
F-ONTOCOM: A Fuzzified Cost Estimation Approach for Ontology Engineering

Sonika Malik^{1,*} and Sarika Jain²

¹*Department of IT, Maharaja Surajmal Institute of Technology, New Delhi;
National Institute of Technology, Kurukshetra, Research Scholar, New Delhi, India*

²*Department of Computer Applications, NIT, Kurukshetra, New Delhi, India
E-mail: sonika.malik@gmail.com; jasarika@nitkkr.ac.in*

**Corresponding Author*

Received 18 January 2021; Accepted 22 July 2021;
Publication 22 October 2021

Abstract

Estimating effort is an essential prerequisite for the wide-scale dispersal of ontologies. Not much attention has yet been paid to this essential aspect of ontology building. To date, ONTOCOM is the most prominent model for ontology cost estimation. Many factors influencing the building cost of an ontology are depicted by linguistic terms like Very High, High, ... and so on; making them vague and indistinct. This fuzziness is quite uncertain and must be taken into consideration. The available effort estimation models do not consider the uncertainty of fuzziness. In this work, we propose an effort estimation methodology for ontology engineering using Fuzzy Logic i.e. F-ONTOCOM (Fuzzy-ONTOCOM) to overcome of uncertainty and imprecision. We have defined the corresponding Fuzzy sets for each effort multiplier and its associated linguistic value, and represented the same by triangular membership functions. F-ONTOCOM is applied to a dataset of 148 ontology projects and evaluated over various evaluation criteria. F-ONTOCOM outperforms the existing effort-estimation models; it has been concluded that F-ONTOCOM improves the cost estimation accuracy and estimated cost is very close to actual cost.

Journal of Web Engineering, Vol. 20.7, 2169–2198.

doi: 10.13052/jwe1540-9589.2076

© 2021 River Publishers

Keywords: Ontology engineering; effort estimation; fuzzy logic, ontology cost model, uncertainty, ONTOCOM.

1 Introduction

The popularity of ontologies and their large-scale dissemination is crossing the boundaries of the academic and research community. Ontologies are the controlled vocabularies that serve as a schema for specifying the terms of the domain of interest. They serve as the semantic data model for knowledge management, information retrieval, information integration, semantic web applications, and intelligent information systems. Knowledge Engineers build ontologies in cooperation with the domain experts with the process generally termed as Ontology Engineering. Ontology Engineering is moving from a pure research topic to real applications. The wide range of projects involving large industry sectors as well as the growing interest of Small and Medium Enterprises seek to be consulted in this field [1, 45]. The availability of demonstrated techniques is an essential condition for all such efforts, which will allow for efficient production of high-quality ontology, whether through reuse, new building, or automatic extraction. During the past decades, many methods have emerged to develop specific application-based ontologies [2]. By now, several methodologies, languages, and tools for building ontologies are already standardized.

In addition to the technical and organizational feasibility of any artefact, its economic feasibility is also of utmost importance. This growth in the wide acceptance of ontologies will still increase if the knowledge engineer can perform effective cost-benefit analysis of the ontology engineering process. To build and maintain large-scale ontologies requires not only technology and tools to support the development process but also means to estimate the overall effort [3]. Effort estimation allows companies to know how much effort is required to create an application on time and within budget. There are so many techniques available from the last decades for effort estimation of software development projects. These are mainly grouped into two categories: algorithmic and non-algorithmic techniques. In algorithmic techniques, a single formulation with constant values can be used. It includes arithmetic calculations derived from the analysis of historical projects. An array of models have been developed using algorithmic techniques, like the COCOMO model, Putnam Model, and Function Point Analysis Model [7, 34, 35]. These traditional approaches lack in terms of efficiency and robustness. These approaches are continuously improved but still not precise

as the non-algorithmic models are. The non-algorithmic methods are seen as highly sophisticated, as artificial intelligence is incorporated in reaching their findings. They are based on soft computing techniques such as artificial neural networks, fuzzy logic models, and genetic algorithms. There are some methods based on non-algorithmic techniques like analogy method, Delphi Method, Top-down Estimation method, and Bottom-up method. Non-algorithmic techniques are currently used for the effort estimation of ontology engineering. Because the empirical information is not anticipated to be complete, certain constraints of the individual methods are likely to be overcome by the mixture of these methods. Currently, based on existing case studies, it may be possible to analyse the key elements affecting the effort of the ontology engineering process. The compilation of some effort multipliers is an important step towards developing an ontology engineering cost estimation tool.

The first effective and also the latest attempt to effort estimation of ontology building, reuse and maintenance is the ONTOCOM [5, 6] approach. There is always refinement and adjustment in the ONTOCOM as more information is available on man-month effort spent in growing genuine ontologies. It is required to handle uncertainty in the ONTOCOM model [44]. Although ONTOCOM's evaluation methodology takes into account the uncertainties of randomness; however, it does not take into account the uncertainty of fuzziness involved in the evaluation phase [41]. ONTOCOM is still open to further development from this point of perspective. The parameters used to estimate ONTOCOM Model have vagueness that adds some uncertainty in the modeling of algorithms. In this study, we have investigated the following Research Questions:

RQ-1 How to check the economic feasibility of an Ontology Engineering project? Do some effort estimation approaches to building ontologies exist?

To answer this research question, we have done a literature survey of the existing methodologies for building ontologies and presented it in Section 2.

RQ-2 Does the vagueness in data affect the effort estimation process? Can the effort estimation based on fuzzy logic be more suitable for handling vague and imprecise data?

To answer this research question, fuzzy inference system is applied to the effort multipliers. As some factors are always shown by grades Extra High, Very High, High, Nominal, Very low, etc. with natural language descriptions.

We convert these grades into numerical forms to evaluate the relevance of the program. It is always useful to describe the natural language assessment by fuzzy numbers rather than crisp ones. A fuzzy rule base has been written for these effort multipliers and tested over a dataset of 148 ontology projects.

RQ-3 *Can the accuracy of the existing ontology effort estimation models be further improved?*

To answer this research question, some scaling factors are introduced and more effort multipliers are introduced to tune the fuzzy inference system and provide better results.

The main contributions of this work are as follows:

1. To conduct a literature survey of the existing cost estimation models for building ontologies.
2. To propose an effort estimation model named F-ONTOCOM, which addresses the fuzziness involved in ontology Engineering effort estimation and its significant role is highlighted through the ONTOCOM's various aspects.
3. To enhance the effort estimation, six effort multipliers and scale factors have been added to the ones listed by ONTOCOM.

The remaining paper is structured as follows: Section 2 describes the related work; Section 3 discusses the Theoretical background; Section 4 gives the Proposed work i.e. F-ONTOCOM; Section 5 provides the experimental setup; Section 6 presents the results and discussions; Section 7 defines the threats of validity and finally, last Section 8 concludes the paper.

