FRAMEWORK FOR DISTRIBUTED DEVELOPMENT OF KNOWLEDGE-BASED ENGINEERING SYSTEMS & OUTSOURCING IN CONTEXT OF PRODUCT DESIGN

Research Paper

Manoj Kumar, SSBM, Geneva, Switzerland, manoj_nits@yahoo.com

Dr. Anna Provodnikova, SSBM, Geneva, Switzerland, anna@ssbm.ch

Abstract

The physical products that are designed and manufactured have one or more components involved. Every component has a geometry that is built with certain relevance to mathematics or heuristics. This provides a way to document the mathematics, or rule behind building any geometry that a component takes shape of. Usually such knowledge is preserved by way of documentation, or as a Knowledge Based Engineering systems that operate with a Computer Aided Design (CAD) software or design authoring tool. Capturing and maintaining the product knowledge is generally a documentation work. To have the knowledge represented as ontology is desirable. This is due to the fact that it helps the automated programs to create and validate different aspects of changes to the product and its building blocks i.e. components. This research is carried out to formulate practice and framework to enable distributed knowledge capturing, transforming the captured knowledge from informal knowledge to a formal model, then use the formal model to develop the knowledge based engineering application. The complete process can be recursive should there be a need to revise or evolve design. The research has demonstrated that KBE applications can be built using modified ICARE form that can directly be used to produce a program template. This challenges the previous researches where ICARE was considered to be informal model. If ICARE form is recorded in XML or JSON format, it can be used to generate the KBE program template. Since the data is recorded in XML/JSON git can be used to record the information and maintain the versions. The generated code template uses open source CSG library independent of design authoring tools. This code can be submitted to git, where all the changes can be tracked and sententiously it allows distributed KBE application development using either Github or GitLab. This satisfies the need of distributed KBE application development as mentioned in few literature. Outsourcing can be enabled on top of the distributed KBE application development with restrictions and security, and at the same time expertise of the domain experts due to open and distributed nature of the development.

Keywords: KBE, Knowledge Based Engineering, Outsourcing, Product Design, Distributed Development.

1 Introduction

The physical products that are designed and manufactured have one or more components involved. Every component has a geometry that is built with certain relevance to mathematics or heuristics. This provides a way to document the mathematics, or rule behind building any geometry that a component takes shape of. Usually such knowledge is preserved by way of documentation, or in some cases as Knowledge Based Engineering systems that operate with a Computer Aided Design (CAD) software or design authoring tool. Evolution of such systems are heavily dependent on how a given CAD software evolves. Often this is also limited by the newer versions of CAD applications. A dependency on CAD also limits overall idea-tion and building of Knowledge Based Engineering Systems in terms of licensing much before the design thinking process starts. Since the deep coupling of CAD and KBE systems are in place, it often gives a circular dependency between CAD and KBE as to which is the parent system. Looking from an organization perspective the Knowledge Based Engineering systems need to derive the CAD, whereas the KBE systems are built to run in a licensed CAD environment. It is often the case that the organizations involved in the design of products rely on various software vendors to build the Knowledge Based Engineering systems due to lack of expertise in building software systems. This usually needs the organizations to share the CAD licenses due to the cost involved in procuring the CAD licence. This research project will provide a framework to build the Knowledge Based Engineering system outside the CAD environment. And, it will also provide a practical implementation of the framework for outsourcing the KBE development independent of CAD systems.

Through the technological advances capturing of data has become seamless, the latest addition is the LIDAR. So, the electronic gadgets can now capture the text, image, audio, video,gps, sensor readings and 3D data. There are machine learning innovations happening in converting the captured data to information. Converting the information to knowledge is still a field vast open for research. The engineers designing the products have had long recognized the and have been using the knowledge based engineering (KBE) to solve their problems.

By the year 2021, the world has the best hardware for computations, cloud computing has become a norm. The CAD applications however are a different domain, while various authoring tools are available as web applications the CAD domain is catching up. CAD applications always need more hardware than what other web applications need.

While various software applications chose the way of open source software, commercial CADs have been largely closed source, and there have been licence costs attached to the CAD applications, which has resulted in limited exposure to the developer -communities.

KBE often needs the domain experts and knowledge engineers to interact. The knowledge engineers are from an experience of software development and the domain experts are the product or design owners. To give a sense of what design authoring tools are capable of, consider two geometry, a cube and a sphere. For simplicity let’s assume that both the geometries have similar size. A design authoring tools can help generate the combination of shapes by applying series of boolean operations such as union i.e. adding up the geometry, subtraction which is removing equivalent materials from geometry to another. Another operation is the intersection in which the common bounding area of the geometries are retained. Image below demonstrates the operations. The set of operations come under the name of constructive solid geometry (CSG).

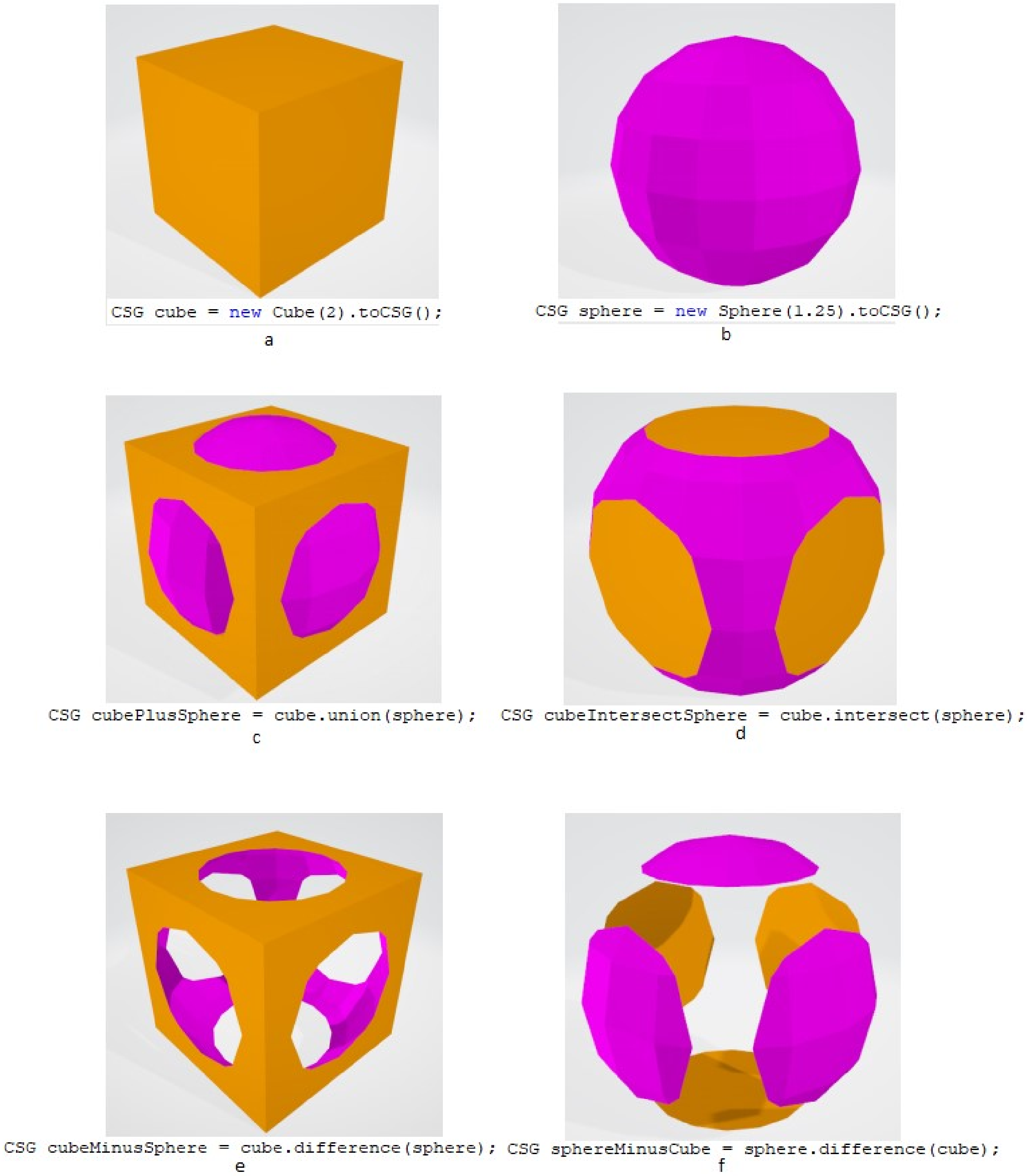


Figure 1 Constructive Solid Geometry Example

1.1 Significance of the Study

Knowledge generated in the domain of building KBE applications is constraint to the given design authoring tool. There can always be common reusable components. The review of literature has brought forward specific need for providing KBE frameworks that can support the distributed deployment of KBE systems (Verhagen et al., 2012), and allow the collaboration between the various domain experts to further make the KBE field take up design challenge of the future (Rocca, 2012).

1.2 Research Questions

Capturing and maintaining the product knowledge is generally a documentation work. To have the knowledge represented as ontology is desirable. This is due to the fact that it helps the automated programs to create and validate different aspects of changes to the product and its building blocks i.e. components. MOKA and KNOMAD are methodologies that are in practice to develop the KBS in the context of KBE. These are largely theoretical in their scope of definition. These frameworks support the iteration in designs, but do not provide a way to version the process of evolution of the designs. The research aim to answer following questions:

1. Can knowledge capturing be linked to ontology development in a parametric framework?
2. Can the versioning of knowledge capturing be done within the scope of ontology development?
3. Can the ontology development be independent of the design authoring tool's environment?

2 Review Of Literature

2.1 Introduction

With the product design every component would need ontology of its own. This would be like a fractal of ontologies. Consider a table as a product represented by certain ontology that inherently will have ontology for design of the legs and the top of the table. An interesting use case comes up when these become reusable for other designs, for instance the ontology of the leg can be reused for a design of chair leg and ontology of table top can be used for chair seat. Now with the simple example it can be observed that although the table and chair can visually look very different but can use the same child ontologies.

Such re-usability of the child components to design new components is commonly used in software development. This is something that has been in use in very limited fashion in CAD designs. Some literature(Studer, Benjamins and Fensel, 1998) have indicated that the experts have sought for such frameworks and repositories of ontologies that can reuse the KBE models for newer designs.

Looking at the productivity improvement data from few literature, it is evident that the success of KBE applications is measured by the time saved in reduction of total design time (Guo et al., 2015; Dongming Xu et al). Some literature also mention the advantages of using KBE to generate the new never seen before designs in aircraft wing designs. One aspect that seems to be not appearing as a significant contribution of KBE is the consistent quality output against the same designs being done by a designer.

