7-2008

DESIGN PROCESS MODELING: TOWARDS AN ONTOLOGY OF ENGINEERING DESIGN ACTIVITIES

Pavan Kumar
Clemson University, pkumar@clemson.edu

DESIGN PROCESS MODELING: TOWARDS AN
ONTOLOGY OF ENGINEERING DESIGN ACTIVITIES

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Mechanical Engineering

---

by
Pavan Prasanna Kumar
August 2008

Accepted by:
Dr. Gregory M. Mocko, Committee Chair
Dr. Joshua D. Summers
Dr. William G. Ferrell Jr.

ABSTRACT

An ontology of engineering design activities, called the Design Activity Ontology (DAO), is developed in this research. The DAO models 82 information flows and 25 design activities. These activities cover phases of the design process from conceptual phase through detail design phase. The ontology provides a formalized and structured vocabulary of design activities for consistency and exchange of design process models. The DAO enables design processes to be modeled, analyzed and optimized. The DAO is constructed using information flows identified in current design literature, commonly accepted engineering design textbooks, and an existing activity ontology. Specifically, the DAO is an extension and refinement of the ontology proposed by Sim and Duffy. The DAO addresses several shortcomings of the Sim and Duffy ontology including: (1) lack of computational representation, (2) inability to construct process models from defined design activities, (3) redundant and semantically equivalent information flows, (4) complex information flows, and (5) inconsistent classification. These shortcomings are identified through Design Structure Matrix (DSM) modeling and analysis, and certain protocols for the analysis of the individual information flows. A total of 112 information flows and 26 activities from the Sim and Duffy ontology are reduced to 82 and 25 respectively. The DAO is implemented in the Protégé using the Web Ontology Language (OWL) and Description Logic (DL). The implemented DAO is analyzed using DL's subsumption property through the Fact++ reasoner. Finally, the DAO is exercised through two demonstration examples: (1) the design of a trash truck and (2) the design of an automotive tail light installation fixture. Results from the example support the completeness of the ontology; ability to formulate design processes; and identify "dead-end" information flows, information flows required in design but not generated and critical information flows.

DEDICATION

Dedicated to my loving parents, Radha P. Kumar (Mom) and C.R. Prasanna Kumar (Dad); and my adorable sister, Pooja. I would also like to dedicate this piece of work to my special friends, who have stood beside me, always; and have taken all my tantrums gleefully. Prajwala, AJ, Mike, Batri, Chida, Deepu, Akki, Baldy (Chethan), Boda (Desai), Divya, Aneesh, Beedu, Rakshit Jain, Ravel, Anupama, Arundhati ,Dangani (GP) and for all the members of the Drunkpals Family, this is for you guys! In some way or the other, you folks have been instrumental in transforming my US dream into a reality; I thank you for your unconditional love and support.

I would also like to mention a significant few others, Godha, Govind, Vijay, Balaji, Srikanth, Sripriya (Buddi), and Vikram (Vicky) for providing me with some good career advice that has helped me to hone my professional outlook.

ACKNOWLEDGMENTS

First of all, I would like to thank Dr. Gregory M. Mocko for being the torch bearer and the beacon for my research. He has been a great resource and help, not only as an advisor but as a friend. I would like to thank Dr. Joshua D. Summers and Dr. William G. Ferrell for their support and valuable feedback.

I would like to thank the professors and staff from my undergraduate degree for instilling a strong foundation of professional ethics in me and to have provided quality education in Mechanical Engineering.

I thank my room-mates, Abhijith and Santosh for being there for me when I depended on them and for providing me with all the support I needed to get back on track when I drifted away from my goals. I would like to dedicate a special thanks to Santosh for being my inspiration and benchmark in certain areas, which helped me to improve professionally.

I would also like to thank several others, Lindsay, Vijay, Stuart, Ben, Chiradeep, Siva, Parker, my fellow TA's, Todd, David Moline, and the Staff of ME department for supporting me.

TABLE OF CONTENTS

Page

Table of Contents (Continued)

Page

Table of Contents (Continued)

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

List of Figures (Continued)

Figure Page

CHAPTER ONE:

INTRODUCTION

This research aims at providing the framework and guidelines to model design processes. This is achieved by establishing a computational vocabulary that contains design information as the most atomic part and this atomic entity is assembled to compose activities, the activities grouped to create a phase and finally the phases are connected to generate a process. The ontology is a vocabulary and grammar of engineering design activities which is the focus of this research and henceforth will be referred to as the "Design Activity Ontology (DAO)". The DAO is used to model design processes and these process models are supported by demonstration examples to substantiate the research questions. This research provides insight on the rules and guidelines to develop an ontology (for similar domains) and the means for validating the developed ontology. The following section of the thesis presents the motivation and background of the research.

PROBLEM MOTIVATION

We build, extend and formalize the work of Sim and Duffy [47], where the motivation comes from the lack of their model to capture the main intent of such formalisms which is to enable information exchange through process models. Sim and Duffy, 2003 [47], provide us with a base ontology of design activities which is critically analyzed and modified to be compatible with its integration to existing design support systems. There is a consensus amongst researchers in this domain that there must exist a uniform framework to identify differences and similarities in design, which would otherwise obscure information in this domain. Gero and Kannengiesser, [19] state that "a number of ontologies have been developed to represent objects, specifically artifacts. They form the basis for a common understanding and terminological agreement on all relevant properties of a specific artifact or class of artifacts. Ontologies can then be used to represent the evolving states of designing these artifacts or as knowledge representation schemas for systems that support designing." They also state that design research is a field that has traditionally shown particular interest in explicit representations of processes besides objects. A number of process taxonomies have been created that classify different design methods. However, most of this work has not been based on process ontologies, which makes comparison of the different taxonomies difficult. Furthermore they show that ontologies are richer than taxonomic class hierarchies, as they provide definitions and constraints for an entity's properties and relationships.

Authors Gero and Kannengiesser, explicitly state that some of the efforts towards stronger ontological foundations for process representation have been driven by the need to effectively plan, control, design, and construct processes. A large number of process ontologies and representations have been developed, with varying degrees of domain or task specificity. For example, IDEF0 [24] is a format that specifies how to represent an activity and how to layout graphical activity models into processes. IDEF 0 is a high-level ontology for modeling industry processes at a level of detail, distinguishing between input, control, output, and mechanism. Another, more recent high-level ontology is PERT [35] and [67] which is a process representation primarily used for scheduling tasks in projects.

Ahmed and colleagues [1] attribute the motivation for developing an ontology towards knowledge sharing, and developing a standard engineering language. One item of particular interest is to provide a structured basis for navigating, browsing, and searching information through the hierarchical descriptions of the ontology. They sate that the starting point of their research was to identify what taxonomies should be contained within an ontology for engineering design. Most process ontologies and representations have a view of processes that is based on flows of activities and/or sequences of states. Semantics, capturing the processes' applicability in a purposive context, are generally not included in most process ontologies. Such semantics are needed to guide the generation, analysis, and evaluation of a variety of processes. As research increasingly focuses on automating parts of the selection or synthesis of processes, existing process ontologies provide inadequate representations for computational support[19]. Based on the previous discussion the following research questions and hypotheses are formulated.

RESEARCH QUESTIONS AND HYPOTHESES

Research Question 1

What are the basic set of activities and information entities required to represent the engineering design process?

Hypothesis

Eighty two information flows and twenty six activities (derived from those information flows) can be used to represent a complex design process.

To address the hypothesis the following tasks are completed:

Task 1: Evaluate 3 existing design process formalisms

- Sim and Duffy Ontology [47]
- Gero Ontology [19]
- Ahmed, Kim, and Wallace Ontology [1]

Task 2: Select a formalism as the baseline for future development and modification, using certain well defined analysis techniques.

Task 3: Refine the baseline ontology based on the observations made during *Task 2.*

Research Question 2

How can the information entities and activities that form the ontology be represented in a computer interpretable form?

Hypothesis

Protégé and Description Logic can be used to formally and computationally represent the DAO.

To address the hypothesis the following tasks are completed:

Task 1: Implement the developed ontology, the DAO in Protégé

Task 2: Check for consistency of the ontology in Protégé

Task 3: Use the DL to verify the hierarchy and dependencies in the DAO

Research Question 3

How can the DAO be empirically analyzed?

Hypothesis

The implementation of the DAO in 2 example studies (one from a project funded by an external organization and the other from a ME 402 Senior Design Project) and analysis of this implementation can provide the required results to prove the ontology.

To address the hypothesis the following tasks are completed:

Task 1: Implement the DAO in 2 example studies

Task 2: Analyze the example studies to prove the DAO

THESIS OUTLINE

Chapter 2 introduces the reader to the several shortcomings and drawbacks of existing research but is preceded by the introduction to the major topics of discussion relevant to this research such as, Design Processes, Ontology, Design Data and Information Management and the Design Structure Matrix. Chapter 3 proceeds to introduce the Ontology that was selected as the baseline ontology which will be developed or modified based on certain evaluation and analysis techniques performed in the early stages of this research. Chapter 3 would be the answer to RQ 1 (Research Question 1) and would provide a summary of major observations that would lay the foundation for Chapter 4. Chapter 4 provides the complete details of the modified or refined ontology which we call as the Design Activity Ontology (DAO) and the complete ontology is described in this chapter with its properties relationships, and hierarchy. This DAO is also implemented in a computational background and the details of these implementations are presented in Chapter 4. Chapter 4 would be the answer to RQ 2 and this version of the DAO is used for the demonstration of example studies that would be discussed in detail in Chapter 5. Chapter 4 also answers some the basic questions that were constructed based on the application of the DAO. Chapter 5 also provides the details of the demonstration examples along with some important observations in this phase. Chapter 6 would be the conclusion chapter that would discuss in detail as to, the approach used for this research, research contributions, and some suggestions as to where the research can go from here in the future work section.

CHAPTER TWO:

BACKGROUND – LITERATURE REVIEW

OBJECTIVES:

- Discuss existing and motivational literature.
- Introduce Design Process from information capture perspective.
- Discuss Ontological concepts to capturing information related to design processes.
- Discuss Data and Information Management issues in Design.

Capturing information pertaining to design processes has been a topic of design research and there have been several models and theories developed in this regard. Sim and Duffy [47] state that there have been no consensus and wide spread application of one such theory or model around the world. The authors do not intend to describe design processes with the help of developing an ontology of engineering design activities to capture and manage knowledge related to design. But they provide a rich background to support the development of the ontology and the ontology itself is based on several published literatures that are highly used in the industry. Sim and Duffy have done an excellent job in summarizing the works of some of the masterminds in design theory. The ontology is also based on several other branches of design such as cognitive psychology, artificial intelligence in design, design reality, cognitive theory of designing, knowledge level (KL), etc. For a complete understanding of these concepts and the ontology, the readers are directed to read the paper by Sim and Duffy [47].In this research information and knowledge are taken to be the same. Several researchers have defined the differences between Knowledge, Information, and Data; though there is no commonly accepted definition.

Before proceeding, definitions of, *Design Processes, Ontologies, and Design Data and Information Management* are established.

Product modeling plays a crucial role in product development and process management research, [10]. Choi and colleagues [13] state that the effect of the design phase during new product development is very important because more than two-thirds of all product lifecycle cost is determined during the conceptual design process. Although design accounts for only 5% of total costs under traditional cost accounting methods, it influences 70% of total costs during lifecycle. In other words, the majority of total lifecycle costs are influenced during the crucial design phase. Also they imply that the knowledge about the activities helps designers to learn about the importance of initial design phases, thus influencing the reduction of the product's total lifecycle cost. The important aspects of improving the design support systems or enterprise systems that has received little attention is the, efficient and effective accommodation of the systems like ERPs (Enterprise Resource Planning), SCMs (Supply Chain Management), PDMs (Product Data Management) and their integration to the design support system [46]. The major challenge with this integration is that these systems are continuously improved. Researchers also observed that in order to meet new industrial needs, the solutions make use of web based applications and distributed architectures (e-business platforms) that allow both a great integration capability and adaptability. In particular, the evolution of Product Lifecycle Management solutions (PLMs) should be considered, as they influence the design process. Thus the research of implementing design process modeling as a sub-system to enhance the enterprise business systems and design support systems is executed.

DESIGN PROCESS

Design processes are similar to manufacturing or production processes and thus must be planned, analyzed, and optimized. In typical production processes, a work piece flows from one activity to another while undergoing a form change. For example, raw stock may enter a turning activity in which material is removed. This "in-process" part is then passed along to other activities until the raw material is changed into the desired finished product (see Figure 1.) Each activity in this production process has a purpose for its execution and consumes some resources. In this context a process is a sequence of operations/activities/tasks involving time, space and other resources. A process typically produces an outcome; in this case it is the technical specification of the artifact. The activities that constitute a process cannot be merely aggregated together; rather the information flow associated with each activity must be interconnected into a complex web [42].

Design processes represent a similar transformation, but deal with changes in information about the artifact. The flow of information in design processes is analogous to the flow of raw material in production processes (see Figure 1). Thus, design processes must also be analyzed, planned, and optimized to ensure high quality output, while reducing time, cost, and effort.

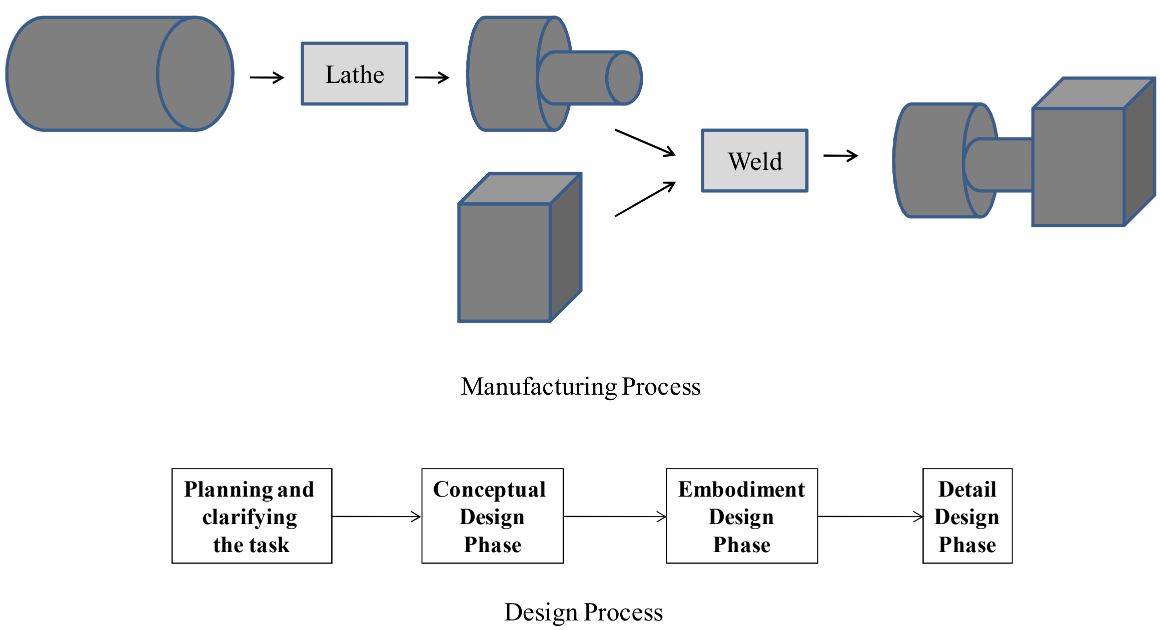


Figure 1: Example of manufacturing and design process

However, there are a few key differences between production processes and design processes that have hindered the ability to develop computational models for the latter. Firstly, in manufacturing or production processes, material is the primary flow between manufacturing activities. The material is passed between events as raw material and "in-process" components. In design processes, the primary flow between activities is information. Secondly, in production processes the discrete manufacturing events are well-understood and can be tied to a specific type of activity with well understood parameters. For example, turning is accomplished by a lathe which can be modeled and simulated using analytical models that describe the turning process. Thus, the behavior of production processes can be modeled, simulated, and optimized. Conversely, the attributes and analytical models that characterize and simulate design activities are neither well understood nor have a common understanding. Finally, the standard vocabulary of production processes is well scoped and repetitive. In other words, complex production processes can be composed of a finite set of production activities. Design processes are often unique and depend on human factors and cognitive psychology where intuition is a major parameter.

A key component of modeling, analyzing, and optimizing processes, whether production or design, is a vocabulary for describing the activities and flows associated with a particular process, [42]. Ontologies are formal representations of a controlled vocabulary and are often expressed in a computer-interpretable representation language. Specifically, an ontology of design activities will enable design processes to be composed from a standardized vocabulary of design activities and to be subsequently analyzed. Researchers have addressed various aspects of process modeling, [30, 42]; in general and design process modeling in particular [39, 40, 47].

ONTOLOGY

Gruber describes ontology as an explicit specification of a shared conceptualization [21], Ahmed and colleagues [1] state that ontologies can be taxonomically or axiomatically based ,

1. Ontologies can be based around a single taxonomy or several taxonomies and the relationships
2. Taxonomies consist of concepts and relationships
3. Taxonomies are organized hierarchically and the concepts can be arranged as classes with subclasses.

In fields such as biology the main effort required to create an ontology is in the population of taxonomies. For example, the taxonomy may be of species and their classification, and effort is focused upon identifying and classifying the species. In engineering design, it is not so clear what taxonomies an engineering design ontology should consist of, i.e., what are the engineering equivalent of species, are these information, activities, functions, behaviors, the physical product, etc?

The authors also describe the process of developing an ontology for engineering design and include the methods employed to identify the taxonomies that form part of the ontology, i.e., identifying the root concepts of the taxonomies and they describe the root concept to be the top-level concept of a taxonomy, for example, species would be the root concept for a taxonomy about species. Furthermore they attribute the motivation for developing an ontology towards knowledge sharing, and developing a standard engineering language. One item of particular interest is to provide a structured basis for navigating, browsing, and searching information through the hierarchical descriptions of the ontology. They state that the starting point of their research was to identify what taxonomies should be contained within an ontology for engineering design. This was addressed in two parts:

1. Identifying the particular application that the ontology is to be used for, and
2. Understanding the root concepts of the taxonomies that can be used to describe engineering design.

They summarize that, ontology for engineering design contains elements which are generic and those that are specific to a project or a product. Taxonomies maybe identified from literature for the root concepts that are generic, and need to be created for those that are specific. A second reason to create a taxonomy is if existing taxonomies are found to be insufficient for the purpose of the ontology. Once the taxonomies have been selected, these are evaluated for the suitability of the ontology. The evaluation needs to be with respect to a criterion for their selection, for example: the completeness of the taxonomy; removal of redundant terms; and mapping of terms on the taxonomy to the conceptual models of the users.

Once all of the taxonomies of the ontology have been identified, the ontology needs to be evaluated for the particular purpose for which the ontology has been developed. There are two parts to this evaluation

1. Check that the root concepts are sufficient for the particular application and
2. Check that the integrated taxonomy, including classes and subclasses, is sufficient for the particular application

This evaluation ensures that the ontology is complete by mapping the classes of the taxonomies to instances. Ontology population means the generation of instances according to ontology definitions. Ontology consists of definitions of concepts, attributes, relations and rules. This research provides the list of activities that can be sequenced to produce the required product in an effective way. Process decisions such as likely or necessary iterations and task dependencies all contribute to the development of the process model. Additional non-engineering factors (related to the process) such as time stamps, organizational factors such as the availability of expertise, systems, resources and tools also contribute to the process model [66].

Ullman [62], states that ontologies are useful in assisting the engineers to plan and schedule their activities for better performance. The ontology is aimed at reducing the time and steps during design. He illustrates that the group considered for his case study first came up with verbal or textual information and then parsed toward physical representation, thus indicating the importance of verbal or textual representation of information. Also the movement was from abstract to concrete or detailed representation. The DAO also directs movement in the same fashion, especially aiding young design engineers by providing the much needed vocabulary and grammar to understand and support the shift. The ontology provides assistance to improve the level of detail in the verbal or textual representation of information. Ullman also quotes that the best level to develop a goal tree is at the activity level. There is also a need for the development of software and technologies that enable engineering designers to communicate and share information between disparate tools and across extended networks. To address this problem, computational ontologies are proposed. They provide an explicit, formal representation of a domain of discourse and establish the semantics and syntax through which intelligent agents can communicate and reason efficiently and effectively [57].

Several ontologies have been developed in the area of engineering design and analysis. For example, ontologies have been developed for the following domains

- *Product functionality,*

  The research in this domain was to improve the accuracy of computer generated design tools, methods of incorporating non-conventional functional representations of artifacts which was proposed to be incorporated and standardized. They argue that without formalized representations of artifact attributes such as manufacturing, feature specific and form specific details; comprehensive design models and tools cannot be generated. They describe a user-need driven approach of addressing shortcomings in product representations by comparing standard hand generated design tools to computer-generated tools [4].

Research was also done to explore the meanings of the terms ‘structure’, ‘behavior’, and, especially, ‘function’. Computers were recognized to assist calculation tasks in engineering practice and for helping with reasoning tasks. There were specific distinctions between function as effect on the environment, and as a device-centered view of device function [12].

There were arguments and discussion on the evolution of design repositories with a backing statement that “All engineering firms maintain archives of previously designed artifacts, often in the form of databases of computer aided design (CAD) data;” and these design repositories being a database to include more heterogeneous information and to provide enhanced capabilities through the application of knowledge representation techniques [27].

Furthermore research was on how product functionality played a role in making a design complex and how it affected collaboration. This approach also emphasized on ensuring comprehensive technical proficiency in a world where trends are toward more multidisciplinary design that can become a costly undertaking for a company. The authors argue that designers are no longer merely exchanging geometric data, but more general knowledge about design and design process, including specifications, design rules, constraints, rationale; etc thus the need for computational design frameworks to support the representation and use of knowledge among distributed designers becomes more critical. This problem was intended to be solved using the Internet and the World Wide Web browsers, along with emphasis to addressing the industry standards [58].

- *Storing geometric CAD models in repositories,*

  As the domain name suggests, this involved a definitive work on the technical and administrative work dedicated to developing a product data exchange standard (commonly known as STEP, Standard for the Exchange of Product model data). This was National Institute of Standards and Technology's (NIST's) effort in product data exchange standardization [33].

  Bohm and Stone also worked on setting up CAD models in Design repositories [4]. The authors along with Szykman published a paper that not only provided a set of heterogeneous product knowledge stored in a coherent design repository but it also supported product design knowledge archival and web-based search, and display. This research was based on design theory where several test products were cataloged to determine what information was essential without being redundant in representation [5].

  The National Design Repository (http://www.designrepository.org, http://repos.mcs.drexel.edu) is a digital library of Computer-Aided Design (CAD) models and engineering designs from a variety of domains. This project was started in 1994 and its objective was to further the state-of-the-art in academic and industrial research in Computer-Aided Engineering. The Design Repository contained over 55,000 files maintained in multiple data file formats (including STEP AP 203, ACIS .sat, DXF, IGES, DGN, Parasolid, .xmt) and growing by 20% every year [45].

- *Engineering analysis models,*

  Due to the lack of existing technology to offer effective solutions for the management and integration of information, there has been an interest to develop abstract computational analysis tools that have become essential to evaluate designs for complex engineering products. Engineers are using more analysis applications to model a wider range of product behavior and they emphasize on the knowledge of how existing analysis applications use and generate information and what their common elements are in order to facilitate the construction of more automated analysis systems [7].

Foundations for exchanging, adapting, and interoperating engineering analysis models is based upon the concept that engineering analysis models are knowledge-based abstractions of physical systems, and therefore knowledge sharing is the key to exchange, adaptation, and interoperability [20].

A paper on the importance of computer simulations and behavioral modeling in product development processes is published. The authors also show how simulations can result in better decisions in less time by providing the designers with greater understanding of the product's behavior. They develop a clean interface that reduces the knowledge gap between engineering design and analysis by facilitating reuse of behavioral models through simple examples [31].

- *Engineering decision problems,*

Kamal, Karandikar, Mistree, and Muster discussed the importance of knowledge representation in decision making and how it affected different disciplines in an complex system design or expert system design [26].

The importance of Product Lifecycle Management (PLM) and its prospects in product design, emphasizing integration, interoperability, and sustainability were also discussed by another research team. The concept of Design Process Lifecycle Management (DPLM) is also introduced with existing state-of-the-art and future recommendations are suggested that have the key elements for enabling the integrated design of products and their underlying design processes in a systematic manner. The motivating factors in the extension of PLM to include the lifecycle considerations of design processes are also presented [40].

An approach for the integrated design of materials, products, and design processes was developed which was based on the use of reusable interaction patterns to model design processes, and the consideration of design process decisions using the value of information metrics. This approach used a multifunctional energetic structural materials (MESM) design example. It was shown that the integrated design of materials and products can be carried out more efficiently by considering the design of design processes [39].

An optimization ontology approach was suggested by another group of researchers where its implementation into a prototype computational knowledge-based tool named ONTOP (ontology for optimization) was created. The tool's salient features have been discussed in this paper along with some case studies [68].

- *Manufacturing services and processes,*

In this research, a multi-agent system (MAS) is developed for enabling intelligent formation of distributed supply chains. The system was proposed to have three major components: 1) An ontology for formal representation of manufacturing services 2) A matchmaking engine for finding matches between suppliers and customers 3) A multi-agent based architecture for system-level operation. The Semantic Web was used as the modeling paradigm and they also used mathematical formalism and fuzzy rationale to calculate semantic proximity of supply and demand entities [2].

Manufacturing Systems Integration Division (MSID) define a neutral representation of product data, most recently realized through the STEP standard with focus on the representation of manufacturing process. They say that "Like product data, process data is also used throughout the life cycle of a product, from early indications of manufacturing process flagged during design, through process planning, validation, production scheduling and control." Thus the Process Specification Language (PSL) defines a neutral representation for manufacturing processes that supports automated reasoning [34].

- *Engineering requirements,*

  In the domain of Engineering Requirements, ontology for representing requirements with support for a generic requirements management process in engineering design domain is developed. The proposed ontology is a part of a more general ontology to capture engineering design knowledge. Objects in the ontology include, parts, features, requirements, and constraints. They have used first-order logic to define the objects and their attributes, and identify the axioms capturing the constraints and relationships among the objects [29]

- *And Mechanical devices,*

  Ontology for the mechanical engineering devices to support a wide range of tasks including analysis and design is developed. The behavior of a mechanical device from a description of its geometry was the primary task. The authors looked for common patterns of behavior and labeled them with the terms that mechanical engineers use to talk about mechanical devices [49].

In addition, researchers have developed vocabularies of generic building blocks for composing business process models and organizational processes, [30, 42]. While these efforts focus on computational representations, they do not specifically address activities in engineering design with design information at its core. Additional work is required for developing specialized design process models. However, the primary focus of current ontology development efforts in engineering is capturing information that describes the artifact and the design process through which the artifact is developed.

Engineering design ontology must enable capture and query of engineering design information and must have the potential for improving the design process and the reuse of captured information. Ullman, emphasizes that the tie between product and process is a major part of concurrent engineering and in the late 1990's this concern became prominent with the development of interest in integrated product and process development (IPPD), the successor to concurrent engineering. He states that project planning and change management has always been a large part of engineering management. Product data management (PDM) systems have made large strides toward integrating the actual design work with what was planned and these systems are still maturing. He proposes an ideal mechanical engineering design support system based on a list of activities and this research enhances the list of activities to be used in such a design support system [60-62]. Ullman also recognizes several key characteristics for a design support system to be successful,

- Support the relationship between the requirements and the development of the product
- Support the development, following, and updating of plans
- Support information about problems or issues addressed (e.g., business issues, planning issues, and artifact design issues)
- Support information about arguments for or against alternatives (e.g., qualitative discussion, quantitative analysis rules, and standards) based on requirements
- Support information about the decisions reached
- Add no cognitive burden while supporting information development
- Capture all types of information with a single entry

- Archive all the types of information so that design intent can be readily recovered
- Support designer query about the design intent for all types of information
- Communicate information in the format, level of abstraction, and level of detail needed
- Guide the designer about what to do next

DESIGN DATA AND INFORMATION MANAGEMENT

Baxter and colleagues [3], categorize existing work in which the design process has a relationship to design information management or design reuse mainly into

- Design process with the information management at its core
- Integrating design rationale process
- Design methodology as design process description or management method
- Design information capture and representation through design processes

The relationship between the design process and the design object is not well understood. Integrating rationale with the design process has relatively little work. Design process models as an integrated part of information management requires further analysis to identify the limits and nature of applicability determined by the type of design process. The authors also comprehensively analyze the CAD / CAE based design reuse methods which include component reuse, parametric design (both generative and variant) and KB systems. They state that most of the Computer Aided Engineering (CAE) systems (such as Unigraphics, Catia, Pro-Engineer and ICAD) provide parameter-driven information modeling capabilities which are normally based on a geometric model. These systems have design rules embedded in the parameters, and are used for very specific engineering calculations. They are very well suited to solving complex, highly structured problems in which a level of optimization is required [3]. Signposting and Design Roadmap (DR) are the tools that are currently available [14] but the authors state that Signposting is a parameter-driven task-based model of the design process and it can be observed that the task model does not have strong precedence links; instead the method uses the level of confidence in key design and performance parameters as the basis for identifying, or signposting, the next design task. The Signposting method is well suited to the development of new technologies in well understood application areas [3, 14]. A formal method to represent the design process is the Design Roadmap (DR) method. This method enables the representation of feedback and feed forward processes, which are common in design yet uncommon in other representations. The DR model enables a variety of graphical representations, or views. Graph, matrix, tree and list views are supported. Additional functions, including resource management, document attachment and notification functions were added to the DR framework [41]. The method mainly addresses project management issues, which implicitly applies product information, and is similar to the DAO but lacking the support on the process side. Thus existing methods to reuse design information are generally not compatible with the whole product design process: some are suitable in conceptual design; most are focused on detail design. Further research is needed to explore the potential of an integrated process and product modeling approach. This should include non geometric information such as problem solving methods, solution generation strategies, design intent and project information. These information types are associated with the variety of tasks in today's dynamic design process [3]. The DAO complements the existing approaches by linking product data to the non geometrical information through the process model, although the CAD based methods are expected to remain highly valuable in supporting detailed design, the other elements are aimed at supporting early stages of product development.

DESIGN STRUCTURE MATRIX

A tool that was extensively used in this research is the Design Structure Matrix (DSM), which is a matrix-based tool for modeling and analyzing complex engineering systems [17, 25, 52, 53]. DSM is a system analysis tool or a project management tool designed for

- Compact and clear representation of a complex system
- Capturing the interactions/interdependencies/interfaces between system elements (i.e. sub-systems and modules).
- Project representation that allows for feedback and cyclic activity dependencies. (Most engineering applications exhibit cyclic property)
- A more realistic execution schedule

DSM's can be classified into the following 4 types:

Component-based DSM: Documents interactions between elements in a complex system architecture. Different types of interactions can be displayed, i.e., Energy or Information or Material (The EMS system from Pahl and Beitz). (Types of interactions will vary from project to project)

Team-based DSM: Used for organizational analysis and design based on information flow among various organizational entities. Individuals and groups participating in a project are the elements being analyzed (rows and columns in the matrix). A Team-based DSM is constructed by identifying the required communication flows and representing them as connections between organizational entities. For the modeling, it is important to specify what is meant by information flow among teams. The information flow can be (a) Level of Detail (b) Direction (c) Frequency or (d) Timing.