2 Related Work

The cost evaluation methods have a long-standing tradition in more mature technical fields like software technology or industrial manufacturing. The most famous algorithmic model is COCOMO released in 1981 by Barry Boehm [7]. It has been developed by analysing 63 software projects. Putnam's model is proposed by Putnam in the late 1970s [34]. This model is based on manpower distribution and the tool used for this model is SLIM (Software Life Cycle Management). The Function Point-based Model is proposed by Albrecht in 1983 [35]. In this model, a metric is given that is known as the function point metric, used to measure the functionality of the project. Approaches in these areas provided useful knowledge regarding methods for defining and evaluating ontology cost models [7, 23, 24]. The

approaches established by ontology engineering focus on the centralized production of static ontologies, i.e. consider only passing the iteration between ontology design/modification and use [39]. Furthermore, these techniques do not address cost-effective processes such as cost management, cost reduction, or analysis of cost benefits. The authors have done the qualitative analysis of cost and ontology usage in different domains and also explored the number of projects based on the ontology approach, but do not provide any cost model [25]. The author Presents observational outcomes for measuring ontology reuse [26]. The authors proposed a template to analyse and evaluate the advantages of ontology in a specified environment, but the model suggested is not quantitative [27]. The authors introduced a model for a cost analysis for people who think in the form of formal statements on studying externalization. The authors attempt to apply their prototype to semantic wikis but don't give any experimental assessment [29]. The authors identified the efforts estimation for web semantic services. They claim that the projects involving semantic web services take less time as compare to conventional web services, but do not serve ontologies [28]. The authors suggested a methodology to evaluate the ontology engineering risks which might be the main component of an ontology project for its possible study [32]. The authors describe gOntt a Gantt chart-based planning tool for ontology projects that maps activities relevant to the life cycle of networked ontologies to current life cycle models [37]. As discussed in [38], these activities can be supplemented by effort-related data. The authors define that ONTOCOM as a framework by which we can analyse the cost and benefits of ontology [12]. The author adjusts the effort multipliers in the web application cost estimation model with the use of ontologies. The effort multipliers, on the other hand, are not adapted to the needs of ontology engineering and there is no evaluation [36]. The authors presented the alignment of the ONTOCOM model with the DILIGENT engineering methodology. They also provided some analytical assessments of application scenarios for the DILIGENT model based on the resulting cost function [11]. In summary, the project manager is supposed to distribute a percentage of effort throughout the project phases established in the scheduling tool, and ONTOCOM estimates, which cover the entire project duration, are distributed accordingly to each phase. The methodology of ONTOCOM considers the uncertainties of randomness; yet, the uncertainty of fuzziness involved in the process of analysis is not taken into account. There is no cost estimation model which can handle uncertainty, so we have given a model i.e., F-ONTOCOM.

3 Theoretical Background

In this section, we present the basic notion of the existing ontology cost estimation models, effort multipliers, and Fuzzy logic.

3.1 Existing Ontology Cost Estimation Model

ONTOCOM is the parametric cost estimating model for ontologies, which predicts the effort invested by predefined effort multipliers in constructing, maintaining, and reusing ontologies. We used a combination of three general-purpose cost estimation approaches [7] to define the relevant factors, which we believe are appropriate to ontology engineering according to the current state of the art in the field [8]. To the ontological cost model, ONTOCOM utilizes three cost estimating techniques. This methodology begins as the Top-down approach, by distinguishing upper-level subtasks of an ontology engineering process and estimates the related costs using the parametric method. The set of effort multipliers associated with each process stage was evaluated using the expert judgment method, and their start values were specified in the a-priori model [8, 9, 21, 40].

The development of Ontology is divided into 3 separate assignments:

Building Ontology: This incorporates all sub-assignments, for example, domain analysis results in the requirement specification, conceptualization results as a conceptual model, Implementation (result: specification of a conceptual model), and Ontology population (result: instantiated ontology).

Maintaining Ontology: This includes costs identified with getting recognizable and making changes in the ontology.

Reuse Ontology: This includes the re-utilization of existing ontologies for the creation of new ontologies and thus entails costs in connection with the finding, evaluation, and adaptation of the former to the latter's demand.

DILIGENT is a methodology for developing ontologies in a distributed, loosely managed, and dynamic manner. The ONTOCOM model and the alignment of the effort multipliers to the activities they affect were used to develop the cost function specific to the DILIGENT process [11]. The cost function is subsequently streamlined to allow the use of cost information to support the decision for three engineering scenarios stated during the project: The best possible size of the original Ontology is found, (2) reuse is extended on local sites and (3) Board meetings are ideal in frequency.

3.2 Effort Multipliers for Ontology Engineering

The following section provides an overview of the effort multipliers involved in the building, maintenance, and reuse of ontologies. We distinguish between effort multipliers associated with product, processes, and personnel. The product category reflects the effect on the overall cost of the product attributes. The category of processes indicates the relevant elements to the cost estimation of the engineering process, while the personnel emphasizes the relevance of their team experience, skill, and continuity for the effort invested [12, 13].

- **Product-Related Effort multipliers:** These effort multipliers represent the effect of the attributes of the product to be built (for example, the ontology) on the total cost. The accompanying effort multipliers were distinguished for the assignment of ontology building [14]:
 - (DCPLX) Domain Analysis Complexity represents the application highlights that impact the diverse nature of the engineering results
 - (CCPLX) Conceptualization Complexity to represent the effect of a complex reasonable model on the total cost,
 - (ICPLX) Implementation Complexity to contemplate the extra endeavours emerged from the use of a particular implementation language,
 - (DATA) Instantiation Complexity to capture the impact on the overall process of the instance data requirements,
 - (REUSE) Required Reusability to catch the extra exertion related to the improvement of a reusable ontology,
 - (OI) Evaluation Complexity to represent the extra endeavours at the end put resources into producing experiments and assessing test results, and
 - (DOCU) Documentation is intended to indicate the additional costs arising from the requirements for detailed information.
- **Personnel Related Effort multipliers:** The effort multipliers underline that team expertise, skill and congruity play a role in the engineering process:
 - (OCAP/DECAP) Ontologist/Domain Expert Capability to represent, as do their cooperation abilities, the apparent capacity and productivity of the performers engaged in the project,
 - (OEXP/DEEXP) Ontologist/Domain Expert Experience to assess the understanding dimension of the engineering team for the results of ontology exercises,

- (LEXP/TEXP) Language/Tool Experience to measure the project team's dimensional knowledge of representation language and ontology management tools,
 - (PCON) Personnel Continuity to represent the repeated changes in the team's workforce.
- Project-Related Effort multipliers: These effort multipliers recognize the broad characteristics and impact of ontology engineering on the complete cost:
 - (TOOL) Support Tool for Ontology Engineering to quantify the consequences in the engineering process of using ontology management tools.
 - (SITE) Multisite Development to reflect the use in the location-distributed group of communication support instruments.
 - (SCED) Required Development Schedule considering the certain time limitations, it considers the special features of the engineering process.
 - The estimation of the given effort multipliers is given by Boehm [7] on the components of extremely low, low, nominal, high, and very high.

