustering of requirements glossary terms, IEEE Trans. Softw. Eng. 43 (10) (2017) 918–945, http://dx.doi.org/10.1109/TSE.2016.2635134.

[4] W. Maalej, H. Nabil, Bug report, feature request, or simply praise? On automatically classifying app reviews, in: 2015 IEEE 23rd International Requirements Engineering Conference (RE), 2015, pp. 116–125, http://dx.doi.org/10.1109/RE.2015.7320414.

[5] E. Guzman, M. Ibrahim, M. Glinz, A little bird told me: Mining tweets for requirements and software evolution, in: 2017 IEEE 25th International Requirements Engineering Conference (RE), 2017, pp. 11–20, http://dx.doi.org/10.1109/RE.2017.88.

[6] M. Robeer, G. Lucassen, J.M.E.M. van der Werf, F. Dalpiaz, S. Brinkkemper, Automated extraction of conceptual models from user stories via NLP, in: 2016 IEEE 24th International Requirements Engineering Conference (RE), 2016, pp. 196–205, http://dx.doi.org/10.1109/RE.2016.40.

[7] T. Breaux, A. Antón, Analyzing regulatory rules for privacy and security requirements, IEEE Trans. Softw. Eng. 34 (1) (2008) 5–20, http://dx.doi.org/10.1109/TSE.2007.70746.

[8] X. Zhang, Z. Ma, Fuzzy RDF knowledge graph embeddings through vector space model, IEEE Trans. Fuzzy Syst. 31 (3) (2023) 835–844, http://dx.doi.org/10.1109/TFUZZ.2022.3190633.

[9] S. Ji, S. Pan, E. Cambria, P. Marttinen, P.S. Yu, A survey on knowledge graphs: Representation, acquisition, and applications, IEEE Trans. Neural Netw. Learn. Syst. 33 (2) (2022) 494–514, http://dx.doi.org/10.1109/TNNLS.2021.3070843.

[10] P. Sae-Ueng, S. Skrbic, Priority fuzzy database management system implementation based on extensions to the XQuery language, J. Intell. Fuzzy Systems 38 (4) (2020) 4107–4118, URL: https://search-ebscohost-com-s.libyc.nudt.edu.cn:443/login.aspx?direct=true&db=asn&AN=143006123&lang=zh-cn&site=ehost-live.

[11] Z. Ma, L. Yan, Data modeling and querying with fuzzy sets: A systematic survey, Fuzzy Sets Syst. 445 (2022) 147–183, URL: https://search-ebscohost-com-s.libyc.nudt.edu.cn:443/login.aspx?direct=true&AuthType=cookie,ip,shib&db=asn&AN=158208251&lang=zh-cn&site=eds-live.

[12] I. Huitzil, F. Bobillo, Fuzzy ontology datatype learning using Datil, Expert Syst. Appl. 228 (2023) 120299, https://doi-org-s.libyc.nudt.edu.cn:443/10.1016/j.eswa.2023.120299, URL: https://www-sciencedirect-com-s.libyc.nudt.edu.cn:443/science/article/pii/S0957417423008011.

[13] L. Bai, J. Wang, L. Zhu, Fixing the inconsistencies of continuous changing operations in fuzzy spatiotemporal RDF graph, Appl. Soft Comput. 131 (2022) 109767, http://dx.doi.org/10.1016/j.asoc.2022.109767, URL: https://linkinghub.elsevier.com/retrieve/pii/S156849462200816X.

[14] L. Bai, N. Li, H. Bai, An integration approach of multi-source heterogeneous fuzzy spatiotemporal data based on RDF, J. Intell. Fuzzy Systems 40 (1) (2021) 1065–1082, http://dx.doi.org/10.3233/JIFS-201258, URL: https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/JIFS-201258.

[15] T. Eiter, G. Kern-Isberner, A brief survey on forgetting from a knowledge representation and reasoning perspective, in: KI-Künstliche Intelligenz, Vol. 33, Springer, 2019, pp. 9–33.

[16] Q. Zhang, Q. Yao, Dynamic uncertain causality graph for knowledge representation and reasoning: Utilization of statistical data and domain knowledge in complex cases, IEEE Trans. Neural Netw. Learn. Syst. 29 (5) (2018) 1637–1651, http://dx.doi.org/10.1109/TNNLS.2017.2673243.

[17] N. Angelopoulos, J. Cussens, Distributional logic programming for Bayesian knowledge representation, Internat. J. Approx. Reason. 80 (2017) 52–66.

[18] R. Quintero, M. Torres-Ruiz, R. Menchaca-Mendez, M.A. Moreno-Armendariz, G. Guzman, M. Moreno-Ibarra, Dis-c: conceptual distance in ontologies, a graph-based approach, Knowl. Inf. Syst. 59 (1) (2019) 33–65.