2.2 Knowledge Representation

Case based reasoning, Frame based reasoning in Semantic Network in knowledge representation. Semantic Net has been considered as the first generation of models to represent knowledge (Kendal and M Creen, 2007). Various type of objects can be linked in a Semantic net, this gives an advantage in representing knowledge as it helps the engineers to capture objects of different characteristics in a single network. Engineering-Based Expert Systems (EBES) and Knowledge-Based Expert Systems (KBES) are compared and have sighted that both the systems suffer from the ambiguity in knowledge representation and also have been contrasted with the simulation engines (Tzafestas, 1993). Historically Artificial Intelligence (AI) and Expert Systems (ES) have been the two fields of study in similar time frames in 1980s as same. They both are largely labeled as same in published literates in early days of researches. This is largely because they both have same objective of representing the knowledge that humans have. With recent advancement in Machine Learning (ML), AI has been able to discover knowledge that was hidden to humans, like helping scientists to discover new planets, performing image operations like generating realistic photos of faces that have never existed. The expert systems could be built with the knowledge captured from the experts in domain and have it automated. These have also been undergoing evolution with help of ML.

2.3 Terminologies

Over the years of some notable KBE methodologies have been practiced. MOKA (Methodology and software tools Oriented to Knowledge based engineering Applications) is a process framework proposed to facilitate the KBE and thus building Knowledge Based System (KBS) (Verhagen et al., 2012; Häußler and Borrmann, 2021). ICARE (Illustration, Constraint, Activity, Rule, Entity) is one such form to capture the information needed for KBE. Overall ontology modeling is the final aim of KBE. The modeled ontology will be required to run as KBS.

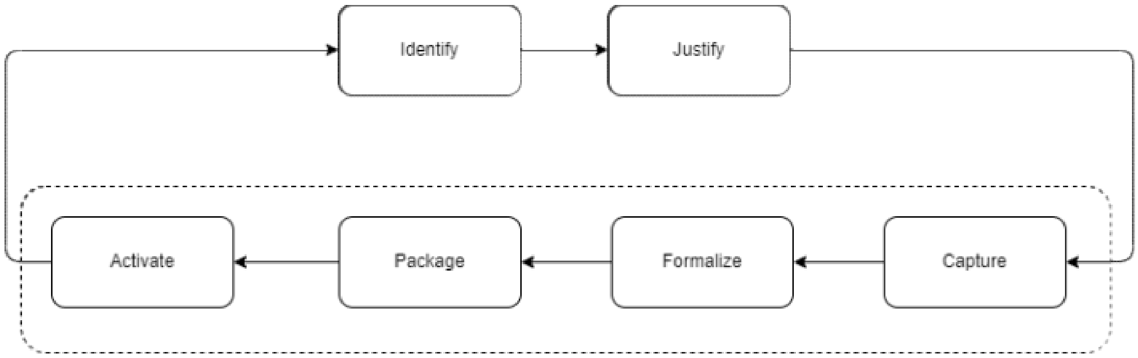


Figure 2 MOKA with modules that can be versioned.

As of the observations, more and more KBE work is happening but in the context of specific CAD applications largely. There are common data exchange formats available for CAD applications, but there isn't a common open format available for KBE applications. It is indicated by (Montali et al., 2017) that KBE is useful in emerging products as well, but it is also difficult to implement. It is also cited that KBE needs to be having provisions for multi-objective design study and predictive design study. The Knowledge Acquisition (KA) is also an important and open aspect of KBE (Leu and Abbass, 2016). Further to this it is also sought that AI tools would be very helpful in extracting Knowledge from the data (Quintana-Amate, Bermell-Garcia and Tiwari, 2015). This further highlights the need of research needed in the integrated domain of KBE, AI and distribution of knowledge (Quintana-Amate, Bermell-Garcia and Tiwari, 2015). A basic framework for KBE is suggested and experimented by (Quintana-Amate et al., 2017), KNOMAD (Knowledge Capture (K), Normalisation (N), Organisation (O), Modelling (M) and Implementation, Analysis (A) and Delivery (D) has been suggested as the preferred methodology, but the implementation of the framework still is limited to CAD specific. MMG Multi Model Generator is also a feature that helps in generating geometries for KBE. KOMPRESSA (Knowledge Oriented Methodology for the Planning and Rapid Engineering of Small-Scale Applications), DEKLARE (Design Knowledge Acquisition and Redesign Environment), DEE (Design and Engineering Engine) is an overall multidisciplinary design optimization approach.

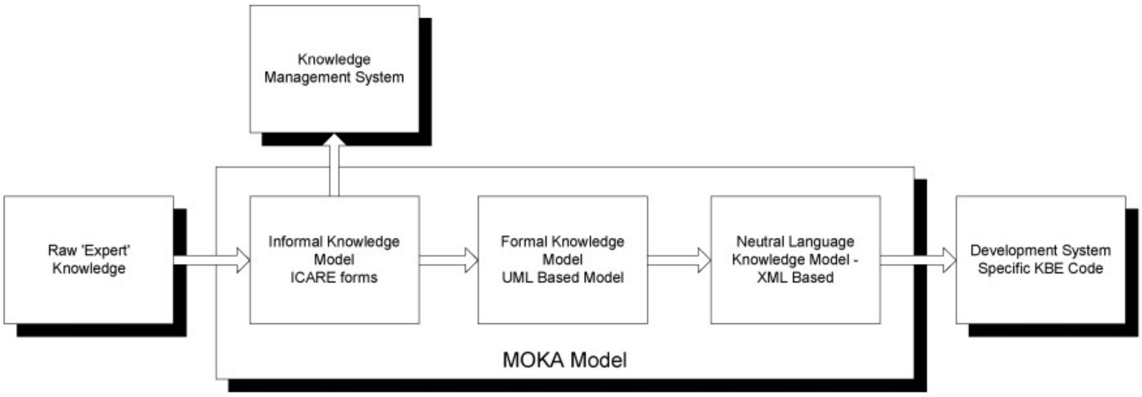


Figure 3 MOKA knowledge models (Preston et al., 2005)

Evolution of newer framework has been proposed to address the inclusion of multi-disciplinary knowledge into the KBE application. As claimed, MOKA has no provision for building KBE applications within multi-disciplinary design optimization (MDO). For addressing the gap researchers have devised the Design and Engineering Engine (DEE) approach (Curran et al., 2010). The DEE consists of three major elements it starts with the design process, which includes multi-disciplinary design and its optimization. It starts with requirement gathering and also perform some baseline calculations for the feasibility of the design. After this, documentation containing product model parameter values is generated. Then it passes on the parameter values to the next element of the DEE, the Multi-Model Generator (MMG). The MMG uses the product model parameter values from the previous step in combination with domain knowledge to generate product geometry.The MMG covers the fully formalized KBE applications that can originate from applying the MOKA KBE Life-cycle. The MMG generates report documents. These are supplied to the next application of the DEE: i.e. analysis modules. These applications take the feed from the MMG and perform the simulation based on the calculations. Designs are finalized by analyzing the Data files in a Converger & Evaluator, which checks for mathematical validity (convergence) and requirements compliance (Evaluator) in a multi-disciplinary fashion (Curran et al., 2010).

2.4 MOKA & ICARE

Investigating further into definitions, MOKA means methodology and tools oriented to knowledge based engineering (KBE) applications. MOKA has cam into being for the purpose of reducing the cost of developing and iterating the KBE applications. This is also a common practice in industry to employ MOKA. It is believed that MOKA adds efficiency to the whole process of the application development. Important aspect of MOKA is driven via ICARE forms (Illustration, Constraint, Activity, Rule and Entity). ICARE provisions standard template for capturing knowledge. Some literature are also of the view that ICARE also provides storage and ontology related capabilities. This however have not be provided with appropriate proofs. The fact that MOKA is largely an informal model, the ontology related capabilities can not be induced. However it can not be ruled out that MOKA in combination with ICARE can also be formalized with ontology related capabilities. This research will aim at demonstrating it.

MOKA ICARE FORMS:	Structural Entity	
Name	2 litre bottle assy	
Reference	ES5.BOTTLE.ASSY.2L	
Related Functions	Containment of liquid EF5/1_LQD.CONTAIN Prevent contamination of contents EF5/2_LQD.CONTAM.PREV Withstand handling and storage EF5/3_ROBUSTNESS	
Behaviour	Ease of assembly EB5/1_ASS	
Context, info, validity	2l bottle for non-carbonated soft drinks	
Description	The bottle assembly consists of two parts: Main bottle body • Bottle cap Main bottle body is blow moulded on-site from recyclable blue or clear PETE depending upon the type of liquid bottle is intended to contain Bottle cap is mid-blue in colour and bought in directly from Bericap	
Related entities	Parent	
	Children	ES5_1.BOTTLE.2L ES5_2.CAP
	Undefined	
Information origin	Interview with D. Smith from design dept. 11-10-03	
Management	Author	STP
	Date	23-10-03
	Version Number	1.1
	Status	In Progress

Figure 4 ICARE, Example of a MOKA entity form (Stokes, 2001)

2.5 Limitations of current KBE development

Although KBE has been into practice from late 1980s, there have been challenges embedded into it. The challenges have been high software license cost of design authoring tools, absence of the documentations on designing KBE application, accessibility of the KBE applications, KBE vendor's involvement and absence of generalized solution and marketing (Rocca, 2012). It is clear that design authoring software's cost and then development of KBE application in the design authoring environment itself is presenting a barrier for KBE vendors to provide the solution's. As an example consider a vendor has developed a KBE application to build a product with some common designs and some organization specific design. Now when the vendor needs to work for another organizations, the common KBE designs become unusable if the next organization doesn't have same design authoring software. In this case the KBE vendor has to redo same application design in again. Overall, in general case scenario, a vendor becomes more efficient over time by building software applications, and reusing common components. However in KBE application development the vendor has no way to use the common design components. This way the development cost of the KBE application always goes up with time, instead to becoming more competitive.

Popular methodology have existed as general procedure, but it has largely been an optional step in KBE application development. Gaining the understanding from various literature, it is imperative that MOKA should be a necessary framework than an optional process. As MOKA can bring some level of requirement to code documentation, it will need to be created as an design authoring tool independent software. As an important quote "Without any doubts, future KBE systems will benefit from any progress in the fields of knowledge modeling and storage frameworks development and the ability to integrate with them. Even further, the impact and diffusion of future KBE systems will benefit from the advances in the development of design rationale systems [61]. A combination of these systems with KBE would offer designers a powerful and more complete toolbox to enhance the level of automation in design, to improve the level of support for decision making, as well as a powerful, dynamic and transparent system for knowledge capturing and re-use.” (Rocca, 2012). More quotes that indicate towards the need of web-based solution for KBE application development “When the first KBE system came into the market, the World Wide Web did not yet exist, nor was the concept of open source software widely known or appreciated. It is natural to expect new KBE systems to be geared toward their use and deployment via the web [46]. The possibility to free the way to open source solutions, from one side, would relieve the end-user to return so often (with cash) to the ‘‘KBE vendor shop’’ and buy dedicated plug-ins (e.g., for graphics generation and visualization, or for the manipulation of XML files). From the other side, it would relieve the KBE vendor from the burden of in house development of typically short life and buggy applications and extensions which may or may not align well with their core focus. Indeed, open source application and extensions have the potential for having fewer bugs than vendor supplied attempts, since a large number of peers have the possibility to bring improvements.” (Rocca, 2012). Then on documentation “The automatic link code-documentation should actually work dynamically and in both directions. Engineers should have the possibility to agilely and interactively generate diagrams representing the structure and the design process of their products, having an interpreter active in the background that is able to generate at least the main structure of the KBE application code.” (Rocca, 2012).