Activity-based DSM: Mainly three types of task interactions can be observed "Independent" – no information is exchanged between the activities and these tasks can be executed simultaneously (in parallel). “Dependent” – sequential information transfer and these tasks would typically be performed in series. “Interdependent or coupled” – mutually dependent information and these are activities often requiring multiple iterations for completion.

Parameter-based DSM: Analyzes system architecture based on parameter interrelationships. Constructed through explicit definition of a system’s decomposed elements and their interactions. A systematic taxonomy and a quantification scheme assist in the analysis by categorizing types of interactions among system elements such as Energy, Material and Signal (EMS) and associating an appropriate weight to each.

Several algorithms have been developed for analyzing and reorganizing the information entities captured in a DSM including partitioning and clustering that was used in this research. *Partitioning* reorders the individual information elements in a DSM to minimize feedback from downstream information elements. Manipulation (i.e. reordering) of the DSM rows and columns such that the new DSM arrangement does not contain any feedback marks (Transforming the DSM into a lower triangular form). For complex engineering systems, it is highly unlikely that simple row and column manipulation will result in a lower triangular form. Therefore, the analyst’s objective changes from eliminating the feedback marks to moving them as close as possible to the diagonal (this form of the matrix is known as block triangular). Results (1) Fewer system elements will be involved in the iteration cycle; and (2) Faster development process. *Clustering* is used to identify strongly related information elements. These information elements can be grouped into modules. Clustering as we have learned in the partitioning section, the goal of partitioning was to render the DSM lower triangular as much as possible. The reason was due to the significance of upper-diagonal marks, which represented feedback information flows. The new goal for the entities that are mutually exclusive or minimally interacting subsets is finding subsets of DSM elements (i.e. clusters or modules). In other words, clusters absorb most, if not all, of the interactions (i.e. DSM marks) internally and the interactions or links between separate clusters is eliminated or at least minimized.

The DSM enables inter-relationships between information elements within a domain to be modeled and analyzed through the matrix-based representation and analysis algorithms. Figure 2 illustrates a typical DSM. The A, B, C, D etc. represent the activities and the green "X" mark (the "X" marks below the diagonal of the matrix) represents a feed forward relationship and the red "X' mark (the "X" marks above the diagonal) the feedback. For a detailed description about DSM, refer Steward [52]. In this research, the DSM is used to capture the model and analyze the ontology of design activities. The following chapter introduces the four different analysis cases for developing a DSM-based representation. The information flows modeled and analyzed using the four different cases are based on definitions established in [8, 47]. A demonstration of the DSM populated by example activities is summarized in Figure 2.

ACTIVITIES		A	B	C	D	E	F	G	H	I	J	K
Receive specification	A	A										
Generate/Select concept	B	X	B									
Design beta cartridges	C	X	X	C								
Produce beta cartridges	D			X	D							
Develop testing program	E	X	X	X		E						
Test beta cartridges	F			X	X	X	F					
Design production cartridge	G	X	X	X			X	G	X	X		
Design Mold	H	X	X				X	X	H	X		
Design assembly tooling	I							X	X	I		
Purchase MFG equipment	J					X		X		X	J	
Fabricate molds	K								X			K

Figure 2: A typical Design Structure Matrix

CHAPTER SUMMARY

The opportunities identified through the discussion in this chapter are in the domain of design data and information management, ontology and design processes. As several researchers concur, there has been a void between realizing design theory to practical design applications and the above sections discussed the issues or roadblocks to construct an ideal design support system. The significant shortcomings identified in the literature review are:

- A survey of published literature about capturing the information related to "Engineering Design Processes" and information about "Design process models" must be conducted to select and refine a formalism to capture the required information pertaining to design.
- An activity model of a typical engineering design activity must be developed based on the certain standards recognized in this research, the standards that education and industry conforms to. Standards defined by IDEF 0 [24] and Browning [10] are selected to perform this task. This activity model represents the typical engineering design activity with the flow of information within that activity. Also additional attributes must be developed to improve the activity's definition.
- Additional applications of the DSM, based on capturing and modeling information must be developed. DSM's must be used more often as they offer analysis and evaluation tools; along with its visualization capability of complex processes, which are not adequately used in the information modeling domain.
- An ontology that is capable of capturing the information related to engineering design processes that can aid the development of an intelligent design support system must be developed. This ontology must be completely described for its attributes, hierarchy, taxonomy and relationships. This ontology must also be explained from a design process point of view with ample examples (from standard projects) for each of the entity

associated with the ontology. This ontology could be made available to students learning design and they can be asked to use it during their design coaching.

- The ontology must also be implemented in a computational background to provide quick access and quick-start for any future work proposed for this research. The computational background must also provide visualization options as design processes are complex in nature. This ontology must also be compatible with web based applications to enable information exchange and interoperability.

CHAPTER THREE:

MODELING AND ANALYSIS OF BASELINE ONTOLOGY

OBJECTIVES:

- Discuss the ontology development lifecycle
- Evaluate the selected ontology for shortcomings and future development
- Model and Analyze the ontology using the Design Structure Matrix (DSM)
- Discuss suggestions for improvement of the analyzed ontology

As discussed in Chapter 2, an ontology is an explicit specification of a shared conceptualization. The proposed Design Activity Ontology (DAO) is an explicit representation of design information derived basically from existing work on design activity ontology by Sim and Duffy [47]. These researchers have developed this ontology based on existing literature and case studies; basically from commonly accepted systematic design methods discussed in publications and textbooks such as Engineering design, [23, 38]; Product design, [44, 54, 63]; Mechanical design, [60]. Its concepts and descriptions have also been taken from conference and journal papers provided a repository of information relating to design research; Protocol analysis of design experiments in different domains such as architectural design, [11]; mechanical design,[50, 51, 59, 62, 65]; and case studies of large complex electromechanical artifacts [15]. The process of developing the ontology can be explained using the following "ontology development life-cycle" presented in Figure 3.

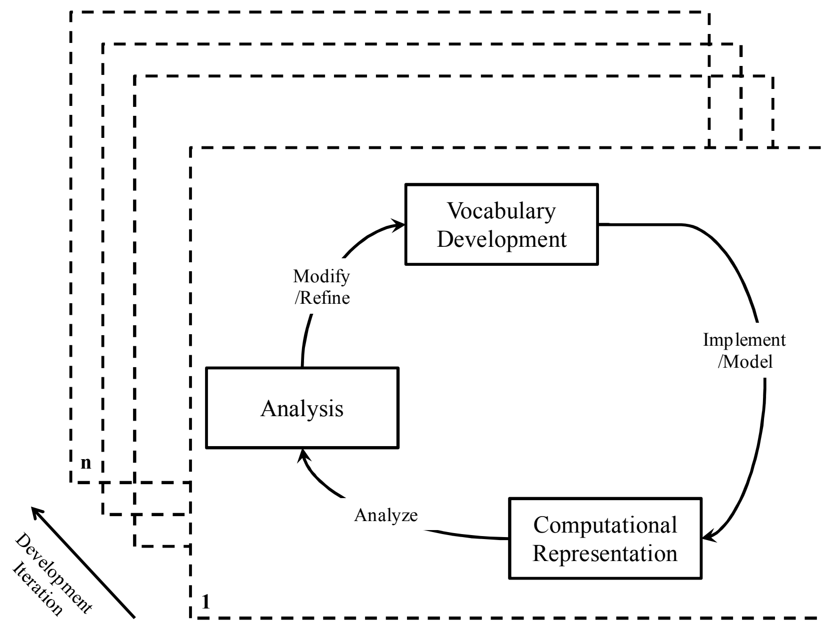


Figure 3: Ontology development life-cycle and scope of research [28]

The phases associated with developing a computer-interpretable ontology are: *Vocabulary Development*, *Computation Representation*, and *Analysis & Validation* [25, 43]. First, the key concepts that exist in a domain and the relationships between them are generated during *Vocabulary Development.* In this research, the Sim/Duffy Ontology is the starting point for vocabulary development phase. Next, computational representation is completed by "implementing" the vocabulary in a computer-interpretable representation language. Initial computational representation is completed using the Design Structure Matrix (DSM). The activities and information flows identified in the vocabulary development phase are modeled using the DSM. A similar method of generation of ontologies attempted here is discussed by Ahmed and colleagues [1], Browning [9], Lin [29], and Pinto [43], thus the lifecycle concurs with established approaches. Finally, analysis & validation of the ontology is completed using several DSM-based analysis algorithms such as partitioning, tearing, and clustering. The results obtained from the analysis are used to refine the vocabulary. As illustrated in Figure 3, the ontology development process is repeated until a stable version results. The following section addresses the first iteration.

ITERATION ONE OF THE ONTOLOGY DEVELOPMENT LIFECYCLE

The specific highlights of this iteration would be:

1. Selection of one ontology for future development by evaluating existing ontologies on design processes
   i. The ontologies proposed by Sim and Duffy [47], Ahmed and colleagues [1] and Gero and Kannengiesser [19] were analyzed
   ii. The ontology proposed by Sim and Duffy [47] was selected for further development
2. Model the information entities defined by Sim and Duffy [47] in the DSM for analysis
3. Refine the ontology based on the observations made during this iteration
   i. Refinement of 26 activities
   ii. Refinement of the 112 information entities

The vocabulary of design activities is modeled and analyzed using the design DSM. Iteration 1 is explicitly illustrated by Figure 4, and it can be seen that Step 1 is the Vocabulary development phase and is completed by the selection and evaluation of an existing ontology by Sim and Duffy; Step 2 is the implementation of this ontology on a computational background like the Design Structure Matrix (DSM); and finally Step 3, the DSM based analysis is conducted before moving to the second iteration.

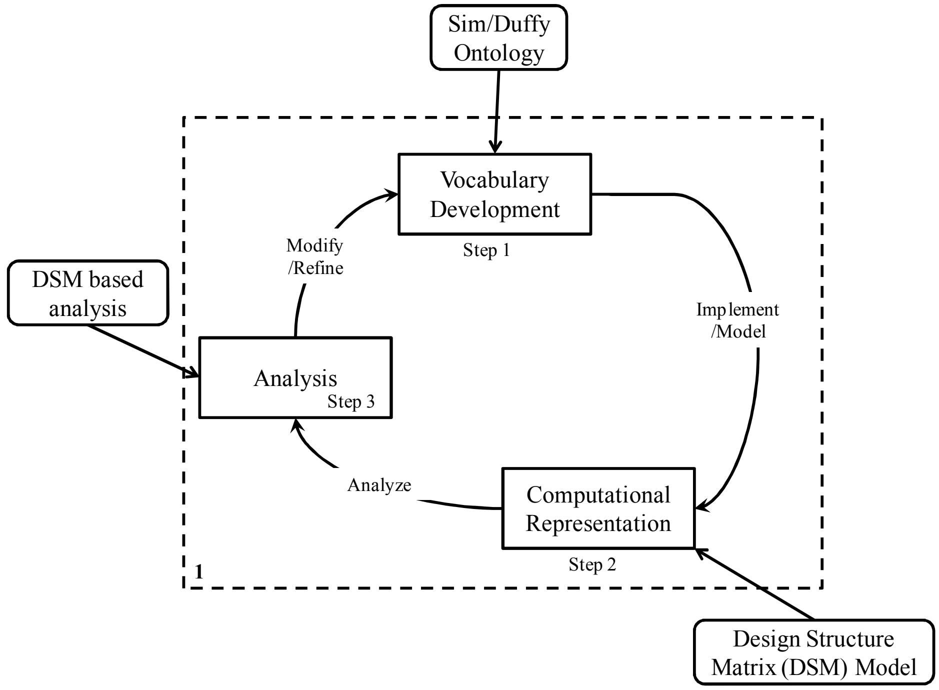


Figure 4: First Iteration of the Ontology Development Lifecycle

Step 1: Vocabulary development phase

The ontology developed by Sim and Duffy is based on several commonly accepted design methods including Hubka [23], Pahl & Beitz [38], Pugh[44], Suh [54], Ullman [60], and Ulrich & Eppinger [64]. The authors argue that no shared understanding of design activities exist and the development of a standardized set of design activities will provide a consistent understanding of design processes and contribute to the development of standard design information representation for process information reuse. Additionally, the authors have suggested that the ontology can serve as the basis for developing design support tools, but have not sufficiently supported this claim. Foundational to the development of the ontology is the basic understanding of how the authors define a design activity (see Figure 5).

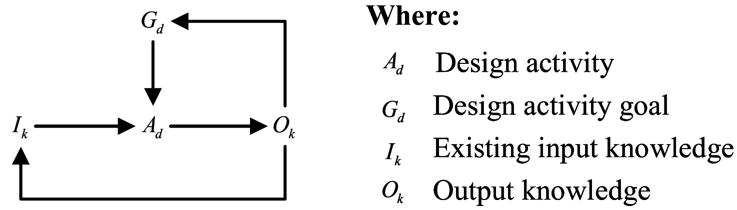


Figure 5: Graphical representation of activity and the information flow in an activity [47]

As illustrated in Figure 5, the activity denoted by A_d consists of a goal G_d which is unique to a particular activity and which determines the fate of an activity. This activity is provided with certain information entities as input denoted as I_k; and after passing through the activity block A_d, these information inputs are transformed to generate a set of output information denoted as O_k. This model also demonstrates a feedback loop where an activity can have a feedback during its execution where the output from an activity has an impact on the input of the same activity. Additionally, the input and output information may be connected to other design activities that lie downstream or upstream from the target activity. Twenty-six design activities are identified to form the vocabulary for describing design processes. For a detailed discussion and explanation of the design activities and information flow between activities, the readers are referred to [47]. The design activities are grouped into three different classifications by Sim and Duffy which are illustrated graphically in Figure 6 through Figure 8.

Design Definition Activities (DDA): Design Definition Activities manage the complexity of design while increasingly defining it. This type of design activity can represent the conceptual Design phase. DDA completes the task of reducing an ill constructed problem into a well structured one and provides some basic design solutions from abstract concepts to concrete ones as the design progresses. We get the information of functions and structures from these activities and we will be able to develop the relationship between them. Synthesizing is the major activity in this classification (see Figure 6).

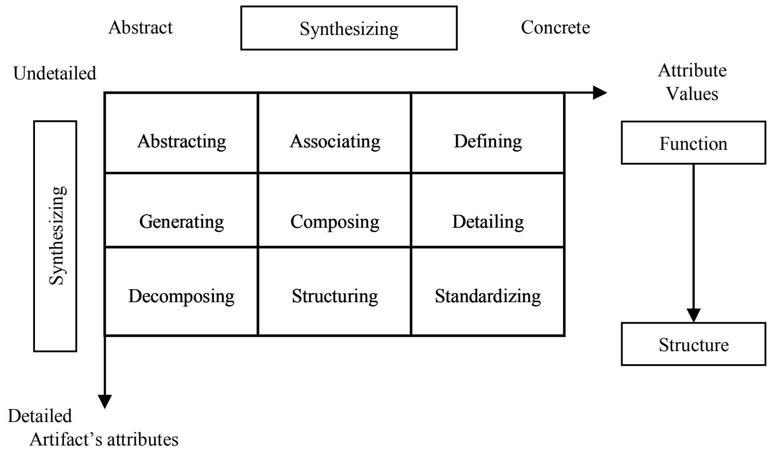


Figure 6: Design definition activities [47]

Design Evaluation Activities (DEA): Design Evaluation Activities analyze and evaluate the developed designs or available designs and to reduce the design solution space. Similar to embodiment design phase and partially similar to the detailed design phase. DEA provides us the information of Functions, Structures and Behavior. And as the process progresses we can focus on a particular design solution from the available set of design solutions by the activities of analyzing, testing and evaluation (see Figure 7).

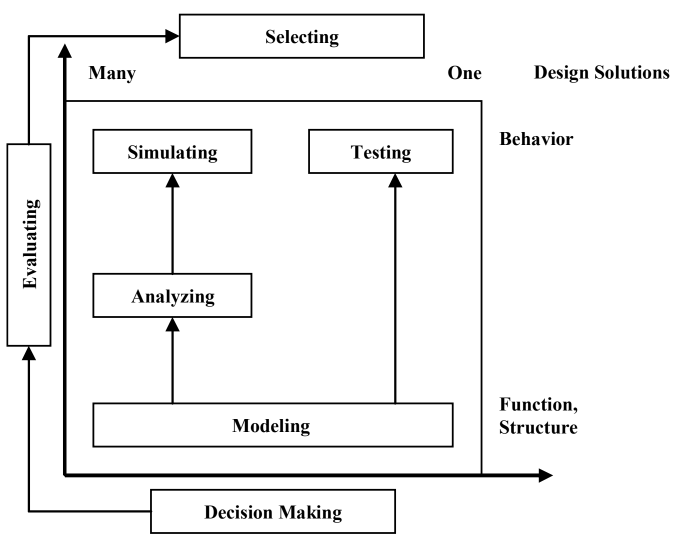


Figure 7: Design evaluation activities [47]

Design Management Activities (DMA): Design Management Activities manage the complexity of coordinating activities related to an evolving design. This classification can be referred when the design has reached embodiment design phase. DMA can be stated as a comprehensive process which provides a set of generic activities which develops the information about the design problem and the design solution, i.e. it changes an ill-structured design problem into a well-defined design problem. It converges to a few or one suitable design solution from the range of design solutions available. It also incorporates design process management activities to schedule and plan the activities accordingly (see Figure 8).

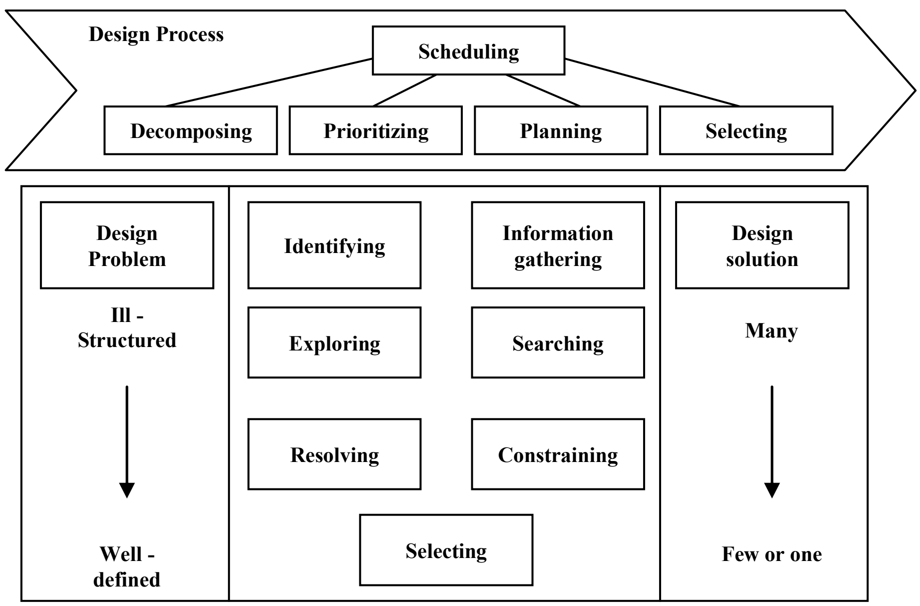


Figure 8: Design management activities [47]

Step 2: Computational representation phase

Several representations of the activities were considered from existing research and were evaluated using empirical studies. The observations from applying these activities to the ontology and analyzing them in the Design Structure Matrix (DSM) for consistency and completeness were used for engineering discussion and to select the most appropriate activity model. The goal of this task is to get a model representation of an engineering design activity that can represent a design activity in reality [28]. The process of developing a DSM-based representation requires

- Understanding of how activities are modeled in general
- Identifying the design activities and their hierarchy
- Identifying the input and output information flows associated with each activity
- Information of analysis algorithms or techniques

Based on the activity definition by Browning [8]; the DSM could be used to model the information illustrated in the Sim and Duffy ontology. The two types of DSM classification that was extensively used were the "Activity Based DSM" and the "Parameter Based DSM." The activities and the activity model specified by Sim and Duffy could be directly translated into the DSM with the help of the basic representation of relationships specified by Browning. And the Information flows themselves could be modeled into the DSM by considering them as parameter in the parameter based DSM. Here the information entities were related to activities and a final relationship matrix could have been created for all the information flows based the classification given by Sim and Duffy, i.e., the DDA, DEA, and DMA. Furthermore 3 activity models were created and tested apart from the activity model from Sim and Duffy. The DSM representations of the four activity models are illustrated. The four different models identified were,

Case 1: Partially Connected with Feedback: A particular input generates a particular output. In addition, feedback exists between input and output information, within an activity (see Figure 9 and Figure 10). It is illustrated in the figure that an input is connected to a particular output by the dotted line and the output is connected back to the input illustrating the feedback. Also the DSM representation of this model that was used for analysis is presented. The "1" indicates the presence of a relationship and the blank cells denotes that there is no relationship between those elements. It can also be seen that the activity is governed by a Goal, which is indicated by Goal acting as an input to the activity but since every activity has only one goal (which is related to the entire activity), it is not considered to be modeled in the DSM. The feedback can be identified in the DSM by the "1's" present in the upper triangular part of the matrix divided by the diagonal (the imaginary line that is created diagonally by the darkened/black colored cells; which just indicate that a relationship need not be expressed between the element itself, as it is well known that every element is related or a subset to itself.) The DSM of this case predominantly exhibit,

- Symmetry across the diagonal
- Sparsely populated near the diagonal
- Only One-to-One relationships exists

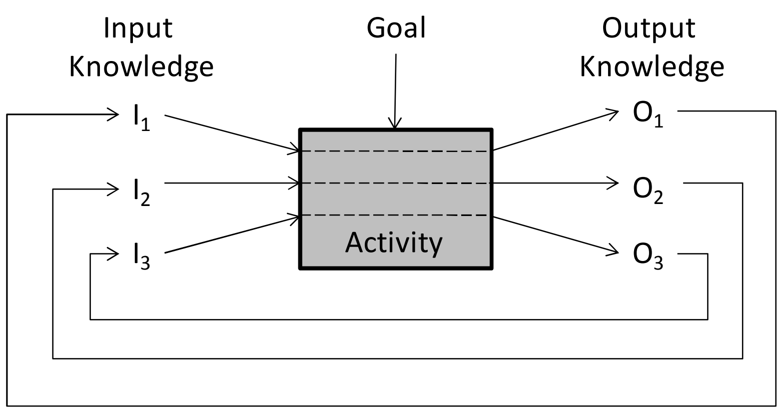


Figure 9: Case 1, design activity and information model [28]

Case 1						
Task/ Info	I_1	I_2	I_3	O_1	O_2	O_3
I_1				1		
I_2					1	
I_3						1
O_1	1					
O_2		1				
O_3			1			

Figure 10: Case 1, Activity model represented in DSM

The DSM created for analysis for this case and its analyzed matrix is presented in Appendix A. Since the matrix is a 112x112 matrix, which is a very large matrix to be presented in the given format of the document, the elements and the components of the matrix are weakly presented.

Case 2: Completely Connected with Feedback: The entire set of input information/s is used to generate the set of output information/s. (i.e., all the inputs are connected to all the outputs). Specific relationships between a particular input and a particular output are tough to capture. Feedback between this entire set of output information/s is given to the set of input information/s. The DSM representation of this activity is presented and the interactions can be seen as a block of "1's" above and below the diagonal (see Figure 11 and Figure 12.) Similarly the DSM created for analysis for this case and its analyzed matrix is presented in Appendix A. The DSM of this case predominantly exhibit,

- Symmetry across the diagonal
- Densely populated near the diagonal
- All kinds of relationships exist (One-to-One, One-to-Many, Many-to-One and Many-to-Many)

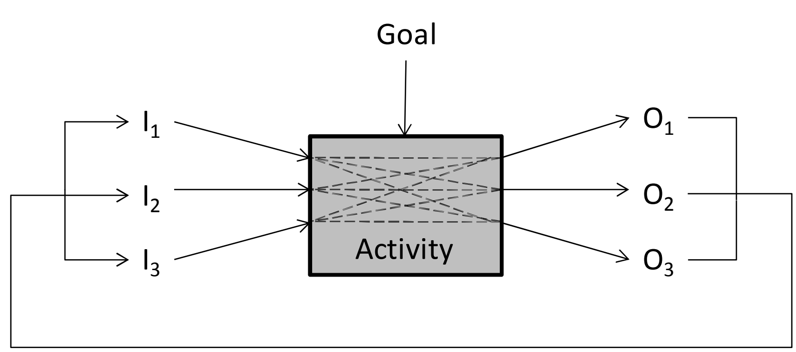
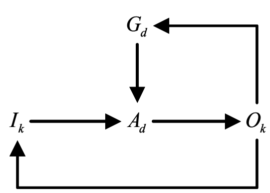


Figure 11: Case 2, design activity and information model [28]

Case 2						
Task/ Info	I_1	I_2	I_3	O_1	O_2	O_3
I_1				1	1	1
I_2				1	1	1
I_3				1	1	1
O_1	1	1	1			
O_2	1	1	1			
O_3	1	1	1			

Figure 12: Case 2, Activity model represented in DSM

Also, when the ontology was studied it was observed that, multiple inputs were being used in the activity to generate a single output, thus we assume that this case represents the generic activity model developed by Sim and Duffy. Activity model defined by Sim and Duffy is represented in Figure 13, which is analogous to Case 2. This is a more realistic representation of a design activity but it has a feedback with in an activity which prohibits its selection for our research.



Where:

A_d Design activity
G_d Design activity goal
I_k Existing input knowledge
O_k Output knowledge

Figure 13: Graphical representation of activity and the information flow in an activity (Same as Figure 5)

Case 3: Partially Connected with No Feedback: The relationship between a single input and a single output is captured explicitly i.e., when particular input information enters an activity, it transforms into a particular output information, that was solely generated from that input information. This case is similar to Case 1 but the only difference between Case 1 and Case 3 is that, there exists no feedback loop between the output and input information. The DSM representation of this activity model is also presented (see Figure 14 and Figure 15.) Similarly the

DSM created for analysis for this case and its analyzed matrix is presented in Appendix A. The DSM of this case predominantly exhibit,

- No symmetry across the diagonal
- Sparsely populated near the diagonal
- Only One-to-One relationships exist

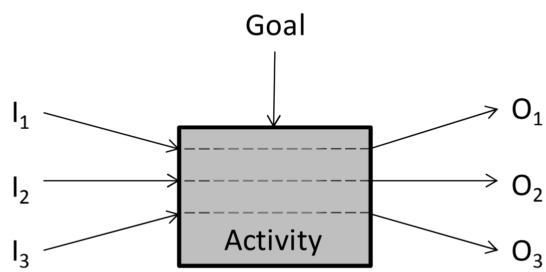


Figure 14: Case 3, design activity and information model [28]

Case 3						
Task/ Info	I_1	I_2	I_3	O_1	O_2	O_3
I_1	■					
I_2		■				
I_3			■			
O_1	1			■		
O_2		1			■	
O_3			1			■

Figure 15: Case 3, Activity model represented in DSM

Case 4: Completely Connected with No Feedback: This case is similar to Case 2, where all the inputs were connected to all the outputs but with a major difference that, there can exist no feedback between the set of output and input information/s within an activity. The DSM representation of this activity model is also presented (see Figure 16 and Figure 17.) Similarly the DSM created for analysis for this case and its analyzed matrix is presented in Appendix A. The DSM of this case predominantly exhibit,

- No symmetry across the diagonal
- Dense population near the diagonal
- All kinds of relationships exist (One-to-One, One-to-Many, Many-to-One and Many-to-Many)

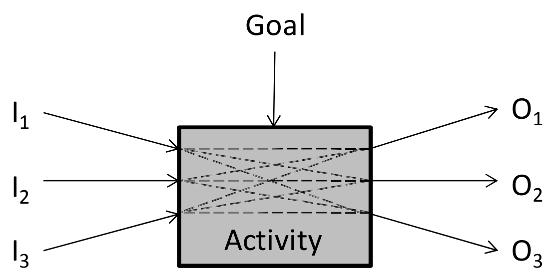


Figure 16: Case 4, design activity and information model [28]

Case 4						
Task/ Info	I_1	I_2	I_3	O_1	O_2	O_3
I_1	■					
I_2		■				
I_3			■			
O_1	1	1	1	■		
O_2	1	1	1		■	
O_3	1	1	1			■

Figure 17: Case 4, Activity model represented in DSM

Note: It has to be observed here that this activity model does not imply that feedback in a process does not occur; it simply means that a feedback within an activity cannot occur.

Step 3: Analysis phase

The design activities proposed by Sim and Duffy are analyzed using four different representations of input and output information flow. These analysis cases are based on the definitions and models of information flow proposed by Browning and colleagues [8]. This DSM is populated with 1's and "empty cells" that represent the existence or nonexistence of a relationship respectively. Cells below the diagonal represent feed-forward information flows and cells above the diagonal represent feed-back flows. DSM-based models of the Sim/Duffy ontology are developed using a Microsoft Excel-based tool[1]. The four cases presented here are proposed to validate the Sim/Duffy ontology.

DSM-Based Analysis: Sim and Duffy ontology is analyzed through DSM analyses algorithms. The activities and related information flows are analyzed using the afore-mentioned DSM algorithms to identify:

- Decomposition of design activities into sub-activities based on independent information flows
- Decomposition of design activities into sub-activities based on feedback information
- Inconsistencies between information flows across design activities
- Grouping of activities into super-activities
- Insufficient information flow and design activities for composing design processes

The specific analysis of the four cases and observations are discussed in greater detail in the following sections. The results are discussed in accordance with the classification proposed by Sim and Duffy.

Case 1: Partial Information Flow & Feedback: The Synthesizing activity is dependent on the information from Generating, Detailing and Standardizing. This supports the original classification. However, the Detailing activity was decomposed into two sub-activities with independent information flows. Additionally, Sim and Duffy assert that Synthesizing is a compound activity related to nine other activities (see Figure 6). However, the Synthesizing activity was found with strong relationships to three activities (Generating, Detailing, and

[1] DSM program – an excel based tool developed by DSMWEB.ORG

http://www.dsmweb.org/index.php?option=com_content&task=view&id=50&Itemid=38

Standardizing) but not to other similarly classified activities. The Decision-making activity is decomposed into two sub-activities with independent information flows. The Identifying activity was decomposed into three parts.

Case 2: Complete Information Flow & Feedback: The partitioning and tearing activities had no effect because of the strong feed-forward and feed-back relationships. Several groupings of design activities were identified in accordance with the established classifications. However, the relationships assumed in this case seem unrealistic because all the information inputs and outputs are assumed to be related and there is a strong feedback. Notably, this analysis case is identical to activity definition presented in Figure 13.

Case 3: Partial Information Flow & No Feedback: All 10 of the design definition activities (DDA), and all seven of the design evaluation activities (DEA), and all 12 design management activities (DMA) were decomposed into sub-activities after tearing and partitioning. Several groupings of design activities were observed. These observations indicate the Sim/Duffy ontology must be decomposed into smaller "atomic" activities.

Case 4: Complete Information Flow & No Feedback: The design definition activities were not decomposed into sub-activities in this analysis. However, the design evaluation and design management activities are decomposed into sub-activities and the classification are scattered around design definition activities.

