Knowledge-based Methods for Evaluation of Engineering Changes

by

Chandresh Rajnikant Mehta

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in The University of Michigan
2010

Doctoral Committee:

Adjunct Professor Debasish Dutta, Co-Chair
Senior Research Fellow Lalit Patil, Co-Chair
Professor Dragomir Radkov Radev
Associate Professor Kazuhiro Saitou

To all my teachers

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the support and encouragement of several individuals.

I am grateful to my co-advisor Prof. Debasish Dutta for his guidance, personal and financial support, constructive criticism and encouragement throughout this dissertation. I would like to express my deep gratitude to my co-advisor Dr. Lalit Patil for his enormous guidance and support. He has provided continuous and timely feedback on all aspects of this dissertation. Numerous discussions with him has helped me to gain confidence and overcome obstacles in this research. Through my co-advisors, I have gained the valuable skills of a matured thinker and an independent researcher.

I am thankful to my committee members Prof. Kazuhiro Saitou and Prof. Dragomir Radev for their insights, comments and suggestions. Prof. Saitou's suggestions where particularly helpful in creating example change knowledge-base. Prof. Radev's insights on the topic of Data Mining were helpful in gaining a better understanding of the research problem.

For my financial support, I am very thankful to National Science Foundation, the Product Lifecycle Management Alliance and the Department of Mechanical Engineering at University of Michigan.

Thanks to all the former and current members of PLM Alliance. This research was inspired from the earlier work of Nikhil Joshi. I would like to thank him for his insights and suggestions on conducting research in this field. Il Yeo and Seung-Cheol Yang are good friends who have been available in variety of ways ranging from critiquing my work and sharing of research ideas to providing personal support. Farhad Ameri, Qi Gao, Bryan Yamasaki, Sisi Xuanyuan, Minh Nguyen and Kameshwari Danthurthi contributed in maintaining a friendly and supportive research environment.

My experience as a Graduate Student Instructor of ME-250 course in the Department of Mechanical Engineering has proved to be immensely rewarding. It has helped me to develop and refine my teaching skills as well as gain some new insights in the fields of Product Development and Manufacturing. I would like to thank all the fellow GSIs, faculty instructors, workshop supervisors, students of ME-250 course during various semesters and the staff at the Center for Research on Learning and Teaching for making this a truly wonderful experience.

I am indebted to Daniel Goldstein at the counseling center of University of Michigan for his counseling and support through the difficult times.

Thanks to all my friends and acquaintances in Ann Arbor for a wonderful and memorable time. I was very fortunate to have Rahul Rattan, Sanjoy Khan, Hari Vemuri, Sridhar Valluri, Ramamurthy Vaidyanathan and Hitesh Gupta as my house mates. I would like to thank Siddharth Athreya, Gauri Khanna, Neha Kaul, Varun Gudiseva, Bindi Dharia, Nidhi Jaiswal, Sunny Surlaker, Satish Dandhapani, Solomon Pulapkura, Rachna Lal, Katie Hanlon, Sunil Tripathi, Amit Goje, Ashish Agarwal, Ketan Jajal, Minal Jajal, Amit Kaushik, Sudhanshu Srivastava, Sunil Tahilramani, Kasak Tahilramani, Anurag Tripathi and Trushal Chokshi for providing a friendly and supportive environment. I want to also thank Milind Modi, Parimal Bhatt, Piyush Anam and Vaishali Anam for providing an environment that made me feel at home. I have truly enjoyed my time on racquetball and squash courts with several people including Prof. John Hart. Thanks to my good friend Medha Dharne for all her support and encouragement over past several years.

I owe thanks most of all to my parents for their enormous love and encouragement. Their patience and support during the course of my doctoral studies is exemplary. Thanks to my sisters, brothers-in-law and parents-in-law for their love, support and encouragement.

I want to express my deep gratitude to my wife Jigna. I do not think that I would had been able to complete this dissertation without her love, support and encouragement. I feel very lucky, when I think of all the happiness and joy that she brings into each day of my life.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

Knowledge-based Methods for Evaluation of Engineering Changes

by
Chandresh Rajnikant Mehta

Co-Chairs: Debasish Dutta and Lalit Patil

Engineering Changes (ECs) are an integral part of a product's lifecycle. A proposed EC can affect several lifecycle-wide components. Detailed evaluation of each proposed EC or its effect is time-consuming and inefficient. Therefore, enterprises plan detailed evaluation of only those EC effects that might have a significant impact. Currently, domain experts decide which effects should undergo a detailed evaluation process. Such an approach relies heavily on personal experience and is less reliable. To address this problem, this research develops a systematic knowledge-based approach for determining whether a proposed EC effect has high expected cost impact and would require a detailed evaluation. An example EC knowledge-base is created to evaluate approaches developed in this research.

Only some of the large number of EC attributes are important for retrieving past ECs, which can be used to evaluate the impact of a proposed EC. This research formulates the problem of determining important EC attributes as a multi-objective optimization problem. Information-theoretic concepts are used to define measures for quantifying importance of an attribute subset. The domain knowledge and the information in EC database are combined to estimate probability distributions, which are required in computation of measures. An Ant Colony Optimization (ACO)-based search approach is developed for efficiently locating the important attribute set. A case study demonstrates the application of our approach to an example EC scenario. The example EC knowledge-base is utilized for evaluating the measures and the overall approach to determine important EC attributes. The evaluation results show that our measures perform better than the state-of-the-art evaluation criteria. The results obtained using our overall approach are analyzed based on the manual observation. The analysis of results show that when the important attributes identified using our approach are utilized to retrieve similar ECs with a goal of predicting impact, the success rate in predicting impact is 83.33%.

Utilizing past EC knowledge to predict the impact of proposed EC effect requires an approach to compute similarity between ECs. The second part of this research presents an approach to compute similarity between ECs that are defined by a set of disparate attributes. Since the available information is probabilistic, the measures of information are utilized for defining measures to compute similarity between two attribute values or ECs. The semantics associated with attribute values are utilized to compute similarity between attribute values. A case study is presented to demonstrate the applicability of our approach. The results of evaluating our approach against state-of-the-art approaches show that there is a statistically significant improvement in precision in retrieving similar ECs as well as success rate in predicting impact using our approach as compared to that using state-of-the-art approaches.

In the last part of this research, an approach is developed to predict impact of proposed EC effect based on the similar past ECs. The approach incorporates a technique to quantify differences between important attribute values in proposed EC and a similar past EC. The Bayes's rule is used to determine differences in impact value from the differences in attribute values. The probability values required in the Bayes's rule are determined based on the minimum cross entropy principle. A case study demonstrates the application of our approach to an example EC scenario. The results of evaluating our approach against state-of-the-art approaches show that there is a statistically significant improvement in success rate in predicting impact obtained using our approach as compared to that obtained using the state-of-the-art approaches. Based on the analysis of results, it can be inferred with 90% confidence that for a very large number of proposed ECs, i.e., $N > 100$, the success rate in predicting impact using our approach shall be greater than that obtained using state-of-the-art approaches.

CHAPTER I

Introduction

1.1 Background and motivation

An Engineering Change (EC) refers to a change to the process or product attribute, such as shape, structure, material or manufacturing process, after the initial design has been released [1]. ECs have always been an integral part of the product lifecyle. It is common within an enterprise to change one or more features of an existing product or process in order to fulfill the evolving customer requirements, technological innovations, and environmental regulations. Several times, distributed manufacturing enterprises encourage Engineering Changes to explore and use opportunities to reduce costs. For example, the US Department of Defense (DOD) promotes a Value Engineering Change Proposal (VECP) through which a contractor can propose changes and share the resultant cost savings. In 1997, the life cycle savings to the DOD were estimated to be nearly $25 million after VECP implementation and development costs were paid to Raytheon, Inc. out of the contract savings [2].

A typical EC process flow is shown in Figure 1.1 (adapted from [3]). The first step in the EC process is to identify and create the change proposal. Once the request for change is initiated, various alternative solutions are developed. Thereafter, the proposed change and each of its solutions are evaluated. The evaluation process typically involves determining the cost/time *impacts* of proposed change and its *effects*. The decision about the acceptance or rejection of the change is made based on the results of its evaluation. The acceptance of the EC is followed by its implementation.

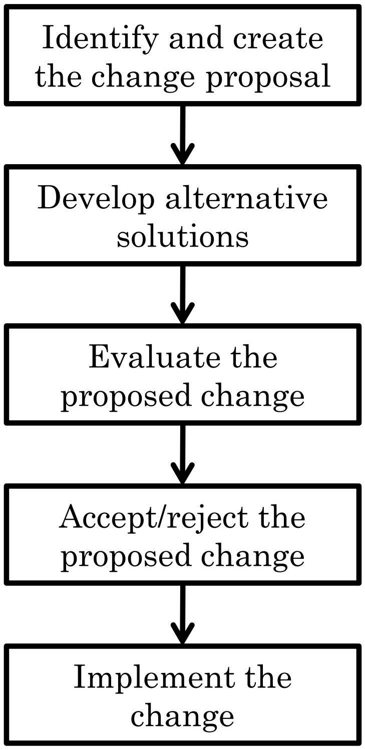

Identify and create the change proposal
Develop alternative solutions
Evaluate the proposed change
Accept/reject the proposed change
Implement the change

Figure 1.1: Typical engineering change process flow. Adapted from [3]

A seemingly simple EC can have several *effects*, since it can affect various product lifecycle elements, such as associated assemblies, manufacturing processes, inventory, and end-of-life treatment plans. A detailed evaluation of each proposed EC and its effects is time-consuming, inefficient and cumbersome due to sheer complexity of EC data and number of ECs typically handled by an enterprise. The complexity results because several distributed stakeholders, including designers, manufacturers, suppliers, and sales partners create an enormous amount of product data throughout the lifecycle - right from the specification of product requirements to product disposal. The number of changes typically handled by an enterprise is large. For example, in one survey of a large international corporation it was found that there are about 200 changes per month [4]. In another survey it was found that an average automotive company handles around 330 design changes per month [5].

As a result of complexity of EC data, cost and time spent on evaluating each effect of a proposed EC is typically high. For example, a survey of a few US and European companies found that the average administrative cost of processing each engineering change is $1400 [5]. In a similar survey of four Hong Kong manufacturing industries, it was found that the time invested in processing an engineering change varies from 2 to 36 person days [6]. To address this, enterprises handle insignificant ECs or effects, i.e., those with low expected cost/time impacts, through a *fast track* process, which aims to achieve faster implementation of a change by reducing its evaluation and implementation time [7]. Currently, domain experts decide which changes or effects should undergo a fast track or detailed evaluation process. This approach relies heavily on personal experience and expertise, and is less reliable. In this context, a problem that motivates this research is that there lacks a systematic approach, which is less dependent on the experience of the involved personnel, to identify exactly those effects of proposed EC which have high expected cost impact and would require a detailed evaluation.

1.2 Knowledge-based system for EC evaluation

Typically, proposed EC and its effects are manually evaluated by a group of domain experts, often with aid of techniques, such as Quality Function Deployment (QFD), Design for Manufacture and Assembly (DFMA), Failure Mode and Effect Analysis (FMEA) and Value Analysis (VA) [8, 9]. Utilizing such an approach for evaluating each effect of proposed EC is time-consuming, inefficient and less reliable.

In a different approach, a knowledge-based system can be developed to evaluate each proposed EC effect. A knowledge-based system emulates the decision-making process of human experts [10]. As compared to manual evaluation, a knowledge-based system is fast, reliable and less dependent on the experience of involved personnel. In the context of this research, a knowledge-based system can combine and utilize the knowledge captured in past change implementations and the knowledge/heuristics specified by other sources, such as domain experts or manufacturing handbooks.

A knowledge-based system is suitable for EC evaluation, since similar changes are likely to have similar effects and impacts, particularly within the same manufacturing enterprise. For example, a change made to a molded cover of one cell phone model is likely to have effects and impacts similar to those of a change to the molded cover of another cell phone model within the same manufacturing enterprise [11].

Technologically, a knowledge-based system for EC evaluation is more feasible now since the enterprises are increasingly embracing the philosophy of Product Lifecycle Management (PLM). PLM promises a framework based on emerging software technologies in the areas, such as knowledge management or web-based collaboration, to facilitate innovation by allowing faster and effective information exchange, knowledge reuse, and seamless collaboration between various stakeholders of an enterprise.

The knowledge-based evaluation of a proposed EC effect can be achieved through following steps, shown in Figure 1.2,

1. **Determining important EC attributes:** The input to a knowledge-based system for EC evaluation is the database containing past ECs and the relevant domain knowledge specified by sources, such as domain experts or manufacturing handbooks. Utilizing past engineering changes to predict the expected cost impact of a proposed EC effect requires a methodology to compute the

similarity between ECs. One of the major challenges in computing similarity between ECs is the computational burden, since the knowledge about each engineering change is captured using a large number (in hundreds) of disparate and interdependent attributes. In addition, some of the attributes might negatively affect the similarity computations. Therefore, there is a need for an approach to identify the important EC attributes that should be compared to compute similarity between the proposed change and each past EC.

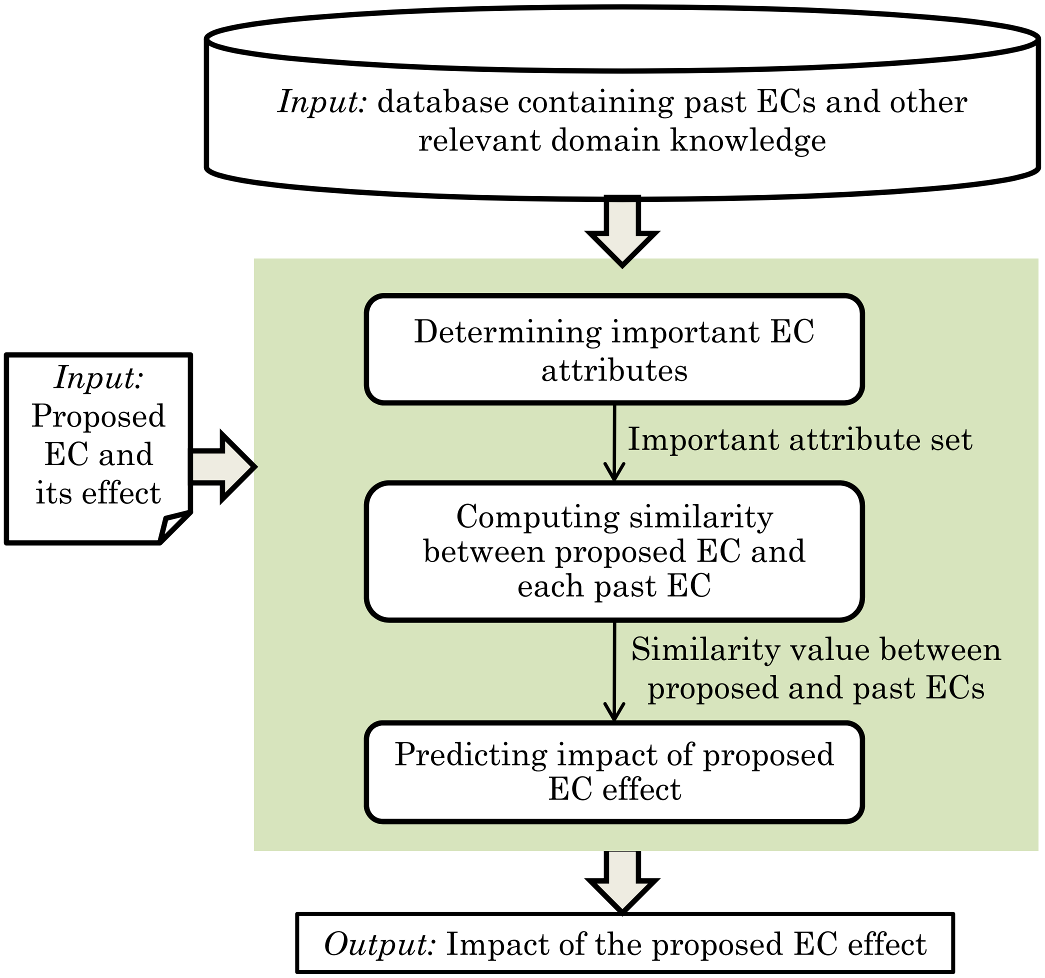

Input: database containing past ECs and other relevant domain knowledge
Input: Proposed EC and its effect
Determining important EC attributes
Important attribute set
Computing similarity between proposed EC and each past EC
Similarity value between proposed and past ECs
Predicting impact of proposed EC effect
Output: Impact of the proposed EC effect

Figure 1.2: Input, output and steps in a knowledge-based approach for predicting impact of proposed EC effect

2. **Computing similarity between proposed EC and each past EC:** Given

the important attributes, the similarity between proposed EC and each past EC must be determined, so that a set of past changes that are most similar to the proposed EC can be identified to evaluate impact of proposed EC effect. Therefore, there is a need for an approach to compute similarity between ECs in context of predicting impact of proposed EC effect.

3. **Predicting impact of proposed EC effect:** Once the important attributes and the similarity values between proposed and past ECs are obtained, the impact of proposed EC effect can be predicted. Therefore, the last step is to develop an approach to predict the impact of proposed EC effect based on the information in the similar past ECs.

Before discussing the dissertation goals and challenges, we shall define a few important terms and concepts with the aid of a typical representation to capture the EC data.