3.3 Fuzzy Logic

Fuzzy Logic is a method for solving issues that are too complicated for a quantitative understanding. It handles the problems with imprecise and incomplete data which is based on the fuzzy set theory and formalized by Prof. Lofti Zadeh in 1965 [15]. It is a class theory with unsharp limits and extends the classical set theory [16, 17]. The membership $\mu_A(x)$ of component x of a traditional set A , a subset of the universe X , is characterized by:

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases} \quad (1)$$

In other words, x is a part of the set A ($\mu_A(x) = 1$) or not ($\mu_A(x) = 0$). The classical sets are either zero or one is referred to as crisp sets as in Equation (1).

Partial membership is allowed in Fuzzy sets. A fuzzy set A is defined by a reference set X called the universe and a mapping;

$$\mu_A: X \rightarrow [0, 1] \quad (2)$$

A fuzzy set A feature called $\mu_A(x)$ membership function for x , X is interpreted as x in the fuzzy set A as the degree of membership. A member function is a curve that describes how each point is mapped between 0 and 1 in the input area. The greater the x of the membership, the more apparent is that x is A . The Membership Functions can be Triangular, Gaussian, Trapezoidal, and Parabolic.

Any framework that has an instant connection with fuzzy concepts is referred to as the fuzzy logic system. There are three possible sorts of the most prominent fuzzy logical frameworks in writing: pure, fuzzy logic systems of Takagi and Sugeno and fuzzy logic systems with fuzzifier and de-fuzzifier known as the Mamdani system. The Mamdani framework [18, 19] is the most widely used in which the fuzzifier maps new inputs into fuzzy sets, and the fluffy de-fuzzifier mappings into crisp outputs as a big part of the engineering apps uses new data as input and produces new data. The primary 4 components of the Fuzzy Inference System (FIS) are Fuzzifier, Fuzzy rule base, Fuzzy Inference Engine, and De-Fuzzifier.

One of the main aspects of the fuzzy logic of uncertainty management is that it gives a systematic framework for dealing with fuzzy quantifiers, for instance, most, many, few, few, almost all, rarely, approximately 0.8, etc. Fuzzy logic thus subsumes both the predicate and the probability theory and allows multiple sorts of uncertainty to be addressed within a single conceptual framework [42]. Fuzzy logic systems are the branch of computer science that focuses on imprecision, uncertainty, and approximation to achieve robustness and low-cost solutions. The Fuzzy logic system performs strategies that mimic the human mind's ability to deal with reasoning and approximation problems rather than more accurate. To handle uncertainty, the Fuzzy Logic System deals with fuzzy parameters, uncertainties by mapping out the path of a given input to an output using the computing framework called the Fuzzy Inference System.

4 Working of F-ONTOCOM

This work proposes an effort estimation methodology for ontology engineering using Fuzzy Logic i.e., F-ONTOCOM. Though the analysis, methodology of ONTOCOM considers the uncertainties of randomness; yet, the uncertainty of fuzziness involved in the process of analysis is not taken into account. From this point of view, the ONTOCOM model is still open to be developed so far. The core of ontology engineering economics is effort estimation. The reason for fuzziness to be considered in the ONTOCOM

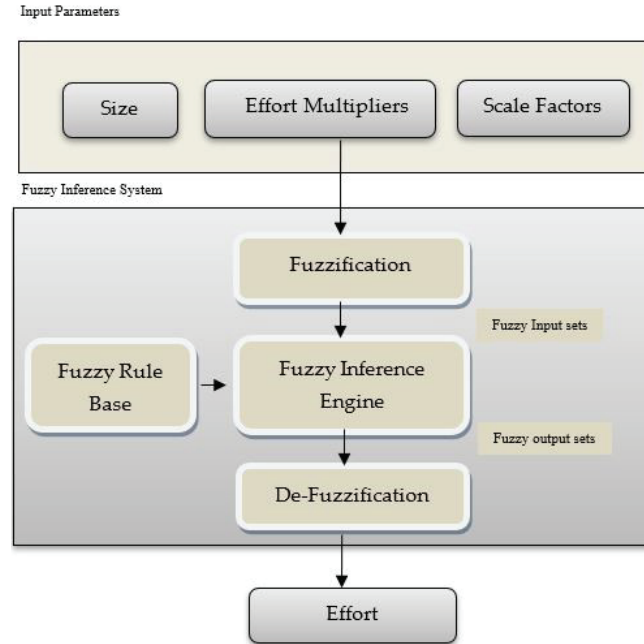


Figure 1 F-ONTOCOM: An ontology cost estimation model.

model lies in the fact that the division of evaluation and rating of some involved factors, which have an important influence upon development cost, are vague and indistinct. They are always depicted by grades with natural language descriptions as Extra High, Very High, High, Very low ... and so on. To evaluate the merit of a program, these grades are transferred into quantitative forms. It is practical to represent the natural language descriptions of evaluation and assessment by fuzzy numbers rather than crisp ones. So we have proposed F-ONTOCOM as shown in Figure 1.

Thus the estimate of ontology costs is the sum of the cost incurred in the building and the maintenance of ontologies (with or without reuse) as mentioned in (1).

$$MM = MM_B + MM_M + MM_R \quad (3)$$

Where MM_B , MM_M , and MM_R are the effort involved in the development, maintenance, and reuse of ontologies. The costs of ontology are calculated in a COCOMO-like way, as follows:

$$MM_X = A * (Size_x)^B * \prod EM_{xi} \quad (4)$$

1. Input Parameters: In this model, we have three input variables like size, Effort multipliers, and scale factors as shown in Figure 1. These input variables are changed to fuzzy variables using fuzzy sets for each linguistic value such as extra high, very high, high, nominal, low, very low. Previously authors had also taken the values of A & B as constant.

We have considered the value of A depends on the complexity of the ontology model either it can be organic, semi-detached, or embedded. At the time of evaluation, these input values are taken from a dataset.

Boehm's definition of organic, semidetached, and embedded systems:

- **Organic:** Any project is said to be organic if the problem is well understood and size is small and team members associated are nominal experienced.
- **Semi-Detached:** A project is called semi-detached if the problem is difficult as compared to organic ones. The team size, experience and knowledge lie in-between organic and embedded.
- **Embedded:** This type of project has the highest level of complexity, required larger team size, very high experience as compared to other two models.

For the organic model the values of A & B are 3.2, 1.05 respectively, for Semi-detached model the values are 3.0, 1.12 and for the embedded model it is 2.8,1.2.

- **Scale Factors:** Ontology development has various characteristics and a scale factor is one of them. It decides the amount of effort involved in development. Development efforts include flexibility in development, resolution of risks identified, Precedentness, maturity of processes, and interconnection between the teams. All these factors are measured as extra high, very high, high, nominal, low, very low. The range of all these scale factors is given in Table 1. The value of B depends on the COCOMO-II scale factors which are given by Boehm [7].

The B Value can be displayed as:

$$B = 0.91 + 0.01 * \text{Sum of rating on scale factors} \quad (5)$$

The value of B can be within the range of 0.91 to 1.23. The Value of the scale factors as given by Boehm in [7].

- **Precedentness (PREC):** It tells about the previous experience on the same kind of projects developed by individuals or organizations. If there is no earlier experience then the scale factor is taken as very low and if

Table 1 Scale factors range

S No	Scaling Factors	Range
1	PREC	1.0–7.0
2	FLEX	1.0–6.0
3	RESL	1.0–8.0
4	TEAM	1.0–6.0
5	PMAT	1.0–8.0

organizations or individuals are completely aware of the domain, then the scale factor is taken as extra high.