3 Research Proposal

3.1 Methodology

The primary research method for this study is literature review and conceptual modeling using software application development. The proof of concept (POC) will be built as a web application. Setting up the CAD independent knowledge capturing is the first step in building a KBE framework for outsourcing. Next will be the need to convert the captured knowledge in the form of a Rule Based System (RBS) & Frame Based System (FBS) (Rocca, 2012). The generated RBS or FBS code will be available for adding the rules that will use the design parameters captured via the first step of knowledge acquisition. A constructive solid geometry (CSG) module will be used to generate the 3D models based on the RBS/FBS code. Once the 3D geometry is created a 2D drawing will be created in a CAD environment. This will then be made available as ontology to generate different variations of the geometry based on design needs. This will also demonstrate the iterative process by applying changes in knowledge base.

3.2 Proposed Framework

A software model will be build to attain the objective of research and there by answer the important research questions around KBE data capture, formalization of data , and thus use the formalized data to generate object oriented model for further application development. A distributed model will also be built to demonstrate the deployment of the KBE application on cloud independent of design authoring tool.

Following diagram represents the model

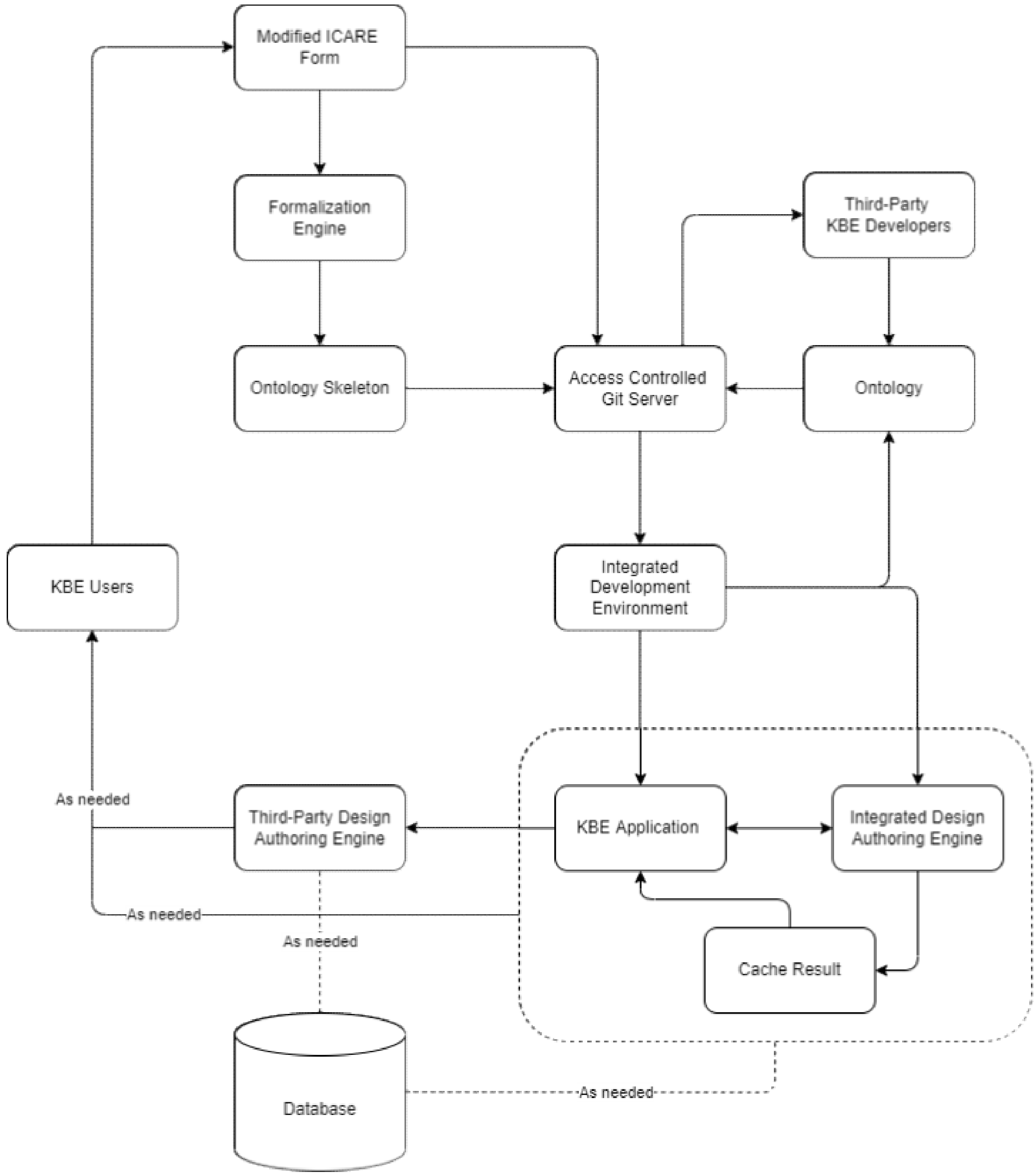


Figure 5 Proposed model for this research.

The KBE development framework MOKA covers the basic and cyclic nature of the complete project life cycle. This research adds the versioning component in each iteration of the KBE application development cycle. Here by the document will address this modified framework as versioned MOKA . As indicated in the figure, under MOKA identify ans justify phase is an activity to be carried out by the business needs, hence it is kept out of the formal versioning scope. And in context of outsourcing, it is safe to keep the versioning and documentation of business decisions independent of the KBE application development process. This will act as a security measure by design in the versioned MOKA methodology.

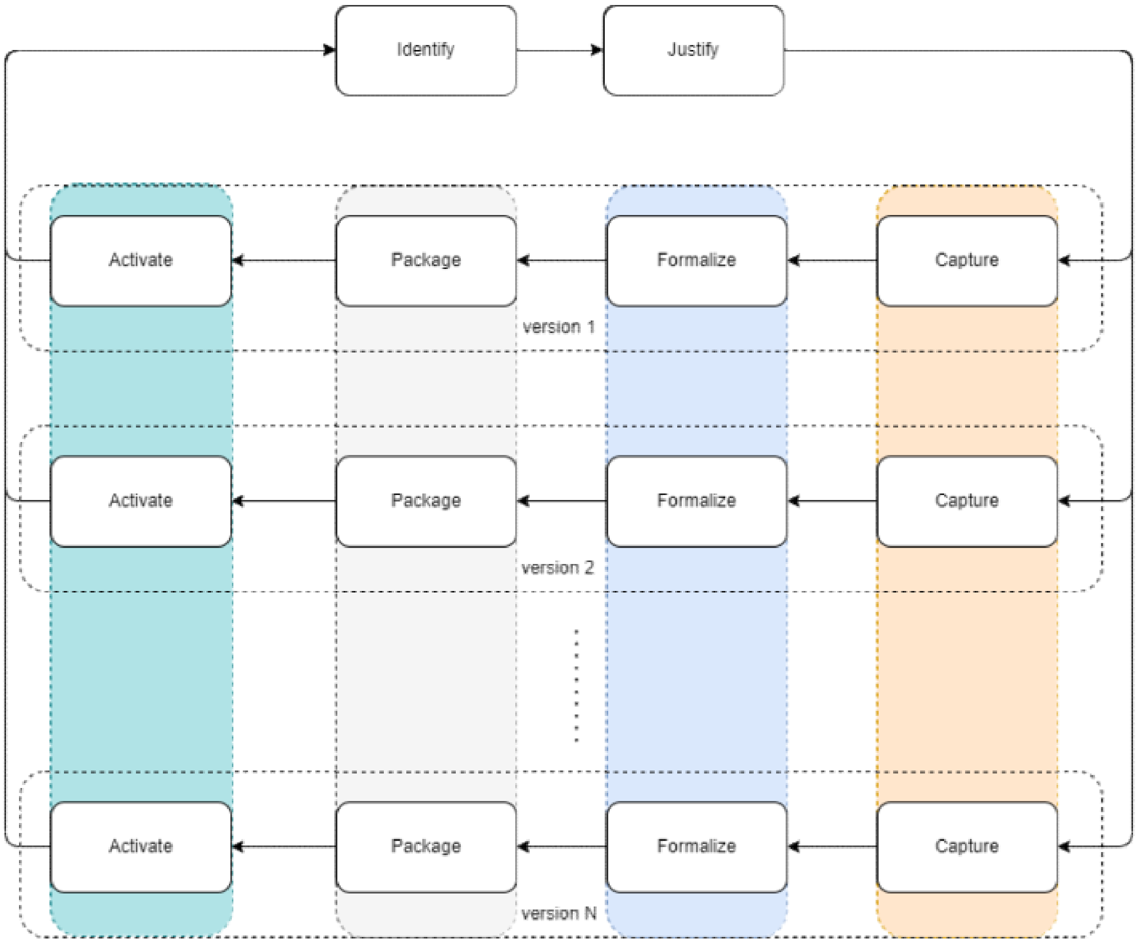


Figure 6 Versioned MOKA enables versioning for each step

3.3 Code & Data Security aspects

For collaboration on development projects needs the git to be hosted on a server. The server hosting has standardized process and can be referred from various available documentations. For standardized writing, the research will refer to book by Chacon and Straub named "Pro Git" published in the year 2014. The hosted server can be accessed via four protocols namely, Local, HTTP, Secure Shell (SSH) and Git. The Local protocol is used when git repository is hosted on same computer server where different users can log in and collaborate in coding. This can be used by organizations that need very strict control on code transfer from its network. However it will come with cost of hosting high-end computer servers. Second protocol is HTTP, which is most widely used protocol for collaboration on public repositories. In context of KBE it can be referred as the components of standards provided by institutes like BIS and ASME. Here, since the design specs are already available in public domain so it is truly distributed open source collaboration ob KBE. Third protocol is SSH, in the aspect of restricting access to code data it is similar to Local protocol, with a difference that, in SSH one can check out the code to their local computer for development. The last protocol is Git, this is the fastest protocol in terms of data transfer, but less secure than previous protocols.

It will need a one time architecture setup of organizations public and private KBE projects. So, it is expected that this setup would be done by experienced and security aware professionals for the given organization. There are standards practices available for collaboration on git, including testing and merge conflicts with release controls. It is expected that Knowledge Based Engineers are aware of the such practices.

3.4 Outsourcing

There has been detailed process research on KBE , and also the detailed work related to Outsourcing in Software development. But there has been limited or no research done in the KBE outsourcing framework. This will be one for the prime objective to develop the framework so that a supporting framework can be developed. Scope, code and model versioning will need to be unified and MOKA methodology will be supported to start with. Version control systems such as Git plays a major role in distributed development of the applications and at the same time it provides the security. Git has been adopted as primary tool for open source application development and also for the closed source application. It is a distributed version control system as illustrated in figure (Chacon and Straub, 2014).