1.3 Terms and concepts

Figure 1.3 depicts a partial EXPRESS-G [12] illustration of a model to capture the data associated with an EC and its evaluation process. Information modeling methodology of STandard for the Exchange of Product model data (STEP) [13] is utilized to represent this data model. The important elements of a STEP data model are the entities, relations and attributes. The entities, e.g., *Part* or *Shape*, represent the main concepts in the domain, the attributes define the entities, and the relations define the linkages between the entities. Depending on the number of values it takes, the attribute data types are classified into simple data types and aggregation data types, e.g., set and array [12]. Depending on the type of value a simple data type takes, it can be classified into quantitative, i.e., integer and real, or qualitative, i.e., categorical or ordinal. For the sake of simplicity of explanation, only a few elements, i.e., entities, relations and attributes, are shown in Figure 1.3.

Following paragraphs define a set of key EC-related terms and concepts that appear in Figure 1.3.

Change

The root entity in Figure 1.3 represents the concept of Change (or EC). The information associated with state of the product before the change is proposed is captured by all the elements that appear beyond the relation *old_ configuration*; whereas the information associated with state of the product after the change is implemented is captured by all the elements beyond *new_ configuration.* Typically, values of some of the attributes in the *new_ configuration* will be unknown while the change is under evaluation. The change in the value of an attribute of an entity beyond the relation *old_ configuration* represents the change in that entity. For example, change in the value of *old_ configuration.part.shape.features* represents the *change in shape.* The new value of this attribute shall be stored in *new_ configuration.part.shape.features.* For simplicity, the entity being changed will be referred to as *change entity.*

Effect

The change in an entity will affect several other entities. For example, change in entity *Shape* can have an effect on entities *Process* and *Assembly.* The affected entities shall be referred as the *effect entities* and the corresponding cascaded changes will be referred as *effects.* For example, if *Process* and *Assembly* are the effect entities due

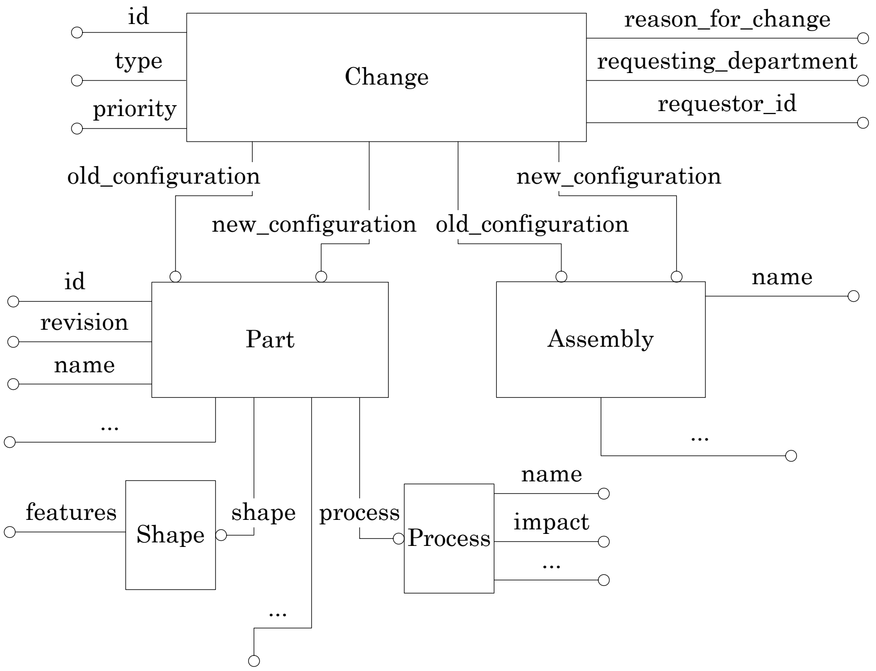


Figure 1.3: Partial illustration of a STEP-compliant data model to capture the knowledge associated with an EC and its evaluation process. The root entity represents the concept of Change. The information about the state of the product before the change is captured by all the elements that appear beyond the relation *old_ configuration*, and the information about the state of the product after the change is captured by all the elements beyond *new_ configuration*

to proposed *change in shape*, then the corresponding cascaded changes, i.e., *change in process* and *change in assembly* are effects of proposed change.

Impact

Associated with each effect entity is an attribute called *impact*, which captures a measurable consequence, i.e., overall cost, of proposed change on that effect entity. Impacts of various proposed change effects are unknown initially, but enterprises can estimate or measure them at the end of implementing the proposed change. In this research, an effect is classified as significant if the value of its cost impact is high;

whereas it is classified as insignificant if the value of its cost impact is low. The thresholds for high and low impact values can be specified by an enterprise.

1.4 Dissertation goals

As discussed in section 1.1, only the effects of proposed EC that might have significant impact should be evaluated in detail. In this context, the goal of this research is to develop a knowledge-based system to predict whether the expected cost impact of a proposed EC effect is significant (high) or insignificant (low).

To achieve this goal, we focus on following specific tasks:

1. Develop an approach to determine important EC attributes that should be compared to compute similarity between the proposed change and each past EC with the goal of ultimately using this knowledge in evaluating the impact of proposed EC effect.

2. Develop a method to compute similarity between ECs in context of predicting impact of proposed EC effect.

3. Develop an approach to predict the impact of proposed EC effect based on the similar past ECs.

Evaluation

To the best of our knowledge, there does not exist a benchmark EC database that can be used to evaluate/validate various approaches developed in this research. In addition, we are not aware of any other research/literature in the area of Engineering Change Management (ECM) which presents or evaluates approaches to the problems discussed in this dissertation. Therefore, for this dissertation, we create an example EC knowledge-base and utilize it to evaluate the various approaches discussed in this dissertation against possibly competitive techniques from other areas, such as machine learning or data mining.

1.5 Research challenges

An EC incorporates attributes of both quantitative and qualitative types. A qualitative attribute can be of ordinal or categorical type [14]. A categorical attribute is either of simple or aggregate type. Table 1.1 illustrates an example of each of these attribute types. The approach for each of the three research tasks identified in previous section should be suitable for such disparate attribute types.

EC attribute data type	**Example**
Categorical - simple	part.material.name
Categorical - aggregate	part.process.name
Quantitative	part.production_rate.unit_quantity
Ordinal	part.surface_finish.value

Table 1.1: Example of various EC attribute types in a typical EC data model

Following paragraphs presents the research challenges that are specific to each of the three tasks identified in previous section.

1.5.1 Determining important EC attributes

The problem of determining important EC attributes is challenging due to following reasons:

1. **Handling two interrelated target tasks:** There exists two interrelated target tasks, namely determining similar ECs and evaluating the impact of proposed EC effect. The approach to determine important EC attributes should be capable of considering both these tasks.

2. **Handling large number of interdependent attributes:** An attribute, by itself, may not be important in evaluating impacts of EC effects. It might be

important when considered in association with other attributes. In other words, associations between a set of interdependent attributes should be considered while identifying important attributes. As a consequence of this, the candidates for the set of important attributes include all the elements of the power set of n attributes which is of size 2^n. Since n is typically large (in order of hundreds), the approach should be capable of efficiently locating the set of important attributes.

1.5.2 Computing similarity between engineering changes

Following are the primary challenges in developing an approach to compute similarity between engineering changes:

1. **Handling the context of predicting impact:** The overall goal is to predict the impact of a proposed EC effect. The approach should compute the similarity between attribute values or ECs in the context of this overall goal.

2. **Utilizing semantics associated with attribute values:** The information available in EC knowledge-base to determine similarity in context of impact between two attribute values is the observed probability distributions of impact given the value of attribute. It is unknown whether an observed probability distribution conforms to the actual distribution of impact associated with the attribute values. Therefore, in addition to utilizing observed probability distributions of impact, the proposed approach should utilize semantics (or meaning) associated with attribute values.

3. **Accounting for variation in the similarity perception:** The perception about similarity value between attribute-values can vary among enterprises or within an enterprise over a period of time. For example, an enterprise might

perceive the values *cast iron - grade 100* and *cast iron - grade 220* to be more similar than the values *carbon steel - AISI 1030* and *carbon steel - AISI 1080*; whereas the opposite might be true in another enterprise. In a different example, within an enterprise the relative similarity between values *cast iron - grade 100* and *cast iron - grade 220* might decrease with the introduction of a new value - *cast iron - grade 150.* The proposed approach should account for such variations in the similarity perception.

1.5.3 Predicting impact of proposed EC effect

The key challenges in developing an approach to predict impact of proposed EC effect are:

1. **Accounting for differences in context of impact between attribute values:** Two changes that have a high value of similarity between them might not have same impact due to differences in context of impact between some of its attribute values. The proposed approach should be capable of accounting the differences in context of impact between attribute values in two changes. For simplicity, in remaining portion of this dissertation the term *difference* and the expression *difference in context of impact* are used interchangeably.

2. **Handling unknown relationship between attribute-values differences and differences in impact:** The nature of relationship between attribute-value differences and differences in impact is unknown. In addition, there is no formal approach to predetermine such relationship.

1.6 Research scope

Engineering changes includes product changes as well as process changes. The product changes are the alterations made to a product attribute, such as shape or manufacturing process. The process changes are the changes made to state and control of manufacturing processes within an enterprise which can affect one or more products. For example, changes in a manufacturing process techniques that is applicable to multiple products.

We limit the scope of this dissertation to product changes. The changes made to a single prismatic part of a mechanical product shall be studied and utilized to evaluate the approaches developed in this research.

1.7 Dissertation Outline

This chapter discussed the need and the problem of identifying proposed EC effect that have high expected cost impact and would require a detailed evaluation. It introduced the concept of knowledge-based system for evaluating a proposed EC or its effects. The important terms and concepts associated with the problem were defined. It also discussed the dissertation goals, three research tasks and associated challenges. The remainder of this dissertation is organized as follows.

Chapter II discusses the work related to the overall dissertation problem of evaluating impact of a proposed EC effect. It also presents a review of the relevant literature on the problems of determining important engineering change attributes, computing similarity between engineering changes and predicting impact of proposed EC effect.

Chapter III discusses an example EC knowledge-base and our approach for creating it. The example EC knowledge-base will be utilized to evaluate various approaches developed in this research.

Chapter IV presents a knowledge-based approach to determine important attributes of an engineering change. It formulates the problem of determining important attributes as the Multi-objective Optimization Problem. Concepts from the field of information theory are utilized to define measures for quantifying importance of an attribute set. An ACO-based search approach is presented to efficiently locate the important set of attributes.

Chapter V addresses the problem of computing similarity between ECs. Since the information available in the EC knowledge-base to compute similarity between attribute values or ECs is probabilistic, fundamental measures of information are utilized to define measures to compute similarity between two attribute values and two ECs.

Chapter VI discusses an approach to predict impact of proposed EC effect based on the similar past ECs. It presents our approach to account for the differences in context of impact between attribute values in two changes. The Bayes' rule is utilized to determine differences in impact value from the differences in attribute-values. The principle of minimum cross entropy is applied to determine the probability values required in the Bayes' rule.

Chapter VII presents the evaluation of the overall approach to predict impact of proposed EC effect against a few state-of-the-art supervised learning approaches to the classification problem.

Chapter VIII summarizes the contributions of this research. It highlights the applications in other areas that can benefit from this research and also discusses avenues for future research.

CHAPTER II

Literature review

This chapter discusses the work related to the overall dissertation problem of evaluating impact of a proposed EC effect. It also presents a review of the relevant literature on the problems of determining important engineering change attributes, computing similarity between engineering changes and predicting impact of proposed EC effect.

2.1 Evaluating impact of proposed EC or its effect

The commercial systems in the area of Engineering Change Management (ECM) typically support organization and control of documentation associated with the process of making ECs [15]. These systems do not implement a systematic approach to evaluate impact of proposed EC. The research efforts in the area of ECM have traditionally focused on enhancing the approach of carefully modeling all the linkages between different components of a product from the EC perspective and conducting a detailed analysis of EC propagation to determine its impact [16, 17]. In [18], a detailed conceptual analysis is followed to determine impacts of an engineering change on supply chain and materials planning. Such approaches rely heavily on the experience/expertise of the involved personnel, and are cumbersome, inefficient and error-prone if repeated for every proposed EC effect. [11] discusses a knowledge-based decision support system to evaluate the impact of a proposed Engineering Change. The past ECs that are similar to a proposed EC are retrieved based on the similarity of a set of a few predetermined attribute values. A simple majority voting procedure is suggested to predict the impact of proposed EC based on the set of past ECs that are similar to it.

The overall problem of this dissertation is similar to a typical classification problem in the fields of Data mining, Machine learning and Pattern recognition. A typical classification problem is to classify a new instance based on past classified examples. To address the problem of classification, there exists several supervised learning approaches, such as, Naïve Bayes classifier, decision tree learner, support vector classifier, neural networks, and k-Nearest Neighbor (NN). A detailed discussion of these approaches appear in the standard data mining and pattern recognition textbooks, e.g., [19, 20]. There also exists, several software tools, e.g., Waikato Environment for Knowledge Analysis (WEKA) [21], that implement the state-of-the-art supervised learning approaches to classification problem. A major drawback of a typical supervised learning approach is that the results obtained using it are purely statistical, since it relies completely on the past examples and seldom utilizes any domain-specific knowledge. The existing supervised learning approaches to classification problem are unsuitable to solve the overall problem of this dissertation, since they do not address the challenges discussed in section 1.5.

2.2 Determining important engineering change attributes

To the best of our knowledge, the problem of determining important attributes of an EC for evaluating its impacts has not been addressed before. Existing knowledge-based approaches [22, 11] propose to determine similarity by comparing every attribute associated with the Engineering Changes. Such exhaustive approaches are computationally expensive. In addition, the methods determine similarities across irrelevant attributes that might negatively affect the purpose.

The problem of determining important attributes has been addressed extensively in the areas of statistical pattern recognition [23], data mining [19], machine learning [24], and information retrieval and text categorization [25]. The attribute (or feature) selection in these areas is typically done to aid the overall task of classification, concept learning or clustering. A typical attribute selection approach can handle large number of attributes. However, as discussed at the end of this section, the existing approaches are not suitable for our problem, primarily because they cannot compute importance of an EC attribute set for two target tasks, namely retrieving similar ECs and evaluating the impact. Figure 2.1 illustrates the fundamental steps in a typical attribute selection method. In each iteration of subset generation, a candidate attribute subset is created based on a search strategy. The candidate attribute subset is evaluated to determine its effectiveness in fulfilling the target task, e.g., classification or learning. The subset generation and evaluation steps are repeated until a stopping criterion is satisfied. Several strategies exist each for subset generation and evaluation, and based on the combinations of these strategies various attribute selection methods have been proposed. Following sections summarize various state-of-the-art subset generation and evaluation methods.

2.2.1 Subset Generation

The methods for subset generation can be categorized into complete, sequential and metaheuristic methods [26]. The complete search technique guarantees to find an

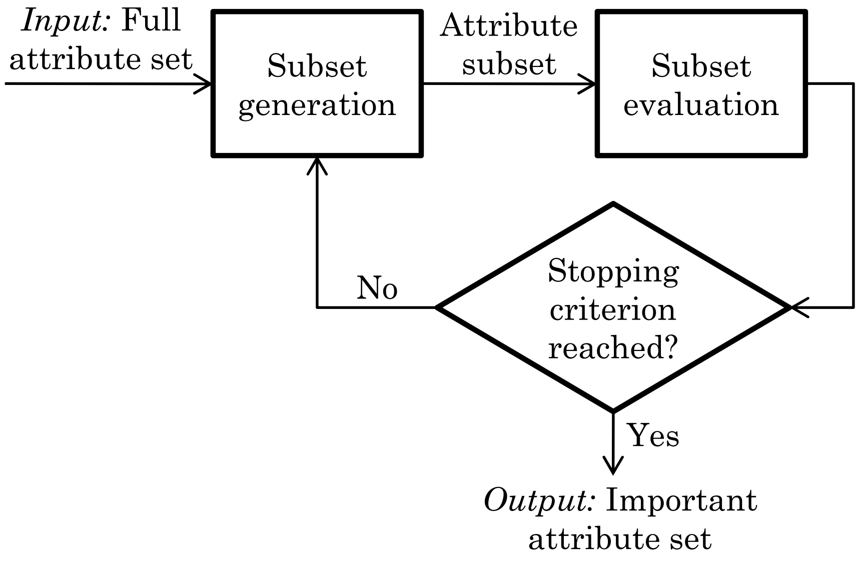

Input: Full attribute set
Subset generation
Attribute subset
Subset evaluation
Stopping criterion reached?
No
Yes
Output: Important attribute set

Figure 2.1: Fundamental steps in a typical attribute selection method. Adapted from [26]

optimal subset of attributes according to the evaluation criterion used. *Branch and bound* [27] is the most popular complete search technique which assumes that the evaluation function is monotonic. This assumption limits the application of *branch and bound* method, since most commonly used evaluation functions do not satisfy the monotonicity property. The *exhaustive* search is also complete; however, it requires examining 2^n candidate subsets for a data set that has n attributes. This becomes impractical even for moderate values of n.

Sequential search methods are greedy algorithms that trade off optimality of the selected subset for computational efficiency. There are two fundamental sequential search strategies, namely, *forward selection* and *backward selection.* The *forward selection* starts with an empty subset and sequentially adds the attributes until the stopping criterion is reached; whereas *backward selection* starts by considering entire attribute set as a candidate important subset and sequentially eliminates one attribute at a time until the stopping criterion is fulfilled. *Plus r take away l* is a bidirectional search that combines the *forward selection* and *backward selection* methods; it recursively adds r and eliminates l attributes to the current set of important attributes until a stopping criterion is satisfied. The best first and the beam search are sequential strategies that maintain a list of attribute sets evaluated earlier, so that it can be revisited, if required, in the process of search. The search itself can be in forward or backward or both the directions.