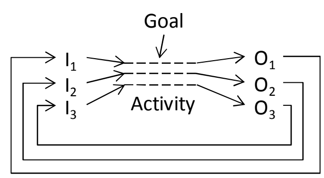
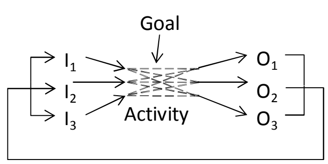
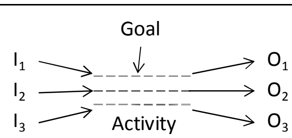
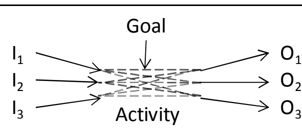
Selection of an activity model:

In the next section, the ontology is modeled based on the most practical activity model, which was selected after a screening process where a DSM analysis of the four cases implementing the ontology was conducted. This task was performed as a first pass analysis, to get a base ontology that would be stable for the computational implementation. The following section also presents a table (see Table 1) of the results obtained after the analysis of the four cases.

In addition to the observations from the DSM analysis, several fundamental limitations of the ontology include:

- Lack of examples illustrating the use of design activity ontology,
- A process "composed of" the suggested design activities
- Study of the ontology for describing engineering design processes
- And, inadequate analysis of the resultant vocabulary.

Table 1: Analyzed cases with representation of information flows in design activities

Case	Activity Model	Analysis observations
Case 1		•Feedback should not be present within an activity •Representation shows it can be decomposed further •8/27 activities decomposed; and 4 new activities and 2 large circuits/groupings were formed
Case 2		•Feedback should not be present within an activity •Represents Sim/Duffy definition of activity •2/27 activities decomposed; and 2 new activities and 2 large circuits were formed
Case 3		•Representation shows it can be decomposed further •27/27 activities were decomposed; 6 new activities were formed; and 2 small circuits and 1 large circuit were formed
Case 4		•Represents a typical design activity •22/27 activities were decomposed; 3 new activities were formed; and 2 large circuits were formed

The ontology must be changed to ensure that the activities represent reality; it must be redefined into a broader and more stable set of information entities; and to eliminate "dead-end" information flows that may not be used by other design activities. Based on the literature survey and empirical studies it was also evident that although it is possible to have a feedback loop in a *process* to elaborate on this, a feedback can exist between two activities; feedback cannot exist within an activity itself. The major reasons for this observation is that,

- This would increase the redundancy in the information generated within that activity
- This would generate unwanted results or cause deviations from expected flow patterns
- Creates greater number of instances on the feedback section of the DSM, which are always tried to be minimized by the use of algorithms, to get an optimal arrangement of the elements.

The base vocabulary in the ontology is orthogonal at the activity level and not orthogonal at the information level. (Orthogonality means uniqueness.)

- This can be interpreted as the definitions of the activities in the ontology are unique which is good but they are also contradictory (as Pentland [42] emphasizes to maintain consistency in the grammar developed). Several activities just have some information flows that are essential but are not carried forward to the downstream activities. Thus creating unique activities that act as dead ends for information flows.
- The base information flows are not atomic and can be classified under many different categories which make it not orthogonal thus creating duplicates of information flows. Thus we expect the information flows to be as unique as possible as it is the most atomic part of the ontology and that creates the ontology.

  The activity definitions do not capture essential attributes (like entry or exit criteria, time, metrics, resources, tools etc.) thus limiting computational implementation. An exhaustive list of attributes related to an activity is described by Browning [8, 9]). The information flows between activities are inadequately explained and lacking examples. There are not many repeating information flows for the activities. This indicates that the activities are not well defined and there are large gaps in the connectivity of the information, which also states that there is no overlapping of activities found illustrating the fact that these activities are merely aggregated rather than being integrated. Most of the activities that decomposed formed an alliance with other disintegrated activities to generate a large circuit or loop; these relationships are not covered in the existing ontology and the activities that housed many different activities under it, called as sub-activities, did not prove to have any relationship with its parents in the DSM analysis.

The activity and its goal illustrate a one-to-one mapping. But an activity can have "m" number of inputs and "n" number of outputs and the inputs and outputs are information that is dependent on each other. The input cannot be the same as the output, although some of the activities have the same information represented as input and output, the output is considered to be an enhanced/updated/upgraded version of the input containing more details to it than when it was used as the input for that particular activity. For e.g., In Constraining the "Constraints hierarchy" is present as an input and as an output, which can be interpreted as, when this information was used as an input, it would be associated with the hard and soft constraints that would have been developed for a project and would be classified based on a crude manner or a designers discretion and when the activity of constraining is executed the output "Constraints Hierarchy" would be the constraints classified based on the design requirements or as the team's discretion.

Based on the several observations made in the previous section and some of the key elements recognized above, Case 4 seems to be a valid representation of a typical design activity and the pattern of information flowing within an activity. Also the following figures graphically illustrate Case 4 and a more generic activity model of Case 4 is also shown in Figure 18 and Figure 19.

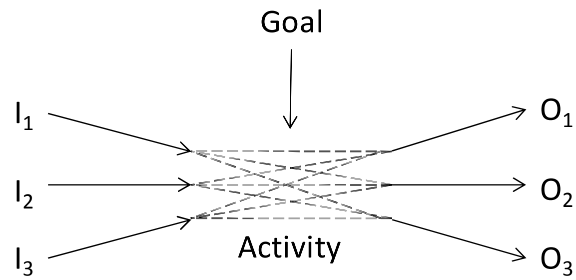


Figure 18: Case 4, design activity and information model

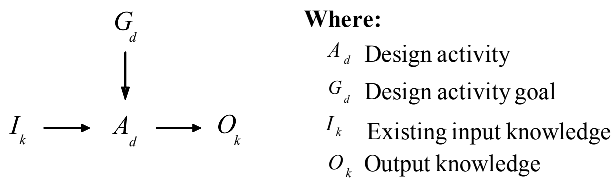


Figure 19: Model of engineering design activity

Figure 19 illustrates the activity model developed in this research. This model also conforms with the IDEF 0 standards [24] and the activity definition by Browning [10]. Thus this chapter concludes by providing the basic foundation to develop and refine the Sim and Duffy ontology based on the DSM analyses and evaluation.

Refinement of the 112 information entities

Table 2 provides the exhaustive list of input and output information developed by Sim and Duffy [47] in their ontology.

Table 2: Vocabulary of Input or Output Information by Sim and Duffy

Input or Output Information
1. Knowledge of function to behavior to structure mapping
2. Knowledge of different level of abstractions
3. Knowledge of the appropriate representation of abstractions
4. Knowledge of relevant domains for different aspects of product
5. Knowledge of the design space
6. Knowledge of product configuration
7. Knowledge of relationships of design properties
8. Knowledge of integrating physical building blocks
9. Types of abstraction

Table 2: Vocabulary of Input or Output Information by Sim and Duffy (Contd.)

Input or Output Information
10. Domain knowledge
11. Appropriate abstractions of design object (e.g. sketches, schematics)
12. Knowledge of function/sub-function hierarchy
13. Design catalogues which map function to solution principle component
14. Knowledge of function to solution principle mapping/structural building block
15. Knowledge of function/sub function decomposition
16. Knowledge of function to design parameters to structural forms
17. Knowledge of mapping between function and physical hierarchies
18. Knowledge of causal relationship between function and behavior
19. Knowledge of rules of combination
20. Knowledge of embodiments
21. Examples of knowledge of function to solution principle/component mapping
22. Knowledge of F to WP to S
23. Knowledge of F to DP
24. Specific qualitative causal and relational knowledge of concepts in terms of F to B to S mapping(s)
25. Existing similar product structure in terms of part/ sub-part relationship, system/sub-system relationship
26. Knowledge of function requirements
27. Existing knowledge of function means mapping, function component mapping.
28. Specific taxonomy of new design in terms of system/sub-system part/sub-part dependencies or independence
29. Specific taxonomy of complex function in terms of sub-functions mapping, function means mapping, function component mapping
30. Domain knowledge
31. Methods for generating ideas (e.g. brainstorming, Gallery method)
32. Ideas, concepts and their links
33. Domain knowledge
34. Combination tables, function modules
35. Concepts or modules that satisfy the overall functions
36. Knowledge of interfaces/interactions between parts, systems
37. Knowledge of specifications components/parts, systems of the product
38. Knowledge of product architecture in terms of chunks and their interactions
39. Reasons for fundamental and incidental interactions
40. Domain knowledge relating the manufacturing, assembly and testing of product
41. Design requirements
42. Detail drawings in terms of part structure
43. Documentation (e.g. design specification, assembly procedure, etc.)
44. Knowledge of design requirements
45. Summary of design decisions made at the end of design iterations or milestones in the design process
46. Various statements of specifications (e.g. specification of requirements, product specification, interface specification, bill of materials)
47. Knowledge of detail design of the product

Table 2: Vocabulary of Input or Output Information by Sim and Duffy (Contd.)

Input or Output Information
48. Knowledge of specifications of components/parts of the product
49. Knowledge of classes of standard components
50. A set of standard components selected
51. Criteria or explanation for the standardization
52. Knowledge of design requirements
53. Knowledge of design alternatives
54. Knowledge of the appropriate analysis method and/or experimental/simulation techniques
55. Knowledge of given criteria, introduced or derived criteria
56. Knowledge of fundamental decisions made (i.e. design rational)
57. Knowledge of selecting the appropriate methods and analysis methods/techniques to support analysis
58. Knowledge of design specification and objectives
59. Knowledge of the appropriate evaluation method and/or experimental/simulation techniques
60. Knowledge of the behavior performance of the artifact compared to the design specification
61. Knowledge of attributes of alternatives
62. Knowledge of attribute-defining requirements
63. Knowledge of choice criteria
64. Knowledge of object selected
65. Knowledge of criteria used
66. Knowledge relating to the physical phenomena and theories
67. The constraints, assumptions made and degree of accuracy required
68. The structure/form of the design
69. The working environment of the design
70. Methods of analysis related to the physical phenomena
71. Knowledge of the behavior of the design
72. Knowledge of the appropriate modeling techniques for the types of analysis required
73. Knowledge of the appropriate models
74. Knowledge of design of the appropriate simulation models
75. Knowledge of the design requirements
76. Knowledge of the expected behavior under certain testing environment
77. Design specification
78. Testing criteria
79. Test results (e.g. resistance against speed curve)
80. Specific design requirements
81. Knowledge of types of constraints applicable
82. Knowledge of specific constraints (hard or soft constraints) applied
83. Rationale for applying the constraints
84. Knowledge of past designs and solutions
85. Client and design brief
86. Knowledge of problem structure and any missing information and knowledge
87. Domain knowledge
88. Past design cases
89. Design methods/methodology
90. Relevant domain knowledge

Table 2: Vocabulary of Input or Output Information by Sim and Duffy (Contd.)

Input or Output Information
91. Specific design case
92. Specific design method(s)/methodology
93. In-house or vendor s depository of information/knowledge
94. Specific knowledge/information related to the design
95. Knowledge of possible type of conflicts
96. Knowledge of conflict resolution strategies
97. Knowledge of hard constraints enforced and/or relaxation of soft constraints
98. Knowledge of search strategy
99. Knowledge/information sources (e.g. patents, lead user interview etc.)
100. End result of the search (e.g. information, patent, component etc.)
101. Knowledge of interrelated activities and their precedence orders
102. Knowledge of algorithms for clustering activities (e.g. triangularisation algorithm)
103. Knowledge of sequencing of decoupled and/or coupled activities
104. Knowledge of relative importance of goals
105. Knowledge of the information requirements of each sub-task and resource
106. Knowledge of agenda of goals in order of priority
107. Design tasks, resources (e.g. design staff), tools (e.g. CAD systems, design software)
108. Planning method/algorithms (e.g. PERT)
109. Sequence of design tasks and allocation of resources and tools
110. Design tasks, resources (e.g. design staff), tools (e.g. CAD systems, design software)
111. Scheduling method/algorithms
112. Sequence of design tasks and allocation of resources and tools in terms of time stamps and due dates

The protocol used for analysis and refinement of information flows are,

- Reduction of verbiage

  The information entities used as input or output knowledge were all referred as "knowledge" yet the names contained "knowledge of …" in them, thus the information entities were changed to reflect this known entity. All the information entities that had the prefix or suffix "knowledge of …" were changed to just indicate the information entity. But there were certain information entities where the knowledge term had to be retained with the name to preserve its meaning. The information entities were, Domain Knowledge, Embodiments Knowledge, Combination Knowledge, and Mapping

Knowledge. There were a total of 64 information entities affected by this transformation. Table 3 provides some examples of this implementation of name change.

Table 3: Examples of name change for information entities

Information Entities with "Knowledge of …" as suffix or prefix	**Changed Information entity**
Knowledge of the appropriate representation of abstractions	Appropriate representation of abstractions
Knowledge of design alternatives	Design Alternatives
Knowledge of the design requirements	Design Requirements
Knowledge of search strategy	Search Strategy

- Establish information as a noun instead of verb

This type of conversion for the information entities involved the removal of verb from the information flow's name and providing a noun in the name. Thus the task of eliminating "Mapping" from information entities and replacing it with "Map" was exercised. Verb is ideally associated to tasks and activities alone. The changes were implemented based on the activity they were involved with; and the goal and description of that activity. There was one exception where "Mapping Knowledge" had the term mapping as it had to be retained to give that information entity its original meaning. There were a total of 6 information entities affected by this transformation and some examples are discussed in Table 4.

Table 4: Examples of mapping to map information entities transformation

Information Entities with Knowledge as suffix or prefix	**Changed Information entity**
Specific qualitative causal and relational knowledge of concepts in terms of F to B to S mapping(s)	Function to Behavior Map
	Behavior to Structure map
Examples of knowledge of function to solution principle/component mapping	Function to Solution Principle Map
	Function to Component Map
Knowledge of mapping between function and physical hierarchies	Function to Physical Hierarchies Map

- Redundant information flow (Semantic Equivalence)

  There were several repeats of information flows, which could be represented by retaining just one counterpart. For example, “Knowledge of function/sub-function hierarchy” is considered to be equivalent to “Knowledge of function/sub function decomposition;” these can simply be replaced by “Function/sub-function hierarchy.” This transformation helped to eliminate several redundant information entities that were being generated but never used. Some examples for the redundant information flows that could be eliminated from the list which neither contributed to the activity nor was grouped during this transformation are; Specific taxonomy of new design in terms of system/sub-system part/sub-part dependencies or independence; Specific taxonomy of complex function in terms of sub-functions mapping, function means mapping, function component mapping; Concepts or modules that satisfy the overall functions; Knowledge of fundamental decisions made (i.e. design rational) etc. There were a total of 20 information entities affected by this transformation.

- Normalization of information entities

  Several information entities had a complex representation. It was a grouping of information entities which were composed of some similar transformation or theme (having a common goal and description). There was decomposition of a single information entity into several individual elements and it was observed that they could survive on their own and still contributed to the formation of activities based on their description. These information entities could be separated into individual information flows. There were a total of 10 information entities affected by this transformation and Table 5 provides some examples for such transformations.

Table 5: Examples of complex information entities transformation

Complex Information Entities	**Changed Information entity**
Knowledge of F to WP to S	Function to Working Principle Map
	Working Principle to Structures Map
Specific qualitative causal and relational knowledge of concepts in terms of F to B to S mapping(s)	Function to Behavior Map
	Behavior to Structure Map
Design tasks, resources (e.g. design staff), tools (e.g. CAD systems, design software)	Design Task Hierarchy
	Tools Map
	Resource Map

- Hierarchies

Some of the information entities could be grouped to form certain hierarchical information entities that could provide a complete set of information from which it was formed. These information entities along with the hierarchy contained the data pertaining to the attributes that form the hierarchy. There were a total of 11 information entities affected by this transformation. The information entities that provided the hierarchical information are, Constraints Hierarchy (Hard constraints and Soft constraints); Activities Hierarchy; Hierarchy of Goals; Design Task Hierarchy; Standard Components Hierarchy; Function/Sub-Function Hierarchy; Part/Sub-Part Hierarchy; System/Sub-System Hierarchy; Function to Physical Hierarchies Map; Abstractions Hierarchy; and Hierarchy of design decisions. A specific example of this transformation is illustrated in Table 6.

Table 6: Specific example of hierarchy transformation

Information Entities that exhibit hierarchy	**Changed Information entity**
Knowledge of function/sub-function hierarchy	Function/Sub-Function Hierarchy
Knowledge of function/sub function decomposition	
Specific taxonomy of complex function in terms of sub-functions mapping, function means mapping, function component mapping	

- The definition of the input or output knowledge with the use of "()" and "etc."

Several information entities contained the use of "()" or "etc" in their name or a shot explanation and in some cases it were found to be a standalone information entity.

Thus we sorted out such information entities and have created individuals for those. For example, "Documentation (e.g. design specification, assembly procedure, etc.)" could be decomposed into Design Specification; Assembly procedure; and Drawings. There were a total of 12 information entities affected by this transformation.

After the exhaustive task of transformation, 112 information flows were reduced to 82 unique information flows that could be used to model the activities in the ontology. A detailed list of the 82 information flows is provided in Chapter four, where the developed ontology is explained in detail.

Refinement of Activities

Similar to the modification of information flows, several activities that had their information modified or deleted had to be analyzed to provide connectivity in the DAO. Thus the Activities had to be reorganized based on its description and goals (the classification provided by Sim and Duffy did not have a strong background.) Also, the most important observation to the change in activities list was the integration of the "Planning" and "Scheduling" activities. These two activities control the occurrence and sequence of an activity and thus must be present to govern a process. This is omitted in Sim and Duffy's ontological classification. The planning and scheduling activities are only present in DMA and not in DDA or DEA. Thus new classification or process models have to be developed to incorporate the DDA (similar to conceptual design phase: changing an ill-structured problem into a well-structured problem) and DEA (similar to embodiment design phase: developing a solution that was selected from a group of feasible solutions) along with the planning and scheduling activities. The information entities related to activities, Planning and Scheduling are:

Planning

- Design Task Hierarchy

- Resource Map
- Tools Map
- Algorithms and methods for planning

Scheduling

- Design Task Hierarchy
- Resource Map
- Tools Map
- Algorithms and methods for scheduling

It can be seen that they are almost identical with just a difference of one information entity. Thus these two activities were grouped to for a super-set activity known as "Planning and Scheduling." Thus the list of 26 activities was reduced to 25 activities based on the merger of Planning and Scheduling activities. The list of 25 activities with its respective information flows is illustrated in Chapter four.

CHAPTER FOUR:

DESIGN ACTIVITY ONTOLOGY – DAO

OBJECTIVES:

- Refine Sim and Duffy Ontology to generate the Design Activity Ontology (DAO)
- Discuss the model or the template for a design activity
- Complete illustration of the DAO
- Implement and Analyze the DAO in Protégé and Description Logic (DL)

The DAO is a stable version of the Sim and Duffy ontology which is ready to be implemented in a computational background. The DAO has a classification much different to that of the Sim and Duffy classification, wherein the classification in the DAO is based on the goals, input and output information associated with the activity, thus enabling a clear and complete representation of the activities in a hierarchical manner. The DAO has a better model for a design activity which has no feedback present within the same activity and each of the activities has been represented graphically depicting the exact number of input and output information flows, along with an abbreviation to quickly recognize the activity. The descriptions for the DAO are basically the summary of descriptions provided by Sim and Duffy [47]. Thus the DAO was evolved from the Sim and Duffy ontology, which was considered as the baseline ontology, but henceforth the DAO would be referred to as the baseline ontology as we completed one iteration of the ontology development lifecycle. The following section starts with the second iteration of this cycle.

ITERATION TWO OF THE ONTOLOGY DEVELOPMENT LIFECYCLE

Figure 20 illustrates the complete cycle of Iteration 2; wherein the Vocabulary development phase is completed by developing the DAO by refining the Sim and Duffy ontology (basically the output of Iteration 1); the Computational representation phase is completed by implementing and modeling the DAO in Protégé; and Analysis phase will be completed by analyzing and evaluating the developed models in Protégé using DL and the tools and plug-ins available in Protégé.

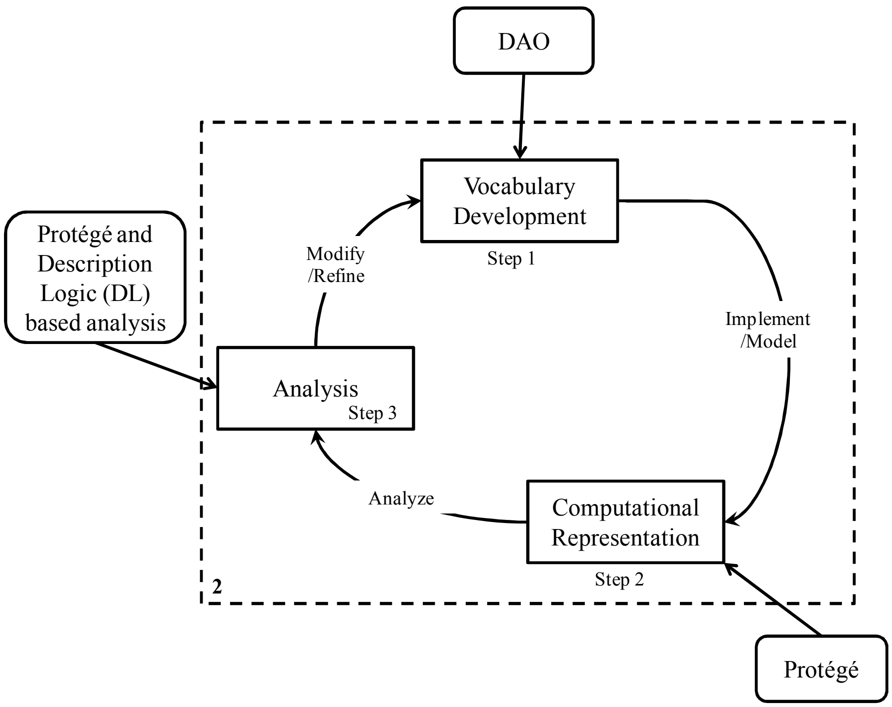


Figure 20: Second Iteration of the Ontology Development Lifecycle

Step 1: Vocabulary development phase

The DAO has attributes and an expanded version of the attributes with its relationships is established. The vocabulary of the activities and the hierarchy of the DAO is discussed in detail in the taxonomy section.

Attributes

As stated earlier, the information entities are considered to be the most atomic part of the DAO, the attributes recognized are:

Input Knowledge: The input knowledge is categorized into tacit and explicit knowledge,[36]. Tacit knowledge is context-specific and therefore hard to formalize and communicate. In the domain of engineering design, three types of knowledge have been commonly referred, design object knowledge, design process knowledge and design rationale knowledge. Collectively, it is also known as experiential knowledge. Explicit knowledge, on the other hand, refers to knowledge that is transmittable in formal, systematic language. This would include scientific and technological knowledge which can be differentiated and structured into different types for different design activities[47]. This ontology has recognized 82 information entities and has shown that these entities are a substantial and exhaustive list.

Output Knowledge: Stems from the application of the appropriate activity based upon the input knowledge, to enable the design to progress towards the design goal and hence towards the ultimate goal, the design solution. The output is the change in information. With the help of the acquired knowledge the person or the group in charge of the design may act rationally or competently by invoking the next activity that may bring the design nearer to the final solution. The nature of the output knowledge is therefore dependent on the design activity and the evolving design solution, [47].

Goal: The design problem has been described as a goal-directed or a goal-oriented process. The goals can be specified or derived. Specified goals are goals inferred from the design requirements that must be complied with. Derived goals are goals invoked in the course of the design process. This may lead to a goal sub-goal hierarchical relationship. Here the goal of the design activity will be influenced by the nature of the activity being considered,[47].

Resources: Typically a design activity or a design process is executed by an individual or a group. The resources in this context of the research and the DAO can be referred to the number of people responsible to execute the activity.

Tools: A design activity is performed using a tool which may be a software (for e.g., CAD/CAM/CAE software); or a design method (for e.g., Brainstorming or Gallery method). These elements can be classified as the tools used to complete the activity.

Time Stamps: It is also true that any design activity conducted is according to a deadline, thus the activity would have a starting point and a finishing point that can be measured using time, and thus we can consider the start and end times of an activity to be a part of the definition of a particular activity.

Currently the first three attributes were considered to be modeled and last three were suggested to be developed in future work explained in the future work section in Chapter six. Also for the modeling and analyses purposes of the ontology, attributes such as Tools, Times and Resources are not considered as they may be used or shared by multiple activities and tasks at any given point in a project, thus making it difficult to query on a particular activity. Thus the interest lies on activities and the attributes that govern these activities such as Goal, Input Knowledge and Output Knowledge. The Case four from Chapter three (see Figure 18 and Figure 19) shows the use of these attributes in the activity model suggested.

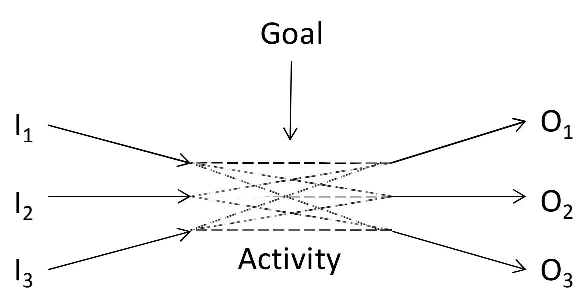


Figure 18: Case 4, design activity and information model

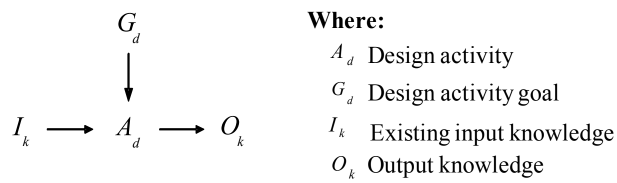


Figure 19: Model of engineering design activity

Vocabulary

The various vocabularies of the DAO are presented in Table 7 and Table 8. The taxonomy (the relationships and the hierarchy) basically is built in these vocabularies. The vocabularies are split into two sections, the vocabulary of Information entities that compose the input and output information which in turn are used to compose design activities expanded in Table 8. Thus there can be 82 possibilities for input information. Similar to input information the output information can be chosen from the same exhaustive list of 82 information entities. Table 8 illustrates the DAO, this table is a summary of all the information pertaining to design activities that is integrated and organized to generate the DAO which can be used to develop process models. The vocabulary of goals of the activities that are used to classify the ontology presented in Table 8. Since there are 25 activities listed in this ontology the number of goals described are 25 (as discussed previously that the mapping between the activities and its goal is one to one.) The vocabulary of design activities are used to develop hierarchies in the ontology. As stated above, there are 25 activities described in this ontology. The information entities are organized according to how they would be generated during the design process. This is just an approximation and is not a strict list as we also specify that this is just the second iteration in the ontology development lifecycle. Design information evolves and it can change depending on various design parameters, design problems, or company's customized or established design process. Table 7 presents the information entities that form the atomic elements of this DAO.

Table 7: The vocabulary of Input or Output Information

Input or Output Information
1. Client and Design Brief
2. Design Requirements
3. Design Objectives
4. Information sources
5. Past Designs and Past Design Cases
6. Design Information
7. Domain Knowledge
8. Repository of design information
9. Problem Structure
10. Degree of accuracy required
11. Product Architecture and Interactions
12. Reasons for fundamental and incidental interactions
13. Physical phenomena and Theories
14. Modeling Techniques
15. Design Methodology
16. Criteria Map
17. Constraints Hierarchy (Hard constraints and Soft constraints)
18. Assumptions
19. Activities Hierarchy
20. Hierarchy of Goals
21. Design Task Hierarchy
22. Mapping Knowledge
23. Missing Information
24. Conflict Resolution Strategies
25. Search Strategy
26. Search Results
27. Standard Components Hierarchy
28. Behavior to Design Specification Map
29. Resource Map
30. Tools Map
31. Information requirements hierarchy
32. Function/Sub-Function Hierarchy
33. Part/Sub-Part Hierarchy
34. System/Sub-System Hierarchy
35. Functional Requirements
36. Functional Means Map
37. Function to Physical Hierarchies Map
38. Abstractions Hierarchy
39. Appropriate representation of abstractions
40. Function to Behavior Map
41. Behavior to Structure map
42. Function to Solution Principle Map
43. Function to Component Map

Table 7: The vocabulary of Input or Output Information (contd.)

Input or Output Information
44. Function to Working Principle Map
45. Working Principle to Structures Map
46. Function to Design Parameters Map
47. Design Parameters to Structure Map
48. Integrating physical building blocks
49. Design Properties and Relationships
50. Methods/Tools for generating ideas
51. Ideas
52. Concepts
53. Function Modules
54. Parts and Systems Interaction
55. Embodiments Knowledge
56. Design Space
57. Product Configuration
58. Combination Knowledge
59. Design Geometry
60. Design Behavior
61. Design Environment
62. Detail Drawings
63. Design Specifications
64. Assembly Procedure
65. Hierarchy of design decisions
66. Detail Design
67. Set of standard components selected
68. Criteria for Standardization
69. Design Alternatives
70. Analysis Methods/Techniques (Experimental and/or Simulation)
71. Evaluation Methods/Techniques (Experimental and/or Simulation)
72. Object Selected
73. Criteria Used
74. Appropriate/developed models
75. Simulation Models
76. Testing Environment
77. Test Results
78. Relaxation of soft constraints
79. Algorithms for activities
80. Decoupled and/or Coupled Activities
81. Algorithms and methods for planning
82. Algorithms and methods for scheduling

Table 8 provides the details on the various activities in the DAO in the alphabetical order, the table provides details such as the Information flows (inputs and outputs) associated with the activities, its abbreviation, its goal, its graphical representation, and a brief description. Several of the descriptions for the activity are derived from Sim and Duffy's [47] definition of activities.

The modifications to these activities are based on the observations in Chapter 3 and thus there will be some difference in the description of each of these activities. The description section of the following table also provides a list of activities that can precede or succeed the activity at focus, this can be used to verify the flow of information when the activities are put together to form a process.

Table 8: The vocabulary of activities

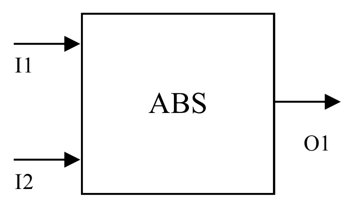
Activity Name: Abstracting	**Abbreviation: ABS**
Activity Goal: To simplify the complexity of design object	
Input Information	**Output Information**
I1. Abstractions Hierarchy	O1. Appropriate representation of abstractions Sketches, Schematics, etc.
I2. Domain Knowledge	
Description Sim and Duffy summarize that Abstracting as an activity is to abstract knowledge that can depict useful relationships of the evolving design concept and to reduce the complexity of the designing object. It is also used to ignore the particulars and emphasize the generic. Abstractions are more than mere simplification of form and behavior, they are information booths that can be used for better decision making and for the evolution of solutions. Preceding Activities: Identifying Succeeding Activities: Synthesizing and Composing	**Graphical Representation** 
Activity Name: Analyzing	**Abbreviation: ANA**
Activity Goal: Prediction of the behavior of a design	
Input Information	**Output Information**
I1. Physical phenomena and Theories	O1. Design Behavior
I2. Constraints Hierarchy (Hard constraints and Soft constraints)	

Table 8: The vocabulary of activities (Contd.)