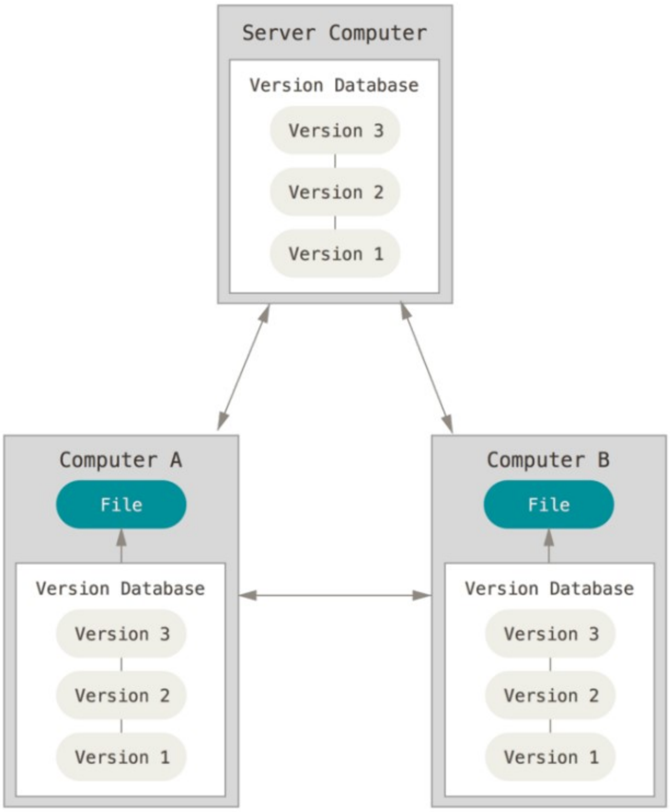


Figure 7 Distributed version control.(Chacon and Straub, 2014)

Another, notable property git brings in is the way versions are stored. Git stores the data as snapshots of the project/repository.

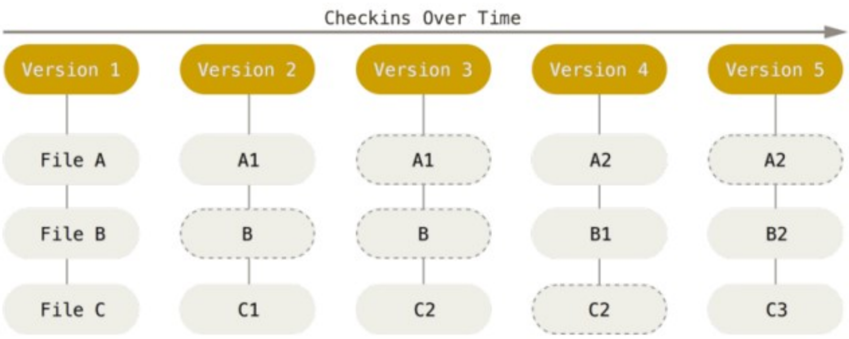


Figure 8 Storing data as snapshots of the project over time.(Chacon and Straub, 2014)

3.5 Data

Product design data is collected from Bureau of Indian Standards (BIS). About the BIS as per its own definition "BIS is the National Standard Body of India established under the BIS Act 2016 for the harmonious development of the activities of standardization, marking and quality certification of goods and for matters connected therewith or incidental thereto.BIS has been providing traceability and tangibility benefits to the national economy in a number of ways – providing safe reliable quality goods; minimizing health hazards to consumers; promoting exports and imports substitute; control over proliferation of varieties etc. through standardization, certification and testing." (Bureau of Indian Standards, 2020)

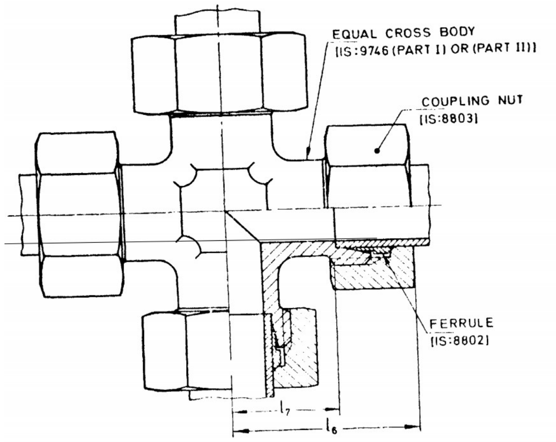


Figure 9 SPECIFICATION FOR EQUAL CROSS COUPLING ASSEMBLY FOR OIL-HYDRAULIC SYSTEMS, IS : 10417 - 1983 (Bureau of Indian Standards, 2020)

The data collected from BIS is standard data set that is available in public domain. Any researcher or organization can obtain this data for results verification. The variables have the various combinations. The data is mostly available in tabular form, and needs manually to be translated to the modified ICARE. In general practice the data is captured in documents in unstructured format and later captured into the ICARE, so this process resembles the actual design specification capturing.

4 Results

4.1 Research Question One

The question was "Can knowledge capturing be linked to ontology development in a parametric framework?". To answer this question, first the informal model (ICARE) was created to present the existing convention of capturing the data for KBE application development as per MOKA. The ICARE form can be compared against Example of a MOKA entity form (Stokes, 2001)

The form below is an attempt to capture IS 3175 : 2013 from Bureau of Indian Standards into the informal model ICARE form.

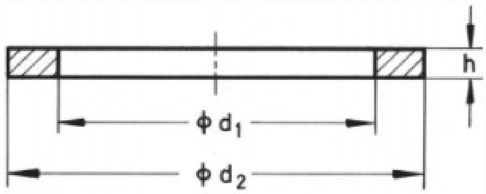
MOKA ICARE FORMS	Entity
Name	IS 3175 : 2013
References	INTERNAL COMBUSTION ENGINE, PIPE UNIONS
Entity Type	part
Functions	INTERNAL COMBUSTION ENGINE — SEALING WASHERS FOR PIPE UNIONS
Behaviour	Provides sealing
Context, Information, validity	For pipe unions in internal combustion engine, A sealing washer of nominal size 10 × 15 made of copper shall be designated as: Sealing Washer 10 × 15 IS 3175 Cu
Description	 Unless otherwise agreed to between the manufacturer and the purchaser, the sampling scheme and criteria for conformity with this standard shall be as laid down in IS 1367 (Part 17).
Related Activities	IS 1367 (Part 17)
Related Entities	Copper of minimum purity 99.5 percent conforming to IS 191 or aluminium grade 19000 H2 conforming to IS 737.
Related Constraints	2643 : 2005 Pipe threads where pressure-tight joints are not made on the threads — Dimensions, tolerances and designation (third revision)
Related Rules	1367 (Part 17) : Industrial fasteners — Threaded steel fasteners — Technical supply conditions: Part 17 Inspection, sampling and acceptance procedure (fourth revision)
Related Illustrations	Table 1 Dimensions and Tolerances for Sealing Washer
Information Origin	Bureau of Indian Standards
Management	BIS is a statutory institution established under the Bureau of Indian Standards Act, 1986 to promote harmonious development of the activities of standardization, marking and quality certification of goods and attending to connected matters in the country

Figure 10 MOKA ICARE form to capture the informal model

Modified ICARE for versioned MOKA is an approach towards the creation of KBE application that can have capabilities to bridge the gap between the application code and the captured knowledge.

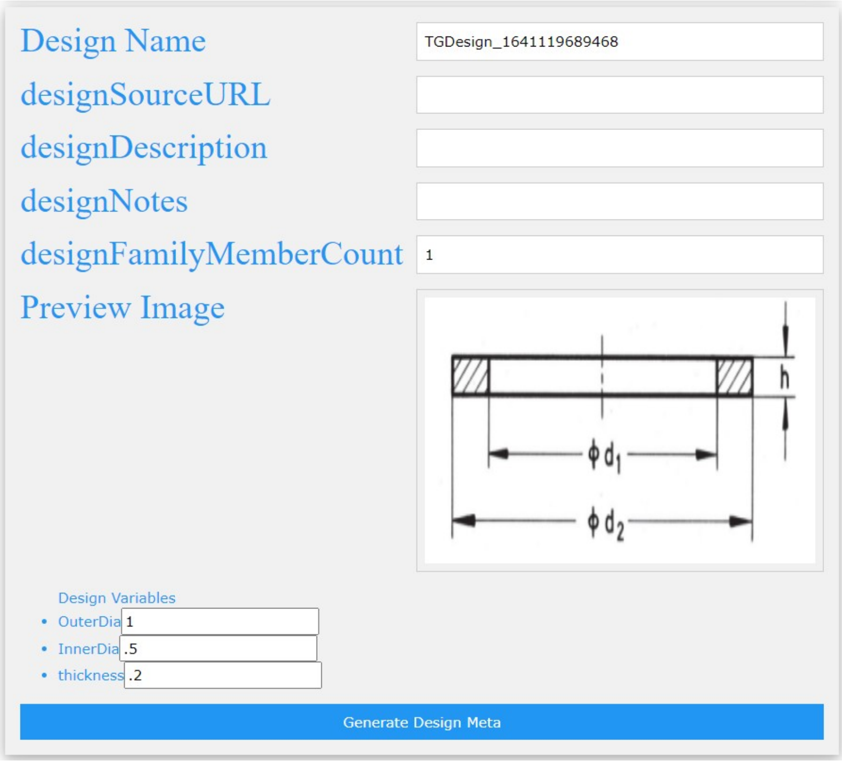


Figure 11 Modified ICARE form for formal model

Then editable modified ICARE form generated from previous step. Important aspect that makes the modified ICARE form different from ICARE is the inclusion of the design variables along with the formal definition. As an example “OuterDia”, “InnerDia” and “thickness” variables are captured in the modified ICARE form itself. This helps the developers in defining the independent design variables. Independent design variables are understood to be the variables that users can supply from the KBE application consoles and these variables define the rest of the product geometry and material selections.