Typically, the sequential methods either generate a limited number of different solutions or stop at poor-quality local optima [28]. These limitations can be bypassed by using random population-based metaheuristic techniques, such as Swarm Intelligence (SI) methods and genetic algorithm [29]. The metaheuristic techniques are general algorithmic framework that can be adapted to specific search problems with relatively few modifications. Unlike single solution metaheuristics, such as simulated annealing, tabu search, guided local search, etc., the population-based metaheuristics manipulate a population of solutions. None of the existing random population-based metaheuristic techniques can be used off-the-shelf to solve our specific attribute selection problem. However we believe that each of these methods can be modified to solve our specific attribute selection problem.

Among the various random population-based metaheuristic techniques, the SI-based methods, in particular, the Ant Colony Optimization (ACO) possess several characteristics that makes it a suitable approach for the problem at hand. SI-based methods constructively builds multiple solutions in each iteration, exploits computer memory to direct future search, and can often be combined with a local search to obtain high-quality solutions. As compared to Genetic Algorithms (GAs), SI-based methods are easier to implement and have fewer parameters to adjust [30]. The three common SI-based search techniques include ACO, Stochastic Diffusion Search (SDS), and Particle Swarm Optimization (PSO) [30, 28]. Among the three common SI-based methods, the PSO, similar to GA, is a global metaheuristic algorithm with weak local search capabilities. SDS is typically suitable for problems, e.g., pattern matching, in which the goal is to locate a predefined target in a large solution space [31]. The ACO can be modified to solve our specific problem. Following section presents a detailed discussion on the ACO.

The Ant Colony Optimization Metaheuristic

The foraging behavior of real ant colonies has inspired development of ACO algorithm to solve various combinatorial optimization problems [32]. In ACO, a colony of artificial ants build solutions to an optimization problem at hand and exchange the information about these solutions using a communication scheme that is reminiscent of stigmergy used by real ants, with an overall quest for finding global optimal solution.

Given an optimization problem to be solved, the ACO algorithm iteratively simulates, in parallel, the movement of a number of artificial ants on a graph that encodes the problem. The set of solution components which serve as the building blocks for producing the solution to the problem are represented by either the vertices or the edges of the graph. Associated with each solution component is a value of *artificial pheromone trail* (or simply, *pheromone trail*), which indicates its utility in building the solution to the problem. In each iteration of ACO algorithm, an artificial ant starts with an empty solution set; moves from vertex to vertex of the graph, extending the partial solution set, based on the probabilistic decision policy until the stopping criterion is reached. The probabilistic decision policy is based on the pheromone trail values and the heuristic information about the problem at hand. At the end of each iteration, ants deposit a certain amount of pheromone on its components depending on the quality of the solution built. In addition, a certain quantity of pheromone is removed from all solution components to simulate the pheromone evaporation process. The mechanism of artificial pheromone evaporation avoids quick convergence toward a sub-optimal solution and allows forgetting of poor choices that may have be done in the past. Ants in the subsequent iteration use the updated pheromone value to build the solutions from scratch.

There exists several variants of ACO algorithms. A detailed discussion of these variants is out of the scope of this paper. The interested reader can refer to [28] for further discussion on the topic. Among the various variants, the Ant Colony System (ACS) and the Max-Min Ant System (MMAS) have found to return the solutions with minimal percentage deviation from optimum [33, 28].

ACO and Attribute selection

There exists a few applications of ACO technique to the attribute selection problem in various areas. The existing applications differ in (1) the interpretation of pheromone trail and heuristic information (2) the exact form of probabilistic selection rule and pheromone update rule, and (3) function utilized for validating the selection process. [34] applied a variant of ACO called the Elitist Ant System (EAS) algorithm for attribute selection in the classification tasks. The approximate value of Mutual Information (MI) is utilized as a heuristic function and the attributes are selected based on a measure referred as Updated Selection Measure (USM). Each subset is evaluated based on its mean square error (MSE) of classification. At the end of each iteration, the pheromone trail values of attributes in best k subsets are updated based on the corresponding MSE values. [35] recently applied the similar approach to select attributes for predicting post-synaptic activity in proteins. [36] utilize the ACS algorithm with Artificial Neural Networks (ANNs) for attribute selection in the domain of medical diagnosis. Inverse of the cost for taking a medical test is used as heuristic function. The attribute selection applications discussed so far follow a wrapper approach. [37] use a filter approach with a fuzzy-rough metric for the problem of fuzzy-rough dimensionality reduction. The Ant System (AS) algorithm is utilized for selection and pheromone update rules. Each of these applications are very specific to the problem at hand.

2.2.2 Subset Evaluation

The subset evaluation can be accomplished by either a filter or a wrapper approach. In wrapper approach, the performance of a pre-determined mining/learning algorithm is utilized for evaluating a subset; whereas in the filter approach, a pre-determined measure is utilized for evaluating an attribute subset. There exists several learning techniques, e.g., Naive Bayes classifier and decision tree learner, developed with a goal of classification [19]. Any of these learning techniques can be utilized in the wrapper approach of attribute selection that has classification as its target task. A limitation of wrapper approach as compared to the filter approach is that they are computationally expensive.

The measures used in filter approach can be classified into information, distance, dependency, and consistency measures [26]. The dependence measures, e.g., correlation function, quantify the association between two sets of categorical attributes. Distance measures find attributes that can separate the two classes as further as possible; whereas the consistency measures determines attributes that separate classes as consistently as the full set of attributes. Similar to the dependence measures, the information measures quantify the association between two sets of attributes. Among all the measures the information measures are considered to be a favorable subset evaluation criterion, since they can be applied to attributes of various types; they do not make any assumptions about the nature of relationships between attributes; they are non-metric and do not depend on the actual values that an attribute takes, but only on its probability distribution.

There exists several information measures that can be utilized for subset evaluation in the process of attribute selection [38]. These measures are based on the fundamental concept of information entropy, introduced by Shannon to quantify the information content or uncertainty of a probabilistic system [39]. Let A be an attribute (categorical or discrete) whose values follow a probability law $\{p_i : 1 \leq i \leq aa, i \in \mathbb{N}\}$, where aa is the domain size of A. The uncertainty about true value that A takes can be quantified by information entropy, $H(A)$, which is defined as follows:

$$H(A) = -\sum_{i=1}^{aa} p_i \times \log p_i \tag{2.1}$$

Higher is the value of $H(A)$, greater will be the uncertainty about true value that A takes. Note that $0 \leq H(A) \leq \log aa$ and $H(A) = 0$ if and only if A takes one of its values with certainty; whereas $H(A) = \log aa$ if and only if A is uniformly distributed. If B is another attribute with ab as the size of the domain of its outcomes, and if A and B are statistically dependent, then the conditional entropy of A, i.e., the entropy of A when outcome of B is observed, is given by:

$$H(A|B) = -\sum_{j=1}^{ab} \sum_{i=1}^{aa} p(a_i, b_j) \times \log p(a_i|b_j) \tag{2.2}$$

where, $1 \leq j \leq ab, j \in \mathbb{N}$, and $p(a_i, b_j)$ is joint probability of the events $A = a_i$ and $B = b_j$, and $p(a_i|b_j)$ is the conditional probability of event $A = a_i$ given that $B = b_j$. $H(A|B)$ is a measure of uncertainty about the true value of A after B is observed.

Lower the value of $H(A|B)$ higher is the one-way association between B and A, so that B can be utilized for the prediction of A.

The conditional entropy can be utilized for ordering attribute sets in the ascending order. If one wishes to order the attribute sets in the descending order, then following measure can be used [38],

$$M(A \Leftarrow B) = \log |aa| - H(A|B) \tag{2.3}$$

Higher the value of equation (2.3), greater is the usefulness of B in the prediction of A. In place of $\log |aa|$ in equation (2.3), the term $H(A)$ can be used. The resulting measure is referred to as the Information Gain (IG) and is defined as,

$$I(A;B) = H(A) - H(A|B) \tag{2.4}$$

One limitation of utilizing equation (2.2), (2.3) and (2.4) as a subset evaluation criterion is that an attribute set with larger domain size might be incorrectly perceived as the best subset, since an attribute set with larger domain size will have a lower value of conditional entropy. It is non-trivial, and probably impossible, to determine the exact amount of bias induced due to the domain size of an attribute set. The typical approach to correct this bias is to add entropy term $H(B)$ in the denominator of the equation. When the term $H(B)$ is added in the denominator of equation (2.4), the resulting measure is referred to as Information Gain Ratio (IGR) [24]. The IG and IGR have been successfully applied as the subset evaluation criterion in various attribute selection problem with a single target task. Since the problem of determining important attributes considered in this research has two target tasks, namely retrieving similar ECs and evaluating the impact, these measures are not applicable in there current form. In addition, the typical approach of computing these measures is to utilize the observed joint, conditional and marginal probability distributions. Such an approach produce results that are entirely statistical.

2.3 Computing similarity between engineering changes

Only a few research efforts have focused on computing similarity between engineering changes. In [22], every EC is composed of product, component, problem types, solutions and process representations. Similarities are measured in each representation type and the results are linearly combined to obtain the overall similarity between change instances. Resnik's information measure [40] is utilized to determine similarity between instances of each ontology. A multi criteria decision making (MCDM) method is utilized to determine the weights for each ontology in the linear combination equation. In [11], past changes that are similar to a proposed change are retrieved based on the similarity of a few specific attribute values. The Issue Based Information System (IBIS) is utilized to determine the similarity between values of *reason for change*, and a predefined look-up table is employed to determine the similarity between all remaining attributes. Based on the similarity value of each attribute, N most similar changes are identified. The existing methods for computing similarity are suitable for ECs that are defined using a few specific attributes. These methods do not focus on determining similarity in the context of predicting impact; nor do they account for variations in the similarity perception among enterprises or within an enterprise over a period of time.

In the area of similarity measurements, two classical approaches for computing similarity between objects based on the attribute values are metric space and set-theoretic [41]. The metric space approach represents objects as points, based on the attribute values, in a multi-dimensional metric space and evaluates similarity between two objects by a measure that is inversely proportional to the distance between the objects. Several metrics exist for determining the distance between two objects. A metric that is suitable for heterogeneous data, such as EC data, is generalized Minkowski metric [42]. Once the distance between two objects is determined, the distance values are transformed into similarity value using an universal law proposed by Shepard [43] as,

$$s_{AB} = \exp(-d_{AB}) \tag{2.5}$$

where, s_{AB} denotes the similarity between objects A and B, and d_{AB} represents the distance between the objects A and B in the multi-dimensional metric space. The metric space approach does not determine the similarity in context of an object attribute; nor does it account for alterations in the similarity perception. The set-theoretic approach of similarity is based on the assumption that the similarity between two objects is the function of saliency of object attributes (or features) [44]. In set-theoretic approach, objects are characterized as sets of binary features. The similarity between two objects A and B is defined as

$$s_{AB} = \theta \times f(A \cap B) - \mu \times f(A - B) - \nu \times f(B - A) \tag{2.6}$$

where, θ, μ and ν are the non-negative parameters, $f(A \cap B)$ is the salience of the features that both A and B have in common and $f(A - B)$ (or $f(B - A)$) is the salience of features that are contained in A (or B) but not B (or A). A major drawback of the set-theoretic approach is the requirement that the features must be characterized as binary and expressed as predicates over the object domain. This avoids the application of the set-theoretic approach to objects, such as EC, which are represented using a predefined list of disparate attributes.

There also exist approaches that utilize the available probability information to compute similarity between objects. Since the probability information available in the database of an enterprise is utilized for computing similarity, such approaches inherently account for alterations in the similarity perception. [45] presents an ordered probability-based similarity measure for determining similarity between objects that are defined using heterogeneous data. In this approach, an order relation is defined for each data type, which is used for determining the list of all attribute-value pairs that are less similar than or equally similar to a given attribute-value pair. The similarity between values of an attribute is computed as the probability of randomly picking an attribute-value pair that is less similar than or equally similar to it. Statistical methods, such as Fisher's transformation, are employed for integrating similarities between attribute values to compute the similarity between objects. The ordered probability-based approach is purely statistical and does not utilize the implicit semantics or explicit information associated with attribute values in the process of computing similarity.

[46] presents an information-theoretic measure for computing similarity between values of attributes based on the available probabilistic information. Starting from a set of six assumptions, the measure of similarity between attribute values A and B is systematically derived as the ratio of the amount of information needed to state the commonality of A and B to the amount of information needed to fully describe what A and B are. The proposed information-theoretic measure is utilized to determine the similarity between ordinal values, categorical aggregate values, words in a text corpus and concepts in a taxonomy. The approach presented in [46] cannot determine similarity in context of impact between two interrelated values of EC attribute, which can be of categorical aggregate type.

2.4 Predicting impact of proposed engineering change effect

The problem of predicting impact of proposed EC effect based on a set of similar past ECs has been addressed earlier in [11]. [11] suggests a simple majority voting method to determine whether the proposed EC effect has significant or insignificant impact. In this method, out of N retrieved similar past ECs, if effect of K changes have a significant impact, then the impact of proposed EC effect is considered to be significant.

In the area of machine learning, the problem of classifying a new object based on a set of similar past objects is commonly referred to as similarity-based classification problem. There exists several approaches to the problem of similarity-based classification [47]. The available approaches can be grouped into five categories. Nearest Neighbor (NN) methods are the simplest category of approaches. In this method, a new object is assigned the majority class of the k most similar objects. A popular variant of this method is weighted k-NN method, which classifies the new instance into a class that is assigned the highest weight [48, 49]. The second category of approaches considers similarity values between test instance and training instances as a feature vector, and utilizes a machine learning technique to classify the test instance [50]. The third category of approaches to the problem of similarity-based classification embeds the dissimilarity values in an Euclidean space using a linear distance-preserving mapping called Multidimensional scaling (MDS) [51]. Once mapped, standard statistical learning methods, such as Fisher Linear Discriminant (FLD) or linear support vector classifier, is used for classification. A popular approach to the problem of similarity-based classification is to consider pairwise similarity matrix as kernel [47]. A kernel is essentially a similarity function with certain mathematical properties. If the pairwise similarity matrix is symmetric and positive semidefinite, then a kernel-based machine learning algorithm, e.g., Support Vector Machine (SVM), can be utilized to solve the classification problem. The fifth category of approaches are called the generative approaches, which model the class-conditional distributions of pre-specified similarity statistics. The class-conditional distributions are then utilized for classification task. A popular generative approach called local Similarity Discriminant Analysis (SDA) follows principle of maximum entropy to estimate the class-conditional distributions [52]. The conditional distributions are utilized to classify a test instance such that the expected misclassification cost is minimized. A limitation of local SDA classifier is that its performance can be negatively affected if there are very few past instances in the neighborhood of the test instance. This limitation is addressed by regularizing a few parameters or class-conditional probabilities [53]. The resulting approach is referred to as regularized local SDA. The existing similarity-based classification approaches are unsuitable for our problem, since these approaches do not account for differences in context of impact between attribute values.

The problem of predicting impact of proposed EC effect addressed in this dissertation is similar to the problem of case adaptation in the area of Case-based reasoning (CBR). In case adaptation the goal is to determine an approach for reusing and revising the retrieved similar cases in context of the new case. There exist several generic approaches for case adaptation based on a single similar case as well as multiple similar cases [54]. At a fundamental level, the existing generic approaches for single/multiple case adaptation can be classified into transformational approaches, which adapt a past case solution, and derivational (or generative) approaches, which adapt a past method of constructing solution [55]. The transformational approaches are further classified into null adaptation, substitutional adaptation and structural adaptation. The null adaptation approach uses the past case solution as it is; whereas in substitutional adaptation, various solution parameters are recalculated based on the relation of the attributes of the problem description of the new case and similar case. Structural adaptation involves the reorganization and addition/deletion of solution elements depending on relations between the problem description of the new and the similar case. The exact technique followed in each of these generic approaches is highly domain dependent and typically requires knowledge about change in solution as a result of change in problem.

2.5 Summary

This chapter discussed the work, within and outside the field of Engineering Change Management (ECM), relevant to the problems addressed in this dissertation. To evaluate impact of proposed EC or its effect, the research efforts in the area of ECM have traditionally focused on enhancing approaches to model the linkages between different component of EC data and conducting a detailed manual analysis of EC propagation. Such approaches rely heavily on personal expertise, and are inefficient and error-prone if repeated for every proposed EC effect. The problem of evaluating impact of proposed EC effect is similar to a typical classification problem in a few Computer Science domain areas, such as Machine learning, Data mining or Statistical pattern recognition. There exists several supervised learning approaches to address the classification problem. A major drawback of a typical supervised learning approach is that the results obtained using it are purely statistical, since it relies completely on past examples.

To the best of our knowledge, the problem of determining important attributes of an EC to evaluate its impact has not been addressed earlier. There exists approaches for attribute selection in the Computer Science domain. These approaches are unsuitable to address our problem, primarily because they cannot quantify importance of an attribute set for two interrelated target tasks. Therefore, there is a need for an approach to determine important EC attributes which can quantify importance of an attribute set for two interrelated target tasks. Such an approach is presented in chapter IV.

In the area of ECM, the existing methods to compute similarity between changes are suitable for ECs that are defined using a few specific attributes. These approaches do not focus on determining similarity in the context of predicting impact nor do they account for variations in the similarity perceptions. The two classical approaches, namely metric space and set-theoretic, to compute similarity are not suitable for our problem, because they do not consider the disparity of the attributes or account for possible alterations in the similarity perception. The ordered probability-based approach to compute similarity between objects, which can be defined using heterogeneous data, is purely statistical and does not utilize the description associated with attribute values. Therefore, there is a need for an approach to compute similarity between ECs that are defined by a set of disparate attributes. Chapter V presents such an approach.

A simple majority voting method has been used in the field of ECM to predict impact of a proposed EC effect based on a set of similar past ECs. The problem of predicting impact of a proposed EC effect based on a set of similar past ECs is similar to the problems of similarity-based classification in the Computer Science domain and case adaptation in the area of CBR. The approaches to these problems are unsuitable, primarily because they do not account for differences in context of impact between attribute values in the process of prediction/classification. Therefore, there is a need for an approach that can account for differences in context of impact between attribute values in two changes. Details on such an approach is given in chapter VI.