[19] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U.R. Acharya, et al., A review of uncertainty quantification in deep learning: Techniques, applications and challenges, Inf. Fusion 76 (2021) 243–297.

[20] Y. Khalifa, D. Mandic, E. Sejdić, A review of Hidden Markov models and Recurrent Neural Networks for event detection and localization in biomedical signals, Inf. Fusion 69 (2021) 52–72.

[21] S. Russell, Unifying logic and probability, Commun. ACM 58 (7) (2015) 88–97.

[22] W. Yue, J. Chai, X. Wan, Y. Xie, X. Chen, W. Gui, Root cause analysis for process industry using causal knowledge map under large group environment, Adv. Eng. Inform. 57 (2023) 102057.

[23] K. Lyu, Y. Tian, Y. Shang, T. Zhou, Z. Yang, Q. Liu, X. Yao, P. Zhang, J. Chen, J. Li, Causal knowledge graph construction and evaluation for clinical decision support of diabetic nephropathy, J. Biomed. Inform. 139 (2023) 104298.

[24] J. Li, Y. Horiguchi, T. Sawaragi, Counterfactual inference to predict causal knowledge graph for relational transfer learning by assimilating expert knowledge–relational feature transfer learning algorithm, Adv. Eng. Inform. 51 (2022) 101516.

[25] D. Dubois, H. Prade, E.P. Klement, Fuzzy Sets, Logics and Reasoning About Knowledge, Vol. 15, Springer Science & Business Media, 2013.

[26] E.C. Tsang, D.S. Yeung, J.W. Lee, D. Huang, X. Wang, Refinement of generated fuzzy production rules by using a fuzzy neural network, IEEE Trans. Syst. Man Cybern. B 34 (1) (2004) 409–418.

[27] H.-C. Liu, J.-X. You, Z. Li, G. Tian, Fuzzy Petri nets for knowledge representation and reasoning: A literature review, Eng. Appl. Artif. Intell. 60 (2017) 45–56.

[28] W. Yue, W. Gui, Y. Xie, Experiential knowledge representation and reasoning based on linguistic Petri nets with application to aluminum electrolysis cell condition identification, Inform. Sci. 529 (2020) 141–165.

[29] W. Yue, X. Wan, S. Li, H. Ren, H. He, Simplified neutrosophic petri nets used for identification of superheat degree, Int. J. Fuzzy Syst. 24 (8) (2022) 3431–3455.

[30] W. Yue, L. Hou, X. Wan, X. Chen, W. Gui, Superheat degree recognition of aluminum electrolysis cell using unbalance double hierarchy hesitant linguistic Petri nets, IEEE Trans. Instrum. Meas. (2023).

[31] A. Harth, K. Hose, R. Schenkel, Linked Data Management, CRC Press, 2014.

[32] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2014.

[33] C. Rolland, C. Proix, A natural language approach for requirements engineering, in: Seminal Contributions to Information Systems Engineering: 25 Years of CAiSE, Springer, 2013, pp. 35–55.

[34] D.M. Fernández, S. Wagner, M. Kalinowski, M. Felderer, P. Mafra, A. Vetrò, T. Conte, M.-T. Christiansson, D. Greer, C. Lassenius, et al., Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice, Empir. Softw. Eng. 22 (2017) 2298–2338.

[35] J. Nanduri, Y. Jia, A. Oka, J. Beaver, Y.-W. Liu, Microsoft uses machine learning and optimization to reduce e-commerce fraud, Inf. J. Appl. Anal. 50 (1) (2020) 64–79.

[36] E. Muñoz, V. Nováček, P.-Y. Vandenbussche, Facilitating prediction of adverse drug reactions by using knowledge graphs and multi-label learning models, Brief. Bioinform. 20 (1) (2019) 190–202.

[37] Y. Sun, Z. Tian, M. Li, C. Zhu, N. Guizani, Automated attack and defense framework toward 5G security, IEEE Netw. 34 (5) (2020) 247–253.

[38] Y. Liu, Q. Zeng, J. Ordieres Meré, H. Yang, Anticipating stock market of the renowned companies: A knowledge graph approach, Complexity 2019 (2019).