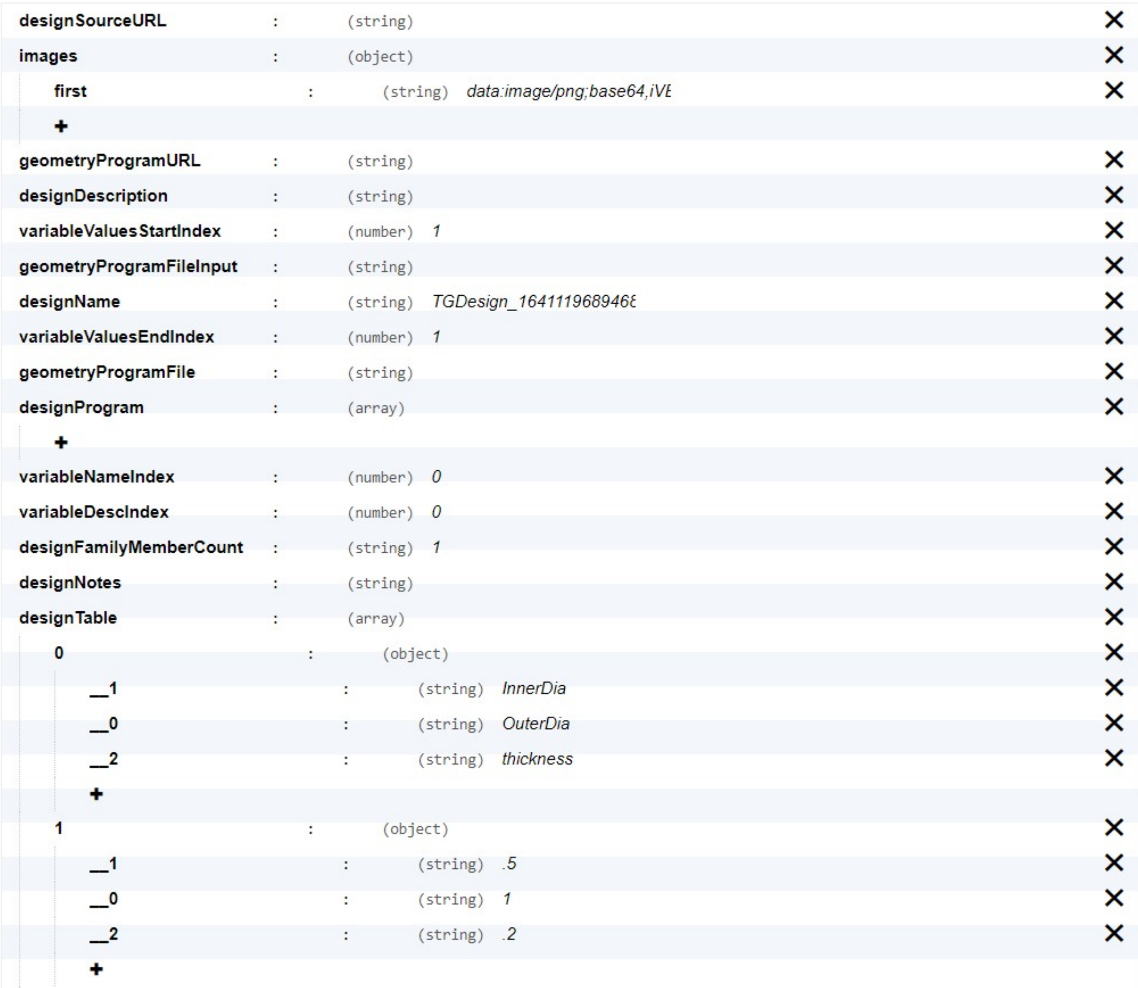


Figure 12 Editable ICARE form for formal model

The editable ICARE form is introduced as part of the KBE development life-cycle along with the code itself to track the changes in formal model due to the informal information. Also the code was written conventionally. Example code shown below is to demonstrate the creation of the geometry independent of the design authoring tool. It can be observed that the geometry creation for a sealing washer geometry required few line of instructions.

```java
double d2=10;
double d1=6;
double h=1;
CSG outerCyl=new Cylinder(d2, h, 30).toCSG();
CSG innerCyl=new Cylinder(d1, h, 30).toCSG();

try {
        outerCyl.difference(innerCyl).toObj().toFiles(Paths.get("sealingWasher.obj"));
} catch (IOException ex) {
        Logger.getLogger(SealingWasher.class.getName()).log(Level.SEVERE, null, ex);
}
```

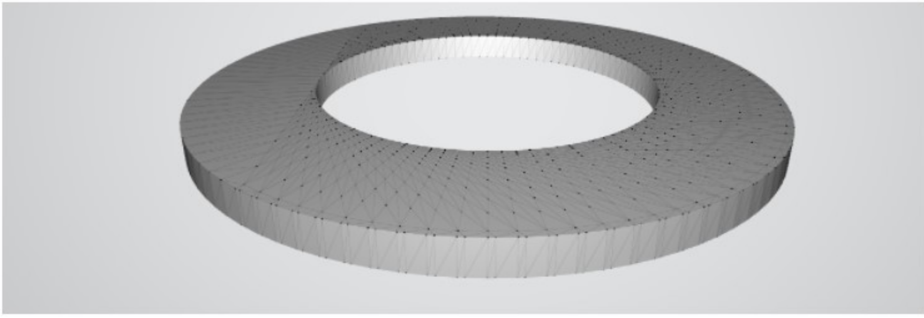
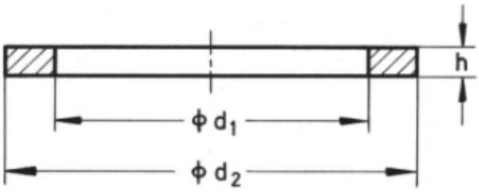


Figure 13 Code sample for creating a sealing washer geometry

4.2 Research Question Three

Automatic generation of the code template. This eliminates the need of manual creation/formalization of captured knowledge using UML.



Save

```java
import eu.mihosoft.jcsg.CSG;
import eu.mihosoft.jcsg.*;
//Template Generated by GeneralFlatPattern.class
//Copyright True Geometry (OPC) Private Limited
public class TGDesign_1641119689468 {
  int designFamilyMemberCount = 1; //Refer to 2D diagram for variations
  int designFamilyMember = 0; //Variable to be supplied, it is subset of designFamilyMemberCount
  double __1 = .5; //InnerDia => InnerDia
  double __0 = 1; //OuterDia => OuterDia
  double __2 = .2; //thickness => thickness

  public CSG toCSG() {
    if (designFamilyMember == 0) {
      return new Cylinder(10, 10, 10).toCSG();
    }
    return new Cylinder(10, 10, 10).toCSG();
  }


  //Generate values of design parameters
  public String[] toDesignText() {
    return null;
  }

  public TGDesign_1641119689468(int designFamilyMember, double __1, double __0, double __2) {
    this.designFamilyMember = designFamilyMember;
    this.__1 = __1;
    this.__0 = __0;
    this.__2 = __2;
  }
}
```

Figure 14 Code template generated from ICARE formal model

Then the code is modified as shown below:

Save Run

```java
import eu.mihosoft.jcsg.CSG;
import eu.mihosoft.jcsg.*;
//Template Generated by GeneralFlatPattern.class
//Copyright True Geometry (OPC) Private Limited
public class TGDesign_1641631289653 {
  int designFamilyMemberCount = 1; //Refer to 2D diagram for variations
  int designFamilyMember = 0; //Variable to be supplied, it is subset of designFamilyMemberCount
  double __1 = 10; //OuterDia => OuterDia
  double __0 = 8; //InnerDia => InnerDia
  double __2 = 1; //thickness => thickness

  public CSG toCSG() {
     CSG outerCyl=new Cylinder(__1, __2, 100).toCSG();
        CSG innerCyl=new Cylinder(__0,  __2, 100).toCSG();

        try {
            outerCyl=outerCyl.difference(innerCyl);
        } catch (IOException ex) {
            Logger.getLogger(SealingWasher.class.getName()).log(Level.SEVERE, null, ex);
        }

    return outerCyl;
  }


  //Generate values of design parameters
  public String[] toDesignText() {
    return null;
  }

  public TGDesign_1641631289653(int designFamilyMember, double __1, double __0, double __2) {
    this.designFamilyMember = designFamilyMember;
    this.__1 = __1;
    this.__0 = __0;
    this.__2 = __2;
  }
}
//result=new TGDesign_1641631289653(int designFamilyMember ,double __1,double __0,double __2).toCSG()
//result=new TGDesign_1641631289653(0,__1,__0,__2).toCSG()
//result=new TGDesign_1641631289653(10,8,1,).toCSG()
```

Figure 15 Completed KBE code for geometry creation

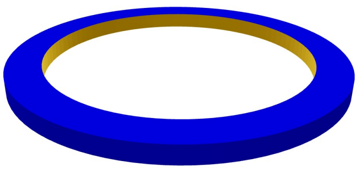


Figure 16 Geometry generated from the CSG code

Similarly nut geometry was generated with code

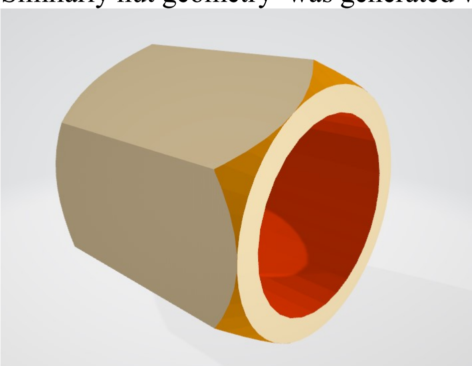


Figure 17 Nut geometry generated from the CSG code

Similarly for FERRULES geometry was created

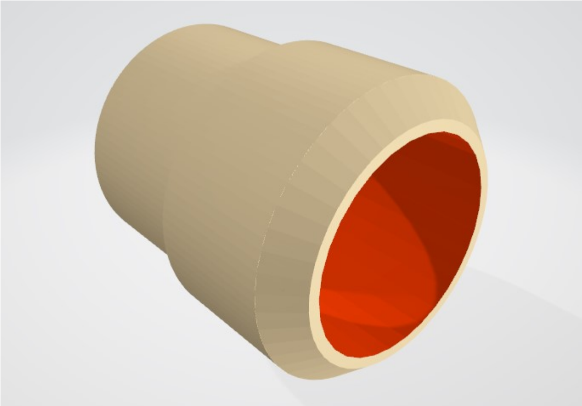
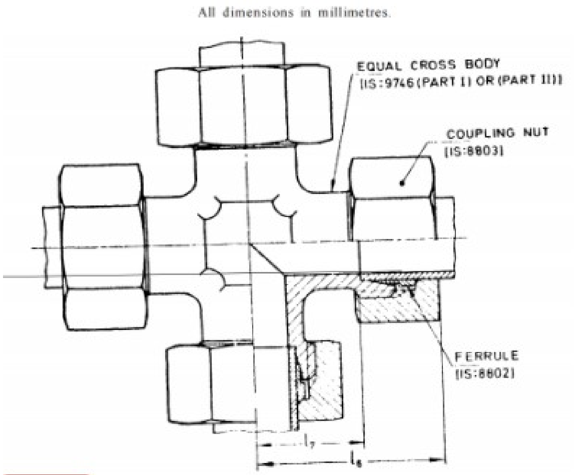


Figure 18 Ferrules geometry created from CSG code.