CHAPTER III

Example EC knowledge-base

As discussed in section 1.6, to the best of our knowledge, there does not exist a benchmark EC knowledge-base that can be used to evaluate various approaches developed in this research. Therefore, for this research, we created an example EC knowledge-base, which is discussed in this chapter. The example EC knowledge-base includes a database of engineering changes and the relevant domain knowledge.

3.1 EC database

Creating an example EC database requires a data model to capture EC data. Following section addresses this problem and presents a data model that is used to capture ECs in example database.

3.1.1 EC representation

The key components of EC data are the product lifecycle data before the change is proposed and after the change is implemented. There exists several standards that can capture one or more aspects of product lifecycle data; however, none can match ISO 10303 in the depth and breadth of coverage of data [56]. ISO 10303, informally known as STandard for the Exchange of Product model data (STEP), is an international standard for capturing and exchanging the product information generated over its entire lifecycle [13]. An interested reader is referred to [57] for a detailed discussion on STEP. STEP consists of several Application Protocols (APs). Each AP contains data models for representing product data for a defined family of products at a defined stage in its lifecycle. As discussed in section 1.3, the important elements of a STEP data model are the entities, relations and attributes. The entities, e.g., *Part* or *Shape*, represent the main concepts in the domain, the attributes define the entities, and the relations define the linkages between the entities. Depending on the number of values it takes, the attribute data types are classified into simple data types and aggregation data types, e.g., set and array. Depending on the type of value a simple data type takes, it can be classified into quantitative, i.e., integer and real, or qualitative, i.e., categorical or ordinal.

Following section reviews current capability of STEP for capturing EC data.

Capability of STEP for capturing EC data

Information models for representing data relevant to Engineering Changes (ECs) appear in multiple STEP APs, e.g., manufacturing APs - AP 224 [58] and AP 240 [59], systems engineering AP - AP 233 [60], lifecycle AP - AP 239 [61], and product data management APs - AP 203 [62], AP 212 [63], AP 214 [64], and AP 232 [65]. The EC-related data models are similar in scope across these APs. Therefore, the remaining part of this section discusses EC-related data model from only one of these APs, i.e., AP 240.

Figure 3.1 shows a partial EXPRESS-G [12] illustration of the core set of entities and associated attributes in AP 240 for representing EC data. The key entities relevant to ECs include *Design_ exception_ notice*, *Engineering_ change_ proposal*, and *Engineering_ change_ order* [59]. The entity, *Design_ exception_ notice* represents a

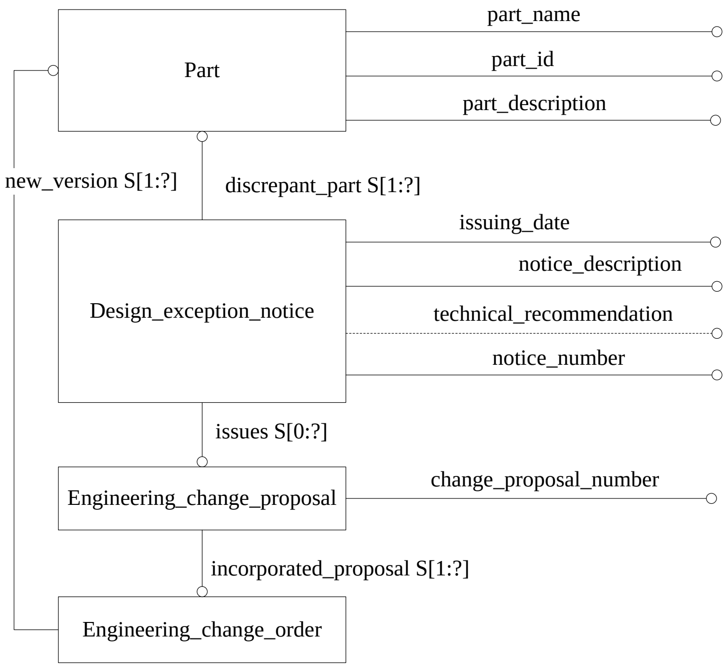


Figure 3.1: EXPRESS-G illustration of the core EC-related entities and attributes in AP 240

notification of a design discrepancy identified while creating the process plans for a given part such that process planning cannot continue until a technical recommendation is made to correct the problem. Each *Design_ exception_ notice* could have issues defined by *Engineering_ change_ proposal* entity. An *Engineering_ change_ proposal* is a document that describes potential alterations to a part and is linked to one or more *Engineering_ change_ order* that represents an authorization for modification of the product data that will result in a new process plan for a part.

STEP contains data models to capture various aspects of product lifecycle data. As discussed above, it also incorporates a model to capture some data about an EC. However, the current data models in STEP are not sufficient to capture all data related to EC or its evaluation. For example, there are no concepts/attributes to capture the data about items, such as impact, change type, priority, and so on. Such information is essential to exchange and reuse the data about past ECs to evaluate a proposed change. While STEP does not currently support representation of all data associated with an EC, it provides fundamental data structures that can be used and extended to capture required EC data [56]. Following section presents a partial illustration of a STEP-compliant data model to capture the knowledge associated with an EC and its evaluation.

STEP-compliant data model to capture EC data

STEP does not currently support representation of all data associated with an EC; it, however, provides fundamental data structures that can be used and extended to capture required EC data. Figure A.1 of Appendix A.1 depicts a EXPRESS-G illustration of a STEP-compliant data model to capture the data associated with an EC and its evaluation process. This data model will be utilized to capture the data associated with ECs in example database. The data model has 100 attributes, out of which 62 are of qualitative, i.e., categorical or ordinal, type and the remaining are of quantitative type. Several elements associated with the entities *Part* and *Assembly* are derived from the STEP manufacturing APs - AP 224 [58] and AP 240 [59]. For the simplicity of explanation, the terminology used for various elements in these APs have been changed in our example data model.

The root entity, which represents the concept of Change (or EC), has attributes: *id*, *type*, *priority*, *reason_for_ change* and *requesting_ department*. The *id* specifies a unique identification for change. The attribute *reason_for_ change* captures the purpose of change. An enterprise might predefine the values for *reason_for_ change* by specifying descriptive labels, such as corrective action, problem prevention, technical improvement and customer request, which informally describe the purpose of change. The *type* and *priority* represents the change type, e.g., change in shape or change in joint, and change implementation urgency, e.g., high, medium or low, respectively. The attribute *requesting_ department* captures the information about the department that proposed the change. The information associated with state of the product before the change is proposed is captured by all the elements that appear beyond the relation *old_ configuration*; whereas the information associated with state of the product after the change is implemented is captured by all the elements beyond *new_ configuration*.

3.1.2 Cost model

Cost impact of proposed EC or its effects depends on several factors. [66] presents a list of factors that should be considered in computing cost impact of a change. This includes cost of scrap, cost of rework/salvage/conversion/retrofit, cost of new/modified tool/equipment, cost of documentation/data creation/communication and administration, cost of schedule disruption and cost of product recall. A similar list of factors is presented in a 2007 survey of a few companies that have been found to efficiently manage engineering changes [67]. The Engineering Change Proposal (ECP) forms, which are utilized to propose the ECs under the VECP, decompose the cost impact of a change into three components, namely production costs, retrofit costs and integrated support logistics cost.

Considering the various factors suggested in different literatures, this section presents a model to compute the impact of an EC effect in the example database. The total impact, denoted as C^T, of an EC effect is computed as sum of three disparate components,

$$C^T = C^A + C^F + C^V \times V \tag{3.1}$$

where, C^A represents the cost per time of analyzing the effect, C^F represents the fixed cost per time of implementing the effect of the change, if it is accepted, C^V represents the variable cost per unit of implementing the effect of the change and V represents the number of manufactured units per time. A proposed EC might have multiple cascaded effects. The aforementioned model for computing impact of an effect assumes that the correct proportion of each cost factor can be assigned to each effect.

The quantity C^A primarily incorporates labor/information cost for analyzing the effect. If an effect is evaluated in detail, then this cost will be higher as compared to the case in which the effect is evaluated by fast-track process. The one-time cost, C^F, of implementing the change effect consist of four components: one-time cost of new tool/equipment/process/technology that is dedicated to the product being changed, the cost of disruptions in manufacturing, the cost of redesign and the cost of training employees. The disruptions in manufacturing include delays in completion of existing project, backorder, higher/lower inventory and obsolescence of a tool/equipment/process/technology. The cost of redesign comprises of the labor/information cost for modifying the product design according to the change. Depending on the proposed EC, the variable cost C^V might be negative. The negative value of C^V represents savings, which might be due to reduction in production time or labor. If the change is implemented, then its total cost impact on an effect is determined using equation (3.1); whereas if the change is not implemented, then its total cost is equal to C^A.

3.1.3 Example engineering changes

The example EC database incorporates 17 changes of type *change in shape. Process* is the effect entity in each of the 17 changes. The knowledge associated with each change is formally captured using the data model shown in Appendix A.1. The domain values for each attribute and the constraints among the attribute values are determined from associated Computer-Aided Design (CAD) models, Cambridge Engineering Selector (CES) [68], and Bralla's manufacturing handbook [69]. Following sequence of steps are followed to assign values to various attributes of an EC. The values of attributes that define *Shape* are determined from associated CAD models. Depending on the value of *Shape* attributes, the CES and Bralla's manufacturing handbook are utilized to assign a value to the attributes that define *Material* and *Process*. The attributes of entities *Tool*, *Machine* and *Fixture* are assigned a value based on the value of *Process* attributes. Also based on the *Process* attribute values, the *Tolerance* and *Surface finish* attribute values are determined from CES. The values of attributes associated with *Minimum Wall Thickness* and *Volume* are determined from associated CAD models. All 17 changes are on various parts from three example products. The CAD models of a relevant product is utilized to assign a value to the attribute *associated_joint_geometry*. Based on the value of this attribute, the values of *Assembly process* attributes are determined from CES and [69]. The remaining attributes, e.g., attributes of *Production rate* and *Person*, are assigned a value after selecting a domain for each of them and ensuring that the assigned value does not conflict with the values of other attributes. For several example changes, there exists a few attributes that are irrelevant. For example, if the process utilized to manufacture a part, associated with an EC, is *Casting*, then the attributes *max_spindle_speed* and *max_feed_rate* are irrelevant. For such irrelevant attributes, a value of 0 is assigned.

Appendix A.2 illustrates the 17 changes in the example database. For each example EC, values of a few attributes, which are typically found to be important, are shown. The relative cost impact of EC on key factors discussed in section 3.1.2 is also summarized. The relative cost impact of EC on various factors are added up to determine total overall cost impact of EC on *Process*. If the impact of an EC on *Process* is above average of all ECs, then it is considered as *high* impact; otherwise it is *low* impact.

3.1.4 Creating multiple datasets

Since the dataset is of very small size, the 0.632 bootstrap technique [19] is utilized for creating 10 datasets from 17 changes. In each run of this technique, the dataset is sampled 17 times, with replacement, to generate a training dataset of 17 instances. The instances that are not part of the training dataset are considered to be proposed changes. The set of proposed changes form the test dataset. There are 6 instances in each test dataset. Appendix A.3 summarizes the training and test instances in various datasets. These datasets are utilized for evaluating various approaches developed in this research.

3.2 Domain knowledge to determine important EC attributes

The knowledge captured by the past ECs is specific to an enterprise. Apart from this enterprise-specific knowledge, there exists domain knowledge that is applicable to several enterprise, maybe to different extents, and is commonly presented in the manufacturing textbooks/handbooks. For example, a typical manufacturing handbook, such as [69], contains the information about the compatibility between the materials and manufacturing processes. At an attribute level, such information explicitly specify the associations and interrelationships among attributes, and can be useful in identifying the important attributes of an EC.

A popular method of capturing and representing knowledge at an attribute level is by means of *if-then* rules [70]. A single ***if-then*** rule has a form ***if*** A ***then*** B, where A, called antecedent, and B, called consequent, can be a conjunction of attribute value pairs. For example,

- ***if*** new_configuration.surface_finish.value = *A* AND new_configuration.tolerance_range.lower_limit = *0.0065* AND new_configuration.tolerance_range.upper_limit = *0.0275* AND new_configuration.min_wall_thickness.value = *0.09* AND new_configuration.production_rate.unit_quantity = *50000* AND old_configuration.material.class = *AL* ***then*** new_configuration.process.name = *die casting*
- ***if*** old_configuration..part.production_rate.unit_quantity = *50000* AND old_configuration.part.process.required_machine.type = *injection molding* ***then*** impact = *high*

We allow an enterprise to identify the relevant knowledge at an attribute level from the various knowledge sources, such as domain experts or manufacturing handbooks. It is, however, assumed that these knowledge is encoded in the form of ***if-then*** rules.

For our example knowledge-base a set of 10 *if-then* rules are derived from CES and [69]. These rules are shown in Appendix B.1.

3.3 Summary

This chapter discussed our approach to create an example EC knowledge-base, which will be utilized to evaluate various approaches developed in this research. A STEP-compliant data model is proposed to capture the data associated with an EC and its evaluation process, since the current data models in STEP are not sufficient to capture these data. A model to compute the cost impact of EC on its effect is proposed. The proposed data model and cost model are utilized in creating 17 example engineering changes of type *change in shape*. The domain values for each attribute and the constraints among the attribute values are determined from associated CAD models, Cambridge Engineering Selector (CES) and a manufacturing handbook. The values were assigned to the attributes of an engineering change, while ensuring that they satisfy the constraints among their domain values. Since the example database has a small number of engineering changes, the 0.632 bootstrap technique is utilized for creating 10 different training and test datasets.

The knowledge that is relevant to the problem of determining important attributes is identified. The identified knowledge is encoded in the form of ***if-then*** rules. A set of 10 *if-then* rules are derived from CES and a manufacturing handbook for our example knowledge-base.

CHAPTER IV

Determining important engineering change attributes

This chapter presents a knowledge-based approach to determine important EC attributes that should be compared to compute similarity between the proposed change and each past EC with the goal of ultimately using this knowledge in evaluating the impact of proposed EC effect.

4.1 Motivation

A large number (in order of hundreds) of disparate and interdependent attributes capture the data about an EC. Utilizing all the attributes to compute similarity between ECs will be computationally expensive. In addition, some of the attributes might negatively affect the similarity computation. Therefore, it is essential to determine important attributes of proposed change and use only those for retrieving similar past changes.

The problem of determining important EC attributes has not been addressed before. Attribute selection problem, however, has been addressed extensively in the Computer Science domain areas, such as machine learning [24], data mining [19] and statistical pattern recognition [23]. The existing approaches to attribute selection in these areas do not address some important challenges in our problem:

1. There exists two interrelated target tasks, namely determining similar ECs and evaluating the impact of proposed EC effect, to identify important EC attributes. The existing attribute selection approaches cannot quantify importance of an attribute set for two interrelated target tasks.

2. To determine important attributes, the information available in EC database is the observed distribution of the attribute values. It is, however, unknown whether the observed probability distribution conforms to the actual distribution of attribute values. Since the existing approaches typically rely on observed distributions to determine important attributes, the results obtained using them might be entirely statistical.

4.2 Objective

The objective of this research phase is to develop an approach, which

For: a proposed Engineering Change

Given: database of past ECs and domain knowledge encoded in the form of *if-then* rules

Determines: which attributes in the proposed EC should be selected to determine similar ECs with the goal of ultimately using this knowledge in evaluating the impacts of the proposed EC effect

Following paragraphs discuss assumptions about the nature of EC database:

1. Each EC is captured using a STEP-compliant data model. In addition, the domain of each attribute associated with ECs is predefined and known.

2. All changes in the database are independent of each other and have same effect as the proposed EC.
3. Depending on the value of impact, the enterprise classifies a past EC effect into significant (high impact) or insignificant (low impact).
4. The enterprise utilizes its internal methods to discretize the quantitative attributes.

For simplicity, in remaining portion of this dissertation the set of important attribute set will be referred to as Important Attribute Set (IAS).

4.3 Multi-objective optimization problem formulation

This section formulates the problem of determining the important attributes of a change as the multi-objective optimization problem. Consider following notations,

- E : effect entity
- U : the attributes of E
- $\{u_i : 1 \leq i \leq au, i \in \mathbb{N}\}$: domain of U
- Z : impact of proposed change on E
- $\{z_i : 1 \leq i \leq az, i \in \mathbb{N}\}$: domain of Z
- $X = \{x_i : 1 \leq i \leq n, i \in \mathbb{N}\}$: n candidate attributes associated with change. This does not include Z and U
- $\mathcal{P}(X)$: power set of X
- $R^U = \left\{r_i^U : 1 \leq i \leq m^U, i \in \mathbb{N}\right\}$: m^U rules among the attribute values that have effect attributes in the consequent

- n_k^U : number of attributes in k^{th} rule from R^U

- $R^Z = \left\{ r_i^Z : 1 \leq i \leq m^Z, i \in \mathbb{N} \right\}$: m^Z rules among the attribute values that have impact in the consequent

- n_k^Z : number of attributes in k^{th} rule from R^Z

- S, T : elements from $\mathcal{P}(X)$. $S \subseteq X$, $T \subseteq X$

- $\{s_i : 1 \leq i \leq as, i \in \mathbb{N}\}$: domain of S

- $\{t_i : 1 \leq i \leq at, i \in \mathbb{N}\}$: domain of T

- Y : the Important Attribute Set (IAS)

4.3.1 Design variables

Corresponding to each attribute x_i, a variable $v_i \in \{0, 1\}$ is defined as,

$$v_i = \begin{cases} 1 & \text{if } x_i \in Y; \\ 0 & \text{if } x_i \notin Y; \end{cases} \tag{4.1}$$

Since the problem is about selecting attributes for inclusion into the IAS, the vector $V = \{v_1, \ldots, v_n\}$ represents the design variables of our optimization problem. An assignment of V is referred to as *complete* if all its element are assigned a value of either 0 or 1; otherwise it shall be referred to as *partial*.