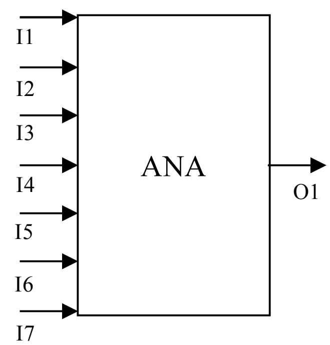
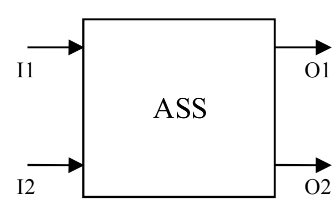
I3. Assumptions	
I4. Degree of accuracy required	
I5. Design Geometry	
I6. Design Environment	
I7. Analysis Methods/Techniques (Experimental and/or Simulation)	
Description Use of models or design related data to answer questions pertaining to the behavior of the design. Though this activity is aimed at producing quantitative results, it is also capable of providing qualitative analyses. The examples include FEA analysis, heuristics-based analysis, approximation analysis, numerical analysis. Preceding Activities: Constraining and Modeling Succeeding Activities: Testing or Experimenting	**Graphical Representation** 
Activity Name: Associating	**Abbreviation: ASS**
Activity Goal: Generate novel or new ideas/concepts through association of ideas/concepts	
Input Information	**Output Information**
I1. Methods/Tools for generating ideas	O1. Ideas
I2. Domain Knowledge	O2. Concepts
Description Generation of new ideas and concepts or relating existing ideas and concepts to generate something useful or different. This activity is associated with the way designers think and thus falls under the classification of a cognitive activity. For e.g., idea generation methods like brainstorming, Gallery method, 656, Morph charts. Preceding Activities: Identifying Succeeding Activities: Composing	**Graphical Representation** 
Activity Name: Composing	**Abbreviation: COM**
Activity Goal: Combine ideas/concepts through association of ideas/concepts that satisfy overall function	
Input Information	**Output Information**

Table 8: The vocabulary of activities (Contd.)

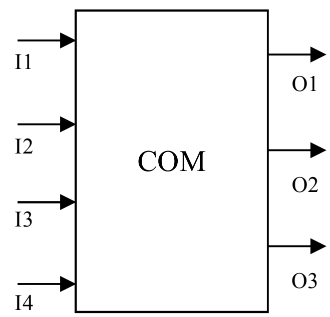
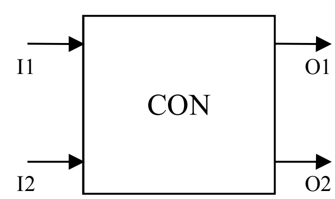
I1. Domain Knowledge	O1. Concepts
I2. Combination Knowledge	O2. Function Modules
I3. Ideas	O3. Design Alternatives
I4. Appropriate representation of abstractions	
Description Combine design ideas or design modules into concepts or to complete the conceptual design cycle. This is predominantly a concept generation activity. This activity also provides the Energy, Material and Signal (EMS) flows which are captured in the function modules. Preceding Activities: Identifying, Abstracting and Associating Succeeding Activities: Detailing	**Graphical Representation** 
Activity Name: Constraining	**Abbreviation: CON**
Activity Goal: To reduce the complexity of the design solution space	
Input Information	**Output Information**
I1. Design Requirements	O1. Constraints Hierarchy (Hard constraints and Soft constraints)
I2. Constraints Hierarchy (Hard constraints and Soft constraints)	O2. Design Space
Description The exploration of the design solution phase thus reducing the complexity of the design itself and producing feasible design solutions. The name for this activity is derived from the fact that designers are constrained with several parameters and entities in design which have to be considered before they can wander away from a common goal. This activity also provides a list of constraints applicable to the design along with its hierarchy. Preceding Activities: None Succeeding Activities: Analyzing, Resolving, Standardizing and Exploring	**Graphical Representation** 
Activity Name: Decision Making	**Abbreviation: DM**
Activity Goal: Choose the best alternative(s) from a set based on some criteria	

Table 8: The vocabulary of activities (Contd.)

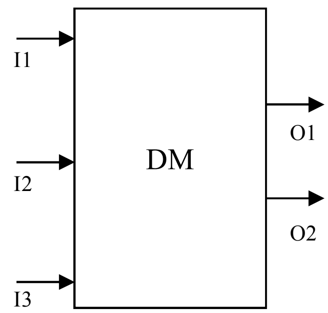
<table>
<tr><th>Input Information</th><th>Output Information</th></tr>
<tr><td>I1. Design Requirements</td><td>O1. Criteria Mapping</td></tr>
<tr><td>I2. Design Alternatives</td><td>O2. Hierarchy of design decisions</td></tr>
<tr><td>I3. Hierarchy of Goals</td><td></td></tr>
<tr><td>Description
As the goal of the activity suggests this activity is aimed at producing results, i.e., selection of the best alternative amongst several qualifying design solutions based on a set of criteria which can be represented using tools such as Design Decision Matrix, Design Structure Matrix, QFD. Furthermore Sim and Duffy have provided 2 types of classifications for decisions, 1. Process oriented, and 2. Product oriented; which self explanatory. This is one of the most important activities in a design process and must be captured explicitly and can be vital in repeating the success for variant and adaptive product designs.
Preceding Activities: Prioritizing and Composing
Succeeding Activities: Selecting, Testing or Experimenting, and Defining</td><td>Graphical Representation

</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Activity Name: Decomposing</td><td>Abbreviation: DEC</td></tr>
<tr><td colspan="2">Activity Goal: 1. Knowledge of product structure or product modularity.
2. Knowledge of functional requirements to design solutions.
3. Maximize decoupling of design activities into tasks/sub-tasks so as to reduce design iteration(s), Minimize information flow between activities.</td></tr>
<tr><th>Input Information</th><th>Output Information</th></tr>
<tr><td>I1. Part/Sub-Part Hierarchy</td><td>O1. Part/Sub-Part Hierarchy</td></tr>
<tr><td>I2. System/Sub-System Hierarchy</td><td>O2. System/Sub-System Hierarchy</td></tr>
<tr><td>I3. Functional Requirements</td><td>O3. Function/Sub-Function Hierarchy</td></tr>
<tr><td>I4. Function to Component Mapping</td><td>O4. Function to Component Mapping</td></tr>
<tr><td>I5. Functional Means mapping</td><td>O5. Functional Means mapping</td></tr>
<tr><td>I6. Activities Hierarchy</td><td>O6. Decoupled and/or Coupled Activities</td></tr>
<tr><td>I7. Mapping Knowledge</td><td rowspan="2"></td></tr>
<tr><td>I8. Algorithms for activities</td></tr>
</table>

Table 8: The vocabulary of activities (Contd.)

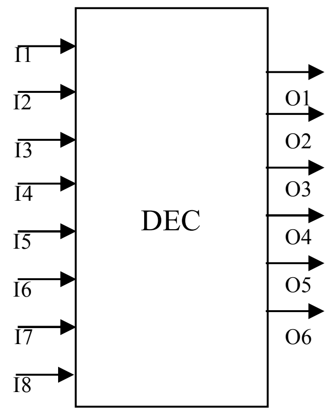
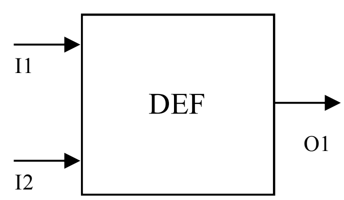
Description The goal of this activity is divided into three parts but the underlying goal of this activity suggest that this is a problem solving activity where complex problems are broken down or decomposed into fragments of simpler problems which can be solved with less or minimum effort. Tasks, problems, objects to be modeled or represented can be simplified or decomposed; but decomposing inherently has a drawback of adding complexity to the design process as the things that were decomposed has to be composed back at the end of this activity to provide that single solution. Preceding Activities: None Succeeding Activities: Generating	**Graphical Representation** 
Activity Name: Defining	**Abbreviation: DEF**
Activity Goal: Definitive decisions representing milestones in the design process that have influence on downstream activities	
Input Information	**Output Information**
I1. Design Requirements	O1. Design Specifications
I2. Hierarchy of design decisions	
Description Defining is an activity to provide design specific documents and specifications that will be used to define the product for downstream activities. The design tasks are oriented in such a manner that tangible outcomes or deliverables will be the end product of this activity. As the goal suggests this activity makes definitive decisions or descriptions of the design to remove uncertainty and to increase the completeness of the product in terms of its description in manuals, reports, specifications list. Preceding Activities: Analyzing, Evaluating, and Decision Making Succeeding Activities: Standardizing, Evaluating, Testing or Experimenting, and Structuring or Integrating	**Graphical Representation** 

Table 8: The vocabulary of activities (Contd.)

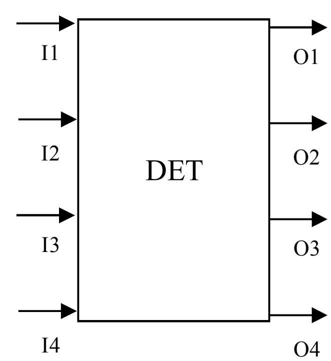
Activity Name: Detailing	**Abbreviation: DET**
Activity Goal: 1. Knowledge of product structure or product modularity. 2. Knowledge of functional requirements to design solutions. 3. Maximize decoupling of design activities into tasks/sub-tasks so as to reduce design iteration(s), Minimize information flow between activities.	
Input Information	**Output Information**
I1. Design Requirements	O1. Detail Drawings
I2. Domain Knowledge	O2. Assembly Procedure
I3. Detail Design	O3. Design Specifications
I4. Concepts	O4. Repository of design information
Description Evolution of design to meet the functional requirements and producing production oriented tasks that can provide details on form dimensions and tolerances; material and surface properties of all the individual parts; and all the drawing and production documents like detail drawings, part lists and instructions for assembly, testing, adjustment, maintenance etc. The eventual output being the complete manufacturing information. This activity is instrumental in the outcome of the project, as lack of detailing can ruin the best of concepts and vice-versa. Preceding Activities: Identifying, Standardizing, Associating, Composing, and Evaluating Succeeding Activities: Structuring or Integrating, Standardizing, Evaluating, and Testing or Experimenting	**Graphical Representation** 
Activity Name: Evaluating	**Abbreviation: EVA**
Activity Goal: A measure of the quality or value of the design solution(s) with respect to a given criterion	
Input Information	**Output Information**
I1. Design Specifications	O1. Behavior to Design Specification Mapping
I2. Design Objectives	O2. Detail Design
I3. Evaluation Methods/Techniques (Experimental and/or Simulation)	

Table 8: The vocabulary of activities (Contd.)

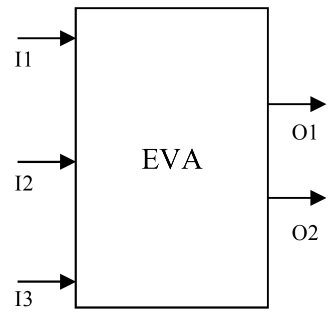
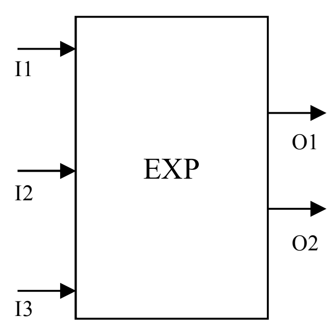
Description	**Graphical Representation**
Concerned with adding quality to the design solution by mapping the solution to the design objectives that were developed based on certain criterions and constraints. This activity is aimed at checking the design to ensure all the requirements have been met and the proposed system will not fail. Some of the examples for the evaluation methods or techniques are, Pugh Selection Matrix, QFD, Mathematical modeling, FEA, CAD, Prototyping. Several DFX rules and approaches can be adopted for this activity too. Preceding Activities: Structuring or Integrating, Standardizing, Detailing, and Testing or Experimenting Succeeding Activities: Detailing	
Activity Name: Exploring	**Abbreviation: EXP**
Activity Goal: To provide a structure to the design problem and explore the design space for solutions	
Input Information	**Output Information**
I1. Past Designs and Past Design Cases	O1. Problem Structure
I2. Client and Design Brief	O2. Missing Information
I3. Design Space	
Description	**Graphical Representation**
Many open-ended practical problems like design, the start state, the goal state and the transformation functions are radically under-specified; and because of the ill-structured nature of the design problem space the solutions are sub-par. Thus exploring as a design activity helps designers to define the structure of the problem space and the potential design solutions. This activity involves tapping of knowledge from several sources to compensate for the missing information, protocol studies, case studies, problem solving techniques, etc. are used. The client and design briefs are the starting point for this activity. Preceding Activities: Identifying, Constraining and Synthesizing Succeeding Activities: Searching	

Table 8: The vocabulary of activities (Contd.)

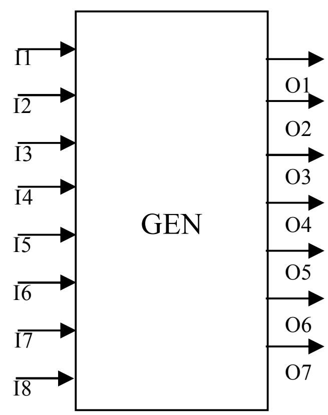
Activity Name: Generating	**Abbreviation: GEN**
Activity Goal: 1. Generate solutions principle(s)/component(s) that satisfy function hierarchy and the solution principles. 2. Generate design solutions using causal reasoning	
Input Information	**Output Information**
I1. Function/Sub-Function Hierarchy	O1. Function to Solution Principle Mapping
I2. Function to Solution Principle Mapping	O2. Function to Working Principle Mapping
I3. Function to Design Parameters Mapping	O3. Working Principle to Structures Mapping
I4. Design Parameters to Structure Mapping	O4. Function to Design Parameters Mapping
I5. Function to Behavior Mapping	O5. Function to Behavior Mapping
I6. Function to Physical Hierarchies Mapping	O6. Behavior to Structure Mapping
I7. Embodiments Knowledge	O7. Design Specifications
I8. Combination Knowledge	
Description Consider the various aspects that go into a concept to satisfy the functional requirements of a design solution. The knowledge about various mappings is the key to this activity. It also provides a hierarchy to generate concepts. This activity inherently uses the design requirements to check its functional factors that contribute to the evolution of the design or the concepts. Morph charts are used in this activity and prior to concept generation, the functional requirements expressed in the form of a function structures or hierarchy is obtained through the functional decomposition activity. Finally able to satisfy the overall functional requirements. Preceding Activities: Decomposing Succeeding Activities: Synthesizing, Structuring or Integrating, Standardizing, Evaluating, and Testing or Experimenting	**Graphical Representation** 
Activity Name: Identifying	**Abbreviation: IDE**
Activity Goal: To mark the relevant and the essential in order to manage the complexity of the design problem	
Input Information	**Output Information**
I1. Domain Knowledge	O1. Domain Knowledge
I2. Past Designs and Past Design Cases	O2. Past Designs and Past Design Cases
I3. Design Methodology	O3. Design Methodology
I4. Design Properties and Relationships	
I5. Design Information	

Table 8: The vocabulary of activities (Contd.)

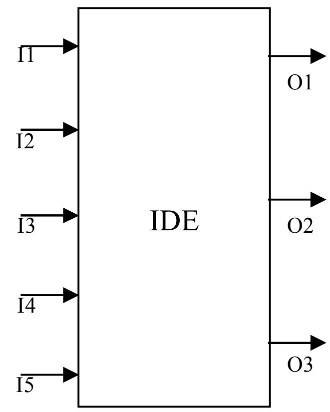
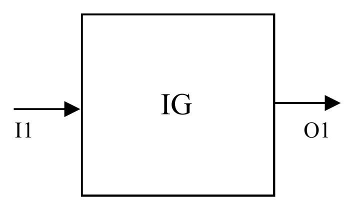
Description	**Graphical Representation**
This is another complexity reducing activity and where the designers are encouraged to work on the procedure and to acquire the relevant resources to generate the solutions. Past cases, Design methodologies and Domain knowledge are the most important knowledge's pertaining to this activity. This activity provides the result for the question, is this design "Original/Novel or Adaptive or Variant?" This activity lays down design roadmap, tools, software, analysis/evaluation methods, etc. and plays a crucial role as to how the design process will be managed and conducted. Preceding Activities: Synthesizing Succeeding Activities: Information gathering	
Activity Name: Information Gathering	**Abbreviation: IG**
Activity Goal: To provide up-to-date knowledge that may progress the design to the next stage or lead to a concrete definition	
Input Information	**Output Information**
I1. Repository of design information	O1. Design Information
Description	**Graphical Representation**
The repository of design information (In-house or Vendor) is accessed for more information regarding the design to push the design to the next stage, Sim and Duffy state that engineering designers spend as much as 30% of the time searching for engineering design information; thus this activity tries to reduce this "Non-productive time." Though this method is too time consuming, it ultimately leads to a better design decisions based on complete data and accurate assumptions as the repositories had up-to-date and relevant information. Preceding Activities: Searching and Detailing Succeeding Activities: Identifying	
Activity Name: Modeling	**Abbreviation: MOD**
Activity Goal: Appropriate modeling of the design dependent on the perspective required for the current design activity	
Input Information	**Output Information**
I1. Modeling Techniques	O1. Appropriate/developed models

Table 8: The vocabulary of activities (Contd.)

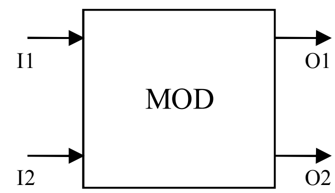
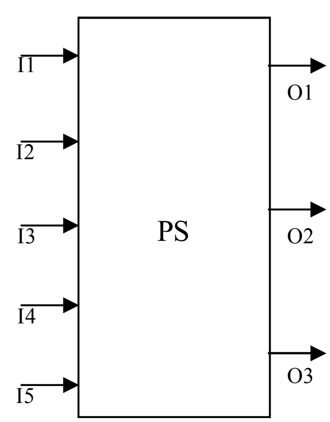
I2. Repository of design information	O2. Design Geometry
Description Designer's representation of some aspects of the intended product to focus attention, through physical modeling, geometrical modeling, CAD modeling, analyses modeling, information modeling, and mathematical modeling. Better communication to help control and predict of the performance of a design and, most importantly, serve as an abstract representation of the design which everybody can relate to. Modeling also helps to evaluate the structure or form and the behavior of a design. The repository of design information is crucial as it might contain existing designs. Preceding Activities: Searching and Detailing Succeeding Activities: Simulating and Analyzing	**Graphical Representation** 
Activity Name: Planning and Scheduling	**Abbreviation: PS**
Activity Goal: Minimize time to market by streamlining design tasks	
Input Information	**Output Information**
I1. Design Task Hierarchy	O1. Design Task Hierarchy
I2. Resource Mapping	O2. Resource Mapping
I3. Tools Mapping	O3. Tools Mapping
I4. Algorithms and methods for planning	
I5. Algorithms and methods for scheduling	
Description Organizes resources for the design process in terms of the order of tasks, resource allocation, the assignment of personnel and tools for each of those tasks and also the activities related to the manufacturing of the product such as purchasing, production, logistics. All these are also accompanied by actual time stamps and time duration specified for each task. Time-sensitive tasks that lead to a critical path can be identified using critical path analysis. For adaptive and variant designs, such plans may be available for reuse with some or no modification. For original designs, triangularisation method of Kusiak and Wang[47] can be used Preceding and Succeeding Activities: None	**Graphical Representation** 

Table 8: The vocabulary of activities (Contd.)

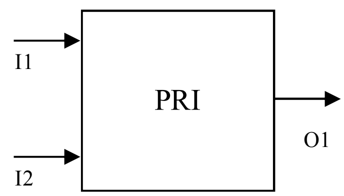
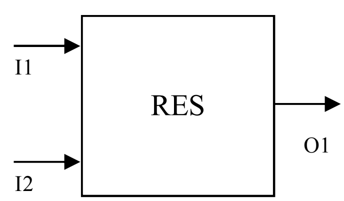
Activity Name: Prioritizing	**Abbreviation: PRI**
Activity Goal: Focus on important goals that have influence on downstream design activities	
Input Information	**Output Information**
I1. Hierarchy of Goals	O1. Hierarchy of Goals
I2. Information requirements hierarchy	
Description Giving importance to those tasks that are goal oriented and these tasks are hierarchically arranged based on this importance factor. Preceding Activities: None Succeeding Activities: Decision Making	**Graphical Representation** 
Activity Name: Resolving	**Abbreviation: RES**
Activity Goal: Focus on important goals that have influence on downstream design activities	
Input Information	**Output Information**
I1. Conflict Resolution Strategies	O1.Relaxation of soft constraints
I2. Constraints Hierarchy (Hard constraints and Soft constraints)	
Description Conflicting interest, requirements and viewpoints are inherent in design. Conflicts exist in individual agents design or in a collaborative design effort. The resolution of such conflicts is important and pervades throughout the design process and requires extensive knowledge of conflict management strategies. Preceding Activities: Constraining Succeeding Activities: None	**Graphical Representation** 
Activity Name: Searching	**Abbreviation: SEA**
Activity Goal: To satisfy some requirement of the design problem or solution	
Input Information	**Output Information**
I1. Search Strategy	O1. Search Results
I2. Information sources	O2. Repository of design information
I3. Missing Information	
I4. Object Selected	
I5. Design Space	

Table 8: The vocabulary of activities (Contd.)

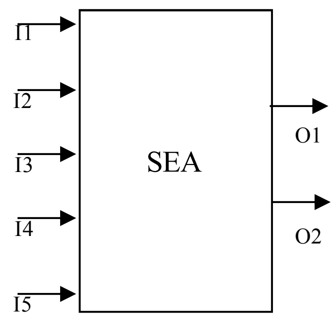
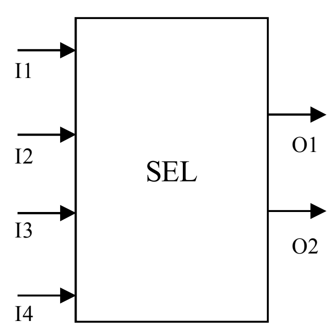
Description	Graphical Representation
Searching activity is similar to exploring but with specific expectations. Done within the design space to look for feasible solutions and alternatives. Searching strategies like genetic algorithms (GA's) and optimization techniques are used. Sometimes external searching is conducted that is an information-gathering process and the sources include lead user interview, expert consultation, patent searches, literature searches and competitive benchmarking; and internal searching involves a process of retrieving a potentially useful piece of information from the designers memory or that of a team of designers and adapting that information to the problem at hand. Preceding Activities: Exploring and Selecting Succeeding Activities: Synthesizing, Testing or Experimenting, Modeling and Information Gathering	
Activity Name: Selecting	**Abbreviation: SEL**
Activity Goal: To choose a feasible design solution or activities from a set of alternatives	
Input Information	**Output Information**
I1. Design Alternatives	O1. Object Selected
I2. Functional Requirements	O2. Criteria Used
I3. Design Requirements	
I4. Criteria Mapping	
Description	**Graphical Representation**
Choosing a design object that satisfies design requirements from a specified set of alternatives. The term object is used in a general sense, covering, for example, the selection of a working principle for a device, a material type, a component from a catalogue, a functional module, a completed design, or a design goal. Selection occurs in all phases of design. Preceding Activities: Composing and Decision Making Succeeding Activities: Searching and Testing or Experimenting	

Table 8: The vocabulary of activities (Contd.)

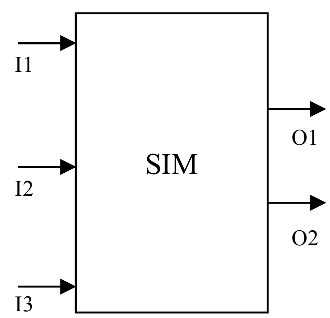
Activity Name: Simulating	**Abbreviation: SIM**
Activity Goal: To form an image or imitation of the behavior and properties of the artifact using appropriate models	
Input Information	**Output Information**
I1. Appropriate/ developed models	O1. Design Behavior
I2. Design Requirements	O2. Testing Environment
I3. Simulation Models	
Description Form an image or imitation of the behavior and properties of the design by reasoning, and/or testing models preceding the manufacture and actual use. It leads to expectations about the actual properties of the design compared to its predicted counterpart. Depending on the behavior under study, there is a wide variety of simulation models, from mathematical models to true-to nature material replicas of conceptualizations. Preceding Activities: Modeling Succeeding Activities: Testing or Experimenting	**Graphical Representation** 
Activity Name: Standardizing	**Abbreviation: STA**
Activity Goal: Reduce complexity in terms of the number of components and parts used in the design; Minimize manufacturing time and cost of products	
Input Information	**Output Information**
I1. Design Specifications	O1. Set of standard components selected
I2. Criteria for Standardization	O2. Detail Design
I3. Design Space	
I4. Standard Components Hierarchy	

Table 8: The vocabulary of activities (Contd.)

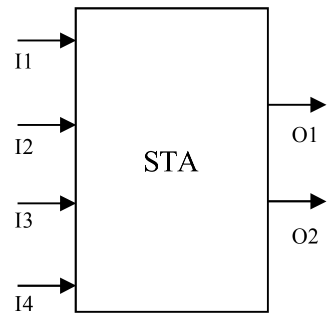
Description	**Graphical Representation**
Special kind of selecting activity that is related to the detail design phase and may be considered as one of the design strategies in an organization to reduce the complexity of the designed products. This activity is guided by the provided functional and performance requirements. Designers should attempt to utilize as many of the existing parts and components as possible in the design to promote uniformity and internal standardization. External standardization occurs when designer use externally supplied component and part that are commonly used by manufacturers of similar product lines. This will lead to standardization of manufacturing and assembly processes. Advantages include elimination of development costs of new components, reduced start-up costs of equipment and machinery, reduced lead time, reduced tooling costs (since tools are already available from previous manufacture of similar parts) and higher production quantities leading to economies of scale. Just-in-time arrangements are made easier due to larger usage quantity and reduction in number of parts/components. Preceding Activities: Detailing, Defining, Generating, Constraining, Synthesizing Succeeding Activities: Detailing	
Activity Name: Structuring or Integrating	**Abbreviation: SI**
Activity Goal: Optimal product architecture that minimizes the complexity of co-ordination required to develop the total product/system	
Input Information	**Output Information**
I1. Design Specifications	O1. Product Architecture and Interactions
I2. Parts and Systems Interaction	O2. Reasons for fundamental and incidental interactions

Table 8: The vocabulary of activities (Contd.)

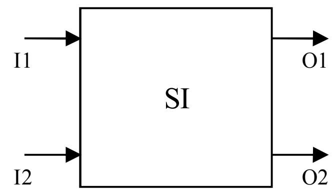
Description	**Graphical Representation**
Describing a product in terms of functional and physical elements and to reduce the complexity to co-ordinate the development of the overall product. The functional elements are usually described in schematic form before implementing them in specific technologies, components or physical working principles; and the physical elements (i.e. parts, components and sub-assemblies) that ultimately implement the product's functions are typically organized into several major physical building blocks called chunks or modules, because these chunks could interact with one another in many ways, to facilitate integration and two categories of interaction are identified between chunks, namely fundamental and incidental interactions. Fundamental interactions are explicitly represented by the schematic showing the clustering of elements into chunks. Incidental interactions are due to the geometric arrangement of the chunks. By identifying chunks with high interactions early in the design process, a design team can choose an architecture that minimizes the complexity of the co-ordination and communication required to develop the system. Knowledge of such incidental interactions can be documented using schematic and interaction graphs or matrices [47]. Preceding Activities: Detailing, Defining, and Generating Succeeding Activities: None	
Activity Name: Synthesizing	**Abbreviation: SYN**
Activity Goal: Totality in the Design of the Product	
Input Information	**Output Information**
I1. Function to Behavior Mapping	O1. Design Space
I2. Behavior to Structure Mapping	O2. Product Configuration
I3. Appropriate representation of abstractions	O3. Design Properties and Relationships
I4. Abstractions Hierarchy	O4. Integrating physical building blocks
I5. Domain Knowledge	

Table 8: The vocabulary of activities (Contd.)

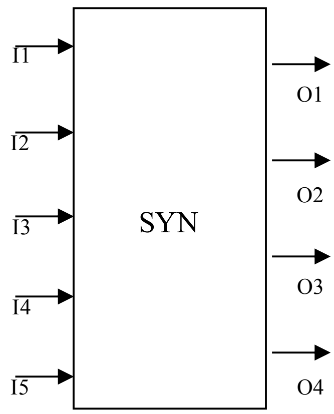
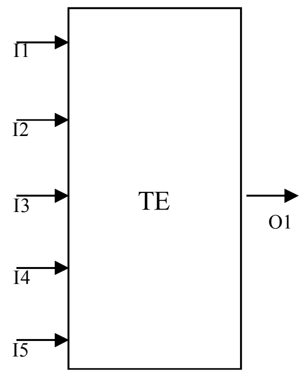
Description	**Graphical Representation**
Mapping between function, behavior and form. But synthesis is more than just a mapping of these entities. Pahl and Beitz [38] describe synthesis as putting together of parts or elements to produce new effects and to demonstrate that these effects create an overall order. Synthesis when applied to the designing of artificial products or systems entails the integration of parts/systems such that the physical laws of the domains involved when acting together in a given environment give rise to a desired behavior and performance. In mechanical design, synthesis is seen as a systematic construction of designs based on generic elements. Design for X (DFX) is considered as a method in the activity of synthesizing in which the designer integrates desirable features and properties in the design throughout the design process. DFX can be defined as a knowledge-based approach that attempts to design products that maximize all desirable characteristics such as – high quality, reliability, serviceability, safety, user friendliness, environmental friendliness, short time-to-market – in a product design while at the same time minimizing lifetime costs, including manufacturing costs [47]. Preceding Activities: Generating and Identifying Succeeding Activities: Standardizing, Exploring, Searching, and Identifying	
Activity Name: Testing and Experimenting	**Abbreviation: TE**
Activity Goal: To verify actual behavior against expected behavior	
Input Information	**Output Information**
I1. Design Specifications	O1. Test Results
I2. Criteria Mapping	
I3. Criteria Used	
I4. Testing Environment	
I5. Design Behavior	

Table 8: The vocabulary of activities (Contd.)

Description	Graphical Representation
Most designs require some form of testing either during the design process, e.g. Stress strain testing on beams/frames used to support systems or components in a product; Products for the consumer or engineering markets usually require a factory test to verify the quality of the product and its compliance with the design specification [44]. Unlike the activity of analysis, in which the behavior of the design is derived through simulation, in testing/experimenting the behavior of the design is derived through measuring the various parameters describing the behavior. Preceding Activities: Generating, Detailing, Defining, Decision Making, Selecting, Simulating, and Analyzing Succeeding Activities: None	

Taxonomy

The relationships in the DAO can be explained in a simple manner by introducing the concept of Taxonomy, which is basically a representation of concepts and relations of an ontology in a structured manner. Ontologies can be based around a single taxonomy or several taxonomies and the relationships; here taxonomies are organized hierarchically and the concepts can be arranged as classes with subclasses [1]. The process followed was,

1. Identifying the taxonomies that form an engineering design ontology (known as root concept);
2. Identify existing taxonomies for each of the root concepts from the previous stage;
3. Creating new taxonomies if no existing taxonomy was found;
4. Testing the taxonomies for the particular application;
5. And refinement of the integrated taxonomy.

The engineering design activities may be termed as the root concept of the taxonomy which has been identified and described in full by Sim and Duffy[47] (step one and two covered), step 3 was skipped. Step 4 was the testing of the taxonomy for the particular application, and here we test the taxonomy for both indexing design information and to apply it to case studies for better evaluation Web Ontology Language (OWL) is used to represent the resulting taxonomy with the help of Protégé; and Description Logic (DL) is used for the analysis of this representation. Step 5 is consistently described in this thesis.