- Development Flexibility (FLEX): This tells about the tractability associated with the process of development. If we have a well-defined process in place, then the scale factor is taken as very low and if generic information is available for a process then the scale factor is considered as extra high.
- Architecture/Risk Resolution (RESL): It provides details about the analysis done on the process. If there is no or minimal analysis done, then the scale factor is very low and if the complete analysis is done, the scale factor value is taken as extra high.
- Team Cohesion (TEAM): It shows the interconnection within the teams. If there is no interconnection or interaction within the team, then the scale factor is very low and if teams are fully interconnected, the scale factor is considered as extra high.
- Process Maturity (PMAT): It defines the maturity in the processes of the organization. If there is no maturity, the scale factor is very low and if organization processes are matured, the scale factor is taken as extra high.
- Effort Multipliers: Effort multipliers are the multiplicative factors that impact the effort required to accomplish the ontology project. The EM_{xi} (effort multipliers) are the central elements of the cost estimation formula as in Equation (4). It gives an exceptionally low to high rating level that communicates its impact on the progressive effort [10, 11]. 17 effort multipliers are being used in ONTOCOM. We have added six more effort multipliers that are not previously defined in the model. The added parameters in the ONTOCOM Model to calculate the cost more accurately are given below in Table 2 and the range of all these multipliers are given in Table 3.

Table 2 Added effort multipliers in ONTOCOM

Effort Multiplier	Purpose	Very Low	Low	Nominal	High	Very High	Extra-High
RELY	The degree to which a model can stretch to complete its functions over some time Only slight inconvenience	Low, easily recoverable losses	Moderate, easily recoverable losses	High Financial loss	Risk to human life	–	
TIME	A measure of the time limit in which execution can be completed	–	–	<=50% use of an available execution time	70%	85%	95%
STOR	A measure of available storage	–	–	<=50% storage utilization available	70%	85%	95%
PVOL	Amount of changes in OS, DBMS and Compiler etc	–	Major modifications every 1 year & Minors every 1 month	Major modifications every 6 months & Minors every 2 weeks	Major modifications every 2 months & Minors 1 week	Major modifications every 2 weeks & Minors every 2 days	–
PEXP	Measure the platform experience	<=2 months	6 months	1 year	3 years	6 years	–
PREX	Measure through the personal experience	5 months	9 months	1 year	2 years	4 years	6 years

Table 3 Effort multipliers range

S. No	Effort Multipliers	Range
1	DCPLX	0.5–1.8
2	CCPLX	0.5–1.8
3	ICPLX	0.7–2.0
4	DATA	0.6–1.8
5	REUSE	0.5–1.4
6	DOCU	0.5–1.4
7	OI	0.5–1.8
8	OCAP	0.5–1.4
9	DECAP	0.5–1.4
10	OEXP	0.5–1.4
11	DEEXP	0.5–1.4
12	PCON	0.5–1.4
13	LEXP	0.5–1.8
14	TEXP	0.5–1.8
15	TOOL	0.5–1.8
16	SITE	0.5–1.4
17	SCED	0.5–1.4
18	RELY	0.2–2.0
19	TIME	0.5–2.0
20	STOR	0.5–2.0
21	PVOL	0.5–2.0
22	PEXP	0.6–2.0
23	PREX	0.5–2.0

The values of all these cost parameters can be extra high, very high, high, nominal, low, very low as mentioned by Boehm [7].

- $Size_x$: The $size_x$ is It is being calculated as given in Equation (6).

$$Size_x = \frac{size_b + size_m + size_r}{1000} \quad (6)$$

- $Size_b$ represents the size of a newly constructed ontology such as the number of natives on whom the conceptualizing stage depends.

- $Size_m$ depends on the number of modified items that are expected to be used.
- $Size_r$ is the size of the source after the application size has been customized. This applies particularly to the parts of the source ontologies, whose contents have to adjust to the target scope, and the fragments have to be directly integrated into the final representational language.

We are considering only the building size of ontology expressed in thousands of ontological primitives like concepts, relations, axioms, and instances.

2. Fuzzy Inference System: It consists of Fuzzification, Fuzzy inference engine, Fuzzy rule base, and De-Fuzzification.

- Fuzzification – The Fuzzification process consists of a fuzzifier that converts crisp input into a fuzzy set of values using its membership function. The membership function is a curve that transforms the input values to the membership values range between 0 & 1. The membership function can be either triangular, trapezoidal, or Gaussian. Analysis can be done with all the three membership functions like Triangular Membership Function (TMF), Trapezoidal membership Function (Trapmf), and Gaussian Membership Function (GMF); however, we have used the Triangular membership function. The Fuzzification of two effort multipliers using triangular membership is shown in Figures 2 and 3.
- Fuzzy Rule Base – It uses if-then rules. Fuzzy rules of the COCOMO-II logic are defined in the Fuzzification process by linguistic variables and it is based on “AND” connectivity of input variables. Some of the rules are framed: –

If (DCPLXI is VL) then (DCPLXO is VL)

If (RELYI is H) then (RELYO is H)

If (OCAP is VH) then (OCAPO is VL)



Figure 2 Fuzzification of DATA Effort Multipliers using Triangular membership function.

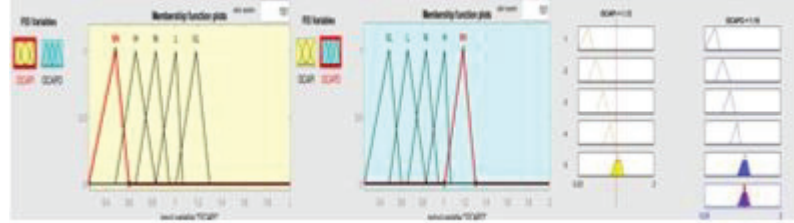


Figure 3 Fuzzification of OCAP Effort Multipliers using Triangular membership function.

If (FLEXI is VH) then (FLEXO is VL)
 If (TIMEI is L) then (TIMEO is L)
 If (PEXPI is L) then (PEXPO is H)
 If (PRECI is VL) then (PRECO is VL)
 If (DECAPI is VL) then (PRECO is VH)
 If (PREXI is EH) then (PREXO is VL)
 If (PRECI is VL) then (PRECO is VL)
 If (TOOLI is VL) then (TOOLO is VH)

- Fuzzy Inference Engine – The fuzzy inference engine is utilized in the inference process to transfer the input from the Fuzzification process to the output based on expert knowledge or rules. The role of fuzzy rules in the inference process is to capture imprecise reasoning styles and serve as a technique of producing fuzzy output from fuzzy input.
- De-Fuzzification – It is the translation of fuzzy output into narrow output.