Let Ω represent the set of all possible complete assignments of V and V^S, V^T represent any two assignment of V. The number of elements in Ω is 2^n. Let $f : \Omega \to \mathcal{P}(X)$ be a function that transforms the element of Ω into the element of $\mathcal{P}(X)$. The f transforms V^S to S such that if the value of i^{th} element in V^S is 1, then the $x_i \in S$. In the remaining portion of this paper, the notations $f\left(V^S\right)$ and S

shall be used interchangeably. Similarly, the notations $f\left(V^T\right)$ and T shall be used interchangeably.

4.3.2 Objective functions

Since the attributes define an entity, in order to determine the similarity between changes with a goal of evaluating the impact of its effect it is essential to compare the attributes that are associated with the effect and change entities before and after the change. The value of effect attributes after the change is unknown while the change is under the evaluation. This can be addressed by identifying and comparing the attributes that enable the prediction of the value of effect attributes after the change is implemented. Similarly, the impact of the proposed change effect can be evaluated by identifying and comparing the attributes that enable the prediction of its value. To this end, the important attribute set shall include the attributes associated with the change and the effect entities before the change, the attributes associated with the change entity after the change, and the attributes that enable the prediction of the value of impact and effect attributes after the change.

The rules among the attribute values that have effect attributes in its consequent shall be useful in identifying the attributes that enable the prediction of the value of U; whereas the rules that have impact in its consequent shall be useful in identifying the attributes that enable the prediction of the value of Z. Since two separate set of rules are useful in identifying the attributes that enable the prediction of the value of Z and U; two different measures shall be defined. Let $\Phi^Z(S)$ and $\Phi^U(T)$ represent a measure that quantifies the usefulness of S and T in predicting the value of Z and U, respectively, such that higher the value greater is the usefulness. An attribute set that is useful for predicting the values of U might affect, negatively or positively, the prediction of value of Z. Let $\Delta^Z(S, T)$ represent the difference between the usefulness of $S \cup T$ and T in predicting the value of U. The positive (or negative) value of $\Delta^Z(S, T)$ implies that the selection of S improves (or deteriorates) the prediction of U. Among the attribute sets with the same value of Φ^Z, one that has larger value of Δ^Z should be chosen for inclusion into the IAS. To enable this, the quantities $\Phi^Z(S)$ and $\Delta^Z(S, T)$ are aggregated as,

$$\Pi^Z(S) = \Phi^Z(S) + \lambda^Z \times \Delta^Z(S, T) \tag{4.2}$$

where, λ^Z is a parameter in the range $[0, 1]$. An attribute set that has maximum value of Π^Z in equation (4.2) should be the part of the IAS. The parameter λ^Z specifies how important it is to reduce the negative effect of S in predicting the value of U. $\lambda^Z = 1$ implies that it is as important to reduce the negative effect of S in predicting the value of U as is maximizing the accuracy of predicting the value of Z. On the other hand, $\lambda^Z = 0$ implies that S should be selected independent of its effect on predicting the value of U.

Similarly, the usefulness of T in predicting the value of U and the difference between the usefulness of $T \cup S$ and S in predicting the value of Z are aggregated as,

$$\Pi^U(T) = \Phi^U(T) + \lambda^U \times \Delta^U(T, S) \tag{4.3}$$

where, $\Delta^U(T, S)$ represents the difference between the usefulness of $T \cup S$ and S in predicting the value of Z, and $\lambda^U \in [0, 1]$ is a parameter that specifies how important it is to reduce the negative effect of S in predicting the value of Z. The two interrelated equations (4.2) and (4.3) represent the measures for identifying the attributes that shall enable the prediction of values of impact and effect attributes after the change is implemented. An attribute set that has maximum value for these measures should be the part of the IAS. To locate such an attribute set, a Multi-objective Optimization Problem (MOOP) is formulated with following two objectives,

1. maximize $(\Pi^Z(S))$, and
2. maximize $(\Pi^U(T))$

Following section presents the constraints in the optimization problem.

4.3.3 Constraints

The constraints on the optimization problem can be classified into hard constraints and soft constraints. The hard constraints formalize the requirements that cannot be violated; whereas soft constraints formalize the desired properties whose violation should be avoided, as much as possible.

Hard constraints

As discussed in previous section, in order to determine the similarity between ECs it is essential to compare the attributes associated with the change and effect entities before the change, and the attributes associated with the the change entities after the change. Such restrictions are captured using constraints of form,

$$v_i = 1; \exists i \in \{1, \ldots, n\} \tag{4.4}$$

In the change/product data model, there exists attributes that should always be compared concurrently. Typically, this shall be the case if the comparison of a set of attributes make sense only when done concurrently. For example, consider the entity *Tolerance range* with attributes *lower_ limit* and *upper_ limit.* Each of these two attributes are crucial in the definition of the entity *Tolerance range*, so that the similarity value between the two instances of *Tolerance range* will make sense only when it is based on the comparison of all its attributes. Similar is the case for entity *Mass* that has attributes *value* and *unit.*

We shall allow an user to indicate the attributes that are mutually inclusive using the $n \times n$ matrix t^{in}. All the diagonal elements in t^{in} are assigned a value of 1. The user can assign a value of 1 to a non-diagonal element t^{in}_{ij}, if the attributes x_i and x_j are mutually inclusive. All unassigned values shall be taken as 0. Based on the matrix t^{in}, the set of n mutually inclusive constraints are modeled as,

$$\text{if } v_i = 1, \text{ then } \sum_{j=1}^{n} v_j \times t^{in}_{ij} = r_i; \forall i \in \{1, \ldots, n\} \tag{4.5}$$

where, r_i is the row-sum of i^{th} row in the t^{in}_{ij}.

Several times there exist two attribute sets that capture the same information. For example, consider the attribute sets *{volume.value, volume.unit, density.value, density.unit}* and *{mass.value, mass.unit}*. It will be redundant to have both these attribute sets to be the part of IAS. In a different example, consider the entity *Material* with following attributes: *name*, *hardness.value*, *hardness.unit*, *strength.value* and *strength.unit.* If the *name* of *Material* is part of the IAS, then it will be redundant to utilize its remaining attributes. Same is true the other way. Such relations among the attributes are captured by the mutually exclusive constraints.

We allow an user to specify the mutually exclusive attributes by filling out the non-diagonal entries in the $n \times n$ matrix t^{ex}. A non-diagonal element t^{ex}_{ij} is assigned a value of 1, if there exists a mutually exclusive relation between the attributes x_i and x_j, otherwise it is assigned a value of 0. All the diagonal elements in t^{ex} are assigned a value of 0. Based on the matrix t^{ex}, the set of n mutually exclusive constraints are represented as,

$$\text{if } v_i = 1, \text{ then } \sum_{j=1}^{n} v_j \times t_{ij}^{ex} = 0; \forall i \in \{1, \ldots, n\} \tag{4.6}$$

Soft constraints

As discussed earlier, the rules among the attribute values specify the association and interrelationships among the attributes. The amount of association between an attribute set and the effect attributes can be quantified based on the proportion of rules between it and the effect attributes. We define a function called *support* to compute the proportion of rules between any two attribute values or a candidate attribute set and the effect attributes. Let $c^U(t_i, u_j)$ denote the support for attribute value pair t_i, u_j in the rules R^U. $c^U(t_i, u_j)$ is defined as,

$$c^U(t_i, u_j) = \sum_{k=1}^{m^U} \frac{\Gamma(t_i, u_j, k)}{m^U \times n_k^U} \tag{4.7}$$

where,

$$\Gamma(t_i, u_j, k) = \begin{cases} 1 & \text{if } t_i, u_j \text{ appear in } r_k^U \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

The support for attribute set T can be computed as,

$$c^U(T) = \sum_{j=1}^{au} \sum_{i=1}^{at} c^U(t_i, u_j) \tag{4.9}$$

The support, $c^U(T = \{x_i\})$, for an attribute x_i among the relations R^U indicates the amount of association between the x_i and the effect attributes U. Higher the value of $c^U(x_i)$ greater is the probability that v_i is equal to 1. This is modeled as the probabilistic constraints on the optimization problem,

$$p(v_i = 1) = c^U(x_i); \forall i \in \{1, \ldots, n\} \tag{4.10}$$

Similarly, let $c^Z(s_i, z_j)$ denote the support for attribute value pair s_i, z_j in the rules R^Z. The n probabilistic constraints based on the rules R^Z are defined as,

$$p(v_i = 1) = c^Z(x_i); \forall i \in \{1, \ldots, n\} \tag{4.11}$$

The probabilistic constraints shown in equation (4.10) are applicable only for maximizing $\Pi^U(T)$; whereas the constraints shown in equation (4.11) are applicable only for maximizing $\Pi^Z(S)$. Such constraints formalize the desired properties whose violation should be avoided, and hence are commonly referred to as soft constraints [71]. On the contrary, since the constraints shown in the equations (4.4), (4.5) and (4.6) formalize the requirements that cannot be violated, they are referred to as the hard constraints. Before discussing an approach for solving the MOOP, following section presents our approach for computing the two measures shown in equations (4.2) and (4.3).

4.4 Computing measures to identify important attributes

As discussed in previous section, $\Phi^Z(S)$ quantifies the usefulness of S in predicting the values of Z. The nature of relationship, linear or non-linear, between the values of S and Z is unknown. The S and Z can include the attributes of both qualitative and quantitative types. Taking these into consideration, $\Phi^Z(S)$ is defined as,

$$\Phi^Z(S) = \log |az| - H(Z|S) \tag{4.12}$$

where, $H(Z|S)$ is the conditional entropy [72]. The $\Delta^Z(S, T)$ represents the difference between the usefulness of $S \cup T$ and T in predicting the value of U. Using conditional entropies, it is defined as,

$$\Delta^Z(S, T) = H(U|T) - H(U|S \cup T) \tag{4.13}$$

Substituting equations (4.12) and (4.13) in equation (4.2) gives the form of measure $\Pi^Z(S)$ as,

$$\Pi^Z(S) = \log|az| - H(Z|S) + \lambda^Z \times [H(U|T) - H(U|S \cup T)] \qquad (4.14)$$

In a similar way, the form of measure $\Pi^U(T)$ is derived as,

$$\Pi^U(T) = \log|au| - H(U|T) + \lambda^U \times [H(Z|S) - H(Z|T \cup S)] \qquad (4.15)$$

Computing conditional entropy values shall require the conditional probability distributions and the joint probability distributions. The typical approach is to utilize the observed distributions from the database. A limitation of such an approach is that it is purely statistical, and can produce erroneous results if the observed distributions does not capture the true nature of association among the attributes. To address this, we shall utilize the rules among the attribute values along with the observed joint and conditional distributions to estimate the unknown joint and conditional distributions. The details on estimating the unknown distributions are discussed in following section.

4.4.1 Estimating unknown probability distributions

This section presents the approach for estimating the joint and conditional probability distribution between the attribute set T and the effect attributes U. The same approach is followed for determining other distributions required in the equations (4.14) and (4.15) of the measure.

Let q_{ij} and $q_{j|i}$ denote the observed joint and conditional probability values between t_i and u_j determined from change database. Let p_{ij} and $p_{j|i}$ denote the unknown joint and conditional probability values between t_i and u_j. In the absence of any other information, the best possible estimation of p_{ij} and $p_{j|i}$ values shall be

q_{ij} and $q_{j|i}$, respectively. However, there is information available in the form of rules among the attribute values. As discussed earlier, $c^U(t_i, u_j)$ denotes the support for value pair t_i, u_j in the relations among the attribute values. If $c^U(t_i, u_j) > 0$, then the probabilities (both joint and conditional) between the two should be higher than the observed probabilities. Greater the value of $c^U(t_i, u_j)$, higher should be the value of p_{ij} as compared to q_{ij}. To fulfill this requirement, the ratio p_{ij}/q_{ij} is computed as,

$$\frac{p_{ij}}{q_{ij}} \geq 1 + a \times c^U(t_i, u_j) \tag{4.16}$$

where, $a \in [0, 1]$ indicates the influence of rules in the determination of joint probability values, such that higher the value of a, greater is the influence. In the case of $a = 1$ the rules among the attribute values and the observed probabilities are equally important in determination of the probability values.

Similarly, the ratio $p_{j|i}/q_{j|i}$ is defined as,

$$\frac{p_{j|i}}{q_{j|i}} \geq 1 + b \times c^U(t_i, u_j) \tag{4.17}$$

where, the parameter b indicates the influence of rules among the attribute values in the determination of conditional probability values. The equations (4.16) and (4.17) are the two constraints on p_{ij} and $p_{j|i}$ values, respectively. There exists additional constraints to ensure that the joint and conditional probability distributions satisfies the basic relations in the probability theory. These constraints are captured in the form,

$$\sum_{i=1}^{at} \sum_{j=1}^{au} p_{ij} = 1 \tag{4.18}$$

$$1 = \sum_{j=1}^{au} p_{j|i}; \forall i \in \{1, \ldots, at\} \tag{4.19}$$

$$1 = \sum_{i=1}^{at} \frac{p_{ij}}{p_{j|i}}; \forall j \in \{1, \ldots, au\} \tag{4.20}$$

There might be several values of p_{ij} and $p_{j|i}$ that satisfy the constraints shown in equations (4.16), (4.17), (4.18), (4.19) and (4.20). Among these several values, the best possible values are determined based on the principle of minimum cross entropy [73], which states that among all the distributions that satisfy the constraints choose one that is closest to the observed distribution. In the principle of minimum cross entropy, the total distance between the unknown and observed distributions is computed using the Kullback-Liebler (KL)-divergence [73] as,

$$KL(p_{U|T}||q_{U|T}) + KL(p_{TU}||q_{TU}) = \sum_{i=1}^{at}\sum_{j=1}^{au} p_{ij} \times \log\left[\frac{p_{ij}}{q_{ij}} \times \frac{p_{j|i}}{q_{j|i}}\right] \tag{4.21}$$

The unknown distributions are determined such that the equation (4.21) is minimized subject to constraints (4.16), (4.17), (4.18), (4.19) and (4.20). Some of the q_{ij} and $q_{j|i}$ values might be zero. This presents an issue in estimating a minimization of equation (4.21). To address this issue, a technique called Laplace estimator will be utilized to convert zero observed probability value to a very small non-zero value [19].

4.5 Search approach

This section discusses our search approach for efficiently locating the IAS. Two important required characteristics of our search approach are,

- **Handling multiple objectives and disparate constraints:** There exist two objective functions in our search problem. Due to the inter-relationships between these two objectives, they cannot be aggregated into a single function.

The search approach should be capable of handling these two objectives in parallel. In addition, the approach should be capable of handling the hard and the soft (probabilistic) constraints.

- **Exploiting past knowledge:** A typical design and manufacturing firm handles large number of changes per month. Once a change is implemented, the knowledge related to it is archived in the database for future use in the evaluation process. The incorporation of new changes into the knowledge-base might change the IAS. However, for the efficacy of evaluation process, the approach should exploit the past knowledge about IAS in determining the new IAS.

As discussed in section 2.2.1, the framework of ACO metahueristic [28] is selected to build our search approach. The conceptual entity that generates and maintains a solution is referred to as *agent*. To ensure that the solution does not get stuck at a local optima, the ACO metahueristic utilizes multiple agents in each cycle to build several solutions. The metaheuristic methods do not guarantee that the solutions determined shall be consistent with all the hard constraints [74]. This issue is addressed by integrating a constraint solver with the ACO framework.

The agents are divided into two groups, namely Z and U, to handle multiple objectives [75, 28]. The goal of agents in Z is to maximize Π^Z and the goal of agents in U is to maximize Π^U. Two agents, one from each group, pair-up together to generate a solution as per the procedure that we refer to as the *tandem solution generation*. The details about this procedure is discussed in section 4.5.1. Each pair of agents can exchange the information among themselves during the selection process. This shall allow handling multiple objectives of the problem. In each cycle, once all the solutions are generated by various pair of agents, the global best solutions are identified by combining the two objective functions using an arithmetic mean.

The ACO framework permits the exploitation of past knowledge and the exploration of search space. In the existing ACO algorithms, the randomness in selection is typically utilized for the exploration of search space. On the contrary in our approach the probabilistic constraints are utilized to guide the exploration.

Two vectors, which are referred to as the *usefulness value* vectors, are defined to exploit the past knowledge. The *usefulness value* of an attribute refers to the past knowledge about its utility in building good, i.e., optimal or near-optimal, solutions. The higher the usefulness value, the more useful an attribute is based on the past knowledge about the optimal solutions. For example, if it is known that each time the effect entity is *Process*, the *material.name* is an important attribute, then *material.name* will have high usefulness value for evaluating impact on *Process*. To build the solution at a current time step, say t, the usefulness value vectors from $t-1$ are utilized. Let τ_i^{t-1} and υ_i^{t-1} denote the usefulness values of an attribute x_i in maximizing Π^Z and Π^U, respectively, at $t-1$. At the end of each cycle, the usefulness values are updated based on following equations [76],

$$\tau_i^t = \begin{cases} (1-\rho) \times \tau_i^t + \rho \times \Pi^{bs} & \text{if } x_i \in Y; \\ \tau_i^t & \text{otherwise ;} \end{cases} \tag{4.22}$$

$$\upsilon_i^t = \begin{cases} (1-\rho) \times \upsilon_i^t + \rho \times \Pi^{bs} & \text{if } x_i \in Y; \\ \upsilon_i^t & \text{otherwise ;} \end{cases} \tag{4.23}$$

where, $\rho \in [0,1]$ is an importance value update parameter and Π^{bs} is the average sum of Π^Z and Π^U for the global best solution. The procedure terminates once the maximum number of cycles Q (specified by user) have been executed.