[39] J. Long, Z. Chen, W. He, T. Wu, J. Ren, An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in Chinese stock exchange market, Appl. Soft Comput. 91 (2020) 106205.
[40] Z. Wang, C.-H. Chen, P. Zheng, X. Li, L.P. Khoo, A graph-based context-aware requirement elicitation approach in smart product-service systems, Int. J. Prod. Res. 59 (2) (2021) 635–651.
[41] Z. Wang, C.-H. Chen, P. Zheng, X. Li, L.P. Khoo, A novel data-driven graph-based requirement elicitation framework in the smart product-service system context, Adv. Eng. Inform. 42 (2019) 100983.
[42] P. Achimugu, A. Selamat, R. Ibrahim, M.N. Mahrin, A systematic literature review of software requirements prioritization research, Inf. Softw. Technol. 56 (6) (2014) 568–585.
[43] O. Ibrahim, A. Nosseir, A combined AHP and source of power schemes for prioritising requirements applied on a human resources, in: MATEC Web of Conferences, Vol. 76, EDP Sciences, 2016, p. 04016.
[44] S. Tumulur, K.V. Kumar, Y. Pavan, A machine learning approach to software requirements prioritization, Int. J. Sci. Eng. Technol. Res. 3 (2014) 6409–6416.
[45] F. Shao, R. Peng, H. Lai, B. Wang, DRank: A semi-automated requirements prioritization method based on preferences and dependencies, J. Syst. Softw. 126 (2017) 141–156.
[46] A. Hudaib, R. Masadeh, M.H. Qasem, A. Alzaqebah, et al., Requirements prioritization techniques comparison, Mod. Appl. Sci. 12 (2) (2018) 62.
[47] X.F. Liu, Y. Sun, C.S. Veera, Y. Kyoya, K. Noguchi, Priority assessment of software process requirements from multiple perspectives, J. Syst. Softw. 79 (11) (2006) 1649–1660.
[48] J. Kim, Y. Geum, Y. Park, Integrating customers' disparate technology readiness into technological requirement analysis: an extended Kano approach, Total Qual. Manag. Bus. Excell. 28 (5–6) (2017) 678–694.
[49] T.M. Fehlmann, New Lanchester theory for requirements prioritization, in: 2008 Second International Workshop on Software Product Management, IEEE, 2008, pp. 35–40.
[50] K.E. Wiegers, J. Beatty, Software Requirements, Pearson Education, 2013.
[51] K. Curcio, T. Navarro, A. Malucelli, S. Reinehr, Requirements engineering: A systematic mapping study in agile software development, J. Syst. Softw. 139 (2018) 32–50.
[52] P. Li, X. Wang, H. Liang, S. Zhang, Y. Zhang, Y. Jiang, Y. Tang, A fuzzy semantic representation and reasoning model for multiple associative predicates in knowledge graph, Inform. Sci. 599 (2022) 208–230.
[53] L. Wang, L. Yu, Z. Ni, A novel IVIF QFD considering both the correlations of customer requirements and the ranking uncertainty of technical attributes, Soft Comput. 26 (9) (2022) 4199–4213.

Yufeng Ma was born in 1997. She received a B.E. degree in industrial engineering from the Northeast Forestry University, Harbin, Heilongjiang, China, in 2020. She is currently pursuing a Ph.D. degree in management science and engineering in National University of Defense Technology. Her research interests include intelligent requirements engineering, knowledge graph, knowledge representation and reasoning, and knowledge embedding.

Yajie Dou was born in 1987. He received his Ph.D. degrees in management science and engineering from National University of Defense Technology, Changsha, Hunan, China, in 2016. He is currently an associate Professor of College of Systems Engineering, National University of Defense Technology. He was a visiting scholar at Katholieke Universiteit Leuven, Leuven, Belgium from 2014 to 2015. His research interests include system-of-systems architecture and engineering management, intelligent requirements engineering, complex project management.

Xiangqian Xu was born in 1995. He received the B.E. degree in systems engineering from the National University of Defense Technology, Changsha, Hunan, China, in 2018. He is currently pursuing the Ph.D. degree in management science and engineering in National University of Defense Technology. His research interests include intelligent requirements engineering, complex project management, defense acquisition, and system-of-systems requirement modelling.

Jiang Jiang was born in 1981. He received his Ph.D. degrees in management science and engineering from National University of Defense Technology, Changsha, Hunan, China, in 2011. He is currently an associate Professor of College of Systems Engineering, National University of Defense Technology. He was a visiting scholar at Harvard University, Boston, MA, USA from 2018 to 2019. His research interests include evidential reasoning, uncertainty decision-making, and risk analysis.

Kewei Yang was born in 1977. He received his Ph.D. degree in management science and engineering from the National University of Defense Technology, Changsha, Hunan, China, in 2004. He is currently a Professor of College of Systems Engineering, National University of Defense Technology. He was a Visiting Scholar at the Department of Computer Science, University of York, York, U.K. from 2011 to 2012. His current research interests include intelligent agent simulation, defense acquisition, and system-of-systems requirement modelling.

Yuejin Tan was born in 1958. He received his M.E. degree in systems engineering from the National University of Defense Technology, Changsha, Hunan, China, in 1985. He is currently a Professor of College of Systems Engineering, National University of Defense Technology. He was a Visiting Scholar at the Complex Systems Management Centre, Cranfield University, Bedfordshire, U.K. from 1993 to 1994. His research interests include system-of-systems architecture and engineering management, intelligent requirements engineering, complex project management, defense acquisition, and system-of-systems requirement modelling.