3. Output: The output of this model is Effort that can be calculated as given in Equation (4).

5 Experimental Evaluation

In this section, we have evaluated the proposed technique using synthetic and real datasets. In a synthetic dataset size and effort of example ontology, i.e. Super Ontology [20, 43, 46] as shown in Figure 4 is calculated. The real dataset that is given by the university of INNSBRUCK [31] consists of 148 ontology projects with all the input parameters like the size of each ontology, the value of all the effort multipliers, an actual effort for all the ontology projects, and finally the effort estimation of all the existing effort estimation models like ONTOCOM, Diligent model. We have compared our proposed technique with all the existing effort estimation models and found that the F-ONTOCOM gives the best results.

5.1 Synthetic Data

The example ontology has been taken from [20, 43]. In this ontology, there are 120 concepts and 454 axioms, and 119 relations. We can calculate the size of ontology for building only from Equation (6).

$$\text{Size} = \frac{120 + 454 + 119}{1000} = 0.693$$

and the value is 0.693 also calculates all the effort multipliers for the same ontology. One Size is available, then ratings of all the effort multipliers and their specifications are taken as mentioned in Table 1. The DCPLX Effort Multipliers value is computed by the average sum of domain complexity, requirement complexity, and information complexity and the value is 1.2. Similarly, we can find the value of all the effort multipliers according to the ontology structure. The Value of all the effort multipliers that have been used in calculating the Effort is given in Table 4.

Table 4 Effort multipliers values for example ontology

S. No	Effort Multipliers	Value
1	DCPLX	1.2
2	CCPLX	1
3	ICPLX	1
4	DATA	0.90
5	REUSE	1.3
6	DOCU	0.7
7	OI	1
8	OCAP	1
9	DECAP	1
10	OEXP	0.70
11	DEEXP	1
12	PCON	0.85
13	LEXP	1.3
14	TEXP	1
15	TOOL	0.90
16	SITE	1
17	SCED	1

The values of A & B are taken 3.0 & 1.12 respectively as we are considering for semi-detached model. After applying the previous parameters to ONTOCOM in Equation (4) the effort is 1.23.

$$\begin{aligned}\text{Effort} &= 3.0 * (0.693)^{1.12} * 1.2 * (1)^8 * (0.90)^3 * (1.3)^2 * (0.7)^2 * 0.85 \\ &= 1.85 * (0.693)^{1.12} \\ &= 1.23\end{aligned}$$

The value of effort after applying the Fuzzification process is 1.18. For this synthetic dataset, we found that after applying Fuzzification the estimated effort is optimized.

The values of added effort multipliers and scale factors are given in Tables 5 and 6 respectively. The effort calculated after adding more effort multipliers is 1.52, slightly more than the earlier one. The fuzzified value for the same is 1.18, which is almost equivalent to the calculated value. For fuzzifying the results of cost from above, we need to apply fuzzy logic & we need to define the fuzzy ranges to all the effort multipliers. The range for the Scale factors and effort multipliers are given in Tables 5 and 6 respectively.

Table 5 Scale factor values

S. No	Scaling Factors	Value
1	Precedentness	3.72
2	Development Flexibility	3.04
3	Architecture/Risk Resolution	4.24
4	Team Cohesion	3.29
5	Process Maturity	4.60

Table 6 Added effort multipliers values

S. No	Effort Multipliers	Value
1	RELY	1.15
2	TIME	1.0
3	STOR	1.0
4	PVOL	1.15
5	PEXP	1.0
6	PREX	1.0

5.2 Real Dataset

The dataset used in this paper is from the University of INNSBRUCK [31, 40]. The dataset consists of 148 ontology projects with their actual size, actual effort in man-months, and value of 17 effort multipliers in the range of very low to extra high. The assessment is made through a comparison with the real effort of the correctness of the projected effort.

A model's evaluation indicates how effectively our model is performing. If the difference between the actual and estimated values is substantial, it will result in inaccurate estimation, which will increase the cost of software development. The study employs well-known evaluation criteria that have been used in the literature, which are MRE (magnitude of relative error), MMRE (mean magnitude relative error), MdMRE (median magnitude of relative error), PRED (prediction).

These evaluation criteria are calculated as follows:

$$MRE = \sum_1^n \frac{|actual - estimated|}{actual} \quad (7)$$

$$MMRE = \frac{1}{n} \sum_1^n MRE \quad (8)$$

Where n is the total number of ontology projects

$$MdMRE = \text{Median (MRE)} \quad (9)$$

$$PRED(n) = \frac{1}{N} \sum_1^n \begin{cases} 1 & \text{if } MRE \leq n \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

The proposed technique is tested on the real dataset as well as all four evaluation criteria.

6 Results and Discussion

This section shows the results when the proposed model is applied to the ontology dataset. To implement this model MATLAB is used. The MRE, MMRE, MdMRE, and Prediction (n) values are computed for the 21 randomly chosen ontology projects. The results of the current effort compared to the estimated effort for the ontology dataset are shown in Table 7. It also includes the results of ONTOCOM and the Diligent Model effort.

Table 7 Actual and estimated effort of few randomly selected projects

S No	Ontologies	A	Size	Actual Effort	Diligent	ONTOCOM	F-ONTOCOM
1	swpathol	4.17472	1.040041	5	6.7068397	3.524081092	4.66
2	Opjk	4.17472	0.948026	2.6	13.763627	1.689793362	1.86
3	OMV	4.17472	0.835133	2.5	5.2918234	1.151013337	1.93
4	Hochschul	4.17472	1.109294	7	7.1436075	3.709762524	7.04
5	Kompetenz management	4.17472	1.014365	5	6.2459774	6.505230457	5.31
6	SESAM	4.17472	0.790595	2	4.2474832	1.302825696	1.37
7	Context	4.17472	0.835133	1.25	6.8757838	0.997488855	1.2
8	life event	4.17472	1.062553	3	7.9606103	4.611297173	2.14
9	web services process ontology	4.17472	0.752851	6	10.518151	3.156844019	4.03
10	BT Digital Library	4.17472	0.884394	2	1.1933657	2.189181074	1.35
11	prm.kaon	4.17472	0.926408	3.5	4.26046	4.597391114	3.11
12	HR	4.17472	1.379875	6	7.4513853	7.055084943	5.46
13	Produktiv	4.17472	1.062553	5	7.809886	7.45802211	5.9
14	REWERSE	4.17472	0.906781	4	1.9220416	2.949027287	3.38
15	Opinions and argumentation	4.17472	0.664297	2.5	4.0120281	0.989119424	1.19
16	Municipal Sociology	4.17472	0.835133	2	4.4323357	0.993112384	1.27
17	Not Public	4.17472	1.109294	4	7.908887	0.2498649	3.76
18	The BEST	4.17472	0.937571	6	4.6982148	3.394756105	5.39
19	Hypothesis	4.17472	0.638722	6	6.4345636	7.271058891	5.6
20	YI	4.17472	0.96716	1	0.8999965	0.246254662	0.73
21	Brain Anatomy	4.17472	0.948026	8	7.0486611	5.606054588	6.64

Case Study 1: Ontology Project 5 named “Kompetenz management”

Input parameters: A = 4.17, Size = 1.01, B = 1

Effort multipliers: DCPLX, CCPLX, ICPLX, DATA, REUSE, DOCU, OI, OCAP, DECAP, OEXP, DEEXP, PCON, LEXP, TEXP, TOOL, SITE, SCED

Output (Estimated Effort)

$$\begin{aligned}
 E &= 4.17 * (1.01)^1 * 0.91 * (0.98)^2 * 0.77 * (0.99)^4 * 1.23 * (1.16)^3 \\
 &\quad * 0.72 * 1.06 * 1.11 * 1.06 * 1.14 = 5.31
 \end{aligned}$$

These values are provided as an input to the Fuzzy Inference system using Triangular membership and if-then rules and depicted by grades of natural language predictions as very high, high, nominal, low, very low, and are in the range as mentioned in Table 3. The actual effort for this project is 5, the effort calculated by Diligent Model is 6.24, Effort by ONTOCOM is 6.5, and effort by the proposed model is 5.31, which is much closer to the actual value compared to other methods.