Activity Taxonomy: To provide the details on the taxonomy developed, we establish, Design Activity Cluster/Group, Design Phase and a Design Process. At a high level it can be stated that an activity is the most atomic element in design analogous to an atom being the most atomic part of an element, but the level of specificity required by designers would eliminate activity's odds of being the most atomic element which can be analogous to the requirement of science to sometimes go beyond atoms depending on the situations and say that the neutrons, protons and electrons are the most atomic entity. Thus the research at certain levels states that the information entities that form the activities are the most atomic element. As discussed earlier, let us consider for the moment that, activities are the most atomic part of a design, we can say that activities form a cluster or grouping also known as Design Activity Cluster/Group and these groups come together to form a Design Phase and The design phases align to form a Design Process. Figure 21 illustrates the formation of this taxonomy and the design process at various levels of abstraction.

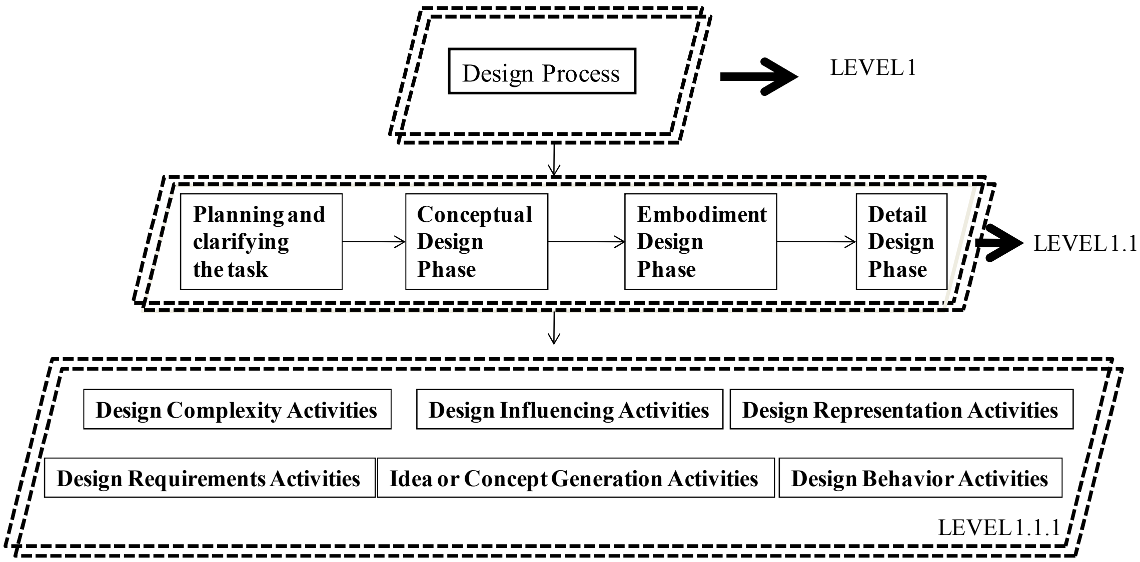


Figure 21: Hierarchy of the DAO

Step 2: Computational representation phase

Computational representation phase is the implementation of the refined ontology into a computer based application for future use and revisions. This phase is executed by implementing the refined ontology using "Protégé" which along with "Description Logic (DL)" can be used to evaluate, and apply ontologies for design engineering problems. But before we go into the details of the computational representation phase a quick introduction of Protégé and DL is provided.

Protégé and Description Logic (DL) Overview

Protégé: Protégé[2] is an open-source ontology development environment with functionality for editing classes, properties, and instances; it provides a growing user community with a suite of tools to construct domain models and information-based applications with ontologies. Protégé implements a rich set of information-modeling structures and actions that support the creation, visualization, and manipulation of ontologies. Protégé can be customized to provide domain-friendly support for creating information models and entering data. Protégé can

[2] http://protege.stanford.edu/overview / Date visited: 05/15/08

be extended by way of a plug-in architecture and a Java-based Application Programming Interface (API) for building information-based tools and applications. Protégé ontology development environment uses a Web Ontology Language (OWL) plugin, wherein OWL provides powerful reasoning capabilities that help in building, maintaining and query of domain models. OWL supports purposes such as, to define classes and properties, to edit logical class expressions, to invoke reasoners, and to link ontologies into the Semantic Web.

According to Gennaria, J. H. and colleagues, [18], Protégé supports the idea that the labor of information-base construction can be divided into: (1) overall ontology construction by a information engineer and then (2) information-base filling-in by a domain expert. However, via meta classes, the sorts of objects can be extended by domain experts who create and edit; unlike other tools, where domain experts were limited to creating instances. The authors state that Protégé is more agnostic about what sort of objects (classes or instances) get created when. The user interface is simple and the meta-class capability is largely hidden from naïve users, and the class/instance distinction is retained by the user interface. Protégé is flexible and powerful for developers and information engineers, yet also support simple tools that are easy for the domain specialist to understand and use. Any of the tabs in Protégé can be configured as ‘‘hidden’’, so that end-users see only the functionality that they need. The use of plug-ins helps us to distribute the development workload across multiple programmers, including external programmers. It also provides a greater range of custom-tailored features to the Protégé user interface. The development and publication of this API is what allows independent developers to build plug-in components that add to or modify the functionality of Protégé [18].

Description Logic (DL): Description Logics (DLs) forms a subfield of information representation and reasoning (KRR) based on formal logic systems. DLs are a family of logics-based information formalisms that facilitate representation and reasoning about information in a structured manner. DLs provide a formal syntax and semantics for describing information within a domain in terms of concepts and properties that specific individuals must satisfy. DL's are extensively used, as it is good for modeling and it enables classification, hierarchy, analysis and provides reasoning. DLs predominantly rely on concepts and properties to model the information within a particular domain. A more detailed description on DL and its specific relation to engineering information modeling can be found in [32]. Each DL[3] defines a number of language constructs (such as intersection, union, role quantification, etc.) that can be used to define new concepts and roles. The main reasoning tasks are classification and satisfiability, subsumption and instance checking. Subsumption represents the is-a relation. Classification is the computation of a concept hierarchy based on subsumption. A whole family of information representation systems have been built using these languages and for most of them complexity results for the main reasoning tasks are known. Description logic systems have been used for building a variety of applications including conceptual modeling, information integration, query mechanisms, view maintenance, software management systems, planning systems, configuration systems, and natural language understanding. The DL tool used was Fact ++ which is a free (GPL/LGPL) open-source C++-based reasoned. It implements a tableau-based decision procedure for general Tboxes (subsumption, satisfiability, classification) and Aboxes (retrieval). OWL-DL with qualifying cardinality restrictions.

Protégé Modeling

The DAO was modeled into a computational background using a software called Protégé. Though this phase was executed initially using a DSM, it was always intended to be used in a information management tool like Protégé. The modeling is simple as the Protégé tool is built to

[3] http://www.ida.liu.se/labs/iislab/people/patla/DL/index.html

Date visited: 05/15/08

represent ontologies and to critically analyze the developed ontologies with the use of DL based reasoners such as Fact ++, Racer Pro etc. The modeling phase of the Protégé representation involved critical thinking of what should be used as the most atomic element from the DAO, the information entities or the activities; for which an immediate answer that surfaced the mind was information entities as they would allow more room for expansion and customization as activities inherently do not describe what tasks were to be executed, i.e., just by reading the name, as different researchers around the globe have different terminologies and descriptions for these activities as explained by [22, 38, 47, 48, 54, 57]. Thus we can consider the information entities to be the most atomic part, (as deduced in iteration one for the representation in the DSM's).

Protégé Modeling Step 1 would be to click on the "OWL Classes" tab and to go to the "OWL Thing" section. This section will be used to model the information entities. Accordingly, the information entities that include, Input Information, Output Information and Goals, which are considered to be the unique identifiers, are modeled under "Design Knowledge" which are commonly grouped as the "OWL Thing". The activities are also modeled alongside Design Knowledge along with the various classifications and hierarchies determined for this ontology as the OWL Thing. The screenshot of the Protégé software with this construction is illustrated in Figure 22 through Figure 24. In Figure 22, the shaded regions shows the topics of interest. The oval shaded region is the "OWL Classes" tab and the rectangular shaded region is the "OWL Thing" section, these are the basic options used for modeling the DAO. This type of modeling is executed because we aim for the DL reasoner to deduce and map, the relationships and hierarchies, specified as rules in the model.

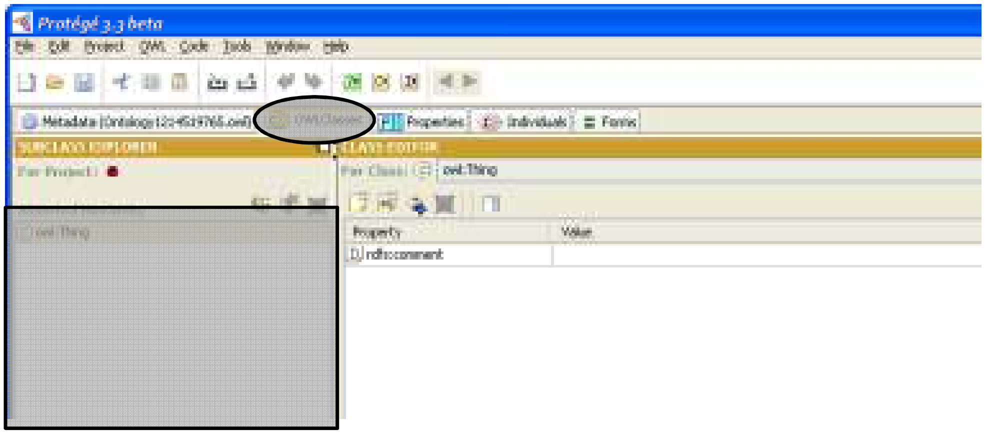


Figure 22: "OWL Classes" tab and "OWL Thing" section used for modeling

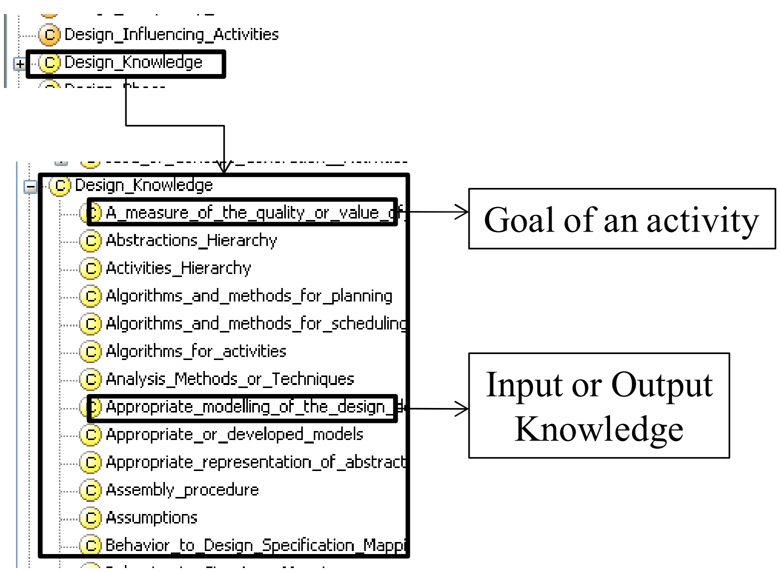


Figure 23: Modeling of Design Knowledge, (where Design Knowledge = Input Knowledge + Output Knowledge + Goals)

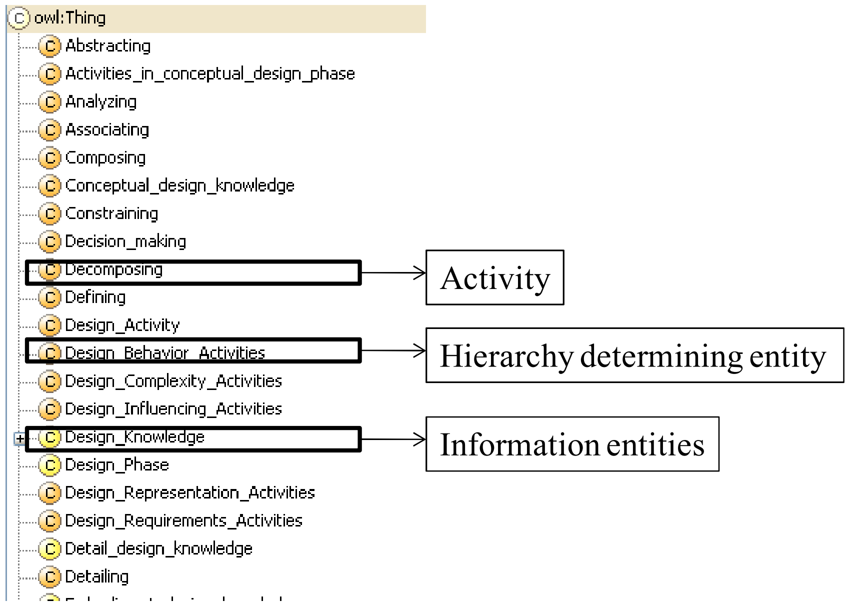


Figure 24: Modeling of Activities alongside Design Knowledge as OWL Things

Protégé Modeling Step 2, the modeling of all the information entities and activities with its respective hierarchical names were executed in the previous step, this involved the linking of the information entities to their respective activities through rules and formal specifications. To proceed further, properties such as Input Information, Output Information and Goals had to be created in the properties tab which could be used by the activities to be specified under the "Necessary & Sufficient" section under "Asserted Conditions." Then the created properties are utilized to define activities. An activity is defined to have the conditions "some" and "only" which denotes the necessary and sufficient type, and it is modeled in this manner because we define an activity to have occurred if one of the information entities from the input information and one of the information entities from the output information have occurred; OR to say that it is not necessary for all of the information entities associated with a particular activity to have occurred, for an activity to be considered as executed in a design process. Thus providing an open world assumption situation to the design process at focus and enabling to modify the process model dynamically. These tasks described above are illustrated in Figure 25 through Figure 27.

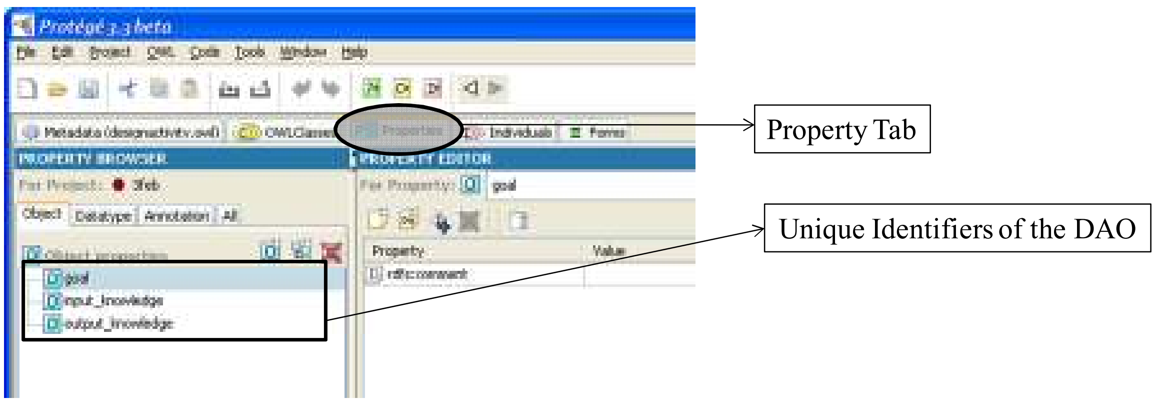


Figure 25: Property Tab and Unique identifiers modeling in Protégé

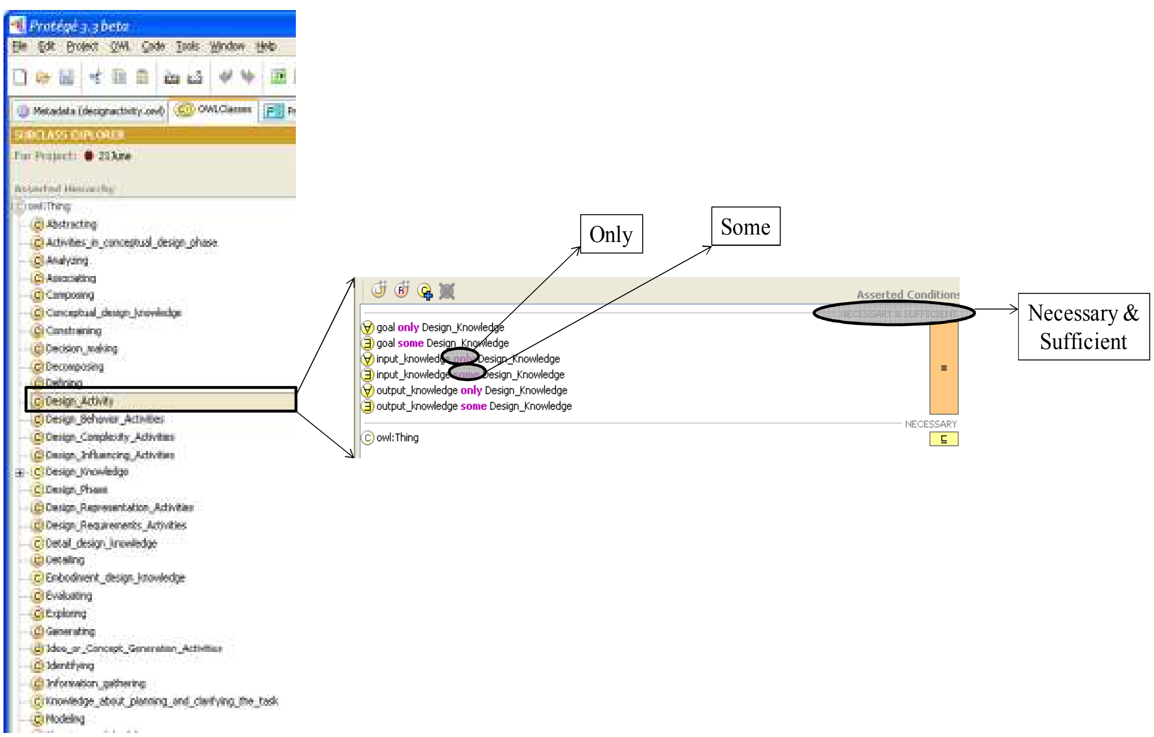


Figure 26: DAO activity model in Protégé

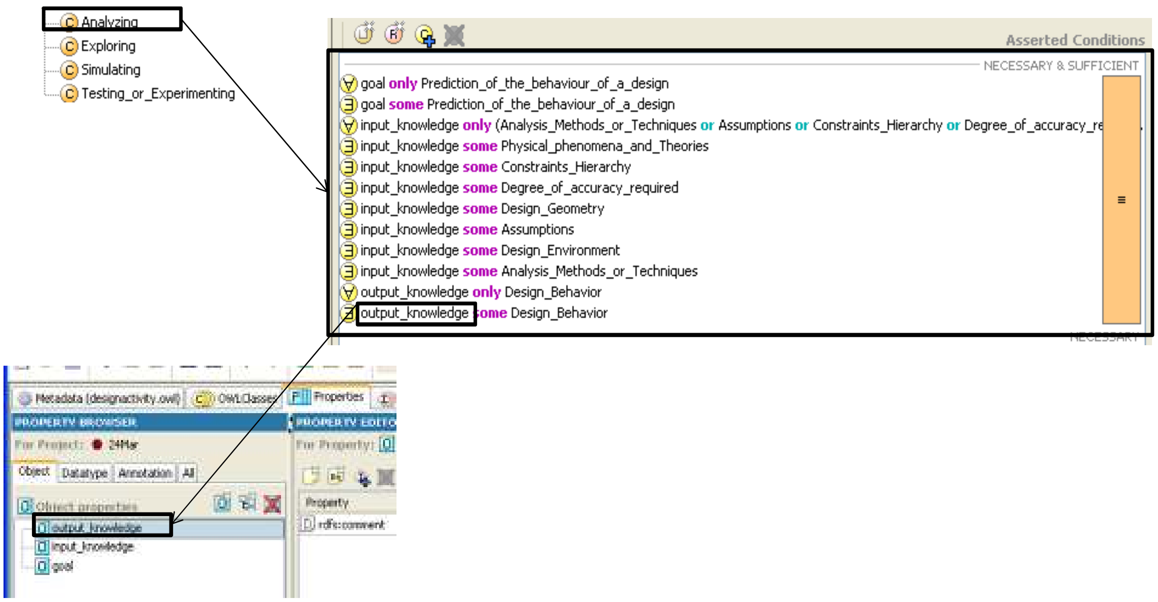


Figure 27: Specific example of an activity (Analyzing)

Protégé Modeling Step 3 involves the modeling of the hierarchy of the ontology in Protégé; which involved defining of the intermediate phases namely, Design Behavior Activities, Design Complexity Activities, Design Influencing Activities, Design Representation Activities, Design Requirements Activities and Idea/Concept Generation Activities. The next higher level would be the design phases described by Pahl and Beitz [38], the Planning and Clarifying the Task phase, Conceptual design phase, Embodiment design phase, and Detail design phase. The intermediate phases are modeled based on the goals of the activities. Activity with similar goals have been grouped as they tend to achieve the same results but with a slight change in orientation or path or direction of the tasks that are executed under those activities. There is also substantial change in the information related to these similar activities. Thus the name of those intermediate phases indicates the general theme of the activities under it. The model is intended to be stopped at the intermediate level of hierarchy since any combination of these intermediate phases can be used in the design phases, thus creating complexity in the modeling and graphical representation of the DAO in Protégé's "OWLViz"plug-in; also the design information is blocked for its entities as it would distort the magnitude of the graphical representation of the hierarchy. The modeling of the hierarchy and a snippet of the graphical representation of the hierarchy via "OWLViz" is illustrated in Figure 28 and Figure 29.

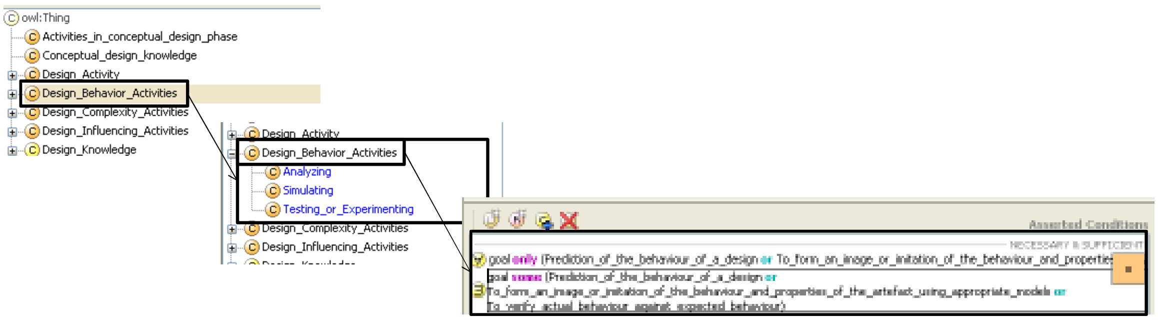


Figure 28: Expanded view of Design Behavior Activities an intermediate phase in the DAO

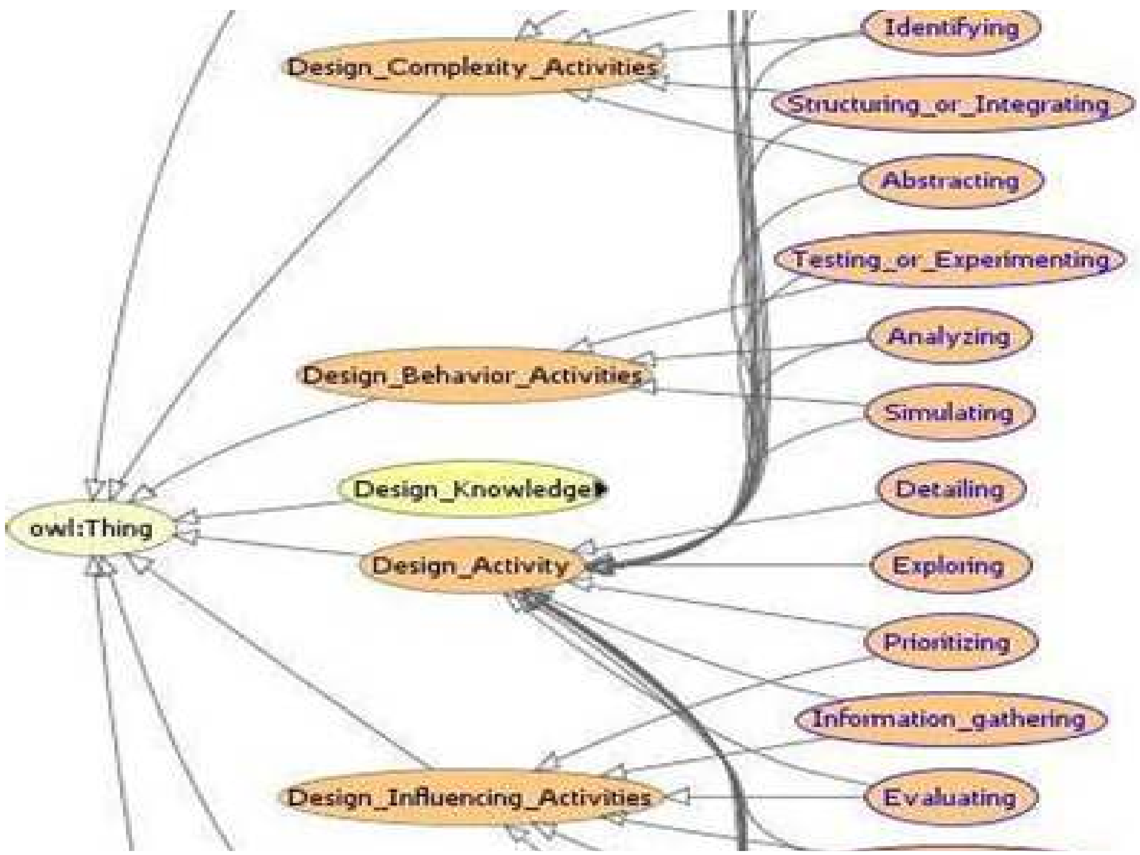


Figure 29: Snapshot of the hierarchy of the DAO in OWLViz

Step 3: Analysis phase

Analysis of the DAO through Protégé and DL is as basic as the software and the DL reasoner attached to the software. The created models were checked for consistency, taxonomy and inferred classes using the option available in Protégé as shown in Figure 30.

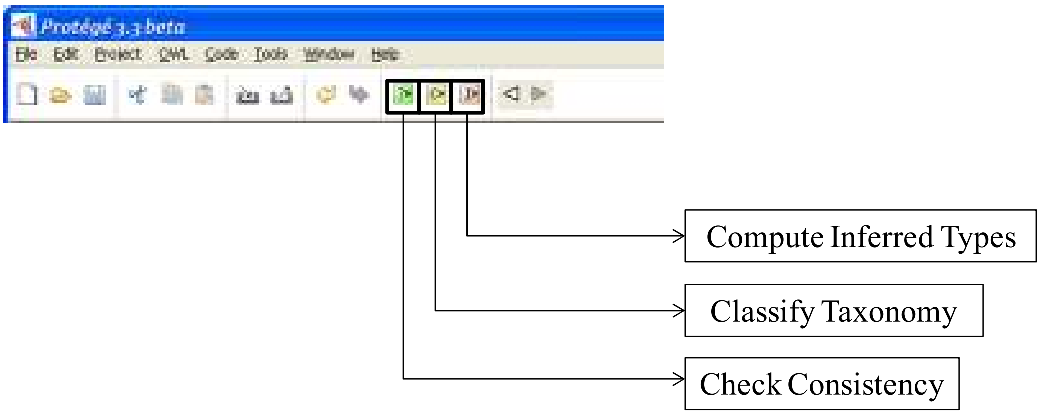


Figure 30: Protégé and DL as tools for analysis of the DAO

The results of all the three options were clean with no discrepancies or failures. But basically analysis was performed based on visual evaluation during modeling; as Protégé does not allow duplicates of the same entity to be modeled, the orthogonality in the design information side and with respect to activities names were maintained. Also by creating dummy activities that had just the test information as input or output was verified to be in multiple activities (so as to not maintain orthogonality here, and to see if there could be a flow of information between activities,) thus showing that an input for an activity was basically an output of another activity. These are the basic analysis that was conducted on the DAO to ensure that the issues and observations made in chapter 3 were addressed. Similarly the classification or the hierarchy provided in this research, i.e., the grouping of the activities based on the goal of a particular activity to provide an intermediate class was developed in Protégé, thus proving the validity of this classification. To be specific, the goals were made the essence of these intermediate phases model and when Protégé was called to solve for the inherent dependencies of those common goals, it provided us with a list of activities with the common goals based on the inputs provided on those goals. The lists and classification provided by Protégé were used as the hierarchy classification. Thus we can check the major issues like connectivity, orthogonality and completeness.

Based on the Protégé modeling and analysis of the DAO, I would also like to discuss some observations,

- Planning and Scheduling activity does not have any preceding or succeeding activities, thus making it a standalone activity and raises the question as to, "why it should be executed?" The answer is that this is a management type activity, this activity have to be executed by the designers to keep themselves on track or working towards a deadline, as it has been statistically proved that people work better when deadlines are imposed.
- Not all activities have succeeding or preceding activities; which is the very nature of design process that some activities are conducted as checks to the design process which might not have an effect on the product by it has its importance to the people and the process involved with the design. Activities such as, Constraining, Decomposing, Resolving, Prioritizing, Testing or Experimenting, and Structuring or Integrating do not have either succeeding or preceding activities; Planning and Scheduling does not have both, succeeding or preceding activities.
- Analyzing has only one output Design Behavior, which can cause a discrepancy in the DAO. The issue on hand is that if this output information were not to occur in a design process or in a design project, and even if one of the 7 input information's related to analyzing occurred, it can be said that the activity "Analyzing" did not occur which would contradict to rules

specified in the DAO and thus breaking the chain of activities related to Analyzing. An in-depth study on this issue can resolve it.

- The phase's classification could not be modeled and tested in Protégé, as explained earlier we state that all the intermediate phases i.e.,

    1. Design Behavior Activities,
    2. Design Complexity Activities,
    3. Design Influencing Activities,
    4. Design Representation Activities,
    5. Design Requirements Activities
    6. And Idea/Concept Generation Activities

    Can contribute to the formation of the 4 phases i.e.,

    1. Planning and Clarifying the task phase,
    2. Conceptual design phase,
    3. Embodiment/System-level design phase
    4. And Detail design phase

This gives us a network of those 6 intermediate phases in the 4 different phase class, forcing us to opt out of this task as it would have caused confusion in the DAO and its representation. Thus future work would be to verify the actual activities and intermediate phases that contribute to the individual phases in the design process.

Scope of the DAO

One of the ways to determine the scope of the DAO is to sketch a list of questions that a information base or a design support system intended to be built on this DAO, should be able to answer, i.e., competency questions. These questions will serve as the litmus test later and also state that these competency questions are just a sketch and do not need to be exhaustive[22].

Several questions were devised and answered to increase the transparency and the understanding of the DAO.

