International Journal of Decision Support System Technology
Volume 15 • Issue 1

Fuzzy Distance-Based Approach for the Assessment and Selection of Programming Languages:
Fuzzy-Based Hybrid Approach for Selection of PL

Rakesh Garg, Amity University, Noida, India
https://orcid.org/0000-0001-7037-4209

Supriya Raheja, Amity University, Noida, India*

ABSTRACT

The desire to develop software with more and more functionalities to make human work easier pushes the industry towards developing various programming languages. The existence of the various programming languages in today's scenario raises the need for their evaluation. The motive of this research is the development of a deterministic decision support framework to solve the object-oriented programming (OOP) language's selection problem. In the present study, OOP language's selection problem is modeled as a multi-criteria decision-making, and a novel fuzzy-distance based approach is anticipated to solve the same. To demonstrate the working of developed framework, a case study consisting of the selection of seven programming languages is presented. The results of this study depict that Python is the most preferred language compared to other object-oriented programming languages. Selection of OOP languages helps to select the most appropriate language, which provides better opportunities in the business domain and will result in high success for engineering students.

KEYWORDS

Computer Engineering, DBA, Decision Support framework, Engineering Education, Fuzzy, Hybrid, Multi-criteria Decision Making, Object Oriented, Validation

Introduction

In the era of technology, computer Technocrats in engineering education desire to excel by being at par with the latest technologies and digital advancement. Programming languages are the necessary and inevitable tools to design new algorithms and software systems. The efficiency of these systems depends on the programmer and features provided by the programming language. Many programming languages are prevalent in the digital world, today offering a wide range of features. With the advancement in technology, software companies are upgrading their systems and programming languages that provide the best features applied to their business domain. Programming languages form the foundation of any software system, and their presence can be traced back to several decades. 'FORTRAN,' 'COBOL,' and 'LISP' were the oldest procedure-oriented programming languages developed during 1957-1960. In the 1970s, another procedure-oriented language, 'PASCAL,' came

DOI: 10.4018/IJDSST.315761 *Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

into existence, followed by the 'C' language that became the base language for many new prominent languages. Further, many new languages like 'C++,' 'Java,' 'Pearl,' 'Ruby,' 'Python,' 'PHP' etc. came into existence for the software development known as OOP languages (Cook, 1986; Parker et al., 2006; Henderson & Zom, 1994; Bosch, 1997; Lesani et al., 2014; Maplesden et al., 2015; Onu F.U et al., 2016; Anfurrutia et al., 2017; Yadav et al.,2017).

Further, to get placed in the best position in software companies, engineering students need to know the latest trends in programming languages. OOP languages occupy a significant position due to their modularity and reusability. There are many OOP languages present with a different set of features. So, it becomes a tedious task for the students who aspire to become a programmer to narrow their decision to a single OOPL that will set their career path. Further, selecting a suitable OOP language may depend on many conflicting attributes. Hence, the task of choosing an OOP language from the trending languages can be considered as an MCDM problem. MCDM is an optimization technique that helps to deduce a single optimum solution from the set of options available (Mardani et al.,2017; Biswas et al., 2021; Goswami et al., 2021; Garg et al., 2021; Chodha et al., 2022; Bansal et al., 2022). Further, the high use of various MCDM approaches is observed for solving many selection problems as university/school selection, teachers evaluation and selection, e-learning website selection, and funds allocation in the field of Education (Erdoğan, & Kaya, 2014; Baykasoglu & Durmusoglu, 2014; Gürbüz, & Albayrak, 2014; Chiang, 2015; Karmaker & Saha, 2015; Jain et al., 2015; Jain et al., 2016; Chang, & Wang, 2016; Chakraborty et al., 2017; Özdemir, & Tüysüz, 2017; Garg & Jain, 2017; Garg, 2017; Handoko et al., 2017; Garg et al., 2018; Hasan et al., 2021; Ghorui et al., 2021).

In this study, the FDBA method is proposed for solving the problem of OOP language selection based on ten selection parameters. The proposed FDBA method integrates fuzzy set theory (FST) and Distance-based approach (DBA). Here, fuzzy set theory is used to get the priority weights of selection criteria and the performance rating of OOP languages concerning selection criteria. DBA is used to get the comprehensive ranking of OOP languages for their selection purpose.

The rest of the paper is structured as follows: Section 2 highlights the selection criteria and MCDM methods used by the researchers to solve the present selection problem, whereas section 3 describes the proposed MCDM method, i.e., FDBA and the developed decision support framework. An illustrated example for OOP language selection is provided in section 4, followed by methodology validation in section 5. The results are discussed in section 6, followed by the significance of the FDBA in section 7. The conclusion and future scope of the research are given in section 8.

Literature Review

The present study focuses on implementing an integrated MCDM approach to solve the programming languages selection problem. A comprehensive literature review is carried out to get detailed information about the various selection criteria and methodologies used in the past. This section is divided into two subsections as (i) Related work and (ii) Motivation for present research.

Related Work

In 1994, authors modeled the programming language selection as an MCDM problem and compared four different languages which majorly supports the inheritance, dynamic dispatch, code reuse and information hiding based on five selection criteria: compiler, optimized time, compile time, object size, and binary size (Henderson & Zorn, 1994). Al Ahmar (2010) presented the prototype of an expert system that supports software project managers and software engineers in selecting the appropriate software development methodology. However, the author discussed on the development methodologies and does not discuss regarding programming languages.

In the contemporary work, Parker et al. (2006) introduced some more selection criteria and implemented the Analytical Hierarchical Process (AHP) approach to solve the programming language selection problem. AHP mainly works on comparing the alternatives to each other (Bakır et al., 2021;

17 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/article/fuzzy-distance-based-approach-for-the-assessment-and-selection-of-programming-languages/315761

Related Content

A Literature Survey for Hazardous Materials Transportation
Serpil Eroland Zafer Yilmaz (2016). *Using Decision Support Systems for Transportation Planning Efficiency (pp. 371-393).*
www.irma-international.org/chapter/a-literature-survey-for-hazardous-materials-transportation/135405

Supporting Collaborative Multi-Criteria Evaluation: The VIP Analysis Plug-In for Decision Deck
João N. Clímaco, João A. Costa, Luis C. Diasand Paulo Melo (2009). *International Journal of Decision Support System Technology (pp. 1-15).*
www.irma-international.org/article/supporting-collaborative-multi-criteria-evaluation/37430

Applying Agility and Living Service Systems Thinking to Enterprise Architecture
Asif Qumer Gill (2017). *Decision Management: Concepts, Methodologies, Tools, and Applications (pp. 487-502).*
www.irma-international.org/chapter/applying-agility-and-living-service-systems-thinking-to-enterprise-architecture/176768

Debiasing Decision Makers Through Knowledge Management
Meliha Handzic (2008). *Encyclopedia of Decision Making and Decision Support Technologies (pp. 134-147).*
www.irma-international.org/chapter/debiasing-decision-makers-through-knowledge/11249

Quantitative Concession Behavior Analysis and Prediction for Decision Support in Electronic Negotiations
Réal A. Carbonneau, Rustam Vahidovand Gregory E. Kersten (2014). *International Journal of Decision Support System Technology (pp. 16-30).*
www.irma-international.org/article/quantitative-concession-behavior-analysis-and-prediction-for-decision-support-in-electronic-negotiations/124310