Case Study 2: Ontology Project 1 named “swpatho1”

Input Parameters: A = 4.17; Size = 1.04; B = 1;

Effort multipliers: DCPLX, CCPLX, ICPLX, DATA, REUSE, DOCU, OI, OCAP, DECAP, OEXP, DEEXP, PCON, LEXP, TEXP, TOOL, SITE, SCED

Output (Estimated Effort)

$$E = 4.17 * (1.04)^1 * 0.91 * 1.25 * 1.19 * (0.99)^2 * 0.77 * 0.97 * 1.16 \\ * (0.96)^2 * 1.10 * 0.93 * 1.32 * 1.51 * 0.99 * 0.47 * 0.98 = 4.66$$

Here in this ontology project 1 the value of actual effort is 5, the effort calculated by ONTOCOM Model is 3.52, Effort by Diligent Model is 6.7 and effort by the proposed model is 4.66 which is closer to the actual effort.

The graphical representation for the comparison of effort by different techniques is shown in Figure 4.

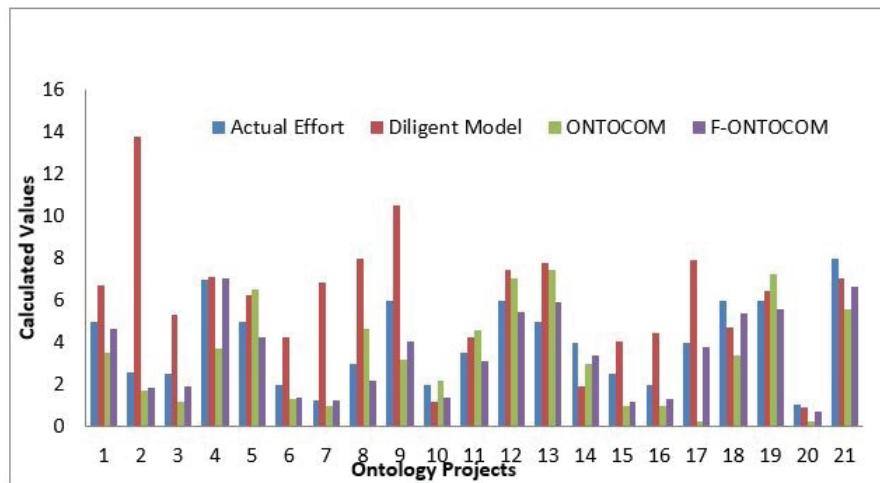


Figure 4 Comparison with Ontology datasets using different techniques.

Table 8 Comparison of MRE for the three models

S No	Ontologies	MRE Using Diligent	MRE Using ONTOCOM	MRE Using F-ONTOCOM
1	swpathol	0.34	0.30	0.07
2	Opjk	4.29	0.35	0.28
3	OMV	1.12	0.54	0.23
4	Hochschul	0.02	0.47	0.01
5	Kompetenz management	0.25	0.30	0.06
6	SESAM	1.12	0.35	0.32
7	Context	4.50	0.20	0.04
8	life event	1.65	0.54	0.29
9	web services process	0.75	0.47	0.33
10	BT Digital Library	0.40	0.09	0.33
11	prm.kaon	0.22	0.31	0.11
12	HR	0.24	0.18	0.09
13	Produktiv	0.56	0.49	0.18
14	REWERSE researcher	0.52	0.26	0.16
15	Opinions and argumentation	0.60	0.60	0.52
16	Municipal Sociology	1.22	0.50	0.37
17	Not Public	0.98	0.94	0.06
18	BEST	0.22	0.43	0.10
19	Hypothesis	0.48	0.15	0.18
20	YI	0.10	0.75	0.27
21	Brain Anatomy	0.12	0.30	0.17

The MRE value for the Ontology Project 5 is 0.25 for the Diligent Model, 0.20 for ONTOCOM, and 0.06 for the proposed model as given in Table 8. Figure 5 provides a graphical illustration of the MRE values for all three models.

The MMRE of the entire dataset is 3.18 for the Diligent Model, 0.56 for ONTOCOM, and 0.41 for the proposed model as given in Table 9. Similarly, The MdMRE of the entire ontology dataset is 1.10 for the Diligent Model, 0.47 for ONTOCOM Model, and 0.20 for the proposed model as given in Table 9 and the graphical representation is shown in Figure 6.

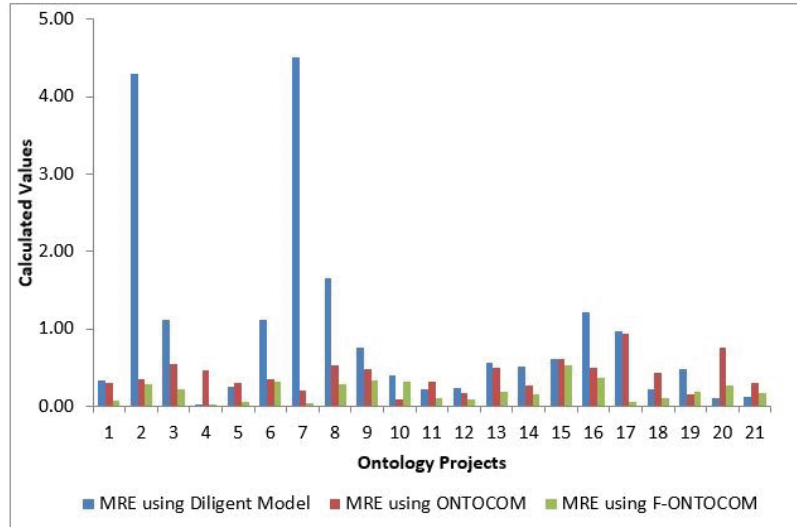


Figure 5 Graphical representation of comparison of MRE for three models.