The questions and the answers to those questions were:

What is the domain that the ontology is intended to cover?

The domain intended for the DAO is "Mechanical Engineering Design." To be precise, the DAO is aimed at capturing the information required to represent design processes at a given level of detail such that not only the product's geometry is captured but the technical "know-how" of how the product was designed is captured. It will provide assistance to amateurs in design domain, be it the industry or education to understand and apply concepts related to design and design processes.

For what are we going to use the ontology?

The DAO is used as a database for information repositories to access, query, and retrieve information pertaining to design. It would facilitate the evolution of design software to develop the process aspect of design and to provide more options to the user to not just the capture geometric data or design process data, but to manipulate it based on his discretion. Basically to enhance the design support systems.

For what types of questions the information in the ontology should provide answers?

The DAO is used to provide answers to questions pertaining to, activities executed by design engineers, hierarchy of activities, information available to design engineers, terminologies used by design engineers, capturing information and its presentation, deliverables of every activities, design roadmaps for adaptive and variant design etc.

Who will use and maintain the ontology?

People in education or industry would be the intended audience for this ontology. Since the ontology is predicted to be dynamic and it would be required to be updated often to make it more stable. Although there would be a time where this ontology and a tool integrated to it with customization options; would be made available to design engineers across the globe, and to organizations who intend to use this ontology at the organization level; but initially they would only be provided with a standard template created by an organization that sets international standards for such technologies, such as ISO or NIST etc.

CHAPTER FIVE:

DEMONSTRATION OF DAO

OBJECTIVES:

- Discuss two example studies with respect to the DAO
- Statistics on the usage of the DAO in the example studies
- Specific examples of the example study where the DAO implemented

To support the completeness and correctness of the DAO, two design projects were studied and analyzed for the use of the DAO, with or without its knowledge, but the example studies discussed here are the ones where the teams did not have prior knowledge about the DAO nor of its existence. Example study based analysis is conducted to empirically prove the use of the ontology; with that statement the following observations in the example study will be based on whether an instance of the information entities mentioned in the DAO were used or not, or if the activities defined in the DAO was executed or not. Two example studies are presented in the following sections and the first example study is a industry funded project and the second one is a ME 402 Senior Design (Capstone Design) project. These example studies have provided sufficient data to discuss the pro's and con's of the DAO.

EXAMPLE STUDY 1: EAI TRASH TRUCK DESIGN PROJECT

This project entailed the design of a trash truck from EAI. This project used concepts like, market analysis and task clarification; and along with the implementation of design methods to generate the design of the vehicle; all of these activities were conducted conforming to the requirements. The final deliverables included a detailed report on the impact of the suggested system on the community, infrastructural needs, residual design tasks, 3-D models in SolidWorks, Simulations in CosmosWorks, Bill of materials (BOM), etc.

Example of Information Flows

The information entities are explained using examples from a funded project conducted at Clemson University is provided in the following Table 99.

Table 9: Vocabulary of Information entities and their examples

Information Entity	Example
Client and Design Brief	EAI's briefing to Clemson university student team
Design Requirements	The design must be as low cost as possible
Design Objectives	Process trash and recyclables on one truck
Information sources	Patents, Lead user interview, Internet, Intranet etc.
Past Designs and Past Design Cases	Existing compactors with EAI
Design Information	Process times for recycles
Domain Knowledge	Knowledge about SolidWorks
Repository of design information	Vendors and/or In-house repository (Referred to manufacturers for hydraulic cylinder sizing)
77Problem Structure	To define the structure, the bailer and trash compactor
Degree of accuracy required	The structure must have a tolerance of 0.x inches
Product Architecture and Interactions	The structure sits behind the trash compactor which sits behind the baler
Reasons for fundamental and incidental interactions	The principal reason is for access and weight distribution
Physical phenomena and Theories	Theory of stresses, bending and deflection
Modeling Techniques	Geometrical 3D modeling technique
Design Methodology	Design methodology by Pahl and Beitz
Criteria Map	Criteria - Should allow for one operator Criteria Map - This criteria maps to the design layout of the trash truck
Constraints Hierarchy (Hard constraints and Soft constraints)	Hard constraint - Had to service x number of households; Soft constraint - A shredder had to be installed on the truck; Constraints Hierarchy - The servicing of x number of households influenced the volume of trash the truck could hold
Assumptions	The operators are skilled

Table 9: Vocabulary of Information entities and their examples (Contd.)

Information Entity	Example
Activities Hierarchy	Defining the design space took precedence over division of 3 systems amongst 3 engineers and the required activities to complete the systems individually. The above mentioned things can be classified into the activities known as, Exploring, Planning and Scheduling, and Prioritizing with respect to the DAO
Hierarchy of Goals	Goal of "fit in certain area" lead to "design of structure and bailer" according to that, that in turn lead to the goal of "designing the truck" presents a hierarchy of goals
Design Task Hierarchy	The 3 systems divided amongst the 3 engineers required several design tasks before the systems were completed and each engineer had different priorities for the tasks which provides a basic design task hierarchy like firstly concepts would be generated for the systems and then they would be analyzed
Mapping Knowledge	Knowledge of mapping requirements to functions
Missing Information	Information regarding compaction properties of trash had to be researched as it was not readily available
Conflict Resolution Strategies	Reports, research, data collection and analysis were used to resolve a conflict of "whether to retain the shredder on the truck or remove it?"
Search Strategy	Look at all resources available for material prices for recyclables
Search Results	Material prices for recyclables
Standard Components Hierarchy	Standard cylinders chosen for the trash compaction influenced the Standard C channels for the balers which shows that a hierarchy was present in selecting standard components
Behavior to Design Specification Map	The mapping of wall deflection to manufacturing methods, where wall deflection is termed as a behavior and manufacturing methods belongs to the class of design specifications
Resource Map	Mapping of work, i.e., individuals were assigned to do work in logistics, on particular system components etc.
Tools Map	Geometric models were created in SolidWorks which in turn can be mapped to modeling activity or Meetings were scheduled using email etc.

Table 9: Vocabulary of Information entities and their examples (Contd.)

Information Entity	Example
Information requirements hierarchy	Acquiring information about the design problem took precedence over information about tools, computers or meeting times
Function/Sub-Function Hierarchy	The function is, rapid unloading of materials – the sub-function related to this would be, the trash should eject out of the side
Part/Sub-Part Hierarchy	Cylinder (sub-part) is part of the bailer (Part)
System/Sub-System Hierarchy	The bailing system has the hydraulic system as the sub system
Functional Requirements	Compact PET plastic to a specific density
Functional Means Map	A large bore hydraulic cylinder was the means to achieve the functional requirement of “compacting PET plastic to a specific density”
Function to Physical Hierarchies Map	Can be obtained from the task of “Functional Decomposition”
Abstractions Hierarchy	The design of the baler is at a higher abstract level than the sizing of the baler heads
Appropriate representation of abstractions	Through construction of prototypes
Function to Behavior Map	When you are compacting trash (which is the function), the forces that it exerts (is the behavior) has to be taken into account
Behavior to Structure map	Compacting the trash is going to generate some forces that are going to shear door pins
Function to Solution Principle Map	The number of households must be supported by the volume of the trash chamber
Function to Component Map	The bailer rails should only allow the use of fewer bailer heads
Function to Working Principle Map	Compaction was achieved by Bernoulli's equation
Working Principle to Structures Map	Can be developed from the descriptive sections of the report
Function to Design Parameters Map	Multiple bailing bins (which is the Design Parameter) supports various recyclable materials (is the function)
Design Parameters to Structure Map	Max height of the vehicle limits the height of the structure
Integrating physical building blocks	Eliminate all electrical systems and use only hydraulics systems on truck, to make integration of building blocks or individual modules easier
Design Properties and Relationships	The draft angle of the baler affects the cylinder shape
Methods/Tools for generating ideas	Random idea generation or Brainstorming etc.

Table 9: Vocabulary of Information entities and their examples (Contd.)

Information Entity	Example
Ideas	The ideas generated for the “Ram face” of the compactor
Concepts	The concepts generated for the design of the bailer's cylinder head
Function Modules	The 3 areas the trash, structure and bailer can be termed as 3 function modules
Parts and Systems Interaction	The slide and cylinder head allowed loading of different materials
Embodiments Knowledge	The knowledge of the clearance of bailer doors and the issues related to it were not required until the embodiments phase, which comes after the conceptual phase
Design Space	The possible solutions that could have been developed for the baler’s cylinder head and the problem structure determines the design space for the baler head
Product Configuration	Product configuration were provided in the drawing packages
Combination Knowledge	Combination Rules or Combination Tables that can be combines concepts or ideas for a better concept or idea
Design Geometry	Components like Cam locks, Cable Locks and Shear pins were geometrically defined
Design Behavior	All the components such as Cam locks, Cable Locks and Shear pins must latch
Design Environment	Waste, trash and hazardous materials
Detail Drawings	Detail drawings of all the systems and components would be presented in the drawing packages
Design Specifications	Specifications were provided for parts such as pumps, alternators, cylinders etc.
Assembly Procedure	Assembly procedures for the systems were described in the drawing packages
Hierarchy of design decisions	Design decisions were defined in the minutes of the meeting, a hierarchy could be established based on the definitions
Detail Design	The structure and Bailer plans can be considered as detail design
Set of standard components selected	The standard components selected for this project is specified in the BOM
Criteria for Standardization	Simple criteria for standardization can be “To fit bolt grades and beam specifications.”
Design Alternatives	Alternate designs for the cylinder heads were developed

Table 9: Vocabulary of Information entities and their examples (Contd.)

Information Entity	Example
Analysis Methods/Techniques (Experimental and/or Simulation)	Performed an experimental test on the bailer to measure deflection
Evaluation Methods/Techniques (Experimental and/or Simulation)	FEA evaluation of designs
Object Selected	"Hydraulics" was selected to do the job of compaction
Criteria Used	Some of the criterions used were "Limited noise, reduce number of systems, increase trash compaction capacity."
Appropriate/developed models	Models developed in SolidWorks
Simulation Models	Simulations done using COSMOSWorks and the models developed using this method
Testing Environment	The testing environment was the "Trash Truck," measurements and data was taken on the prototype of the trash truck
Test Results	One of the test was to check the density of the baled PET, The result was 8 pounds per cubic foot
Relaxation of soft constraints	The soft constraint was "A shredder has to be installed on the truck," the elimination of the shredder from the truck can be an example of the relaxation of a soft constraint
Algorithms for activities	Lean manufacturing techniques and methods used to make the process better and minimize activities
Decoupled and/or Coupled Activities	The fitting of the different modules to come up with the final design required coupling and decoupling of activities that they were assigned and a detailed description about the activities that were coupled or decoupled can be found in the project reports
Algorithms and methods for planning	Used email to plan tasks and activities
Algorithms and methods for scheduling	A Gantt chart can be used for scheduling

Activity Details

From the analysis of the EAI project, all the 25 activities from the DAO were executed by the team. One such example of an activity used is *associating* which is considered for further discussion. Associating was executed by this team based on, *The goal* – "*Generate novel or new ideas/concepts through association of ideas/concepts,"* as the goal suggests if the group is aiming at developing new solutions for the issues or problems that arise during the execution of the project.

The input information –

1. *Methods/Tools for generating ideas* which can be methods like Brainstorming, Gallery Method, 656, Morph charts, etc.,

2. *Domain Knowledge* which can be the information pertaining to any domain related to the project. Particularly any information related to design.

The output information –

1. *Ideas* that are conceptualizations of thoughts presented in a crude manner such as rough sketches, figures etc. Ideas can be the initial part which is further developed to generate concepts.

2. *Concepts* that have clear representation through dimensions, descriptions etc. apart from figures and sketches. Concepts can be refined ideas presented in a polished manner.

As discussed in Chapter four, an activity can be composed of any number of the input or output information as defined in the DAO table, in

Table 7and it is not necessary or mandatory for that activity to be composed of all the inputs and outputs specified in the table. The goal for the project group remained the same, which was to "Generate novel or new ideas/concepts through association of ideas/concepts." The information the group had and the tasks they conducted that composed this activity (specifically) was,

For the classification of input information *"Domain Knowledge,"* the group had,

1. Knowledge about existing designs
2. Knowledge of the infrastructure support
3. Knowledge about Idea generating tools and methods

For the classification of output information *"Ideas,"* the group generated several ideas at various stages, for example:

- How to compact trash, for which a major concern/issue was space and the group generated 5 ideas in total, like Rotating cylinder, power screws etc.
- To design a latch for the trash door; the group had 3 ideas, like Shear pins, wedge lock, etc.
- Packaging for the balers and how to arrange them
- Floor plan for the structure etc.

For the classification of output information *"Concepts,"* the group developed a total of 12 concepts during the project; during task clarification phase, a concept for what processes are critical in the truck was generated. Further the identification of three sub systems for further concept generation was considered; the Baler sub-system was started first and 3 concepts were generated for the baler. The Trash Compactor and Structure sub-system were executed simultaneously for which 4 concepts for Trash Compactor and 3 concepts for the Structure were developed. Finally during testing, 1 concept was generated and a final design solution was selected after several testing and analysis iterations. This can be evaluated against a basic chart created by the team in Microsoft Visio to analyze the design process at a particular level of abstraction or detail and this chart is illustrated in Figure 31, and the marked or shaded regions indicate the generation of a concept.

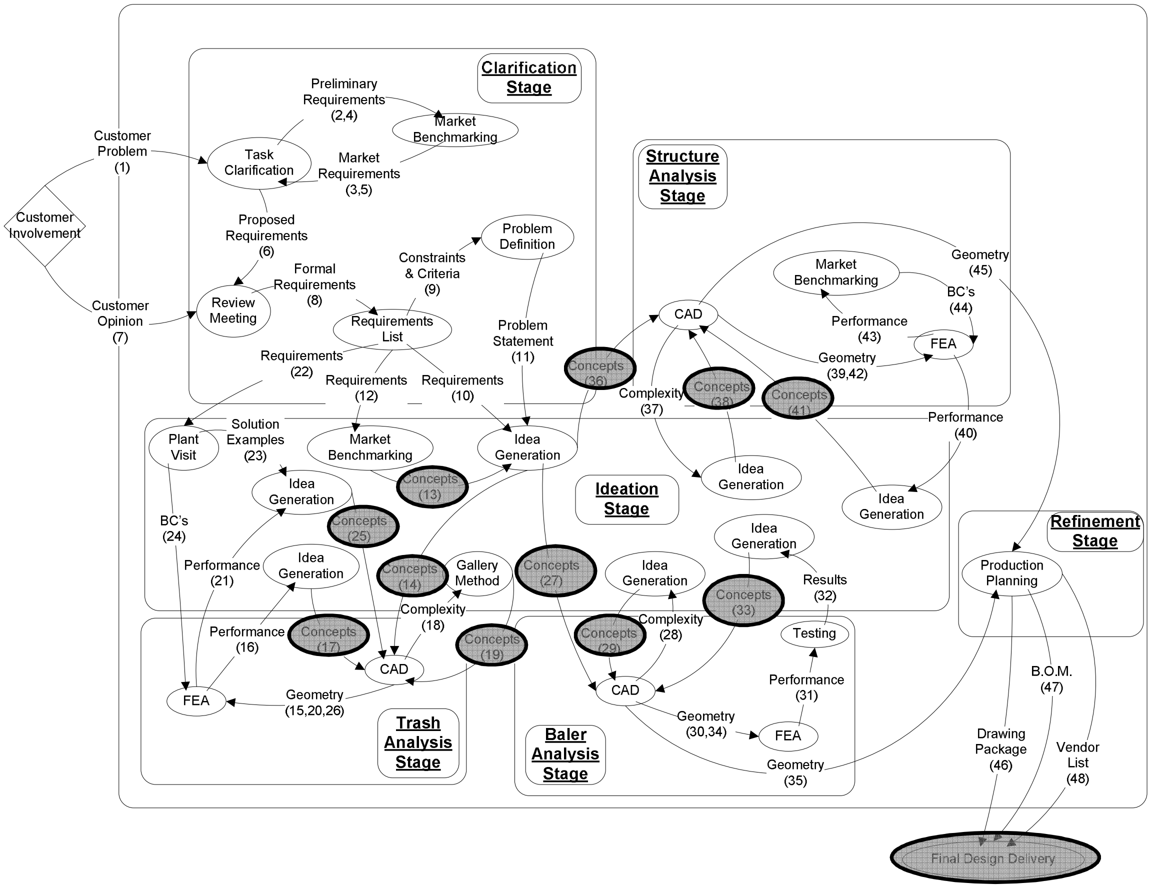


Figure 31: Process model from the EAI group

The example of the "Associating" activity is modeled in Protégé and a screenshot of the graphical visualization of this activity and its instances are presented in the Figure 31. The figure shows one of the several different views available in Protégé to represent the individual instances that occur in an activity. The instances are specified in the individuals tab and the grey shaded areas indicate the individuals in the graphical visualization tool in Protégé.

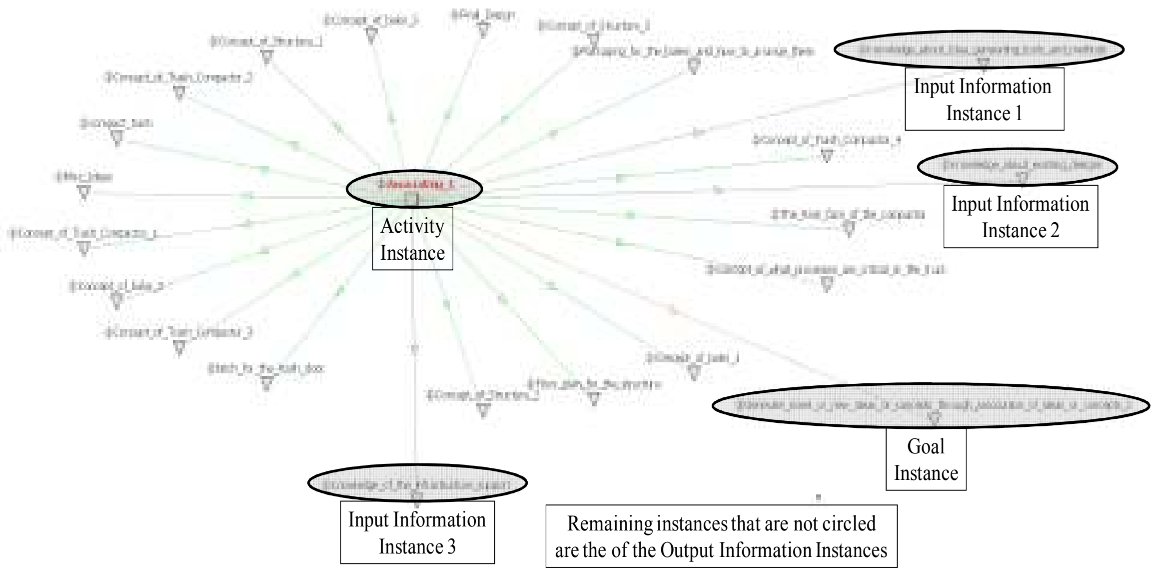


Figure 32: Snapshot of instance visualization in Protégé using Jambalaya

The associating activity described was just one of the 25 activities executed in this project. The remaining activities that were executed could not be presented in detail, because a definitive process model could not be created for this example study as the information captured by the project group which is illustrated as a process model shown in Figure 31 is inadequate. The major problems encountered that prohibit the transformation of the team's process model to the DAO process model are,

- Lack of data, there were no concise and clear project reports. The availability of such reports would have enabled the capture of the design process information. There were just a list of deliverables such as status reports, emails, 3D models, simulation models and miscellaneous documents.
- Lack of information representation in the process model developed, the process model provided for analysis and transformation only captures information in small numbers and explicit to a certain task. The model's definition of an activity is equal to a task that is carried out in an activity in the DAO. The only match that could be made from the

process model provided to the list in DAO is the Idea Generation, Problem Definition and Task Clarification. The others such as FEA, CAD, Market Benchmarking etc., are a part of the activities mentioned in the DAO.

- The process data is not available on a computational background compatible to the DAO, i.e., DSM format or in Protégé. Generally design processes are complex and are not clear in pictorial/graphical representations, thus they are expected to be available in a DSM format where the data can be analyzed and a simpler process model can be created based on a higher level of abstraction. The availability of this projects process in a DSM would have provided enough information to develop a process model using the DAO. Similar to the DSM, it would have been a lot easier to develop a process model if the team had recorded its proceedings and occurrences using Protégé.

Instance Details

Due to the unavailability of the process model for further analysis, another approach of recording the number of instances was conducted using the available data. There were approximately around 174 instances for the use of information flows for this project, for example, for the information entity, Design Requirements, there were 9 total instances recorded, the specific instance classification being:

9 (Total) = 2 (Preliminary Requirements) + 2 (Market Requirements) + 1 (Proposed Requirements) + 1 (Formal Requirements) + 3 (Requirements)

Thus the list of total number of instances was developed based on these recordings. It can be said that the ontology captures design process related information in explicit packets which can be accessed and understood with ease. Also the DAO enables analysis and graphical visualization of the design process if the information captured were to be modeled into Protégé using the existing Protégé model of the DAO. The details of the 174 number of instances can be found in Table 1010, where 174 is the summation of all the number of instances in column named, Number of instances.

Table 10: The number of instances for the Info Flows used in the EAI project

Information Entity	Number of instances
Client and Design Brief	3
Design Requirements	9
Design Objectives	1
Information sources	4
Past Designs and Past Design Cases	7
Design Information	3
Domain Knowledge	6
Repository of design information	3
Problem Structure	1
Degree of accuracy required	1
Product Architecture and Interactions	1
Reasons for fundamental and incidental interactions	1
Physical phenomena and Theories	1
Modeling Techniques	4
Design Methodology	0
Criteria Map	2
Constraints Hierarchy (Hard constraints and Soft constraints)	2
Assumptions	1
Activities Hierarchy	1
Hierarchy of Goals	1
Design Task Hierarchy	1
Mapping Knowledge	1
Missing Information	1
Conflict Resolution Strategies	1
Search Strategy	1
Search Results	1
Standard Components Hierarchy	1
Behavior to Design Specification Map	1
Resource Map	1
Tools Map	1
Information requirements hierarchy	1
Function/Sub-Function Hierarchy	1
Part/Sub-Part Hierarchy	1
System/Sub-System Hierarchy	1
Functional Requirements	1
Functional Means Map	1

Table 10: The number of instances for the Information flows used in the EAI project (Contd.)

Information Entity	Number of instances
Function to Physical Hierarchies Map	0
Abstractions Hierarchy	1
Appropriate representation of abstractions	1
Function to Behavior Map	1
Behavior to Structure map	1
Function to Solution Principle Map	1
Function to Component Map	1
Function to Working Principle Map	1
Working Principle to Structures Map	0
Function to Design Parameters Map	1
Design Parameters to Structure Map	1
Integrating physical building blocks	1
Design Properties and Relationships	1
Methods/Tools for generating ideas	7
Ideas	20
Concepts	12
Function Modules	1
Parts and Systems Interaction	1
Embodiments Knowledge	1
Design Space	0
Product Configuration	1
Combination Knowledge	0
Design Geometry	10
Design Behavior	1
Design Environment	1
Detail Drawings	1
Design Specifications	6
Assembly Procedure	1
Hierarchy of design decisions	1
Detail Design	1
Set of standard components selected	1
Criteria for Standardization	1
Design Alternatives	3
Analysis Methods/Techniques (Experimental and/or Simulation)	4
Evaluation Methods/Techniques (Experimental and/or Simulation)	4
Object Selected	1
Criteria Used	1
Appropriate/developed models	7
Simulation Models	1
Testing Environment	1
Test Results	1
Relaxation of soft constraints	1

Table 10: The number of instances for the Information flows used in the EAI project (Contd.)

Algorithms for activities	1
Decoupled and/or Coupled Activities	1
Algorithms and methods for planning	1
Algorithms and methods for scheduling	1
Total:	174

This example used 77 information flows out of the 83 information flows; which is approximately 94% usage (93.90% to be precise.) This shows that the DAO is extensive and complete in capturing the information flows pertaining to a design project and yet has some information flows that were never used which indirectly show the expanded boundaries of the DAO. There were also no recordings of the DAO unable to capture any of the information related to this project. This example study has proved to be favorable to this version of the DAO.

EXAMPLE STUDY 2: DESIGN OF TAIL-LIGHT FIXTURE FOR THE BMW X5 SAV (ME402)

Design Project Context

Several design project reports were studied and analyzed for the use of the information entities mentioned in the DAO and if it was executed in the project it was considered to be an instance of a particular information. The design projects considered as example studies for this research are the senior design projects such as ME 402 senior design project, capstone projects etc. This particular example study is related to a ME 402 senior design project executed by students of Clemson University to design tail light fixtures for the BMW X5 SAV. This design project was conducted y several student groups and each of the groups data or information that was presented in the form of deliverables was analyzed for the use of the DAO and the statistics of the use of the DAO is presented. The design problem given to the student groups can be explained as follows, "The installation of the BMW X5 Sport Activity Vehicle's outer tail light assembly requires accurate locating of the light with respect to the vehicle's body and proper mating of the light assembly with its corresponding body mating points. The current process for installing the tail light assembly involves installing the tail light by hand, lowering the tail hatch for use as a reference, raising the tail hatch, and tightening the assembly's 3 holding nuts to the specified torque. The current process has been unable to provide an accurate, consistent means of installing the assembly due to human error as well as uncertainty in the location of the hatch. These inconsistencies are resulting in an unacceptable number of vehicles being sent to rework, which costs the customer time in assembly and increases the overall assembly cost.The new fixture must maintain the correct gap between the outer tail light assembly and the body surface, maintain flushness of the light surface and the body surface, and not cause damage to either the outer tail light assembly or the painted vehicle body. Other key issues are ease of use and simplicity of design."

The problem statement for this project is: *Design a fixture and an appropriate process for its use to consistently install the left and right outer tail lights to the location specifications determined by the manufacturer*.

It has to be noted that in this example we are not looking at the exact number of instances executed in the project but we are going to just check the percentage usage of the DAO. The scale considered for this analysis is university level projects or senior class level projects because at this level, the projects reports though are extensive they are not as detailed as at the company level. This project is considered to prove that if there are sufficient entries for the information flows in the DAO at this level, then it can be stated that the level of detail added by the DAO to create process models is sufficient to cover industry level projects with substantial detail. Thus we are just looking at a hit or miss situation here and in the following tables if there is an entry from the project to a particular information flow, it will be discussed or reported as “1” and if there are not instances recorded for a particular information flow, it will have a “-“ next to that information flow.

Use of DAO for Modeling Design of X5 SAV

According to the analysis of the design projects, four project reports were verified for the use of the DAO and two of the recordings taken from teams B and D which were randomly selected to be discussed for each of the findings are presented in the following table. The table presents the details of every instance recorded for teams B and D (see Table 1111), also for teams B and D, the details are quiet similar except for some information flows. Apart from these teams the data for teams A and C are also presented but the specifics are omitted as they are very similar to the information flows presented in Table 11. The “1’s” that are not explained for team D are similar to team B’s explanation, simply represented as 1.

Table 11: Information and their usage by Teams B and D

Information Entity	TEAM B	TEAM D
Client and Design Brief	1(Briefing from professors and BMW)	1
Design Requirements	1(Fast and easy to use)	1
Design Objectives	1(Install both taillights at the same time)	1
Information sources	1(Internet)	1
Past Designs and Past Design Cases	1(Information about BMW's existing fixture)	1
Design Information	1(Can be information from design requirements or design specs etc. like Information regarding assembly procedures)	1
Domain Knowledge	1(Knowledge about taillight fixtures)	1
Repository of design information	1(Documents and handbooks from client and university resources)	1
Problem Structure	1(Problem Definition and Specification chart)	1
Degree of accuracy required	1(Not to cause damage to either the car body or the taillight)	1
Product Architecture and Interactions	1(Functional Decomposition)	1

Table 11: Information and their usage by Teams B and D (Contd.)

Information Entity	TEAM B	TEAM D
Reasons for fundamental and incidental interactions	1(Functional Decomposition)	1
Physical phenomena and Theories	-	1(Design theory)
Modeling Techniques	1(CAD modeling techniques to develop solid models)	1
Design Methodology	1(Design methodology by Pahl and Beitz)	1
Criteria Map	1(Criteria definition in the report)	1
Constraints Hierarchy (Hard constraints and Soft constraints)	1(Constraints definition in the report)	1
Assumptions	-	-
Activities Hierarchy	-	-
Hierarchy of Goals	1 (Objectives tree was developed)	-
Design Task Hierarchy	1(Gantt Chart showed the distribution of design tasks)	1
Mapping Knowledge	1(Mapping functions to behavior)	1
Missing Information	1(Information about quality and finish of the final design)	1
Conflict Resolution Strategies	-	-
Search Strategy	-	-
Search Results	-	-
Standard Components Hierarchy	-	-
Behavior to Design Specification Map	-	-
Resource Map	-	-
Tools Map	-	-
Information requirements hierarchy	-	-
Function/Sub-Function Hierarchy	1(Can be derived from the Functional Decomposition)	1
Part/Sub-Part Hierarchy	1(Concept design's interaction with the Frame, Inward positioning system, Vertical positioning system, Scissors Mechanism)	1
System/Sub-System Hierarchy	1(Frame Design that had interactions with several sub systems like hatch and tailgate)	1
Functional Requirements	1(Can be derived from the Functional Decomposition)	1
Functional Means Map	1(Can be derived from the Morph Chart)	1
Function to Physical Hierarchies Map	1(Can be derived from the Functional Decomposition)	1

Table 11: Information and their usage by Teams B and D (Contd.)

Information Entity	TEAM B	TEAM D
Abstractions Hierarchy	-	1(Giving importance to abstracting information from defined variables, like understanding the design process took precedence over developing design decision matrices based on the given criteria and constraints)
Appropriate representation of abstractions	-	1(Developing the design decision matrix)
Function to Behavior Map	1(Can be derived from the Functional Means Tree)	1
Behavior to Structure map	-	-
Function to Solution Principle Map	1(Can be derived from the Functional Means Tree)	1
Function to Component Map	1(Can be derived from the Morph Chart)	1
Function to Working Principle Map	1(Can be derived from the Functional Means Tree)	1
Working Principle to Structures Map	-	1(Rotating "L" design, which is the Working Principle; is with the horizontal toggle clamp and alignment clamp, which are the structures)
Function to Design Parameters Map	-	1(Function of installing the tail light with the "L" design related to appropriate alignment in x, y and z directions)
Design Parameters to Structure Map	-	-
Integrating physical building blocks	-	-

Table 11: Information and their usage by Teams B and D (Contd.)