Equal-cross body geometry generated with program

Figure 19 Equal Cross Body generated from CSG code



Save

```
import eu.mihosoft.jcsg.CSG;
import eu.mihosoft.jcsg.*;
//Template Generated by GeneralFlatPattern.class
//Copyright True Geometry (OPC) Private Limited
public class TGDesign_1642774674429 {
  int designFamilyMemberCount = 1; //Refer to 2D diagram for variations
  int designFamilyMember = 0; //Variable to be supplied, it is subset of designFamilyMemberCount
  double __1 = 11; //17 => 17
  double __0 = 21; //16 => 16

  public CSG toCSG() {
    if (designFamilyMember == 0) {
      return new Cylinder(10, 10, 10).toCSG();
    }
    return new Cylinder(10, 10, 10).toCSG();
  }


  //Generate values of design parameters
  public String[] toDesignText() {
    return null;
  }

  public TGDesign_1642774674429(int designFamilyMember, double __1, double __0) {
    this.designFamilyMember = designFamilyMember;
    this.__1 = __1;
    this.__0 = __0;
  }
}
//result=new TGDesign_1642774674429(int designFamilyMember ,double __1,double __0).toCSG()
//result=new TGDesign_1642774674429(0,__1,__0).toCSG()
//result=new TGDesign_1642774674429(11,21,).toCSG()
```

Figure 20 Code template for Equal Cross Body assembly

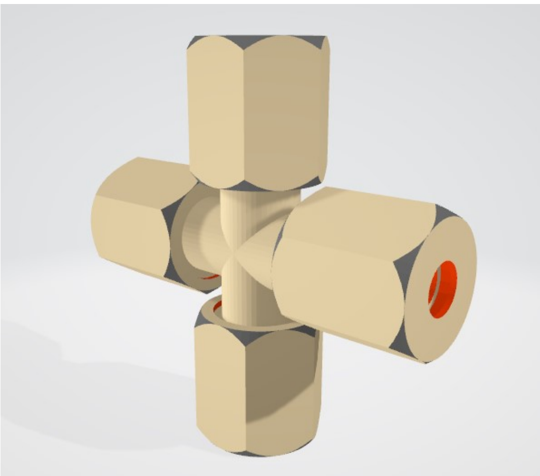


Figure 21 Equal Cross Coupling Assembly generated from CSG code

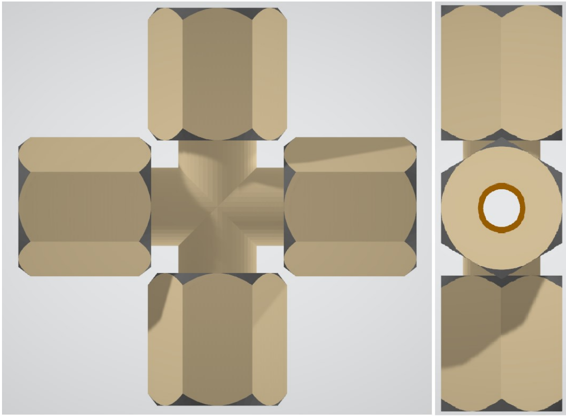


Figure 22 Equal Cross Coupling assembly projections

As it was shown that there are libraries available as part of open source to generate geometry pragmatically independent of the CAD or design authoring systems.The output of 3D models are available in STL and OBJ formats. STL file format is most popular for 3D printing. It is a neutral 3D file format. This file format only stores geometry information. As a scope of future, since 3D printing is becoming more and more mainstream, the drawings of the components may become obsolete, as the manufacturing can directly be done the models generated with the KBE applications. An important aspect to consider is that, once the models are generated via KBE applications they are expected to not to be modified manually, and instead re-create from the KBE applications. This ensures that the information about the changes are preserved for future references. And once this is in place, the 3D models generated from KBE applications can be imported to design authoring tools as well. It also provides the immutability to the designs of the components that are not expected to change during new assembly designs. To further enhance the systems, one can generate alerts and alarms on the geometry checks.

4.2 Research Question Two

Project structure is shown below. This is a typical maven project structure used for developing the software application that are to be written in java. It consists for four category of files, first is the “Source Packages”, it contains all the source code that needs to be written by the application developer. Second category of files are “Dependencies”, these are the set of compiled libraries that are open source and available in a central repository. Third category is “Java Dependencies”, these are the files that come along with the application software development kit abbreviated at sdk. The final category is “Project Files”, this contains the meta data about the project such as, the name of the project, dependencies, and compilation information.

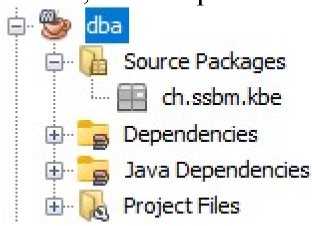


Figure 23 Sample project Structure used for research

Then the project is initialized as git repository. Version control systems such as Git plays a major role in distributed development of the applications and at the same time it provides the security. Git has been adopted as primary tool for open source application development and also for the closed source application. It is a distributed version control system as illustrated in figure (Chacon and Straub, 2014).

```
-------------------------
Entry: DiffEntry[DELETE TGOutput/TGDesign_1641119689468.java], from: AbbreviatedObjectId[e1935aac0208165123d3ba2b2c4723fb5e4cd295], to:
AbbreviatedObjectId[0000000000000000000000000000000000000000]
diff --git a/TGOutput/TGDesign_1641119689468.java b/TGOutput/TGDesign_1641119689468.java
deleted file mode 100644
index e1935aa..0000000
--- a/TGOutput/TGDesign_1641119689468.java
+++ /dev/null
@@ -1,29 +0,0 @@
-import eu.mihosoft.jcsg.CSG;
-import eu.mihosoft.jcsg.*;
-//Template Generated by GeneralFlatPattern.class
-//Copyright True Geometry (OPC) Private Limited
-public class TGDesign_1641119689468{
-int designFamilyMemberCount=1;//Refer to 2D diagram for variations
-int designFamilyMember=0;//Variable to be supplied, it is subset of designFamilyMemberCount
-double __1 =;//InnerDia => InnerDia
-double __0 =;//OuterDia => OuterDia
-double __2 =;//thickness => thickness
-
-public CSG toCSG(){
- if(designFamilyMember==0){return new Cylinder(10,10, 10).toCSG();}
-return new Cylinder(10,10, 10).toCSG();}
-
-
- //Generate values of design parameters
-public String[] toDesignText(){ return null;}
-
-public TGDesign_1641119689468(int designFamilyMember ,double __1,double __0,double __2){
-this.designFamilyMember=designFamilyMember;
-this.__1=__1;
-this.__0=__0;
-this.__2=__2;
-}
-}
-//result=new TGDesign_1641119689468(int designFamilyMember ,double __1,double __0,double __2).toCSG()
-//result=new TGDesign_1641119689468(0,__1,__0,__2).toCSG()
-//result=new TGDesign_1641119689468(,,,).toCSG()
\ No newline at end of file
```

Figure 24 Snapshot of changes in git

5 Summary

Product development organizations use KBE applications to reduce the routine work and generate new combination of variables for product development. Published literature have confirmed that KBE application development is active filed but also limited to the needs of specific organization.

The research demonstrates an approach to develop KBE applications in distributed manner along with translation from knowledge acquisition to application template generation. Added advantage comes up in versioning of the acquired knowledge and the application developed based on the knowledge captured.

It was also observed that the CSG is sensitive to the granularity expected in geometry surface quality. The better quality surface would take longer time to generate. It can be understood that every point in three dimensional space can be represented by three numbers x, y and z. And a minimum of three of these points will be needed to generate one surface which is a triangle. Combination of triangles can be used to form the surfaces. A fully closed surface geometry represents a solid geometry. As a general practice, CSG doesn't necessarily ensure the geometry to be manifold, so the KBE developer needs to do such verification. The versioning aspect has been demonstrated with git. It is evident that ICARE form data, KBE application code , packaging and deployment can be be tracked with each version of changes. Git and open source are allowing the distributed development of the KBE applications. The aspect of data security can be addressed via the options available in git.

Writing the KBE application appeared to be organized due to the formalization of modified ICARE form. It is often the case that the KBE application is developed by the group of engineers. The automated code template generation from the ICARE reduces the ambiguity in scope of application to be developed.

Geometry generated for Equal Cross Body assembly is as per the standards obtained from BIS. This can be double verified with data captured in ICARE. This has presented the very transparent methodology of developing the KBE application, and logical verification of application output as well.

Object oriented programming approach that has been used in research coincides with the Semantic Network in knowledge representation. This is combination of frame based reasoning and case based reasoning. While the individual part design like for nut and equal cross body was based on frame based reasoning, the equal cross body assembly has been the frame based reasoning approach. This has resulted in creation of true ontology for equal cross body coupling.

If an organization decides to execute the knowledge based engineering projects independent of the design authoring systems, it is feasible and also very scalable. It can be used to generate the assemblies of millions of parts. Another advantage will be use of the parallel computing in generating the geometries. Essentially, these models can also be generated using big data computations. So, it is no longer restricted to capacity of one high end computing machine. One can imagine it to be a distributed computing job to generate millions of geometries and store them on a Hadoop cluster. And then from the Hadoop clusters they can be shipped to manufacturers for quote and 3D printing or manufacturing.

The proposed framework of the this research was further used to generate the different component designs and published for spherical gear, spur gear and supercharger. (Manoj,2022a; Manoj, 2022b; Manoj, 2022c).

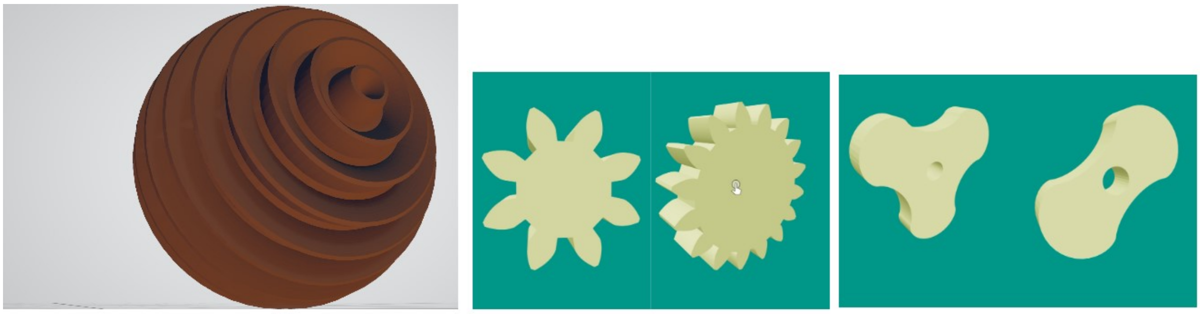


Figure 25 Geometry developed on the proposed KBE framework.(Manoj,2022a; Manoj, 2022b; Manoj, 2022c)

6 Implications

Development of KBE applications are limited to the organization or a department of the organization. This limits the sharing of knowledge of KBE application development. Bringing in the methodology of distributed application development can benefit the organizations in terms of standardization of the process of KBE.

A design authoring tool neutral approach of data capturing or knowledge acquisition places the organization at a better position to switch between the design authoring tools if needed. The research indicates that the code templates can be generated for design authoring tools as well. An example can be the code template in java for Creo, or for UgNX that are some popular design authoring tool.

Developing KBE application requires very specific skill set that includes knowledge of programming language, user interface design, solid geometry, constructive solid geometry, software development life cycle (SDLC), git, high performance computing and cloud computing. Finding all the skill sets in a product development organizations is usually unlikely, and hence outsourcing is something that should be considered to develop the KBE applications.

The questions surrounding the protection of intellectual properties and data security can also be addressed by involving relevant professionals. As the outsourcing of application development is already a practice placed for various domains, the templates to follow can be obtained form such organizations that provide outsourcing.

7 Recommendations for Future Research

This research has focused on the ontology development for 3D models that can be 3D printed. There is scope to extend this research to the drawing and drafting, so that conventional manufacturing can also be achieved with automated drafting. Another research area that will help the knowledge ecosystem grow is to study the open source licensing aspect applied to the product design. As it is already known that large software companies tend to open source the code on which they partially run on with licence like MIT, BSD and more. If the product designs are open sourced in similar fashion, would it benefit the whole ecosystem or not.

It will be recommended that if an organization is going for outsourcing of the KBE applications, they use the design authoring tool independent platforms to generate the 3D geometry and also capture knowledge independent of the design authoring tools. It will protect and preserve the intellectual property of the organizations in case organizations wish to switch to different design authoring tools. This research has shown that it is possible to adopt to and take advantage of the advances in open source revolution and also benefit from the Big Data technologies.