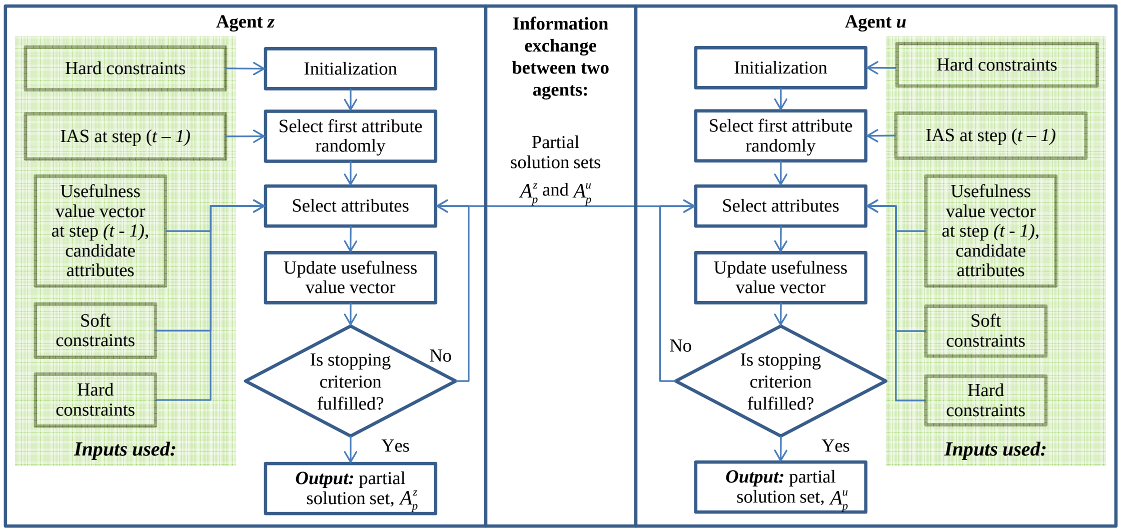

Agent z
Hard constraints
Initialization
IAS at step (t – 1)
Select first attribute randomly
Usefulness value vector at step (t - 1), candidate attributes
Select attributes
Update usefulness value vector
Soft constraints
Is stopping criterion fulfilled?
No
Hard constraints
Inputs used:
Yes
Output: partial solution set, A_p^z
Information exchange between two agents:
Partial solution sets
A_p^z and A_p^u
Agent u
Hard constraints
Initialization
Select first attribute randomly
IAS at step (t – 1)
Select attributes
Usefulness value vector at step (t - 1), candidate attributes
Update usefulness value vector
No
Is stopping criterion fulfilled?
Soft constraints
Hard constraints
Yes
Inputs used:
Output: partial solution set, A_p^u

Figure 4.1: Tandem solution generation procedure followed by a pair of agents. Agent z is from group Z and agent u is from group U. The Important Attribute Set determined by the pair z and u is the set $A_p^z \cup A_p^u$. The two agents exchange information during the iterative selection process

4.5.1 Tandem solution generation

Figure 4.1 illustrates the solution generation method followed by a generic pair of agents z and u. The agent z is from group Z and agent u is from group U. Each agent follows an iterative procedure to identify the important attributes. The dynamically growing set of important attributes is referred to as the *partial solution set* and is denoted as A_p^z corresponding to agent z. The set of attributes that are feasible for inclusion into A_p^z is referred to as the *feasible attribute set*, denoted as A_f^z. Let V^z and V^u represent the solution generated by agents z and u, respectively. The solution vectors can be determined from the partial solution sets using the inverse of f as $V^z = f^{-1}(A_p^z)$ and $V^u = f^{-1}(A_p^u)$.

The input to the tandem solution generation step is the candidate set of attributes X, the hard and soft constraints, the important attribute set and the usefulness values at time $t-1$. The output is the important attribute set, which is $A_p^z \cup A_p^u$, identified by agents z and u. In the initialization step, the variables in V^z and V^u are assigned a value such that the constraints shown in equation (4.4) are satisfied. To explore the search space each agent in a group starts its selection process from a different attribute. Therefore, the first attribute is chosen randomly from the IAS at time $t-1$, while ensuring that each agent in a group picks a different attribute. Once the first attribute is selected, the remaining attributes are selected iteratively until a stopping criterion is reached. Depending on the set of mutually inclusive constraints, more than one attributes might be selected in an iteration. Following section presents the details about the iterative selection procedure.

Selection procedure

In a particular iteration, if the increase in the objective function is not above a desirable level, then the probabilistic constraints are utilized to guide the search process. We shall allow an user to specify the minimum desirable change in the objective function using the parameter $q_0 \in [0,1]$. The decision about value of q_0 can be based on the number of instances in the change database. If the enterprise specific knowledge is small, i.e., there exists few instance in the change database, then the user will want to rely more on the probabilistic constraints, since these constraints model the domain knowledge. For such cases the value of q_0 can be above mean.

Before further discussing the selection policy, quantities that are utilized in it shall be derived. Let $P = \{P_j : 1 \leq j \leq |l|\}$ represent the l candidate attribute sets for agent z in a particular iteration. The candidate attribute sets is obtained from A^z_f using the mutually inclusive and exclusive constraints shown in equations (4.5) and (4.6). Let γ_j denote the probability that the attribute set $A^z_p \cup P_j$ is important.

This probability is determined based on the mean probability of all its elements as,

$$\gamma_j = \frac{\sum_{x_i \in \left\{A_p^z \cup P_j\right\}} c^Z(x_i)}{|A_p^z \cup P_j|} \tag{4.24}$$

Let τ_j^t represent the average usefulness value of all the attributes in partition P_j. To exploit the past knowledge the τ_j^t are aggregated with the γ_j. The aggregation function should be such that an enterprise has the flexibility to assign appropriate weights to τ_j^t and γ_j value depending on the units for time step t. For example, if the units of t is small (say, one week) so that the change in the state of knowledge-base is insignificant, then an enterprise might assign a higher weight to usefulness value, which is based on past knowledge about optimal solution. Accordingly, the τ_j^t and γ_j are aggregated into a single quantity, φ_j, as follows:

$$\varphi_j = \beta \times \tau_j^t + (1 - \beta) \times \gamma_j \tag{4.25}$$

where, β is a parameter that determines the relative influence of usefulness value as compared to the other quantity, i.e., γ_j in this case. Similarly, the objective function and usefulness values are aggregated using into a single quantity, η_j, as shown below,

$$\eta_j = \beta \times \tau_j^t + (1 - \beta) \times \Pi^Z(A_p^z \cup P_j) \tag{4.26}$$

The decision about selecting an attribute set for inclusion in A_p^z is based on following policy,

$$P_s = \begin{cases} \underset{P_j \in P}{\arg\max}\ \eta_j & \text{if } \exists P_j : \Pi^Z(A_p^z \cup P_j) - \Pi(A_p^z) \geq q_0; \\ \underset{P_j \in P}{\arg\max}\ \varphi_j & \text{if } \exists P_j : 0 \leq \Pi^Z(A_p^z \cup P_j) - \Pi(A_p^z) < q_0; \end{cases} \tag{4.27}$$

where, P_s represents the selected attribute set. The first case in equation (4.27) allows selection of an attribute set based on the objective function and the past knowledge; whereas the second case enables the exploration of search space based on the past knowledge and the probabilistic constraints.

If there is no positive change in the value of Π^Z during an iteration, then it implies that the addition of new attributes will not be useful. Thus, agent z stops adding attributes into A^z_P once

$$(\Pi^Z(A^z_p \cup P_j) - \Pi^Z(A^z_p)) \leq 0, \forall P_j \in P \tag{4.28}$$

To facilitate exploration of attributes that have not been visited yet, as soon as an attribute is added to the solution set by z its importance trail value is decreased. Accordingly following linear interpolation equation [76] is utilized to update the importance value of the selected attribute, say x_i,

$$\tau^t_i = (1 - \xi) \times \tau^t_i + \xi \times \tau^{t-1}_i \tag{4.29}$$

where, ξ takes a value in $[0, 1]$ and can be used to control the amount of exploration.

4.6 Case study

Figure 4.2 illustrates an example proposed change in shape from dataset # 8. This proposed EC is same as $EC - 4$, shown in Figure A.5 of Appendix A.2, from our example database. Consider the task of determining important attributes of proposed EC which should be used to retrieve similar past ECs, so that the impact of proposed EC on *Process* can be evaluated. Following section discusses the knowledge-base available for determining important attributes.

4.6.1 Knowledge-base

The knowledge-base of dataset # 8 contains 17 past ECs, listed in Appendix A.3, of type change in shape. The *Process* is the effect entity in each of the 17 past

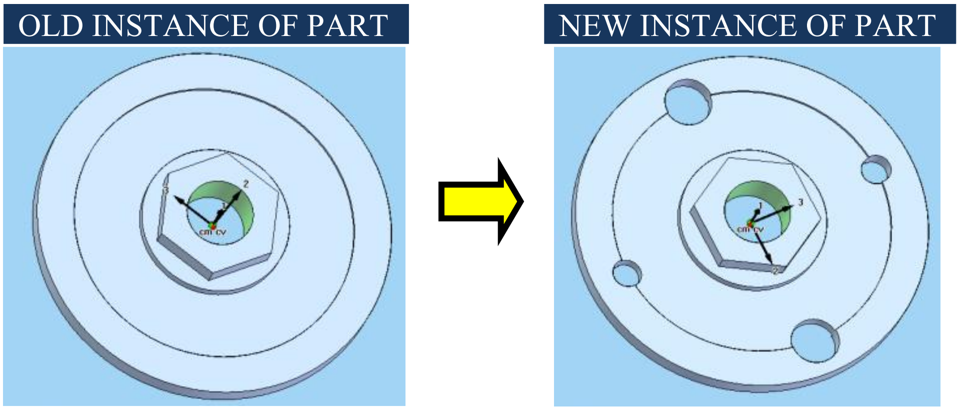


Figure 4.2: Example proposed EC from dataset # 8. This proposed EC is same as $EC - 4$ from our example database

changes. The knowledge about the proposed and each past ECs in database is captured using the data model, shown in Appendix A.1, which has 100 attributes. Among the 100 attributes, 82 attributes are the candidates for being important. The remaining 18 attributes are not candidates for being important, since its value in proposed EC is unknown. The 82 candidate attributes are listed in Appendix B.2. The numeric attributes in each dataset are discretized using a popular discretization approach called the proportional k-interval discretization technique [19]. As seen in Appendix B.2, each attribute is given an unique integer id that shall be utilized while specifying the constraints. The knowledge-base also contains a set of 10 *if-then* rules, which are are shown in Appendix B.1. These rules are derived from CES and [69].

4.6.2 Constraints

The problem of determining important attributes of example EC is modeled as the Multi-objective Optimization Problem. The two objectives of optimization problem are to maximize Π^Z and Π^U, as defined by equations (4.14) and (4.15), respectively. This section discusses the constraints on the optimization problem.

Hard constraints

Based on the discussion in section 4.3.3, in order to compute the similarity between proposed EC and each past EC, so that the impact of proposed EC effect on *Process* can be evaluated, it is essential to compare the attributes associated with the *Shape* and *Process* entities before the change, and the attributes associated with the *Shape* entity after the change. These requirements are captured using following three constraints,

$$v_i = 1; i = 30, 45, 76 \tag{4.30}$$

where, i refers to the unique integer id of a candidate attribute.

A partial illustration of the matrix t^{in}, which captures the information about attributes that are mutually inclusive, is shown in Figure B.1 of Appendix B.3. There are 20 rows in the matrix t^{in} which have atleast two non-zero value. As a result, there are 20 mutually inclusive constraints relevant to this case study. Each constraint is modeled using equation (4.5). For example, consider the 10^{th} row in the matrix t^{in}. Based on the values in this row, a mutually inclusive constraint is modeled as,

$$\text{if } v_{10} = 1, \text{ then } v_{10} + v_{11} = 2 \tag{4.31}$$

A partial illustration of the matrix t^{ex}, which captures the information about attributes that are mutually exclusive, is shown in Figure B.2 of Appendix B.3. There are 66 rows in the matrix t^{ex} which have atleast one non-zero value. Based on this information, there are 66 mutually exclusive constraints relevant to this case study. Equation (4.6) is utilized to model each mutually exclusive constraint. For example, consider the 10^{th} row in the matrix t^{ex}. Based on the values in this row, a mutually exclusive constraint is modeled as,

$$\text{if } v_{10} = 1, \text{ then } v_{56} = 0 \tag{4.32}$$

Soft constraints

Based on the rules that have impact in the consequent, following six probabilistic constraints are identified,

1. $p(v_{55} = 1) = 0.125$
2. $p(v_{34} = 1) = 0.375$
3. $p(v_{45} = 1) = 0.167$
4. $p(v_{76} = 1) = 0.167$
5. $p(v_{39} = 1) = 0.083$
6. $p(v_{42} = 1) = 0.083$

The above six probabilistic constraints are utilized for maximizing $\Pi^Z(S)$. Similarly, based on the rules that have effect attributes in the consequent, following six probabilistic constraints are identified which are utilized for maximizing $\Pi^U(S)$,

1. $p(v_{57} = 1) = 0.248$
2. $p(v_{60} = 1) = 0.248$
3. $p(v_{59} = 1) = 0.126$
4. $p(v_{55} = 1) = 0.192$
5. $p(v_{58} = 1) = 0.122$
6. $p(v_{61} = 1) = 0.067$

4.6.3 Parameter settings

Table 4.1 summarize the parameters in our approach and their values used for the case study.

Parameter	Description	Value used for the evaluation
λ^Z	importance of reducing the negative effect of S in predicting the value of U	0.8
λ^U	importance of reducing the negative effect of U in predicting the value of S	0.8
a	influence of rules in the determination of joint probability value	1.0
b	influence of rules in the determination of conditional probability value	1.0
β	relative influence of usefulness value as compared to the probability of importance of an attribute set	0.5
q_0	minimum desirable change in the objective function	0.8
ρ	global importance value update parameter	0.1
ξ	local importance value update parameter	0.1
τ^0	initial attribute usefulness value	0.01
n^0	number of agents at time step $t = 1$	10
Q	number of search cycles	5

Table 4.1: Parameters in our approach and their values used for the case study and evaluation

λ^Z and λ^U are the important parameters in the measures. A good value for these parameters was determined based on a simple experiment. In this experiment, the loss in the prediction of impact, effect attributes, and impact and effect attributes were determined for six different values of λ^U. Figure 4.3 illustrates the variation in the values of loss in prediction of impact, effect attributes, effect attributes and impact with the change in the value of λ^U. As seen, the mean loss in prediction of process and impact is relatively smaller when the values of λ^U is less than 0.2 or greater than 0.6. Among the two ranges $[0.0, 0.2]$ and $[0.6, 1.0]$, the later is chosen for the case study to ensure that the attribute selected for the prediction of impact does not affect to a large extent the prediction in the value of effect attributes and vice-versa. From the range $[0.6, 1.0]$, a mean value of 0.8 is selected for λ^U as well as λ^Z.

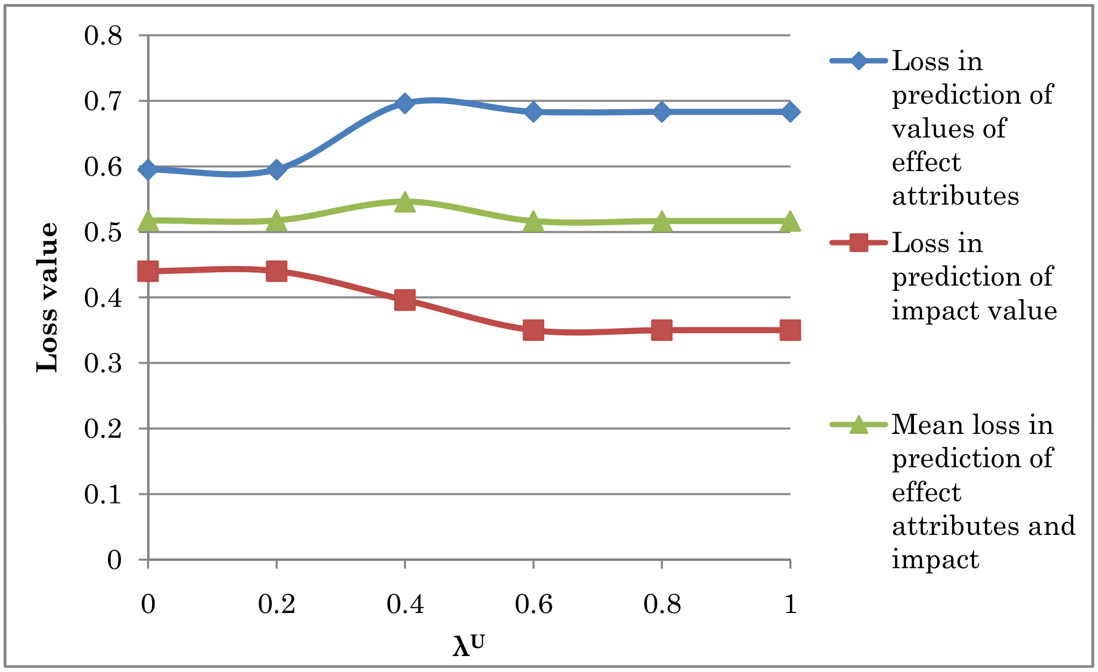


Figure 4.3: Variation of (a) loss in prediction of effect attributes, (b) loss in prediction of impact and (c) the mean loss in prediction of effect attributes and impact with the change in the value of λ^U. Based on this plot, a good value of λ^U shall be less than 0.2 or greater than 0.6

The change database and the rules among the attribute values are considered to be of equal importance in determining the joint and conditional probability distributions. Therefore, both a and b are assigned a value of 1.0. In the absence of information about the units of time step, equal weight is assigned to the usefulness value and the probability of importance of an attribute set in the equation (4.25).