Information Entity	TEAM B	TEAM D
Design Properties and Relationships	1(Description in the report)	1
Methods/Tools for generating ideas	1(Brainstorming and Morph charts)	1
Ideas	1(Ideas for the frame design)	1
Concepts	1(Concepts for the frame)	1
Function Modules	-	-
Parts and Systems Interaction	1(Description in the report)	1
Embodiments Knowledge	1(Knowledge about providing details for the design and its evaluation)	1
Design Space	-	1(As specified by the parameters used in the design decision matrix)
Product Configuration	1(Description in the report)	1
Combination Knowledge	-	-
Design Geometry	1(Design geometry from solid models)	1
Design Behavior	1(Description in the report)	1
Design Environment	1(Available from BMW's briefing)	1
Detail Drawings	1(3D models)	1
Design Specifications	1(Description in the report)	1
Assembly Procedure	1(Description in the report)	1
Hierarchy of design decisions	-	1(Design decision Matrix was created)
Detail Design	1(Description in the report)	1
Set of standard components selected	-	-
Criteria for Standardization	-	-
Design Alternatives	-	1(Hanging design concept)
Analysis Methods/Techniques (Experimental and/or Simulation)	1(FEA analysis)	1
Evaluation Methods/Techniques (Experimental and/or Simulation)	1(Evaluation from client and prototyping)	1(Time trial analysis)
Object Selected	1 (Scissors design selected)	1
Criteria Used	1(Taper, parallelism, and symmetry of design)	1
Appropriate/developed models	1(3D models developed)	1
Simulation Models	1 (Prototypes created)	1
Testing Environment	1(BMW X5 tail light area)	1
Test Results	1(Fit and form discussed in the report)	1

Table 11: Information and their usage by Teams B and D (Contd.)

Information Entity	TEAM B	TEAM D
Relaxation of soft constraints	1(Square tubing does not bend as well as the round tubing, so right angle welded corners were proposed. Shortly after looking at the computer model, it was realized that the bottom corners of the door-jam on the BIW are rounded and would not allow a square corners of the full width fixture to sit in position. The frame design was changed again to include 45 degree angles to join the vertical and horizontal components of the frame)	1(The finish can be sub-par, like scratching of paint may occur which is not a major concern)
Algorithms for activities	1(Information from the Gantt Chart)	1
Decoupled and/or Coupled Activities	-	-
Algorithms and methods for planning	1(Information from the Gantt Chart)	1
Algorithms and methods for scheduling	1(Information from the Gantt Chart)	1
Total:	**56**	**63**

Process Model Representation

This research talks about developing process models using the DAO, Example study 2 is selected to represent some sample process models developed using the DAO. The following figures (see Figure 33 and Figure 34) illustrate the process captured for Teams B and D, based on the DAO. The two process models are almost identical; they have all the 25 activities described in the DAO in their process models, except for a fact that four activities were executed at different times (ABS, SYN, RES and CON)

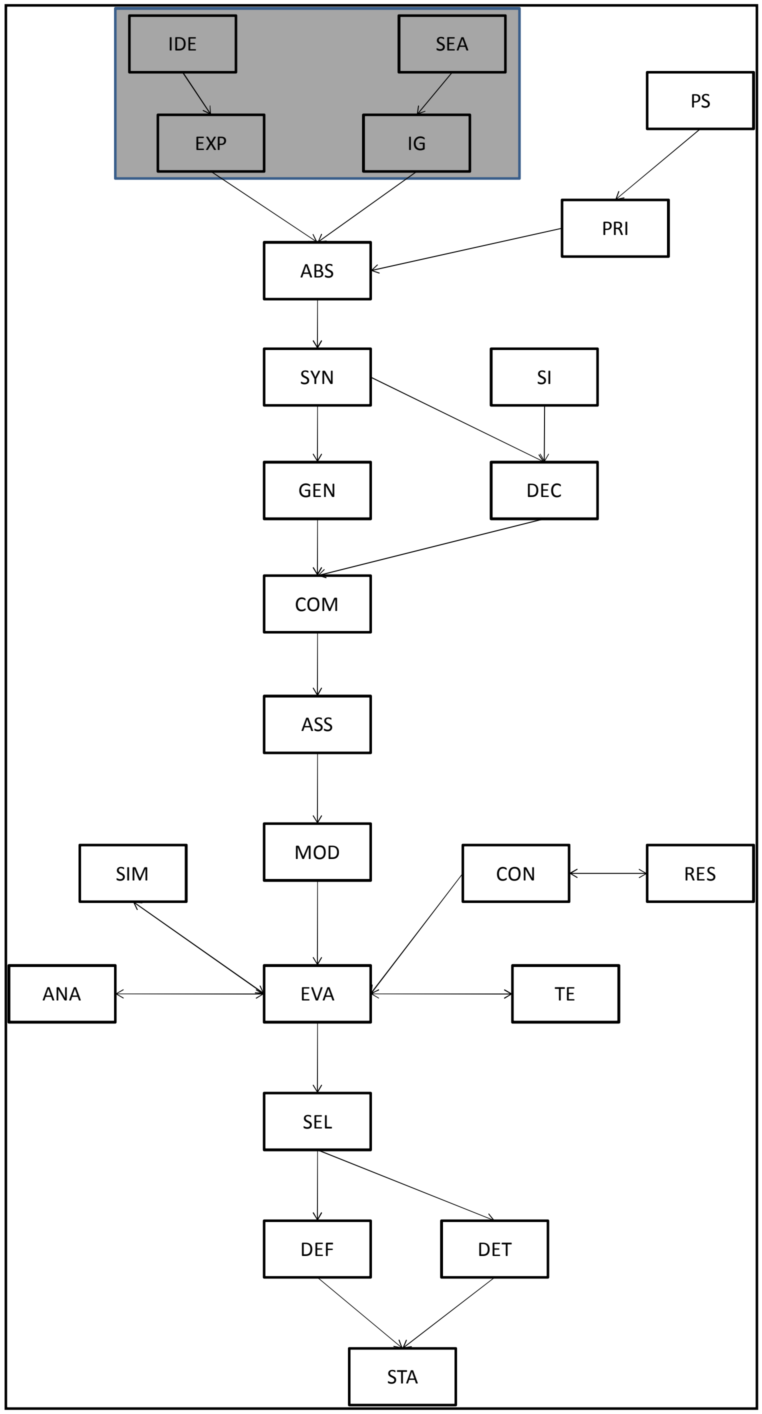


Figure 33: Process model of Team B using the DAO

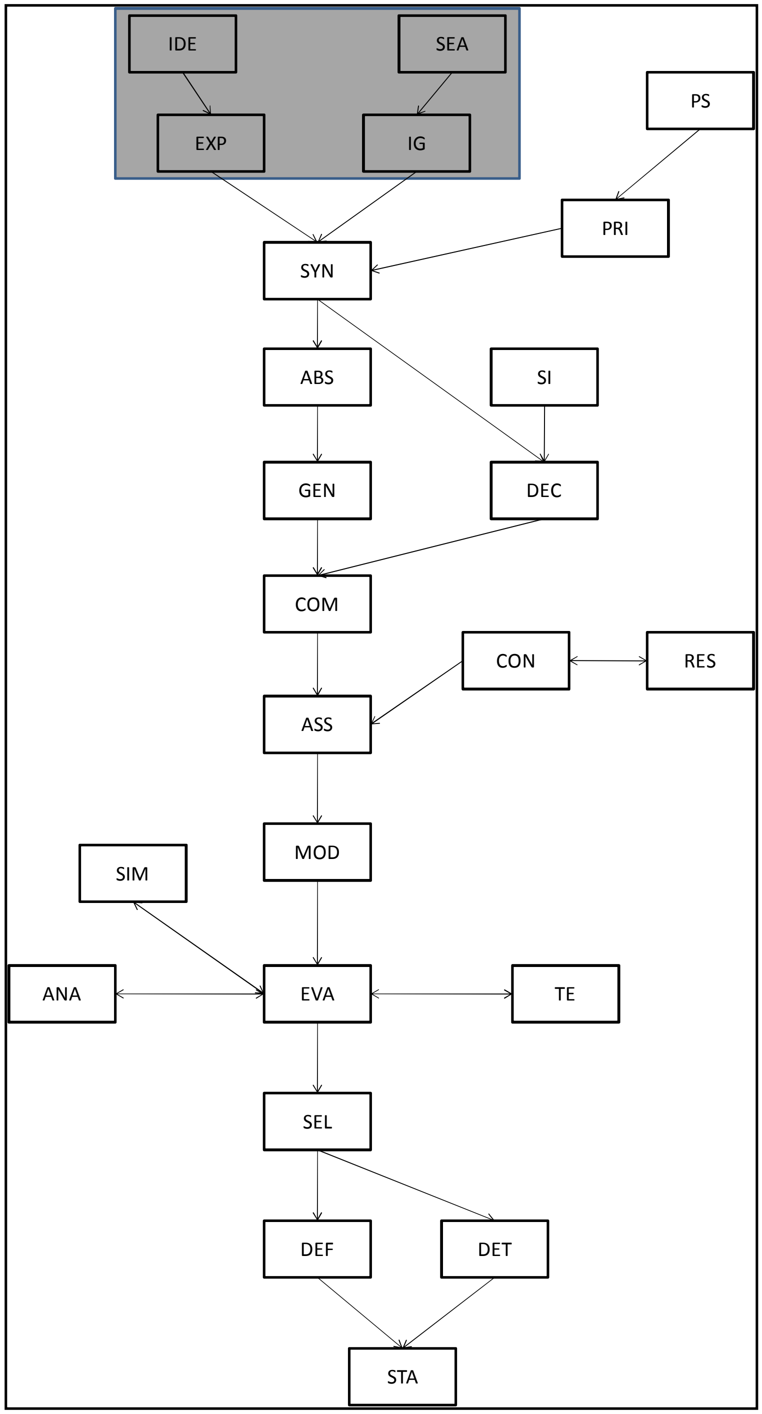


Figure 34: Process model of Team D using the DAO

Since Figure 33 and Figure 34 have the shaded region identical, we shall consider it for future discussions. From the shaded region in Figure 33 and Figure 34, we can see that four activities (IDE, EXP, IG and SEA) are involved in a simultaneous execution. The exact map of these activities along with its graphical representation from the DAO is illustrated in Figure 35. It can be seen from the figure that there are some redundant information entities, those information flows that lead to nowhere; but we can also see those information entities that take the design to the next step or the next activity. It can be seen that the activities were executed using information for an external source, but with the representation from the DAO, and as shown below, the information entity "Repository of design information" is transferred from Searching activity (SEA) to Information Gathering (IG) activity. Thus the entire process could be modeled using the DAO. The details of the information entities involved in Figure 35 are illustrated in Table 12.

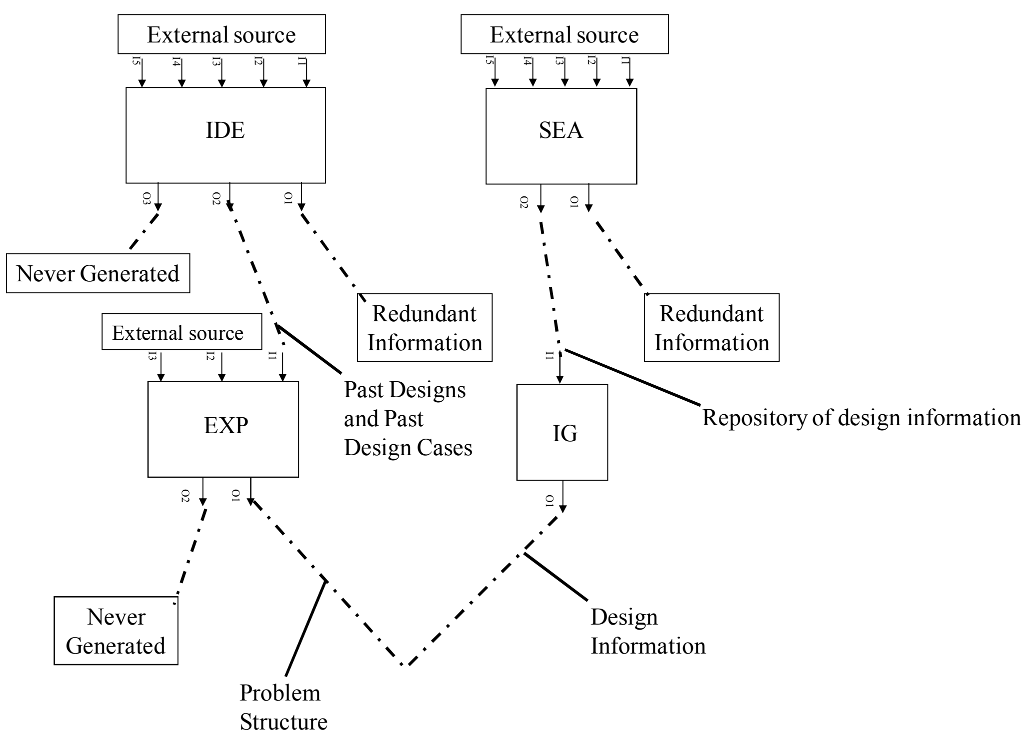


Figure 35: Process model, exploded view (based on Figure 33 and Figure 34)

Table 12: Details of the information entities used or generated from Figure 35

Information Type	Instance Details
Past Designs and Past Design Cases	Information about BMW's existing fixture
Repository of design information	Documents and handbooks; from client and university resources
Problem Structure	Problem Definition and Specification chart
Design Information	From design requirements
Design Methodology	Redundant Information
Domain Knowledge	Never Generated
Search Results	Redundant Information
Others: • Client and design brief • Design space • Search strategy • Design properties and relationships etc.	External Source

CHAPTER SUMMARY

It is evident from the details provided here that there is substantial use of the information flows from the DAO. The entire design process was captured using the DAO and no information pertaining to the design process was left uncaptured and yet there has been only, less than 77% usage of the DAO, showing that the DAO can handle design projects with varying magnitude. It should also be noted that all the activities in the ontology were executed by each of the teams without exceptions. The execution of all the activities is recorded and is termed possible although there were certain information entities that were not utilized or executed because as we have defined in the Protégé modeling section, that it is not necessary nor mandatory for all the information associated with an activity to be executed, if atleast one or more information entity (per input or output) associated with an activity is executed, then the activity is considered to be executed. Thus we can state that there was 100% usage of the activities from the DAO.

The two example studies provide decisive inputs to this version of the DAO. It can be seen that both the example studies were completely covered by the DAO and there was no information that could not have been captured by this DAO, and yet the DAO's bounds were not exceeded. The exact number of instances of Example 2 was omitted, as we were able to obtain a process model for this example, and the demonstrations show that it is enough to argue about the completeness and correctness of the DAO with respect to the example studies conducted. The facts and observations on the example studies presented in this research were recorded after these projects were executed; thus future recommendations would be to provide this DAO to a team of designers for their reference and study; before they handle a project and then put on projects. With the knowledge of the DAO to capture the specifics of design process, the team must be assessed periodically for the use of the DAO and to record its pro's and con's. This would enable a holistic approach towards the DAO, and would certify this version of the DAO from every angle possible. Also, the example studies presented here was not implemented in the Protégé model, the next step would be to implement the findings and instances of all these example studies in the Protégé model of the DAO and to let DL do the analysis of information flows and hierarchy. Also a design repository can be created to store, retrieve and reuse these process models for future research. Furthermore this repository can be built into a database and can be tied to various software modules for quicker access and use.

CHAPTER SIX:

CONCLUSION

The approach used to develop the DAO was the bottom-up approach, where we specified the activities and the information flows related to those activities and then built taxonomies, rules, intermediate phases and phases; rather than a top-down approach where we had to specify the information entities alone and perform a one to one mapping of the information flows to develop activities that were based on applying the information flows to design projects and developing a list of activities based on the observations from each of those projects; which would have led us to finalizing an activities list based on the number of times a particular activity was executed in that project and across projects. Then these activities had to be described with the list of attributes specified earlier. To roughly estimate the number of total possible activities if the top-down approach was carried out, it would be a combination of 82 information flows which is the true minimum for this problem, i.e., "82! X 82!" would be the number of possible activities where the value of "$82! = 4.75364334 \times 10^{122}$" and "82! X 82!" would be a number that is twice of "$4.75364334 \times 10^{122}$" and this would be impossible to perform and analyze. Thus the bottom-up approach was selected, where we could use the 82 information flows and the 25 activities to form the DAO.

There were two iterations of the ontology development lifecycle conducted; and there were some major changes in the ontology that was selected initially i.e., the Sim and Duffy ontology, when compared to the DAO. The DAO after two iterations looks comprehensive and sustained the two analysis example studies and provided the much needed starting points for future work. The DAO is also explicitly discussed in Chapter four, and very specific examples of the information entities are provided in Chapter five. There are several drawbacks observed in the analysis and evaluation performed using the example studies in Chapter five.

- The number of instances recorded and the percentage usage of the DAO was considered to be a metric to assess the demonstration and exercising of the DAO which raises a lot of doubt and question.
- There are several gray areas even after iteration two. The DAO must be further refined to precisely describe what the information entities can have as instances. The activities have been defined in detail but the information entities lack a clear demarcation of what instances can be recorded under it.
- The example studies exercising the DAO could not provide conclusive results on the process models and the design process. Thus more example and case studies must be exercised to capture the design information pertaining to that project as well as recording the activities as and when they are executed (real-time capture)
- The example studies were only visually evaluated and the computational model was not utilized to analyze. The specific instances must be recorded using the computational model and Protégé and DL should be used to analyze and evaluate the DAO.

The answers to the research questions that were formulated in this research are,

Research Question 1

What are the basic set of activities and information entities required to represent the engineering design process?

Answer to Research Question 1

From chapter three we can derive that 82 information flows and the 25 activities developed using the information flows were sufficient to represent design processes. The first iteration of the ontology development lifecycle was discussed in this chapter, and this chapter provided the details on how the selected formalism was modified based on certain established analysis and evaluation techniques. This Chapter also illustrated the transformation of the baseline formalism to generate the DAO.

Research Question 2

How can the information entities and activities that form the ontology be represented in a computer interpretable form?

Answer to Research Question 2

Chapter four provided the answer to this research question as we illustrated that Protégé can be used as the computational background for the DAO. We also showed that protégé along with DL provided tools for analysis, creation and visual representation of design processes, manipulation of design processes, and information exchange through design processes. The models created using protégé can also be uploaded on the internet for the users around the world to modify, edit, and update the created process models.

Research Question 3

How can the DAO be empirically analyzed?

Answer to Research Question 3

Chapter five illustrates the demonstration of the DAO in two example studies that provide some basic explanation of how the DAO can be implemented in design projects and how it can be analyzed for its usage. It also demonstrates instance logging and logical debugging of the information flows to map the occurrences to the information flows. Process models was created for 2 teams from example study 2 as we proved that a process model could not be created with the inadequate information captured by a process model created by another technique in Example Study 1.

The example studies executed to prove the effectiveness of the DAO shows that all the activities mentioned in the DAO were used by all the teams and were used in all the projects that were evaluated. It was also evident that all the information flows in the DAO were not used, thus proving that the DAO is diverse and captures all the relevant information pertaining to a design process. I can also be deduced that the DAO obtained after the second iteration of the ontology development life-cycle has proved to be orthogonal at the information level and overlapping at the activity level to create unbroken process models; and the availability of this DAO in Protégé allows flexibility to model design processes by just adding the exact occurrences from design projects.

RESEARCH CONTRIBUTIONS

This research has provided a survey of published literature about capturing the information related to "Engineering Design Processes" and information about "Design process models." An activity model is developed based on the standards defined by IDEF 0 [24] and Browning [10]. This activity model represents the typical engineering design activity with the flow of information within that activity. This activity also describes the essential attributes that control the uniqueness of an activity. Several additional attributes are recognized to improve the activity definition.

DSM application to information modeling and also as an analysis and evaluation technique/tool is discussed in this research. The activity based DSM and parameter based DSM is applied in this research with the use of analysis options like "Partitioning" and "Tearing." This research also uses the DSM to represent the complex visualization of design processes based on the DAO.

The DAO is completely defined in terms of, attributes, taxonomy, hierarchy, relationships; and its performance and analysis is discussed in detail with examples from industry sponsored projects, and senior design and capstone projects. This thesis also presents the graphical representation for each of the activities, along with its description; and every information flow associated with this ontology is explained with an example. This ontology could be made available to students learning design and thus introducing them to a standard format at an early stage in their design career will help this domain to capture information in the required format for future proceedings.

The DAO is furthermore implemented in Protégé, thus providing a quick access and quick-start with a solid computational background for future work. Protégé is also used in conjunction with DL to prove the hierarchy and completeness of the DAO. The hierarchy is also presented graphically using OWLViz, OntoViz and Jambalaya, which are plug-ins available in Protégé.

FUTURE WORK

This research is intended to aid the development of intelligent design support systems or enhance existing design support systems. It is also observed that there is some overlap between the DAO's vocabulary and the six-sigma (DMAIC) standards model. This ontology can also be used as the standard to automate the capture of information related to design processes; which would be the ultimate impact of this research. Also, by automating the information capture based on the designer's actions, the design support system can provide a summary at the end of every activity for the designer to review and recommend changes for similar processes in the future. The next step towards the realization of this goal would be to,

1. Develop the Protégé model

Develop the Protégé model and to utilize the tools and options available in Protégé to develop good Ontological models. The areas where the Protégé model can be developed are; the definition of information input using the "Forms" tab; mapping of the occurrences to the respective individuals created for the project using the "Individuals" tab; developing good graphical representation of the ontology using the "ONTOViz, OWLViz, and Jambalaya" tabs, especially for the Jambalaya tab which can also show the exact number of instances or individuals that participated in an activity; and to develop certain "Semantic Web Rule Language" rules to generate explicit answers for explicit queries using the "SWRL" tab. The ontology must be well equipped to handle different query questions and should be able to provide only the answer the users are looking for. For example, when a user queries to find out only the redundant information flows, the ontology must be able to provide only those information flows that have individuals created for them, but which were never fed into an activity. Thus the future work in Protégé involves development of rules, constraints and forms to render the DAO operational and well specified.

It is also suggested to implement the DAO in another computational background to check the compatibility and performance. The other software platform suggested is the Web Service Modeling Ontology (WSMO) studio with design process modeling as the environment. This would enable to discover, compose, mediate, and execute newer methods of capturing information pertaining to design processes [16]. This would also provide the availability of DAO on two computational backgrounds for future research. Furthermore the Protégé model of the DAO can also be used to record the exact number of occurrences in future design projects and can be further analyzed to see that the ontology does not disintegrate or decompose as shown in Chapter 3 for the Sim and Duffy Ontology.

2. Demonstrate the DAO in more design projects

The DAO must be used more often in several industry sponsored projects for its effective analysis and should be refined based on the major observations. The project teams must implement the DAO in the projects right from start as to capture every minute detail regarding the process. Thus the project teams must be exposed to the DAO before the project starts. Care should also be taken not to become too specific as it will demotivate the people involved and can compromise the project. The DAO also specifies specificity, but at a fairly abstract level. More example studies implementing the DAO would strengthen the baseline ontology i.e., the DAO.

3. *Develop metrics to evaluate the process models developed*

Metrics for evaluating the complexity of design processes created using the DAO must be developed. The process models developed based on these computational representations and project implementations must be verified for their complexity based on certain standard metrics. Several metrics and measures have been developed in this regard, such as,

- Design Performance [37]
- Measures of Engineering Design Complexity [56]
- Algorithms to assess design complexity [55]
- Measurement of a Design, Structural and Functional Complexity [6]

It is also to be analyzed if the number of instances and the percentage usage of the DAO could be used as a metric to evaluate the DAO and the process models developed.

Also workflow analysis and axioms for analyzing design processes can be implemented to evaluate the DAO and its computational models. The implementation must include and answer questions such as,

- What are the rules for evaluating the goodness of a process?
- Are the information flows that are generated being used?

APPENDIX

DSM's of the four cases to represent the design activities and its information flows

The details of these matrices are provided to illustrate that the classification provided by Sim and Duffy did not hold true for any case, thus proving the Sim and Duffy ontology insuffient and inefficient. The first figure of every case shows how the information was modeled in the DSM, where 1 indicated a relationship between the entities and "null" indicated no relationship. It can be seen from the DSM's that the DSM was modeled based on the description and classification provided by Sim and Duffy. The relationships between activities and the information flows were taken from the three classifications provided by Sim and Duffy, i.e., the DDA, DEA, and DMA. The second figure in every case corresponds to the analyzed matrix where partitioning, tearing and clustering were applied. This DSM shows the disintegration of the classification provided by Sim and Duffy and also shows several new groupings that were developed (indicated in the blue shaded regions.)