The organizations are proposed to have tow distinct set of teams for designs, one that takes care of the designs based on design authoring or CAD systems, and other that develops the KBE applications for the designs. This will ensure that the organizations intellectual property is constantly being translated to repository of ontology. It should enable the organizations to easily allow the upgrades to the ontology with in-house development or with outsourcing. The organizations will have to encourage the users to collaborate on open source KBE applications to reach to the stage where other open source projects are. It is expected that some intellectual property related questions may arise, but they need to be addressed through the patents instead of user restrictions.

Many product organizations already patent the products where certain critical information is already shared with the community by publishing the patent. In that regard the intellectual properties of the organizations are protected. So, combining the open source licensing and protection from patents, how can the ecosystem evolve?

8 Conclusion

The research has demonstrated that KBE applications can be built using modified ICARE form that can directly be used to produce a program template. This challenges the previous researches where ICARE was considered to be informal model. If ICARE form is recorded in XML or JSON format, it can be used to generate the KBE program template. Since the data is recorded in XML/JSON git can be used to record the information and maintain the versions.

The generated code template uses open source CSG library independent of design authoring tools. This code can be submitted to git, where all the changes can be tracked and sententiously it allows distributed KBE application development using either Github or GitLab. This satisfies the need of distributed KBE application development as mentioned in few literature.

Outsourcing can be enabled on top of the distributed KBE application development with restrictions and security, and at the same time expertise of the domain experts due to open and distributed nature of the development.

The software model was built with help of tools available under open source. Interpretation of the data from BIS needed the knowledge of machine drawing knowledge, which is a typical thing for KBE engineers. A variety of KBE knowledge capturing process needs KBE engineer to conduct the interviews and record the data in ICARE. For the purpose of research that has been replaced by directly obtaining the component design data from BIS, under the non-commercial usage of data. Research was able to capture the design data into the modified ICARE form. The modified ICARE from was then fed to a program that could interpret the recorded data and generate the program template for KBE application. The form currently considers the parts and assembly as same. This has a scope of improvement to treat the part and assembly design separately.

References

Abd Wahab, A. and Piak San, T. (2018). A Systematic Literature Review on Risk Factors in Software Development Outsourcing. *International Journal of Engineering & Technology*, 7(3.20), p.76.

Ågerfalk and Fitzgerald (2008). Outsourcing to an Unknown Workforce: Exploring Opensurcing as a Global Sourcing Strategy. *MIS Quarterly*, 32(2), p.385.

Aatmling B. Narayanpure (2017). MULTIPLY YOUR BUSINESS WITH REAL TIME AUTOMATION. *International Journal of Research in Engineering and Technology*, 06(05), pp.92–97.

Akbar, M.A., Khan, A.A., Mahmood, S., Alsanad, A. and Gumaei, A. (2020). A robust framework for cloud-based software development outsourcing factors using analytical hierarchy process. *Journal of Software: Evolution and Process*.

Akagi, S. and Fujita, K. (1990). Building an Expert System for Engineering Design Based on the Object-Oriented Knowledge Representation Concept. *Journal of Mechanical Design*, 112(2), pp.215–222.

Ammar-Khodja, S., Perry, N. and Bernard, A. (2008). Processing Knowledge to Support Knowledge-based Engineering Systems Specification. *Concurrent Engineering*, 16(1), pp.89–101.

Amazon Web Services, Inc. (2019). *AWS Lambda – Serverless Compute - Amazon Web Services*. [online] Available at: https://aws.amazon.com/lambda/.

Anon, (2020). *Top 3D File Formats for 3D Commerce, Social & More | VNTANA*. [online] Available at: https://www.vntana.com/blog/demystifying-3d-file-formats-for-3d-commerce-and-more/ [Accessed 22 Mar. 2022].

Artificial Intelligence and Expert Systems. (2012). *Lees' Loss Prevention in the Process Industries*, pp.2201–2312.

Barreiro, J., Martinez, S., Cuesta, E. and Alvarez, B. (2010). *Conceptual principles and ontology for a KBE implementation in inspection planning. International Journal of Mechatronics and Manufacturing Systems*, 3(5/6), p.451.

Bloomenthal, J. and Chandrajit Bajaj (1997). *Introduction to implicit surfaces*. San Francisco, C.A.: Morgan Kaufmann Publishers.

Bonnardel, N. (2000). Towards understanding and supporting creativity in design: analogies in a constrained cognitive environment. *Knowledge-Based Systems*, 13(7-8), pp.505–513.

Brachman, R.J. and Levesque, H.J. (1985). *Readings in knowledge representation*. Los Altos, California: Morgan Kaufmann Publishers.

Bureau of Indian Standards. (2020). *Home - Bureau of Indian Standards*. [online] Available at: https://www.bis.gov.in/.

businessdocbox.com. (n.d.). *Knowledge Based Engineering across Product Realization. Version 1.0 June, 2011 WHITE PAPER - PDF Free Download*. [online] Available at: https://businessdocbox.com/78628756-Business_Software/Knowledge-based-engineering-across-product-realization-version-1-0-june-2011-white-paper.html [Accessed 14 Oct. 2021].

Case-Based Reasoning Approach. (2008). *Studies in Computational Intelligence*, pp.51–70.

Chacon, S. and Straub, B. (2014). *Pro Git*. Berkeley, Ca: Apress.

Chapman, C.B. and Pinfold, M. (1999). Design engineering—a need to rethink the solution using knowledge based engineering. *Knowledge-Based Systems*, 12(5-6), pp.257–267.

Chapman, C.B. and Pinfold, M. (2001). The application of a knowledge based engineering approach to the rapid design and analysis of an automotive structure. *Advances in Engineering Software*, 32(12), pp.903–912.

Chen, Y., Bharadwaj, A. and Goh, K.-Y. (2017). An Empirical Analysis of Intellectual Property Rights Sharing in Software Development Outsourcing. *MIS Quarterly*, [online] 41(1), pp.131–161. Available at: https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1152&context=icis2009 [Accessed 28 Nov. 2019].

Cheng, X., Fu, S. and de Vreede, G.-J. (2021). Determinants of trust in computer-mediated offshore software-outsourcing collaboration. International Journal of Information Management, 57, p.102301.

Choudhury, V. and Sabherwal, R. (2003). Portfolios of Control in Outsourced Software Development Projects. *Information Systems Research*, 14(3), pp.291–314.

Chau, K.W. and Albermani, F. (2004). Hybrid knowledge representation in a blackboard KBS for liquid retaining structure design. *Engineering Applications of Artificial Intelligence*, 17(1), pp.11–18.

Civilingenjörsprogrammet (n.d.). *Knowledge based engineering for SMEs MASTER'S THESIS*. [online] Available at: https://www.diva-portal.org/smash/get/diva2:1025380/FULLTEXT01.pdf.

Conrad, J., Fox, S., Hofmann, P., Klahn, C. and Meboldt, M. (2021). Enabling non-engineers to use engineering tools: introducing product development to pupils using knowledge-integrating systems. *Procedia CIRP*, 100, pp.265–270.

Corallo, A., Laubacher, R., Margherita, A. and Turrisi, G. (2009). Enhancing product development through knowledge-based engineering (KBE). Journal of Manufacturing Technology Management, 20(8), pp.1070–1083.

Curran, R., Verhagen, W.J.C., van Tooren, M.J.L. and van der Laan, Ton.H. (2010). A multidisciplinary implementation methodology for knowledge based engineering: KNOMAD. Expert Systems with Applications, 37(11), pp.7336–7350.

Das, Dr.S. (2004). *Solid Modeling; Constructive Solid geometry (CSG)*. [online] Available at: http://www.cse.iitm.ac.in/~vplab/courses/CG/PDF/CSG.pdf [Accessed 1 Jan. 2022].

Database support for knowledge-based engineering environments. (1988). Knowledge-Based Systems, 1(4), p.255.

de Vries, M.J. (2009). Translating Customer Requirements into Technical Specifications. *Philosophy of Technology and Engineering Sciences*, pp.489–512.

Enoch, O.F., Shuaib, A.A.B. and Ismail, R. bin (2012). The Application of Computer Aided Design as Tool for Building User-Centered Design in Consumer Ceramics' Product Development. *International Journal of Art, Culture and Design Technologies*, 2(2), pp.30–42.

Estrada, G. and Lloveras, J. (2011). APPLICATION OF MOKA METHODOLOGY TO CAPTURE KNOWLEDGE IN DESIGN FOR POKAYOKE ASSEMBLY. *INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED11*.

Dibbern, Winkler and Heinzl (2008). Explaining Variations in Client Extra Costs between Software Projects Offshored to India. *MIS Quarterly*, 32(2), p.333.

Dibona, C., Ockman, S. and Stone, M. (1999). *Open sources : voices from the open source revolution*. Beijing ; Sebastopol, Ca: O'reilly.

Du, T., Inala, J.P., Pu, Y., Spielberg, A., Schulz, A., Rus, D., Solar-Lezama, A. and Matusik, W. (2018). InverseCSG. *ACM Transactions on Graphics*, 37(6), pp.1–16.

Fayolle, P.-A. and Pasko, A. (2016). An evolutionary approach to the extraction of object construction trees from 3D point clouds. *Computer-Aided Design*, 74, pp.1–17.

Feng Fu, (Structural Engineer (2018). *Design and analysis of tall and complex structures*. Kidlington, Oxford, United Kingdom: Butterworth-Heinemann, An Imprint Of Elsevier.

groups.csail.mit.edu. (n.d.). *Constructive Solid Geometry Methods*. [online] Available at: http://groups.csail.mit.edu/graphics/classes/6.837/F98/talecture/ [Accessed 4 Dec. 2021].

G. Pahl, W. Beitz, and al., Engineering Design: A Systematic Approach, Springer-Verlag, Berlin, 2007

Gaebert, C. and Kautz, K. (2020). The contract-type choice for short-term software development outsourcing: The role of behaviour-based formal control. *Journal of Information Technology*, p.026839622096766.

GeeksforGeeks. (2019). *Object Oriented Programming (OOPs) Concept in Java - GeeksforGeeks*. [online] Available at: https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/.

Ghys, É. (2015). The Butterfly Effect. In: *The Proceedings of the 12th International Congress on Mathematical Education*. [online] pp.19–39. Available at: http://perso.ens-lyon.fr/ghys/articles/butterflyeffect.pdf.

Guo, W., Wen, J., Shao, H. and Wang, L. (2015). Implementation of knowledge-based engineering methodology in hydraulic generator design. *Advances in Mechanical Engineering*, 7(5), p.168781401558423.

Häußler, M. and Borrmann, A. (2021). Knowledge-based engineering in the context of railway design by integrating BIM, BPMN, DMN and the methodology for knowledge-based engineering applications (MOKA). *Journal of Information Technology in Construction*, 26, pp.193–226.

Häußler, M., Esser, S. and Borrmann, A. (2021). Code compliance checking of railway designs by integrating BIM, BPMN and DMN. *Automation in Construction*, 121, p.103427.