Table 9 Comparison of MMRE, MDMRE for the three models

Evaluation Criteria	Diligent Model	ONTOCOM	F-ONTOCOM
MMRE	3.18	0.56	0.41
MdMRE	1.10	0.47	0.20

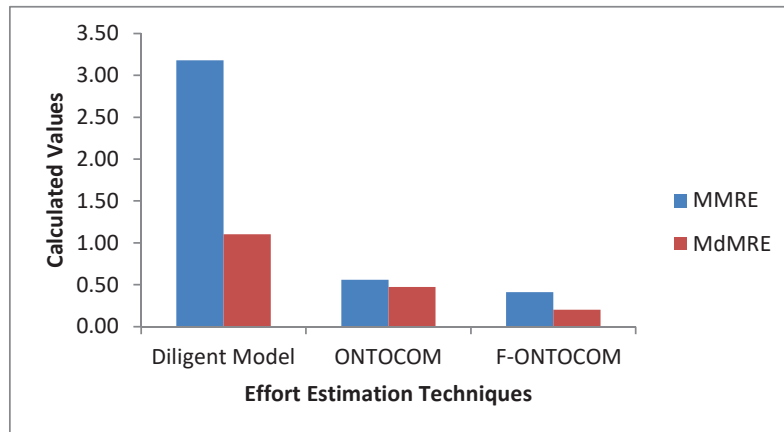
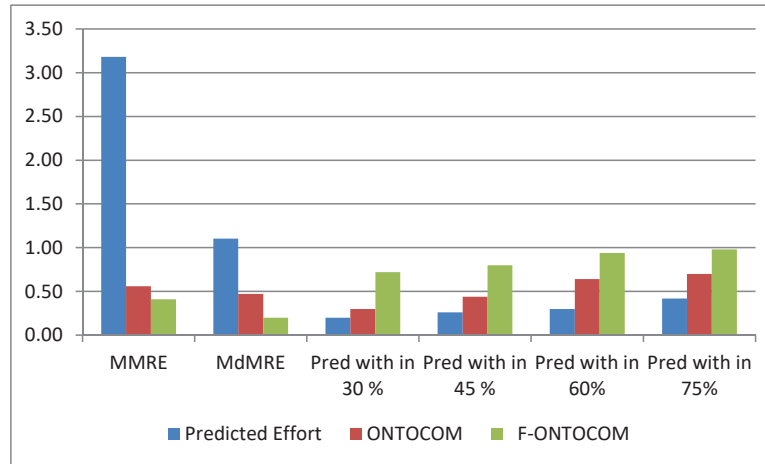


Figure 6 Graphical representation of MMRE & MdMRE for three models.

Table 10 Comparison of Prediction values for the three models

Evaluation Criteria	Diligent Model	ONTOCOM	F-ONTOCOM
Prediction within 30 %	0.20	0.30	0.72
Prediction within 45 %	0.26	0.44	0.80
Prediction within 60%	0.30	0.64	0.94
Prediction within 75%	0.42	0.70	0.98

**Figure 7** Graphical representation of Prediction (n) for three models.

The Prediction (n) is evaluated at different criteria's like Prediction within 30%, 45%, 60%, and 75%. The Prediction (n) within 30% is 0.20, 0.30, and 0.72 for the Diligent Model, ONTOCOM, and proposed model respectively. The pred (n) within 45% is 0.26, 0.44 and 0.80 for the proposed model. Similarly, for the pred (n) within 60% and 75% listed in Table 10. The graphical representation Pred values are shown in Figure 7.

The findings acquired thus show that the model suggested can be used to predict ontology costs correctly.

7 Threats of Validity

This study deploys the F-ONTOCOM for ontology cost estimation. The proposed approach is also supported by the outcomes produced and assessed using various assessment criteria. Moreover, there are some points to be considered regarding internal and external validities.

Internal Validity: This study uses the Fuzzy technique for ontology cost estimation called F-ONTOCOM. This technique gives the best results with the chosen parameters. It may not always work, if we vary the parameters or It give better results by integrating any other optimization technique with Fuzzy.

External Validity: External validity concerns the generality of the results, whether the outcomes of research are generalized or not. In this study, we have used the dataset of 148 ontology projects with the value of actual effort, Effort Multipliers, and size. The limitation of this study is dataset availability. There is only a single dataset that may be used for this method evaluation. The results are more validated if more datasets will be available.

8 Conclusion

Reliable cost assessment techniques are a necessity for widespread ontological dissemination in company environments. In this paper, we have proposed the use of Fuzzy sets rather than classical intervals in the ONTOCOM Model. We have defined the corresponding Fuzzy sets for each effort multiplier and its associated linguistic values and represented by triangular membership functions. The importance of including Fuzziness into ordinary ONTOCOM to get a more realistic output in uncertain conditions. The proposed methodology is compared with the existing ontology cost model ONTOCOM and DILIGENT, which is not fuzzified. Using this technique, the values evaluated for the dataset of 148 ontology projects, yield much lower MMRE, MdmRE, and high Prediction. They are much closer to the actual effort. The experimental results show that F-ONTOCOM performed remarkably better than ONTOCOM in optimizing the cost. Thus this new model for the ontology cost is remarkably improved. Furthermore, the utilization of other optimization techniques can also be applied to ontology cost estimation. This work can also be extended using the Neural approach or combination of these techniques and it can also be tested on other datasets.

References

- [1] Gruber T 1993 A translation approach to portable ontology specifications Knowledge Acquisition, 5: 199–220.
- [2] Fensel D Ontologies 2001 Silver Bullet for Knowledge Management and Electronic Commerce. IEEE intelligent Systems Heidelberg: Springer-Verlag, 16: 54–59.

- [3] Neches R, Fikes R E, Finin T, Gruber T R, Senator T, Swartout, W R 1991 Enabling technology for knowledge sharing. *AI Magazine*, 12: 36–56.
- [4] Ragab, A. H. M.: Cost Estimation Models for Ontology Engineering Based Projects. Bsc Thesis Project, KAU (2010).
- [5] Zia Z, Rashid A, Zaman K 2011 Software Cost Estimation for Component based fourth-generation-language software applications. *IET Software*, 5: 103–110.
- [6] Issa A A 2011 An Algorithmic Software Cost Estimation Model for Early Stages of Software Development. *J. of Academic Research*, 3: 336–341.
- [7] Bw, B. (1981). *Software Engineering Economics*. Inglewood Cliffs,, NJ:, Prentice-Hall, 198(1).
- [8] PaslaruBontas E, Mochol M 2005 A cost model for ontology engineering. Technical Report TR-B-05-03. FU.
- [9] Paslaru Bontas E, Mochol M, Tolksdorf R 2005 Case Studies in Ontology Reuse. In *Proceedings of the 5th International Conference on Knowledge Management*, pp. 345–353.
- [10] PaslaruBontas E, Mochol M 2005 Towards a methodology for ontology reuse. In *Proceedings of the International Conference on Terminology and Knowledge Engineering TKE05*.
- [11] Paslaru Bontas E, Tempich C 2005 How much does it cost? Applying ONTOCOM to DILIGENT. Technical Report TR-B-05-20 pp. 1–64.
- [12] PaslaruBontas E, Mochol M 2006 Ontology Engineering Cost Estimation with ONTOCOM. Berlin, Technical Report TR-B-06-01.
- [13] Buitelaar P, Olejnik D, Sintek M A 2004 Protg Plug-In for Ontology Extraction from Text Based on Linguistic Analysis. In *Proceedings of the European Semantic Web Symposium ESWS04*.
- [14] Dittenbach M, Berger H, Merll D 2004 Improving domain ontologies by mining semantics from text. In *Proceedings of the 1st Asian-Pacific Conference on Conceptual Modelling*, pp. 91–100.
- [15] Zadeh L A 1965 Fuzzy Sets. *Information and Control*. 8: 338–353.
- [16] Wasif Nisar M, Yong-Ji W, Elahi M 2008 Software Development Effort Estimation using Fuzzy Logic – A Survey. In *Proceedings of fifth International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 421–427.
- [17] Wang L X 1994 *Adaptive Fuzzy System and Control: Design and Stability Analysis*. Prentice Hall, Inc., Englewood Cliffs, NJ 07632.