Case 1: Partial Information Flow & Feedback

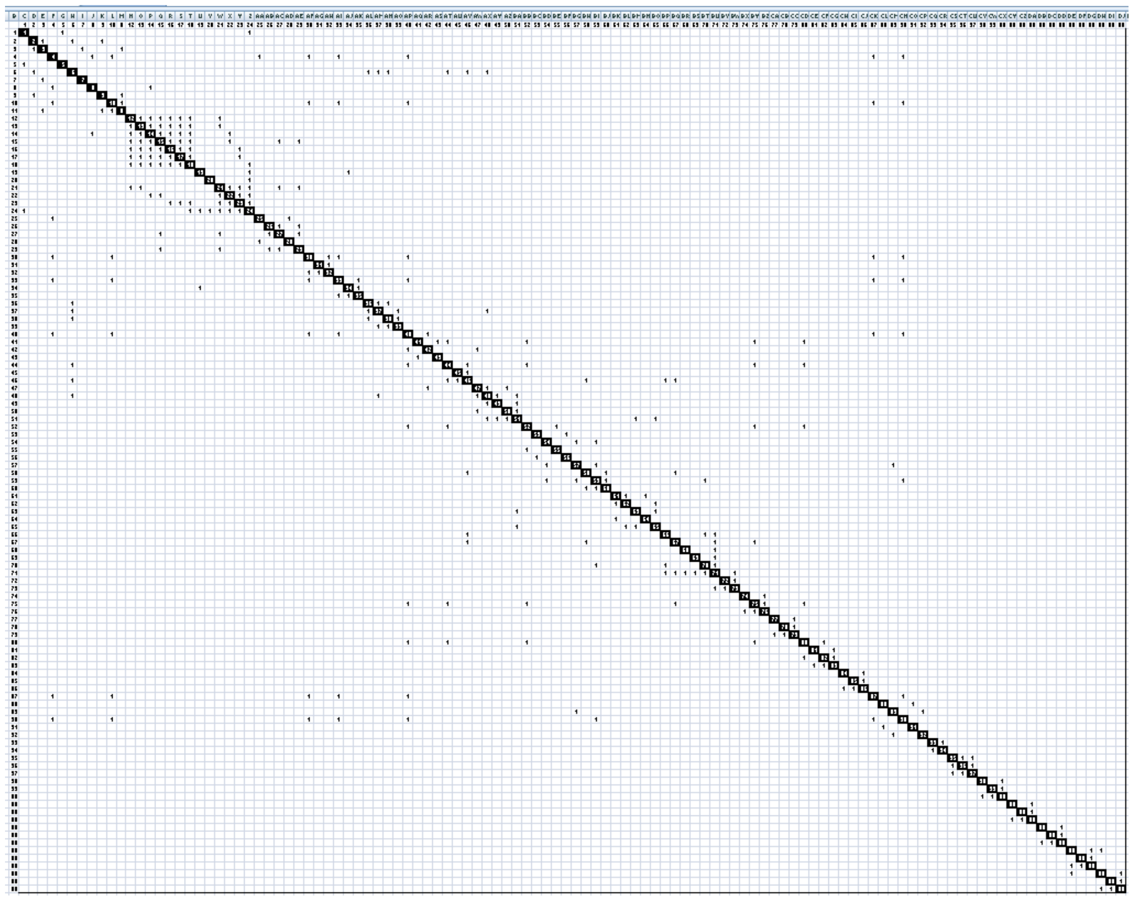


Figure A- 1: Case 1 DSM model

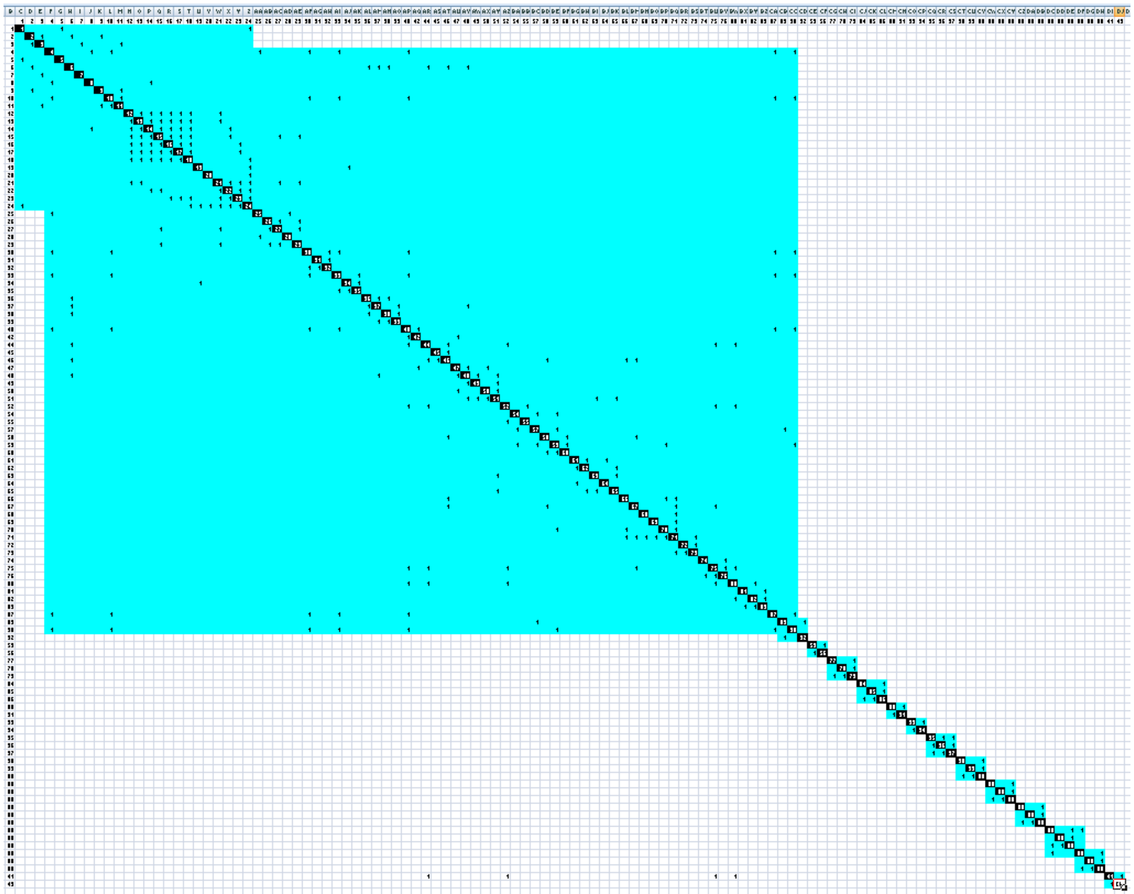


Figure A- 2: Case 1 analyzed DSM model

Case 2: Complete Information Flow & Feedback

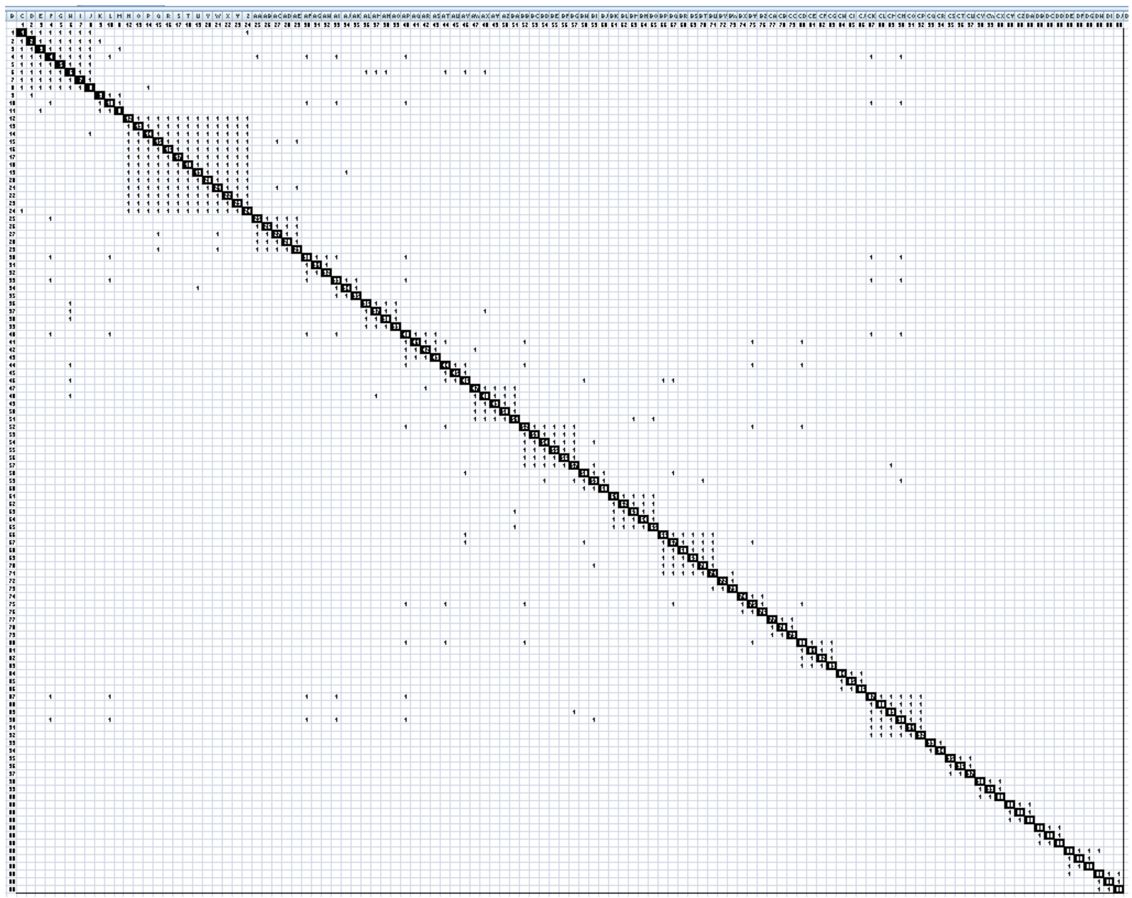


Figure A- 3: Case 2 DSM model

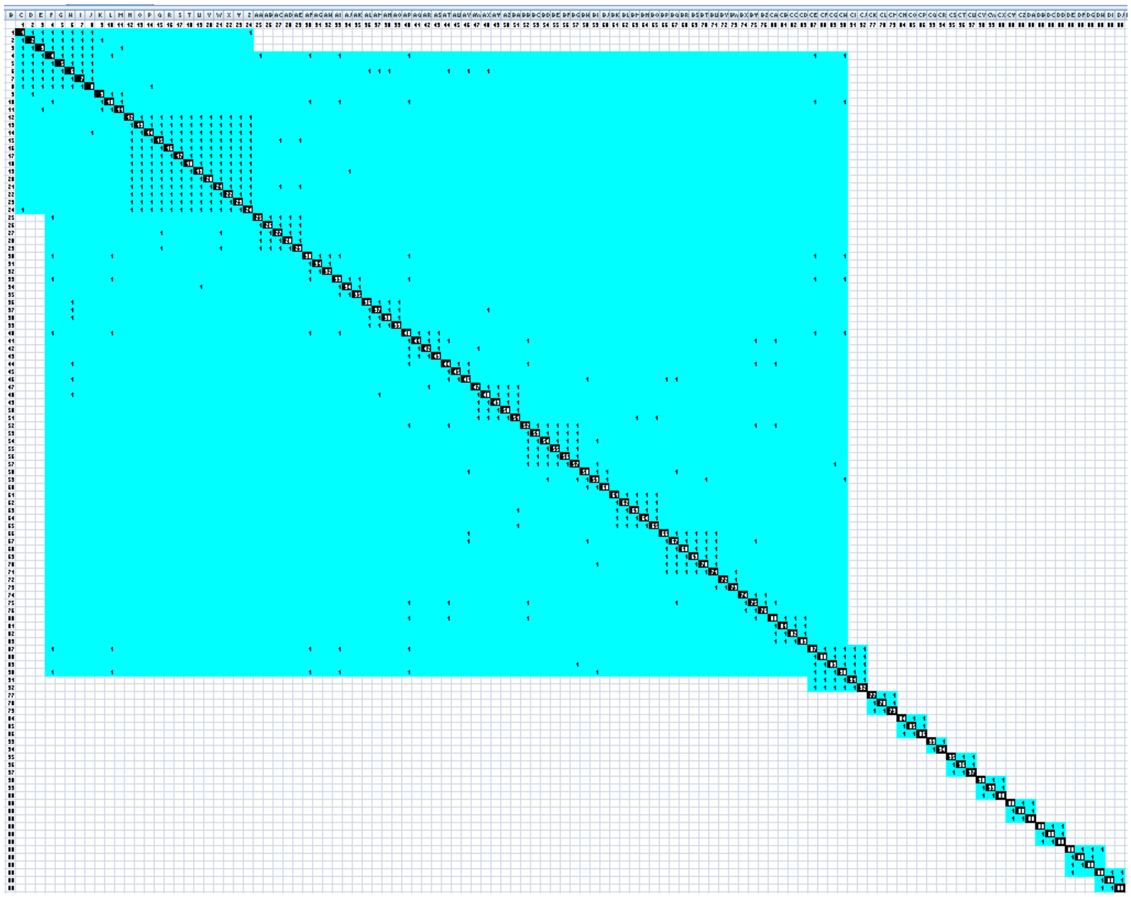


Figure A- 4: Case 2 analyzed DSM model

Case 3: Partial Information Flow & No Feedback

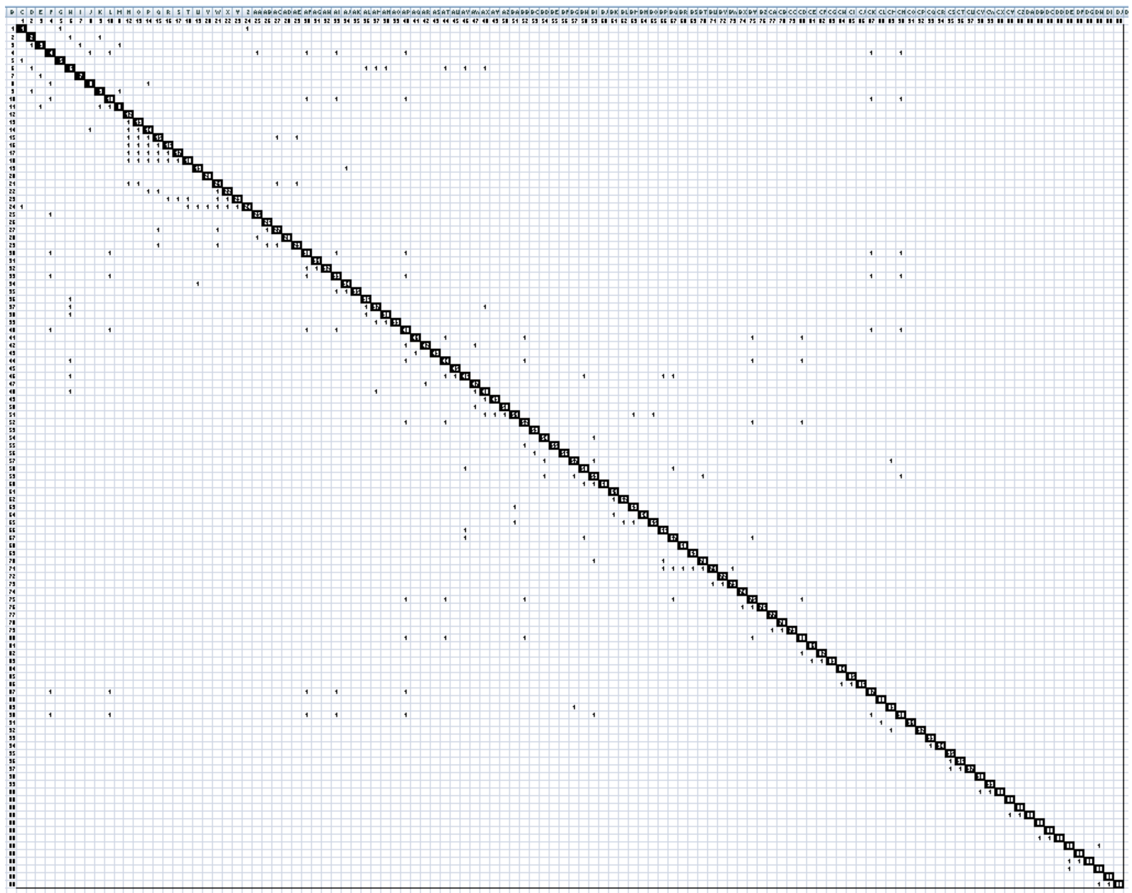


Figure A- 5: Case 3 DSM model

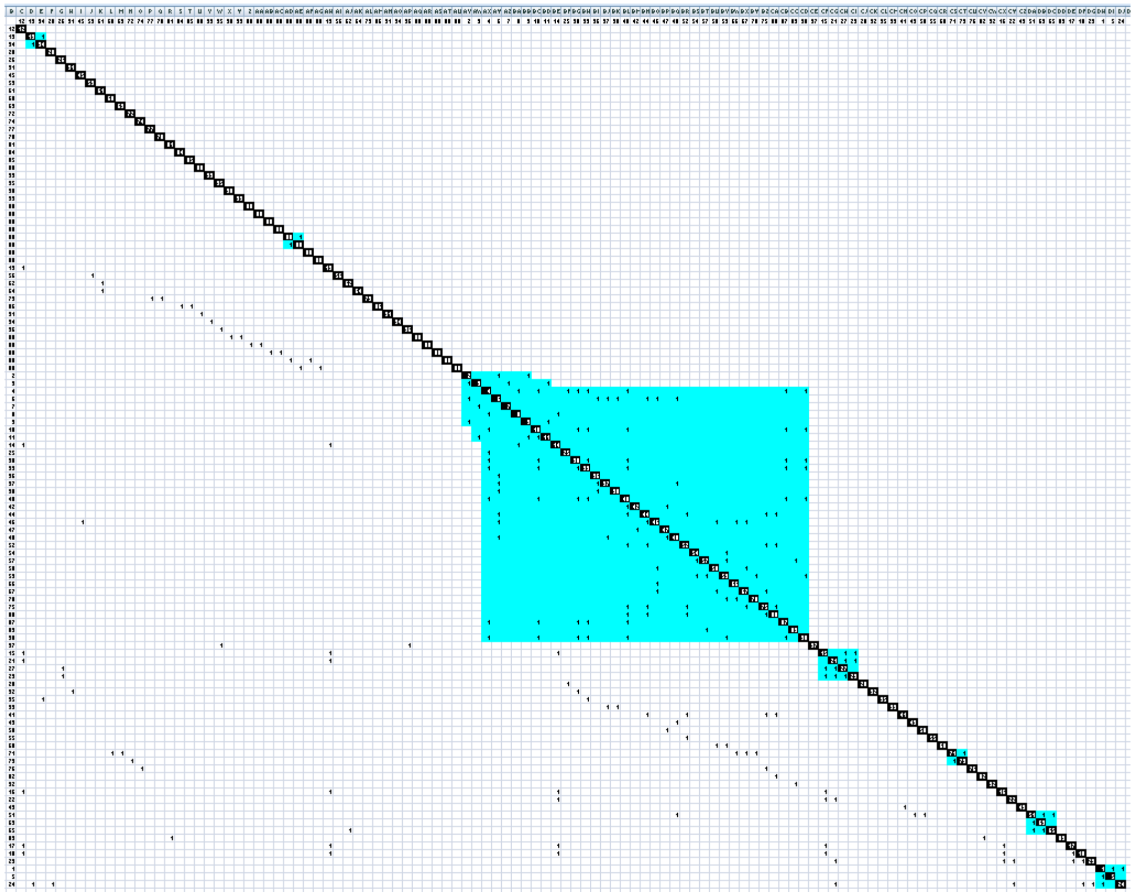


Figure A- 6: Case 3 analyzed DSM model

Case 4: Complete Information Flow & No Feedback

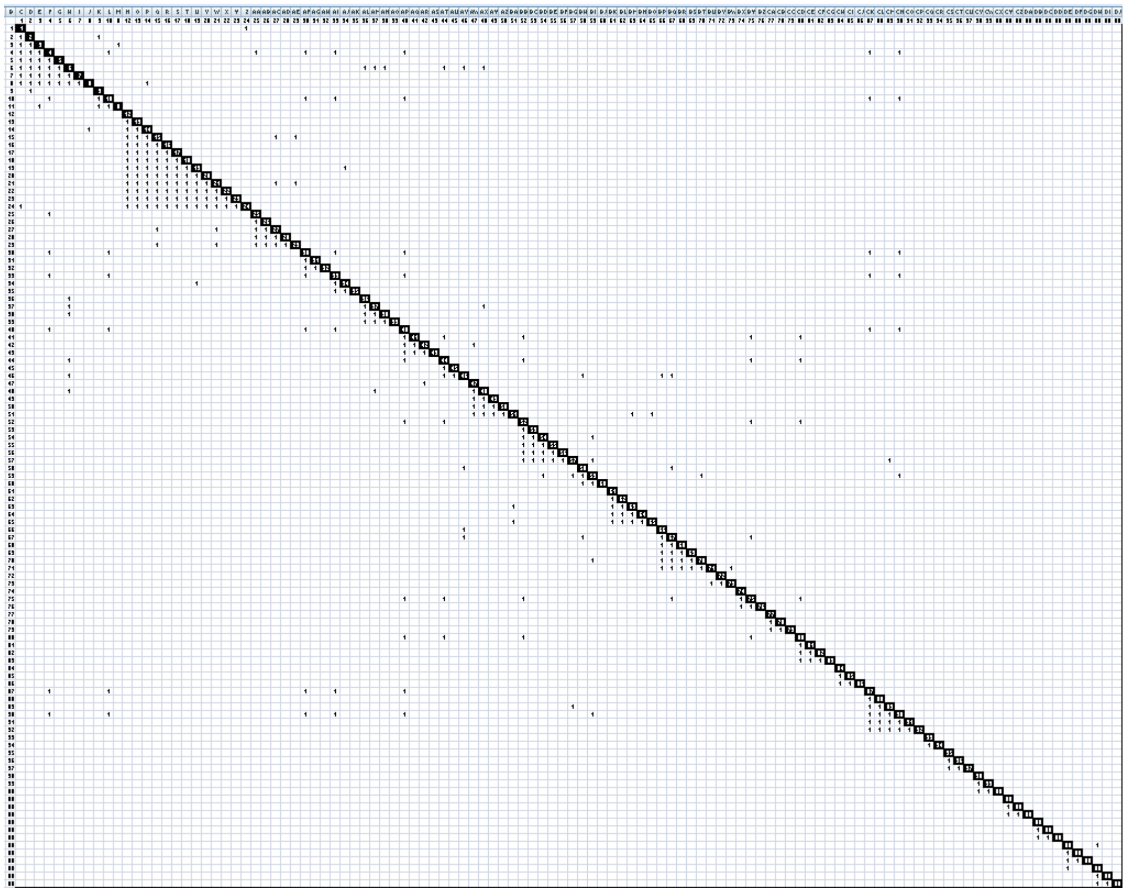


Figure A- 7: Case 4 DSM model

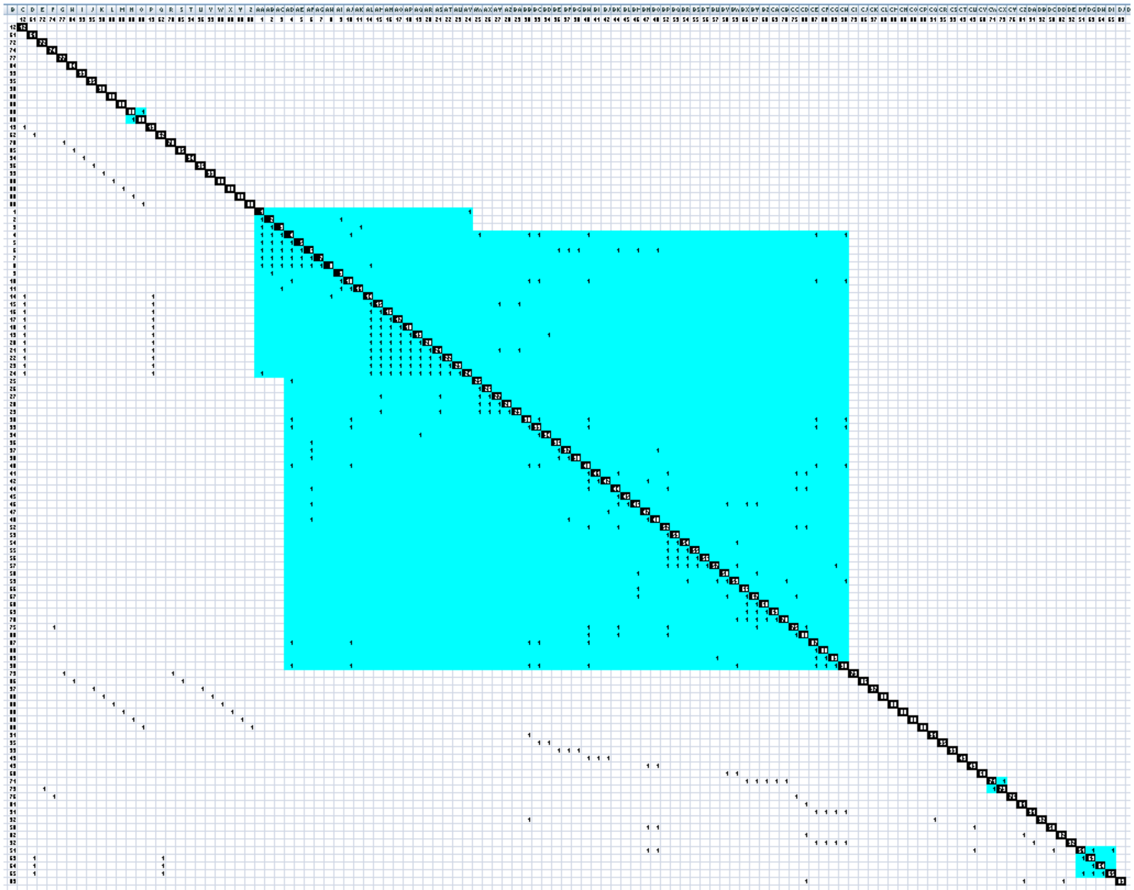


Figure A- 8: Case 4 analyzed DSM model

LIST OF REFERENCES

1. Ahmed, S., S. Kim, and K.M. Wallace, *A Methodology for Creating Ontologies for Engineering Design.* Journal Of Computing Information Science And Engineering, 2007. **Vol. 7**: p. 132 - 140.
2. Ameri, F.a.D.D. *An Upper Ontology for Manufacturing Service Description*. in *Proceedings of ASME International Design Engineering Technical Conference,*. 2006. Philadelphia, Pennsylvania USA.
3. Baxter, D., J. Gao, K. Case, J. Harding, B. Young, S. Cochrane, and S. Dani, *An Engineering Design Knowledge Reuse Methodology Using Process Modelling.* Research in Engineering Design, 2007. **18**(1): p. 37-48.
4. Bohm, M. and R. Stone. *Representing Functionality to Support Reuse: Conceptual and Supporting Functions*. in *24th ASME Computers and Information in Engineering Conference (CIE) 2004.* 2004. Salt Lake City, Utah USA.
5. Bohm, M., R. Stone, and S. Szykman. *Enhancing Virtual Product Representations for Advanced Design Repository Systems*. in *23rd ASME Computers and Information in Engineering Conference (CIE)*. 2003. Chicago, Illinois USA.
6. Braha, D. and O. Maimon, *The Measurement of a Design Structural and Functional Complexity.* IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS-PART A: SYSTEMS AND HUMANS, 1998. **VOL. 28**(NO. 4).
7. Brooke, D.M., A. de Pennington, and M.S. Bloor, *An Ontology for Engineering Analyses.* Engineering with Computers, 1995. **11**(1): p. 36-45.
8. Browning, T.R., *Process Integration Using the Design Structure Matrix.* Systems Engineering, 2002. **Vol. 5**(No. 3): p. pp. 180-193.
9. Browning, T.R., F. E., and N. H., *Key Concepts in Modelling Product Development Processes.* Systems Engineering, 2006. **Vol. 9**(No. 2): p. pp. 104-128.
10. Browning, T.R. and R.V. Ramasesh, *A Survey of Activity Network-Based Process Models for Managing Product Development Projects.* Production And Operations Management, 2007. **16**(2): p. 217-240.
11. Chan, C.S., *Cognitive processes in architectural design problem solving.* Design Studies, 1990. **11**: p. 60–80.
12. Chandrasekaran, B. and J.R. Josephson, *Function in Device Representation.* Engineering with Computers, 2000. **16**(3): p. 162-177.
13. Choi, Y., K. Kim, and C. Kim, *A design chain collaboration framework using reference models.* International Journal for Advanced Manufacturing Technology, 2005. **26**: p. 183-190.
14. Clarkson, P.J. and J.R. Hamilton, *Signposting, a parameter-driven task-based model of the design process.* Research in Engineering Design, 2000. **12**(1): p. 18-38.
15. Crabtree, R.A., M.S. Fox, and N.K. Baid, *Case studies of co-ordination activities and problems in collaborative design.* Research in Engineering Design, 1997. **9**: p. 70–84.
16. Dimitrov, M., A. Simov, S. Stein, and M. Konstantinov, *A BMPO based semantic business process modeling environment.* 2008.
17. Eppinger, S.D., *Model-based Approaches to Managing Concurrent Engineering.* Journal of Engineering Design, 1991. **2**: p. 283-290.

18. Gennaria, J.H., M.A. Musenb, R.W. Fergersonb, W.E. Grossod, M. Crub!ezyb, H. Erikssonc, N.F. Noyb, and S.W. Tub, *The evolution of Protege: an environment for knowledge-based systems development.* International Journal of Human-Computer Studies, 2003. **58**: p. 89-123.
19. Gero, J.S. and U. Kannengiesser, *A function–behavior–structure ontology of processes.* Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 2007 **21**: p. 379–391.
20. Grosse, I.R., J.M. Milton–Benoit, and J.C. Wileden, *Ontologies for Supporting Engineering Analysis Models.* Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 2005. **19**(1): p. 1-18.
21. Gruber, T.R., *Towards Principles for the Design of Ontologies Used for Knowledge Sharing*, in *Knowledge, Systems Laboratory, Stanford University*. 1993: Palo Alto, CA.
22. Grüninger, M. and M.S. Fox. *Methodology for the design and evaluation of ontologies*. in *Workshop on Basic Ontological Issues in Knowledge Sharing*. 1995. Montreal.
23. Hubka, V., *Principles of engineering design*. 1982: Butterworth Scientific, London.
24. Integrated Computer-Aided Manufacturing, *IDEF 0* 1993.
25. Johannson, O. and P. Krus. *Configurable Design Matrices for Systems Engineering Applications*. in *26th ASME Computers and Information in Engineering (CIE) Conference*. 2006. Philadelphia, Pennsylvania USA.
26. Kamal, S.Z., H.M. Karandikar, F. Mistree, and D. Muster, *Knowledge Representation for Discipline-Independent Decision Making, in Expert Systems*, in *Computer-Aided Design*. 1987, J. Gero, Editor, Elsevier Science Publishers B.V.: Amsterdam. p. 289-321.
27. Kopena, J.B. and W.C. Regli. *Design Repositories On The Semantic Web With Description-Logic Enabled Services*. in *Proceedings of VLDB Semantic Web and Databases Workshop*. 2003. Humboldt-Universität, Berlin, Germany.
28. Kumar, P. and G.M. Mocko. *Modeling and Analysis Of Engineering Design Activities Ontology Using the Design Structure Matrix*. in *27th ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. 2007. Las Vegas, Nevada, USA.
29. Lin, J., M.S. Fox, and T. Bilgic, *A Requirement Ontology for Engineering Design.* Concurrent Engineering, 1996. **4**(3): p. 279.
30. Malone, T.W., K. Crowston, J. Lee, B.T. Pentland, C. Dellarocas, G. Wyner, J. Quimby, C.S. Osborn, A. Bernstein, G. Herman, M. Klein, and E. O'Donnell, *Tools for inventing organizations: Towards a handbook of organizational processes.* Management Science, 1999. **Vol. 45**(No. 3): p. pp.425-443.
31. Mocko, G., R. Malak, C.J.J. Paredis, and R.S. Peak. *A Knowledge Repository for Behavioral Models in Engineering Design*. in *24th ASME Computers and Information in Engineering Conference (CIE)*. 2004. Salt Lake City, Utah USA.
32. Mocko, G.M., D.W. Rosen, and F. Mistree. *A critical analysis of description logics for engineering information management*. in *26th ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. 2006. Philadelphia, Pennsylvania, USA.
33. National Institute of Standards and Technology, *STEP: The Grand Experience: Special Publication 939*. 1999: Gaithersburg, Maryland USA.
34. National Institute of Standards and Technology, *Process Specification Language (PSL)*. 2005.
35. National Institute of Standards and Technology (NIST), *Process Specification Language (PSL)*. 2005.

36. Nonaka, I., *A dynamic theory of organizational knowledge creation.* Organizational Science, 1994. **5**(1): p. 14-28.
37. O'Donnell, F.J. and A.H.B. Duffy, *Design Performance*. 2005: Springer-Verlag, London.
38. Pahl, G. and W. Beitz, *Engineering Design: A Systematic Approach* 2nd edition ed. 1996: Springer, New York.
39. Panchal, J.H., H.J. Choi, J.K. Allen, D.L. McDowell, and F. Mistree. *Designing Design Processes for Integrated Materials and Products Realization: A Multifunctional Energetic Structural Materials Example*. in *32nd ASME Design Automation Conference (DAC)*. 2006. Philadelphia, Pennsylvania USA.
40. Panchal, J.H., M.G. Fernández, C.J.J. Paredis, J.K. Allen, and F. Mistree. *Designing Design Processes in Product Lifecycle Management: Research Issues and Strategies*. in *Design Engineering Technical Conferences and Computer and Information in Engineering Conference. 2004.* 2004. Salt Lake City, Utah USA.
41. Park, H. and M.R. Cutosky, *Framework for modeling dependencies in collaborative engineering processes.* Research in Engineering Design, 1999. **11**(2): p. 84-102.
42. Pentland, B.T., *Grammatical Models of Organizational Processes.* Organizational Science, 1995. **Vol. 6**(No.5): p. pp.541-556.
43. Pinto, H.S. and J.P. Martins, *Ontologies: How can They be Built?* Knowledge and Information Systems, 2004. **6**(4): p. 441-464.
44. Pugh, S., *Total design: integrated methods for successful product engineering*. 1991: Addison Wesley, Wokingham.
45. Regli, W.C. *National Design Repository*. 2006 [cited Accessed on January 28, 2006]; Available from: http://edge.cs.drexel.edu/repository/frameset.html.
46. Romero, F., P. Company, M.-J. Agost , and C. Vila, *Activity modelling in a collaborative ceramic tile design chain: an enhanced IDEF0 approach.* Research in Engineering Design, 2008. **19**: p. 1-20.
47. Sim, S.K. and A.H.B. Duffy, *Towards an ontology of generic engineering design activities.* Research in Engineering Design, 2003. **Vol. 14**(No. 4): p. pp. 200-223.
48. Sobek II, D.K. (2004) *System-Level Design: A Missing Link?* **Volume**,
49. Stahovich, T.F., R. Davis, and H. Shrobe. *An Ontology of Mechanical Devices*. in *AAAI-93, Working Notes, Reasoning About Function*. 1993.
50. Stauffer, L., D.G. Ullman, and T.G. Dietterich. *Protocol analysis of mechanical engineering design*. in *Proceedings of the 1987 international conference on engineering design*. 1987. Boston.
51. Stauffer, L.A. and D.G. Ullman, *A comparison of the results of empirical studies into the mechanical design process.* Design Studies, 1988. **9**((2)).
52. Steward, D.V., *The Design Structure System: A Method for Managing the Design of Complex Systems.* IEEE Transactions in Engineering Management, 1981. **Vol. 28** (No. 3): p. pp. 71-84.
53. Steward, D.V., *Systems Analysis and Management; Structure, Strategy and Design, .* 1981: Petrocelli Books, Inc., New York.
54. Suh, N.P., *The principle of design*. 1990: Oxford University Press, New York.
55. Summers, J.D. and F. Ameri. *An Algorithm For Assessing Design Complexity Through A Connectivity View*. in *Proceedings of the TMCE 2008*. 2008. Izmir, Turkey: Edited by I. Horváth and Z. Rusák.
56. Summers, J.D. and J.J. Shah. *Developing Measures of Complexity for Engineering Design*. in *ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 2003. Chicago, Illinois USA.

57. Szykman, S., S.J. Fenves, W. Keirouz, and S.B. Shooter, *A foundation for interoperability in next generation product development systems.* Computer-Aided Design, 2001. **33**: p. 545 to 559.
58. Szykman, S., J. Racz, R.D. Sriram, and C. Bochenek, *Web-based System for Design Artifact Modeling*. 1998.
59. Takeda, H., S. Hamada, T. Tomiyama, and H.A. Yoshikawa. *Cognitive approach to the analysis of design process*. in *Rinderle JR (ed) Design theory and methodology, September, ASME*. 1990. Chicago.
60. Ullman, D.G., *The Mechanical Design Process*. 1992: McGraw Hill, New York.
61. Ullman, D.G., *Toward the ideal mechanical engineering design support system.* Research in Engineering Design, 2002. **13**: p. 55-64.
62. Ullman, D.G., T.G. Dietterich, and L.A. Stauffer, *A Model of the Mechanical Design Process Based on Empirical Data.* AI EDAM, 1988. **2(1)**: p. 33-52.
63. Ulrich, K.T. and S.D. Eppinger, *Product design and development*. 1995: McGraw-Hill Inc., New York. .
64. Ulrich, K.T. and S.D. Eppinger, *Product design and development*. 1999: McGraw-Hill Inc., New York. .
65. Visser, W., *Designers activities examined at three levels: organisation, strategies and problem-solving processes.* Knowledge-based Systems, 1992. **5**: p. 92-104.
66. Whitten, J.L., L.D. Bentley, and K.C. Dittman, *Systems Analysis and Design Methods*. 6th ed. 2004: Irwin/McGraw-Hill, Boston, MA.
67. Wiest, J.D. and F.K. Levy, *A Management Guide to Pert/Cpm: With Gert/Pdm/Dcpm and Other Networks*. 1977: Prentice-Hall, Englewood Cliffs, NJ.
68. Witherell, P., S. Krishnamurty, and I. Grosse. *Ontologies for Supporting Engineering Design Optimization*. in *26th Computers and Information in Engineering Conference (CIE)*. 2006. Philadelphia, Pennsylvania, USA.