Hoffer, M. (2021). JCSG. [online] GitHub. Available at: https://github.com/miho/JCSG [Accessed 29 Nov. 2021].

Jayakiran Reddy, E., Bhanodaya Kiran Babu, N. and Bala Raju, A. (2021). Reverse engineering technique for enhancing the EOT crane rope drum. *IOP Conference Series: Materials Science and Engineering*, 1057(1), p.012029.

Johnson, N.E. (1990). Knowledge based systems series Vol. 1: Knowledge acquisition for knowledge based systems Vol. 2: Knowledge acquisition tools for expert systems. *Knowledge-Based Systems*, 3(3), p.181.

Kendal, S.L. and M Creen (2007). *An introduction to knowledge engineering*. London: Springer.

Kidd, A. (1985). Human Factors Problems in the Design and Use of Expert Systems. *Fundamentals of Human–Computer Interaction*, pp.237–247.

kpkm.polsl.pl. (n.d.). *Knowledge Based Engineering :: Projects :: MOKA KA*. [online] Available at: http://kpkm.polsl.pl/PROJEKTY/KBE/moka_ka/moka_ka_en.html [Accessed 23 Oct. 2021].

Krishnamoorthy, C.S. and S Rajeev (1996). Artificial intelligence and expert systems for engineers. Boca Raton: Crc Press.

La Rocca, G., Krakers, L. and van Tooren, M.J.L. (2002) 'Development of an ICAD generative model for blended wing body aircraft design', *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimisation*, Atlanta, GA, AIAA-2002-5447.

Lee, J.Y., Choi, B.C., Ghauri, P.N. and Park, B.I. (2020). Knowledge centralization and international R&D team performance: Unpacking the moderating roles of team-specific characteristics. *Journal of Business Research*.

Leu, G. and Abbass, H. (2016). A multi-disciplinary review of knowledge acquisition methods: From human to autonomous eliciting agents. *Knowledge-Based Systems*, 105, pp.1–22.

Lovett, P.J., Ingram, A. and Bancroft, C.N. (2000). Knowledge-based engineering for SMEs — a methodology. *Journal of Materials Processing Technology*, 107(1-3), pp.384–389.

Luft, Thomas, Roth, Daniel,Binz, Hansgeorg and Wartzack, Sandro : *A NEW "KNOWLEDGE-BASED ENGINEERING" GUIDELINE* . In: Proceedings of the 21st International Conference on Engineering Design (ICED17), Vol. 6: *Design Information and Knowledge*, Vancouver, Canada, 21.-25.08.2017.

Manoj (2022a). *Drag racing & Supercharger Geometry*. [online] True Geometry's Blog. Available at:

https://blog.truegeometry.com/app/geometry/2022/07/24/RootsTypeSuperchargers.html [Accessed 25 Aug. 2022].

Manoj (2022b). *Robots & Spherical Gear*. [online] True Geometry's Blog. Available at: https://blog.truegeometry.com/app/geometry/2022/07/31/SphericalGearZaxis.html [Accessed 25 Aug. 2022].

Manoj (2022c). *Spur Gear Design in KBE App*. [online] True Geometry's Blog. Available at: https://blog.truegeometry.com/app/geometry/2022/08/21/TechnologyDemoSpurGearKBEApp.html [Accessed 25 Aug. 2022].

Mishra, D. and Ramkumar, M. (2021). Study of project type and team structure in the Indian software outsourcing industry. OPSEARCH.

Montali, J., Overend, M., Pelken, P.M. and Sauchelli, M. (2017). Knowledge-Based Engineering in the design for manufacture of prefabricated façades: current gaps and future trends. *Architectural Engineering and Design Management*, 14(1-2), pp.78–94.

Naya, F., Jorge, J., Conesa, J., Contero, M., María, J. and Gomis (2002). *Direct modeling: from sketches to 3D models*.

Narayan, R. (2014). *Rapid prototyping of biomaterials : principles and applications*. Cambridge: Woodhead Publishing.

Ontotext. (n.d.). *What are Ontologies and What are the Benefits of Using Ontologies*. [online] Available at: https://www.ontotext.com/knowledgehub/fundamentals/what-are-ontologies/.

Oracle.com. (2021). *Download the Latest Java LTS Free*. [online] Available at: https://www.oracle.com/java/technologies/downloads/.

Preston, S., Chapman, C., Pinfold, M. and Smith, G. (2005). Knowledge acquisition for knowledge-based engineering systems. *International Journal of Information Technology and Management*, 4(1), p.1.

Pokojski, J., Szustakiewicz, K., Woźnicki, Ł., Oleksiński, K. and Pruszyński, J. (2021). Industrial application of knowledge-based engineering in commercial CAD / CAE systems. *Journal of Industrial Information Integration*, p.100255.

Pokojski, J., Szustakiewicz, K., Woźnicki, Ł., Oleksiński, K. and Pruszyński, J. (2019). Templates for KBE – Their Real-Life Developments and Applications. Advances in Transdisciplinary Engineering.

Poudelet, V., Chayer, J.-A., Margni, M., Pellerin, R. and Samson, R. (2012). A process-based approach to operationalize life cycle assessment through the development of an eco-design decision-support system. *Journal of Cleaner Production*, 33, pp.192–201.

Quintana-Amate, S., Bermell-Garcia, P., Tiwari, A. and Turner, C.J. (2017). A new knowledge sourcing framework for knowledge-based engineering: An aerospace industry case study. *Computers & Industrial Engineering*, [online] 104, pp.35–50. Available at: https://www.sciencedirect.com/science/article/pii/S0360835216304818.

Quintana-Amate, S., Bermell-Garcia, P. and Tiwari, A. (2015). Transforming expertise into Knowledge-Based Engineering tools: A survey of knowledge sourcing in the context of engineering design. *Knowledge-Based Systems*, 84, pp.89–97.

Rahman, H.U., Raza, M., Afsar, P. and Khan, H.U. (2021). Empirical Investigation of Influencing Factors Regarding Offshore Outsourcing Decision of Application Maintenance. IEEE Access, 9, pp.58589–58608.

Rocca, G.L. (2012). Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design. *Advanced Engineering Informatics*, 26(2), pp.159–179.

Rocca, G.L. and Tooren, M.J.L.V. (2007). Enabling distributed multi-disciplinary design of complex products: a knowledge based engineering approach. *J. of Design Research*, 5(3), p.333.

Rowley, J. (2007). The wisdom hierarchy: representations of the DIKW hierarchy. *Journal of Information Science*, [online] 33(2), pp.163–180. Available at: https://unigis.at/schnuppermodul/modul_gisintro/html/lektion5/media/rowley-2007.pdf.

Sigg, C. (2006). Representation and Rendering of Implicit Surfaces. *ETH Zurich*.

Skarka, W. (2007). Application of MOKA methodology in generative model creation using CATIA. *Engineering Applications of Artificial Intelligence*, 20(5), pp.677–690.

Shang, Y. (2005). Expert Systems. *The Electrical Engineering Handbook*, pp.367–377.

Shaw, K. (1986). The application of expert systems principles to teaching and training. *Technology-Based Training*, pp.123–131.

Stokes, M. (Ed.) (2001) 'MOKA: methodology for knowledge based engineering applications', *Bury St Edmunds, Professional Engineering Publishing, UK*.

Tokhirov, A. (2021). APPLICATION PROCEDURE CAD / CAM / CAE – SYSTEMS IN SCIENTIFIC RESEARCH. *Universum:Technical sciences*, 87(6-5).

Sapuan, S.M. (2001). A knowledge-based system for materials selection in mechanical engineering design. *Materials & Design*, 22(8), pp.687–695.

Savory, S.E. (1988). *Artificial intelligence and expert systems*. Chichester: E. Horwood ; New York.

Steenhuizen, D. and van Tooren, M. (2012). The implementation of a knowledge-based framework for the aerodynamic optimization of a morphing wing device. *Advanced Engineering Informatics*, 26(2), pp.207–218.

Studer, R., Benjamins, V.Richard. and Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, [online] 25(1-2), pp.161–197. Available at: https://www.sciencedirect.com/science/article/pii/S0169023X97000566 [Accessed 1 May 2019].

Tran, T.A. and Lobov, A. (2020). Ontology-based model generation to support customizable KBE frameworks. Procedia Manufacturing, 51, pp.1021–1026.

True Geometry's Blog. (2022). *MOKA informal model (ICARE) form for capturing product knowledge*. [online] Available at: http://blog.truegeometry.com/kbe/moka/icare/2022/01/07/MOKA-ICARE-Form.html [Accessed 9 Jan. 2022].

Tzafestas, S.G. (1993). *Expert systems in engineering applications*. Berlin ; New York: Springer-Verlag.

Verhagen, W.J.C., Bermell-Garcia, P., van Dijk, R.E.C. and Curran, R. (2012). A critical review of Knowledge-Based Engineering: An identification of research challenges. *Advanced Engineering Informatics*, [online] 26(1), pp.5–15. Available at: https://www.sciencedirect.com/science/article/abs/pii/S147403461100036X.

Wan, S., Li, D. and Gao, J. (2016). Exploring the Advantages of Content Management Systems for Managing Engineering Knowledge in Product-service Systems. Procedia CIRP, 56, pp.446–450.

Waldron, M.B. and Waldron, K.J. (1992). A KNOWLEDGE TRANSFER METHODOLOGY IN SELECTION OF MECHANICAL DESIGN COMPONENTS. *Artificial Intelligence in Engineering Design*, pp.57–77.

webprotege.stanford.edu. (n.d.). *WebProtégé*. [online] Available at: https://webprotege.stanford.edu/ [Accessed 23 Oct. 2021].

Wikipedia Contributors (2019). *Butterfly effect*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Butterfly_effect.

Wikipedia. (2021). *Knowledge-based engineering*. [online] Available at: https://en.wikipedia.org/wiki/Knowledge-based_engineering [Accessed 22 Oct. 2021].

Wikipedia. (2021). *Knowledge engineering*. [online] Available at: https://en.wikipedia.org/wiki/Knowledge_engineering [Accessed 22 Oct. 2021].

Wikipedia Contributors (2019). *Knowledge-based systems*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Knowledge-based_systems.

Wikipedia Contributors (2019). *Knowledge representation and reasoning*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Knowledge_representation_and_reasoning.

Wikipedia Contributors (2019). *Knowledge management*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Knowledge_management.

Wikipedia Contributors (2019). *Fractal*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Fractal.

WYVILL, B. and VAN OVERVELD, K. (1996). POLYGONIZATION OF IMPLICIT SURFACES WITH CONSTRUCTIVE SOLID GEOMETRY. *International Journal of Shape Modeling*, 02(04), pp.257–274.

www.json.org. (n.d.). *JSON*. [online] Available at: https://www.json.org/json-en.html.

Xu, Y. and Bernard, A. (2011). Quantifying the value of knowledge within the context of product development. *Knowledge-Based Systems*, 24(1), pp.166–175.

Yadav, A.K., Shukla, R. and Singh, T.R. (2021). Machine learning in expert systems for disease diagnostics in human healthcare. *Machine Learning, Big Data, and IoT for Medical Informatics*, pp.179–200.