- [18] Mamdani H 1974 Applications of fuzzy algorithms for simple dynamic plant. In *Proceedings of IEEE*, 121, pp. 1585–1588.
- [19] Kaushik A, Soni A K, Soni R 2013 A Type-2 Fuzzy Logic Based Framework for Function Points. *International Journal of Intelligent Systems and Applications*. 5: 74–82.
- [20] Malik S, Mishra S, Jain N K , Jain S 2015 Devising a super ontology. *Procedia Computer Science*. 70: 785–792.
- [21] Simperl E, Buerger T, Hangl S, Woelger S, Popov, I 2012 Ontocom: A reliable cost estimation method for ontology development projects. *Web Semantics: Science, Services and Agents on the World Wide Web*. 16: 1–16.
- [22] Simperl E, Mochol M, Brger T 2012 Achieving maturity: the state of practice in ontology engineering. *Int. J. Computer Science Applications*. 7: 45–65.
- [23] Kemerer C F 1987 Empirical Validation of Software Cost Estimation Models. *Communications of the ACM*. 30(5): 416–429.
- [24] Stewart R D, Wyskida R M, Johannes J D 1995 *Cost Estimator's Reference Manual*. Wiley.
- [25] Menzies T 1999 Cost benefits of ontologies. *Intelligence*. 10: 26–32.
- [26] Cohen P R, Chaudhri V K, Pease A, Schrag R 1999 Does prior knowledge facilitate the development of knowledge-based systems? In *Proceedings of the 16th International Conference on Artificial Intelligence*, pp. 221–226.
- [27] Burger T, Simperl E 2008 Measuring the benefits of ontologies. In *Proceedings of Ontology Content and Evaluation in Enterprise*. Monterrey, Mexico, pp. 584–594.
- [28] Wolff F, Oberle D, Lamparter S, Staab S 2005 Economic reflections on managing web service using semantics. In *Proceedings of the Workshop in Klagenfurt, EMISA*, pp. 194–207.
- [29] Suarez-Figueroa M C, Gomez-Perez A 2008 Building ontology networks: How to obtain a particular ontology network life cycle? In *Proceedings of the Third International Conference on Semantic Systems (I-Semantics)*.
- [30] Felfernig A 2004 Effort estimation for knowledge-based configuration systems. In *Proceedings of the 16th International Conference of Software Engineering and Knowledge Engineering*.
- [31] <https://www.sti-innsbruck.at/>
- [32] Ferreira C R, Marques P, Martins A L, Rita S, Grilo B, Arajo R, Sazedj P, Pinto H S 2007 Ontology design risk analysis. In *Proceedings*

- of the OTM Confederated International Conference on the Move to Meaningful Internet Systems. pp. 522–533.
- [33] Kaushik A, Verma S, Singh H J, Chhabra G 2017 Software cost optimization integrating fuzzy system and COA-Cuckoo optimization algorithm. *International Journal of System Assurance Engineering Management*. 8: 1461–1471.
- [34] Putnam, L.H. 1978 A General Empirical Solution to the Macro Software Sizing and Estimating Problem. *IEEE Transactions on Software Engineering*, 4, 345–361. <https://doi.org/10.1109/TSE.1978.231521>
- [35] IFPUG, FPCPM (2000) International Function Point Users Group (IFPUG) Function Point Counting Practices Manual.
- [36] Korotkiy, M 2005 On the Effect of Ontologies on Web Application Development Effort. In *Proc.of the Knowledge Engineering and Software Engineering Workshop*.
- [37] Gómez-Pérez A, Suarez-Figueroa, M and Vigo, M 2009 gOntt: a tool for scheduling ontology development projects. In *Proceedings of the Fifth International Conference on Knowledge Capture*, 2009.
- [38] Simperl E, Siorpaes K, Han S, and Wölger S 2010 Integrating ONTOCOM to gOntt. Technical report, STI Innsbruck, University of Innsbruck.
- [39] Gómez-Pérez A, Fernández-López M, and Corcho O 2003 *Ontological Engineering* Springer.
- [40] Simperl E, Popov I.O, Bürger T 2009 ONTOCOM Revisited: Towards Accurate Cost Predictions for Ontology Development Projects. In: Aroyo L. et al. (eds) *The Semantic Web: Research and Applications. ESWC. Lecture Notes in Computer Science*, vol. 5554. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-02121-3_21
- [41] Fei Z, and Liu X 1992 f-COCOMO: fuzzy constructive cost model in software engineering, *IEEE International Conference on Fuzzy Systems*, pp. 331–337, doi: 10.1109/FUZZY.1992.258637.
- [42] Zadeh L.A 1983 The role of fuzzy logic in the management of uncertainty in expert systems, *Fuzzy Sets and Systems*, vol. 11, Issues 1–3, pp. 199–227, ISSN 0165-0114, [https://doi.org/10.1016/S0165-0114\(83\)80081-5](https://doi.org/10.1016/S0165-0114(83)80081-5)
- [43] Malik, S., & Jain, S. (2021). Sup_Ont: an upper ontology. *International Journal of Web-Based Learning and Teaching Technologies (IJWLTT)*, 16(3), 79–99.

- [44] Malik, S., & Jain, S. (2018, November). A Review on Methods to Handle Uncertainty. In International Conference On Computational Vision and Bio Inspired Computing (pp. 773–781). Springer, Cham.
- [45] Mishra, S., Malik, S., Jain, N. K., & Jain, S. (2015). A realist framework for ontologies and the semantic Web. *Procedia Computer Science*, 70, 483–490.
- [46] Malik, S., & Jain, S. (2017, June). Ontology based context aware model. In 2017 International Conference on Computational Intelligence in Data Science (ICCIDS) (pp. 1–6). IEEE.

Biographies



Sonika Malik has done B.Tech from Kurukshetra University, India in 2004 and did her Masters from MMU in 2010. She is doing her doctorate from National Institute of Technology, Kurukshetra. She has served in the field of education from last 13 years and is currently working at Maharaja Surajmal Institute of Technology, Delhi. Her current research interests are in the area of Semantic Web, Knowledge representation and Ontology Design.



Sarika Jain graduated from Jawaharlal Nehru University (India) in 2001. Her doctorate, awarded in 2011, is in the field of knowledge representation in Artificial Intelligence. She has served in the field of education for over 19 years and is currently in service at the National Institute of Technology Kurukshetra (Institute of National Importance), India. Her current research interests are knowledge management and analytics, the semantic web, ontological engineering, and intelligent systems. She is a senior member of the IEEE, a member of ACM, and a Life Member of CSI.