Important Attribute Set
old_configuration.part.shape.features, old_configuration.part.process.name, old_configuration.part.process.tool.id, new_configuration.part.shape.features, new_configuration.part.production_rate.unit_quantity, new_configuration.part.surface_finish.value, new_configuration.part.tolerance_range.upper_limit, new_configuration.part.tolerance_range.lower_limit

Table 4.2: Important attribute set for proposed changes in dataset # 8 determined using our overall approach

Accordingly, value of β is chosen as 0.5. The values for the remaining parameters are chosen same as those utilized in a typical application of ACO algorithm to the Traveling Salesman Problem (TSP) problem [28].

4.6.4 Results

Table 4.2 summarize a set of 8 attributes of proposed engineering change which are identified as important using our overall approach. It can be verified with manual observation that each of the 8 attributes identified as important shall be useful to determine similar ECs so that the impact of proposed change effect can be evaluated. The 2 important attributes, namely *old_ configuration.part.shape.features* and *old_ configuration.part.process.name*, capture the information about the effect entity - process and the change entity - shape before the change. These attributes are important for determining past changes that are similar to the proposed change with a goal of evaluating the proposed change effect. The other 2 attributes, namely *new_ configuration.part. production_ rate.unit_ quantity* and *old_ configuration.part.process.tool.id* influence the impact of effect on the process due to change in shape. The remaining attributes are typically utilized for selecting process, which is the effect entity, of a product [77, 78].

It should be noted that our overall approach is independent of attribute values in proposed EC. Therefore, the same set of attributes will be identified as important for other proposed ECs in the dataset # 8.

4.7 Evaluation

This section presents the evaluation of our measures and the overall approach for determining the important Engineering Change attributes. Our two interrelated measures, shown in equations (4.14) and (4.15), quantify importance of an attribute set in predicting values of impact and effect attributes. Our overall approach identifies important attributes that should be compared to determine similar ECs with the goal of ultimately using this knowledge in evaluating the impact of proposed EC effect.

10 datasets discussed in Appendix A.3 are utilized for evaluation. The list of candidates for important attributes in all datasets is shown in Appendix B.2. The numeric attributes in each dataset are discretized using a popular discretization approach called the proportional k-interval discretization technique [19]. The domain knowledge and the constraint matrices utilized for evaluation are presented in Appendix B.1 and B.3, respectively. The parameter values shown in Table 4.1 are utilized for evaluation.

4.7.1 Evaluation of proposed measures

Our measures are evaluated against a state-of-the-art filter evaluation criterion, namely Information Gain Ratio (IGR), and two state-of-the-art wrapper evaluation criteria, namely Decision tree classifier and Naïve Bayes classifier. These three state-of-the-art approaches have been commonly used for various attribute selection problems. An implementation of these three state-of-the-art approaches is available in WEKA [21].

Strategy

All the 10 datasets shown in Appendix A.3 are utilized for evaluation of measures. Each dataset contains 17 training and 6 test instances. For each dataset, the Important Attribute Set (IAS) is determined based on the training instances. The test as well as the training instances are used to determine how well IAS, identified using our measures or state-of-the-art approaches, predict the values of impact and effect attributes. For each change instance, a set of training instances are identified that have the value for IAS same as itself. The identified set of training instances are utilized to compute the probability of each value in the domain of impact and effect-attributes. The resulting probabilistic predictions are evaluated using quadratic loss function, which is a popular function for evaluating the probabilistic predictions [19]. For a given dataset, let LV^{Z}_{test} and LV^{U}_{test} represent the average loss values in predicting values of impact and effect attributes, respectively, for all the test instances. The two loss values are aggregated using arithmetic mean,

$$LV_{test} = \frac{LV^{Z}_{test} + LV^{U}_{test}}{2} \tag{4.33}$$

Similarly, the average loss values in predicting values of impact and effect attributes for all training instances are aggregated using arithmetic mean. Let $LV_{training}$ represent the average loss values in predicting impact and effect attributes for all the training instances. In 0.632 bootstrap, the overall loss value, denoted as LV, is computed as,

$$LV = 0.632 \times LV_{test} + 0.368 \times LV_{training} \tag{4.34}$$

The loss value is an important statistical performance metric in evaluating various

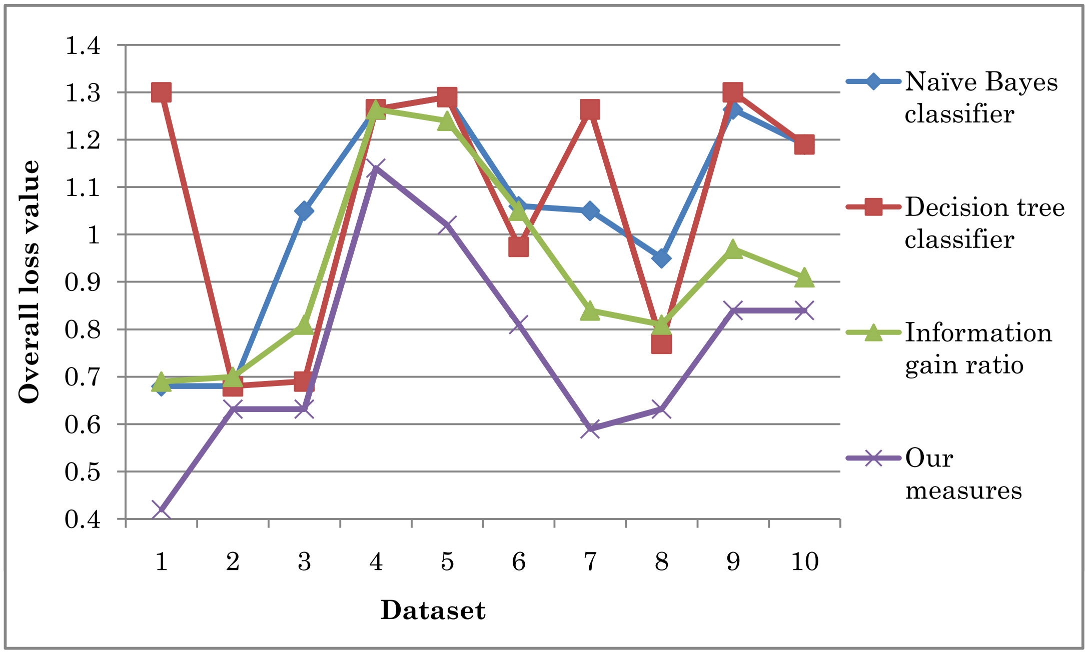


Figure 4.4: Overall loss in the predicting impact and effect attributes values using various subset evaluation criteria

approaches to quantify importance of an attribute set. An approach with the lowest loss value is considered to be superior to all other approaches.

Implementation and results

The minimum cross entropy formulation discussed in section 4.4.1 is encoded in Matlab. The minimum cross entropy formulation requires solving a constrained non-linear optimization problem. The Matlab's implementation of interior-point algorithm is selected to solve our constrained non-linear optimization problem [79]. The remaining computations are performed using MS Excel and standalone programs written in Visual C++.

For all 10 datasets, Appendix B.4 summarizes the IAS determined using various subset evaluation approaches. The overall loss values for various datasets are summarized in Figure 4.4.

Analysis

As seen in Figure 4.4, for each dataset the aggregated loss value from our measure is less than the loss value from other subset evaluation criteria. The average loss value from our measure is 0.7556, whereas the average loss value from Naive Bayes classifier, Decision tree classifier and IGR is 1.0478, 1.0722 and 0.9284, respectively.

Significance test

In order to determine whether the difference in the loss values between our approach and other approaches are statistically significant, and not due to a chance effect in the estimation of loss, a significance test was carried out using the results of 10 bootstrap runs. Since each of the 10 datasets are derived from a single database, the corrected resampled t-test [19] was chosen for the significance test. In this test, the t-statistic is determined using equation,

$$t = \frac{\bar{d}}{\sqrt{\left[\frac{1}{k} + \frac{n_2}{n_1}\right] \times \sigma_d^2}} \tag{4.35}$$

where, $\bar{d}$ is the mean difference in the loss values from our measures and a state-of-the-art measure, σ_d^2 is the variance of the difference in the loss values, n_1 is the number of training instances, n_2 is the number of test instances, and k is the number of times the test is repeated. For our experiment, $k = 10$, $n_1 = 17$, $n_2 = 6$, $\bar{d}$ and σ_d^2 are based on the 10 differences in the loss values presented in Figure 4.4.

Following the corrected resampled t-test, it is determined with 99% confidence that there is a statistically significant difference between the loss values from our measures and Naive Bayes classifier or IGR. Similarly, it is determined with 80% confidence that there is a statistically significant difference between the loss values from our measure and the Decision tree classifier.

Effect size

A standardized effect size statistic, called d statistic [80], was utilized for determining how large is the difference in the expected loss values from our measures and each of the state-of-the-art measures. The value of d statistic for the difference in means between our measure and Naive Bayes or Decision tree classifier is 1.3, and its value for the difference in means between our measure and IGR is 0.8. Based on the d statistic values, it is inferred that the expected loss value from our measures is less than the expected loss value from Naive Bayes classifier (or Decision tree classifier) by 1.3 times the pooled standard deviation in the loss values from the two approaches. Similarly, the expected loss value from our measures is less than the expected loss value from IGR by 0.8 times the pooled standard deviation in the loss values from the two methods.

4.7.2 Evaluation of overall approach to determine important attributes

Results

For all 10 datasets, Appendix B.5 summarizes the Important Attribute Set (IAS) determined using our overall approach. The problem of determining important attributes of an EC has not been addressed before. As a result, there lacks a state-of-the-art approach against which our overall approach can be evaluated. Therefore, the results of using our overall approach for one the 10 datasets will be analyzed based on the manual observation.

Analysis

To determine whether the IAS identified using our approach is suitable for determining similar ECs so that the impact of proposed change effect can be evaluated, we shall follow a majority voting procedure discussed in [11] to evaluate the impact on *Process* of each test instance in a given dataset. In majority voting procedure, for each attribute in the IAS, 2 past changes, i.e., training instances, are determined that have a value most similar to the value in the test instance. Thus, if there are n important attributes, then $2n$ changes will be identified as similar. It should be noted that a past change might occur several times in the $2n$ changes, if it is similar to the test instance based on more than one attributes. If the impact on *Process* is *high* in majority of the similar past changes, then the proposed change effect is considered to be *high*; otherwise the impact is *low*. For this analysis, the similarity between two values of an quantitative attribute is determined based on the euclidean distance; whereas manual observation is used to determine similarity between two values of a qualitative attribute.

For example, consider the test instance $EC-4$ from dataset # 1. Table B.14 of Appendix B.5 summarize the 9 attributes that are identified as important for changes in dataset # 1. Based on the technique discussed above it is found that the 18 past change instances from dataset # 1 that are similar to $EC-4$ include 8 instances of $EC-2$, 4 instances of $EC-11$, 3 instances of $EC-1$, 2 instances of $EC-12$ and 1 instance of $EC-3$. These changes are shown in Appendix A.2. There are 12 instances out of these 18 that have an impact value of *low*. As a result, the impact of proposed EC is predicted as *low*, which is actually the case. Similarly, the impact of other 5 test instances from dataset # 1 is evaluated. It is found that 5 out of 6 test instances from dataset # 1 are correctly predicted. The resulting success rate is 83.33%. It is further determined that for a very large number of changes (i.e., $N > 100$) the true success rate using our overall approach will lie between 50% and 96% with 90% confidence. The range for success rate is large since the number of instances in test set is small.

4.8 Summary and Future work

Summary

This chapter discussed the problem of identifying the important attributes that should be compared to retrieve past engineering changes, which are similar to the proposed change, so that the impact of proposed change effect can be evaluated. The problem of identifying the important attributes of EC is formulated as the Multi-objective Optimization Problem (MOOP). The attribute interrelationships are captured in the form of hard and soft constraints on the optimization problem. Two interrelated measures are defined to quantify importance of an attribute set. The observed distribution and the domain knowledge that is encoded in the form of rules among the attribute values are utilized for estimating the probability distributions required in the computation of measures. A search strategy that is based on the framework of the ACO metaheuristic is developed to efficiently locate the IAS.

An example knowledge-base is utilized for evaluating our measures and approach for determining important attributes of EC. Our measures are compared against three state-of-the-art evaluation criteria. The loss in the prediction of impact and effect attributes is utilized as a criterion for comparison. The evaluation results show that there is a statistically significant reduction in loss value from our measures as compared to other criteria. Furthermore, it is determined that the expected loss value from our measures is less than the expected loss value from Naive Bayes classifier (or

Decision tree classifier) by 1.3 times the pooled standard deviation in the loss values from the two approaches. Similarly, the expected loss value from our measures is less than the expected loss value from IGR by 0.8 times the pooled standard deviation in the loss values from the two methods. Our overall approach has a success rate of 83.33% in correctly predicting the value of impact for 6 proposed changes in an example dataset. It is further determined with 90% confidence that for a very large number of changes the true success rate shall lie between 50% and 96%.

Future work

The approach presented in this chapter has certain limitations. The search approach is based on heuristics, which requires users to specify values of various parameters. The future research on this topic should perform a sensitivity analysis to determine the optimal range for various parameters. The results of this shall serve as a guide to the users in selecting a good value for various parameters.

When mutually inclusive and exclusive relations are specified by multiple users, techniques for aggregating them will be required. There exists general frameworks, e.g., [81, 82], for aggregating binary evaluations from multiple individuals. The applicability of these frameworks to our problem will need to be studied in future. Another potential direction for extending this research is to allow users to associate a probability with mutually inclusive and exclusive relations. In this case, the use of fuzzy set theory to model mutually inclusive and exclusive constraints will need to be studied.

Currently, the problem and the approach discussed in this chapter does not account for the temporal changes in the database or the attribute values. For example, new attributes might be added into the data model or the specification of a material might change over a period of time. Such changes can affect the decision about the important attributes. Accounting for such temporal changes in the process of determining important attributes presents an interesting avenue of future research.

CHAPTER V

Computing similarity between engineering changes

This chapter presents an approach to compute similarity between Engineering Changes (ECs) with the goal of ultimately using this knowledge in evaluating the impact of proposed EC effect.

5.1 Motivation

Utilizing past ECs to predict the impact of proposed EC effect requires an approach to compute similarity between ECs. Since the data about an EC is captured using a set of attributes, computing similarity between ECs will require an approach to compute similarity between its attribute values. The overall goal is to predict the impact of proposed EC effect. Therefore, the similarity between changes or attribute values should be computed in context of predicting impact of proposed EC effect. The approach to compute similarity between attribute values should be suitable for disparate attribute types, since an EC incorporates attributes of both quantitative and qualitative types. The existing approaches [22, 11] in the area of ECM do not focus on determining similarity in context of impact. In addition, these approaches are suitable for ECs that are defined using a few specific attributes.

As discussed in section 1.5, the approach to compute similarity should account for variations in the similarity perception among enterprises or within an enterprise over a period of time. The classical approaches [41, 43] to compute similarity do not account for the possible alterations in the similarity perception, nor do they consider the disparity of the attributes.

The information available in EC knowledge-base to determine similarity in context of impact between two attribute values is the observed probability distributions of impact given the value of attribute. It is unknown whether an observed probability distribution conforms to the actual distribution of impact associated with the attribute values. Therefore, in addition to utilizing observed probability distributions of impact, the approach to compute similarity should utilize semantics associated with attribute values. An ordered probability-based approach [45] considers the disparity of the attributes and accounts for the possible alterations in the similarity perception. It, however, does not utilize the semantics associated with attribute values, nor does it compute similarity in context of predicting impact.

5.2 Objective

The objective of this research phase is to develop an approach, which

For: a proposed EC

Given: database of past ECs and other relevant domain knowledge

Computes: similarity between the proposed change and each past EC with the goal of ultimately using this knowledge in predicting the impact of a proposed change effect

In addition to the assumptions discussed in section 4.2 about the nature of knowledge-base that stores past ECs, following paragraph discusses an assumption made in solving the aforementioned problem.

- EC knowledge is captured using a large number (in hundreds) of disparate and interdependent attributes. As discussed in chapter I, only some of these attributes should be used for similarity computation. It is assumed that a set of EC attributes based on which the similarity between ECs is computed is known. Chapter IV presents an approach to determine important attributes of an EC which should be used to determine similar ECs.

5.3 Attribute values similarity

As mentioned earlier, computing similarity between two ECs requires similarity between its attribute values. This section discusses our approach to compute similarity between attribute values. Following section presents our approach for aggregating attribute value similarities to compute the overall similarity between changes.

Consider an EC attribute X, which can be of qualitative or quantitative type. As discussed in section 5.1, the similarity between values A and B of X should be determined in context of predicting impact as well as based on the semantics associated with values A and B. The information available in EC knowledge-base to determine the similarity in context of impact between values A and B is the conditional probability distributions of impact given the values A and B. In addition, the information available in EC knowledge-base to compute similarity between values A and B based on the semantics is the probability of occurrence of attribute values. Since the available information is probabilistic, the concept of information will be utilized for quantifying the similarity between two values of an attribute.

Consider following notations:

- $s_p(A, B)$: similarity between values A and B determined in context of predicting impact

- $s_s(A, B)$: similarity between values A and B determined in context of semantics associated with values A and B
- $P_{Z|A}$: conditional probability distribution of impact given the value A
- $P_{Z|B}$: conditional probability distribution of impact given the value B
- $I_p^c(A, B)$: amount of information that is common to $P_{Z|A}$ and $P_{Z|B}$
- $I_p^d(A, B)$: information that is different in values $P_{Z|A}$ and $P_{Z|B}$
- $I_p^t(A, B)$: total information in $P_{Z|A}$ and $P_{Z|B}$. The total information is assumed to be the sum of $I_p^c(A, B)$ and $I_p^d(A, B)$, i.e., $I_p^t(A, B) = I_p^c(A, B) + I_p^d(A, B)$, since knowing the differences and commonalities in $P_{Z|A}$ and $P_{Z|B}$ is same as knowing $P_{Z|A}$ and $P_{Z|B}$ [46]
- $I_s^c(A, B)$: information that is common to values A and B
- $I_s^d(A, B)$: information that is different in values A and B
- $I_s^t(A, B)$: total information in values A and B. The total information is assumed to be the sum of $I_s^c(A, B)$ and $I_s^d(A, B)$, i.e., $I_s^t(A, B) = I_s^c(A, B) + I_s^d(A, B)$, since knowing the differences and commonalities in A and B is same as knowing A and B [46]

In principle, the similarity between two objects depends on both the commonalities and the differences [44]. Based on this principle, it is assumed that the $s_p(A, B)$ is the function of $I_p^c(A, B)$ and $I_p^d(A, B)$. In particular,

$$s_p(A, B) = \frac{I_p^c(A, B)}{I_p^t(A, B)} \tag{5.1}$$

Similarly, it is assumed that [46],

$$s_s(A,B) = \frac{I_s^c(A,B)}{I_s^t(A,B)} \tag{5.2}$$

The similarity values between A and B based on semantics and in the context of predicting impact are aggregated using the weighted sum to determine the overall similarity, $s(A,B)$, between A and B as,

$$s(A,B) = w_s \times s_s(A,B) + w_p \times s_p(A,B) \tag{5.3}$$

where, w_s and w_p are the weights associated with s_s and s_p, respectively. For the same values of $I_s^c(A,B)$ and $I_p^c(A,B)$, greater is the value of $I_s^t(A,B)$ as compared to $I_p^t(A,B)$ lower will be the value of s_s as compared to s_p. This effect is compensated by taking the weights in the equation (5.3) as proportional to the total information. That is,

$$w_s = \frac{I_s^t(A,B)}{I_s^t(A,B) + I_p^t(A,B)} \tag{5.4}$$

$$w_p = \frac{I_p^t(A,B)}{I_s^t(A,B) + I_p^t(A,B)} \tag{5.5}$$

In following paragraphs, the quantities $I_p^c(A,B)$ and $I_p^t(A,B)$ are derived such that the similarity is computed in context of predicting impact.

5.3.1 Similarity in context of predicting impact

The average information content in a probability distribution $P = \{p_1, \ldots, p_n\}$ can be quantified as [83],

$$I(P) = -\sum_{k=1}^{n} p_k \times \log p_k \tag{5.6}$$

Based on equation (5.6), the total information content in the probability distributions $P_{Z|A} = \left\{p_{z_1|A}, p_{z_2|A}\right\}$ and $P_{Z|B} = \left\{p_{z_1|B}, p_{z_2|B}\right\}$ is quantified as,

$$I_p^t(A, B) = -\sum_{k=1}^{2} \left(p_{z_k|A} \times \log p_{z_k|A} + p_{z_k|B} \times \log p_{z_k|B}\right) \tag{5.7}$$

The desirable characteristics of function $I_p^c(A, B)$ are,

1. $I_p^t(A, B) \geq I_p^c(A, B) \geq 0$, since the amount of information that is common or different cannot be negative.

2. $I_p^c(A, B) = I_p^t(A, B)$, if and only if $P_{Z|A} = P_{Z|B}$, since if the two probability distributions are exactly same, then there difference is 0.

3. $I_p^c(A, B)$ should be symmetric, since the commonalities in the distributions $P_{Z|A}$ and $P_{Z|B}$ is same as the commonalities in the distributions $P_{Z|B}$ and $P_{Z|A}$, i.e., $I_p^c(A, B) = I_p^c(B, A)$.

4. A generic distributions $P_{Z|X}$ can be shown on the line $p_{z_1|X} + p_{z_2|X} = 1$ as illustrated in Figure 5.1. The term $I_p^d(A, B)$ shall take up a maximum value in two cases: (1) $P_{Z|X=A} = \{1, 0\}$ and $P_{Z|X=B} = \{0, 1\}$ or (2) $P_{Z|X=B} = \{1, 0\}$ and $P_{Z|X=A} = \{0, 1\}$. These two cases are shown in Figure 5.1. For these two cases, following should hold: $I_p^d(A, B) = I_p^t(A, B)$. This ensures that the state of maximum difference is same as the state of no (or zero) commonality.

A function that satisfies all the four aforementioned properties is in the form,

$$I_p^c(A, B) = cos(\theta_{AB}) \times I_p^t(A, B) \tag{5.8}$$

where, $cos(\theta_{AB})$ is the cosine similarity metric defined as [84],

$$cos(\theta_{AB}) = \frac{\sum_{k=1}^{2} p_{z_k|A} \times p_{z_k|B}}{\sqrt{\sum_{k=1}^{2} (p_{z_k|A})^2 \times \sum_{k=1}^{2} (p_{z_k|B})^2}} \tag{5.9}$$

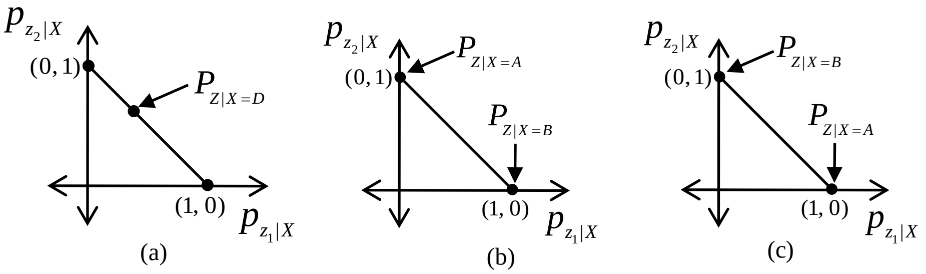


Figure 5.1: Illustration of a conditional probability distribution of impact given an attribute value. In Figure (a), the two coordinates of each point on the line from (0, 1) to (1, 0) represents the possible conditional probability distribution $\{p(Z = z_1|X), p(Z = z_2|X)\}$ of impact Z in the change instances that take the same value for an attribute X. For example, the probability distribution, denoted as $\left\{P_{z_1|X=D}, P_{z_2|X=D}\right\}$, of Z in the change instances that take the value $X = D$ is approximately $\{0.4, 0.6\}$. Figure (b) and (c) illustrate the two cases in which the two distributions $P_{Z|A}$ and $P_{Z|B}$ have maximum difference among them. For these two cases, the term $I_p^c(A, B)$ should zero.

Based on equations (5.1) and (5.8), $s_p(A, B)$ is determined as,

$$s_p(A, B) = cos(\theta_{AB}) \tag{5.10}$$

5.3.2 Similarity based on semantics

In following paragraphs, the equation to compute similarity based on semantics is derived for each attribute type.

Categorical attributes

The categorical attributes can be of simple or aggregate type. This section presents a measure for computing similarity between two values of categorical aggregate attribute. As discussed later in this section, the same measure with a little modification will be applicable to the categorical simple attribute.

Each value of a categorical aggregate attribute contains a combination of values from a list of primitive values. Let $\{x_i : 1 \leq i \leq ax, i \in \mathbb{N}\}$ represent the list of ax primitive values of a categorical aggregate attribute X. Let $\{a_i : 1 \leq i \leq aa\}$ and

$\{b_i : 1 \leq i \leq ab\}$ represent a set of aa and ab primitive values in A and B, respectively. Let $C = \{c_i : 1 \leq i \leq ac\}$ represent a set of ac primitive values that are common to both A and B.

Two forms of explicit semantics associated with a categorical aggregate attribute value, say A, are useful in computing similarity based on semantics. The first is the primitive values that define A. The second is the position of primitive values of A in an IS-A taxonomy, which relates all the primitive values of X. Accordingly, the similarity based on semantics between A and B has two components. The first component is based on the comparison of sets of primitive values that occur in A and B. The second component is based on the relative position of A and B in an IS-A taxonomy, which relates all the primitive values of X. For the sake of simplicity, the first component is referred as similarity based on primitive values and the second component is referred as similarity based on IS-A taxonomy.

Let $s_{s_p}(A, B)$ and $s_{s_t}(A, B)$ denote the similarity between A and B based on primitive values and IS-A taxonomy, respectively. For each of these two components, the equations to quantify the total and common information are different. Following section derives the total and common information terms for similarity based on primitive values. This is followed with derivation of total and common information terms for similarity in an IS-A taxonomy.

Similarity based on primitive values

Given the likelihood, $p(x_i)$, of value x_i, the information content of x_i can be quantified as negative the log likelihood, i.e., $-\log p(x_i)$ [40]. Based on the likelihood of primitive values of X, the total information in the aggregate values A and B can be quantified as,

$$I^t_{s_p}(A,B) = -\sum_{i=1}^{aa} \log p(a_i) - \sum_{i=1}^{ab} \log p(b_i) \tag{5.11}$$

The information that is common to A and B based on its primitive values is quantified as,

$$I^c_{s_p}(A,B) = \begin{cases} 0 & \text{if } C = \emptyset \\ -2 \times \sum_{i=1}^{ac} \log p(c_i) & \text{otherwise} \end{cases} \tag{5.12}$$

2 appears as a multiplicative factor in the second part of equation (5.12), since each element in C appears in both A and B. From equations (5.2), (5.11) and (5.12), $s_{s_p}(A,B)$ is defined as,

$$s_{s_p}(A,B) = \begin{cases} 0 & \text{if } C = \emptyset \\ \frac{2 \times \sum_{i=1}^{ac} \log p(c_i)}{\sum_{i=1}^{aa} \log p(a_i) + \sum_{i=1}^{ab} \log p(b_i)} & \text{otherwise} \end{cases} \tag{5.13}$$

Similarity based on IS-A taxonomy

There exist several categorical aggregate attributes whose primitive values can be related using a taxonomy of IS-A type. For example, consider the attribute *old_ configuration.part.process.name* with six primitive values: *{molding, casting, drilling, milling, planing, turning}*. Figure 5.2 illustrates an example taxonomy that relates the primitive values of the attribute *old_ configuration.part.process.name.* Based on the taxonomy shown in Figure 5.2, it can be inferred that the values *{drilling, milling}* is more similar to *{planing, turning}* as compared to *{casting}*, since the two previous values are of type *Machining process.*

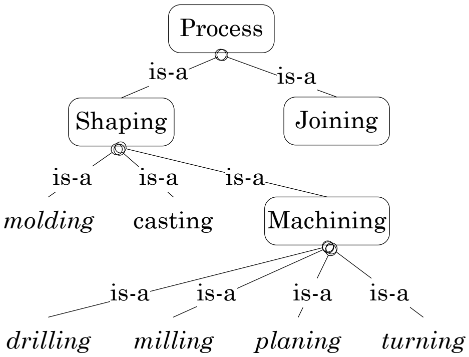


Figure 5.2: Example taxonomy of IS-A type for the primitive values in the domain of categorical aggregate attribute *old_ configuration.part.process.name*

The IS-A taxonomy that relates all the primitives values in the domain of an attribute can vary among the enterprises or within an enterprise over a period of time. Therefore, we allow an enterprise to specify this taxonomy for each relevant categorical aggregate attribute. Let $p(A)$ and $p(B)$ represent the probability of the values A and B, respectively. Let Q be the most specific parent node in the IS-A taxonomy which subsumes A and B. For example, if $A = \{drilling, milling\}$ and $B = \{drilling, casting\}$, then from Figure 5.2, $Q =$ *Shaping process.* The probability, $p(Q)$, can be determined by counting the occurrences of X with all the primitive values as the child of Q. Based on the probability of occurrence of aggregate values A and B, the total information content in values A and B can be quantified as,

$$I^t_{s_t}(A, B) = -\log p(A) - \log p(B) \tag{5.14}$$

whereas, the information content that is common to A and B based on the IS-A taxonomy can be quantified as [46],

$$I^c_{s_t}(A, B) = -2 \times \log p(Q) \tag{5.15}$$

From equations (5.2), (5.14) and (5.15), $s_t(A, B)$ is defined as,

$$s_{s_t}(A, B) = \frac{2 \times \log p(Q)}{\log p(A) + \log p(B)} \tag{5.16}$$

Aggregating similarities based on primitive values and IS-A taxonomy

The similarities between A and B based on primitive values and based on IS-A taxonomy are aggregated using the weighted sum to determine the overall similarity based on semantics between A and B as,

$$s_s(A, B) = w_{s_p} \times s_{s_p}(A, B) + w_{s_t} \times s_{s_t}(A, B) \tag{5.17}$$

where, w_{s_p} and w_{s_t} are the weights associated with s_{s_p} and s_{s_t}, respectively. For the same values of $I^c_{s_p}(A, B)$ and $I^c_{s_t}(A, B)$, greater is the value of $I^t_{s_p}(A, B)$ as compared to $I^t_{s_t}(A, B)$ lower will be the value of s_{s_p} as compared to s_{s_t}. This effect is compensated by taking the weights in the equation (5.17) as proportional to the total information. That is,

$$w_{s_p} = \frac{I^t_{s_p}}{I^t_{s_p} + I^t_{s_t}} \tag{5.18}$$

$$w_{s_t} = \frac{I^t_{s_t}}{I^t_{s_p} + I^t_{s_t}} \tag{5.19}$$

In case of categorical simple attributes, $s_s(A, B) = s_{s_t}(A, B)$. In case of categorical simple attribute whose primitive values do not have an associated meaning, the similarity value is either 1, if the values are identical, or 0, if the values are non-identical.

Ordinal and Quantitative values

The values of an ordinal or quantitative attribute are interrelated through a specific hierarchical order. For example, consider the ordinal attribute *old_ configuration. part.surface_finish.value* that has following three values in its domain: *very-smooth*, *smooth* and *rough*. It is known that the value *very-smooth* is more similar to *smooth* as compared to *rough*. Such information about hierarchical order is utilized to compute similarity based on the semantics. Given the likelihoods, the total information content of A and B can be quantified as $I_s^t(A,B) = -\log p(A) - \log p(B)$. The information content that is common to two ordinal values can be quantified as $-2 \times \log \sum_{i \in R} p(i)$, where R represents all the values that are on the path between the two values including the two values themselves [46]. The similarity based on semantics, $s_s(A,B)$, for ordinal attributes is defined as,

$$s_s(A,B) = \frac{2 \times \log \sum_{i \in R} p(i)}{\log p(A) + \log p(B)} \tag{5.20}$$

Since it is assumed that the continuous quantitative attributes are discretized in a pre-processing step, the equation (5.20) is also applicable to the quantitative attributes.

5.4 Engineering change similarity

This section presents our approach for aggregating attribute-values similarities to compute the similarity between ECs. Let $Y = \{y_i : 1 \leq i \leq ay, i \in \mathbb{N}\}$ represent the set of ay attributes based on which the similarity between changes is computed, and $y_i = \{y_{ik} : 1 \leq k \leq ay_i, k \in \mathbb{N}\}$ represent the ay_i values in the domain of an attribute y_i. Let s_{ji} denote the similarity between value of y_i in the proposed change and a past change c_j, and S_j denote the similarity between the proposed EC and c_j. The three required characteristics of a function used to aggregate attribute-values similarities are [85]:

1. *Preservation of bounds:* If all the attribute-values similarities are 0, then the similarity between corresponding ECs should be 0. That is, if $s_{ji} = 0 : 1 \leq i \leq ay$, then $S_j = 0$. Similarly, if all the attribute-values similarities are 1, then the similarity between corresponding ECs should be 1.

2. *Monotonicity:* As one of the attribute-values similarity increase while the remaining are same, the similarity between ECs should also increase. For example, consider two past changes c_1 and c_2, such that the first $ay - 1$ attribute-values similarities between each of these ECs and the proposed EC, c_0, is same, i.e., $s_{1i} = s_{2i} : 1 \leq i \leq ay - 1$. If the ay^{th} attribute-values similarity between c_0 and c_1 is greater than that between c_0 and c_2, then the similarity between c_0 and c_1 should be greater than the similarity between c_0 and c_2.

3. *Continuity at the boundary points:* None of the attribute-values similarities have a weight of 0. Thus, if there exist atleast one non-zero attribute-values similarity, then the similarity between corresponding ECs should be non-zero.

Considering the three aforementioned required characteristics, the similarity between the proposed change and c_j is computed by aggregating the corresponding attribute-values similarities using weighted sum,

$$S_j = \sum_{i=1}^{ay} w_i \times s_{ji} \tag{5.21}$$

where, w_i is the weight associated with the attribute y_i. Since the goal is to determine similarity between changes in the context of predicting impact, the weights are taken as proportional to the amount of information about impact in an attribute. The measure of Mutual Information (MI) (or Information Gain, as defined in chapter IV) between two attributes quantifies the amount of information about one contained in other [72]. Let $Z = \{z_l : 1 \leq l \leq az, l \in \mathbb{N}\}$ represent the az values in the domain of Z. The probability of occurrence of various combination of values of y_i and Z can be determined from the EC knowledge-base. Based on these probability values, the MI, denoted as I_i, between y_i and Z can be determined as [72],

$$I_i = \sum_{k=1}^{ay_i} \sum_{l=1}^{az} p(y_{ik}, z_l) \times \log \frac{p(y_{ik}, z_l)}{p(y_{ik})p(z_l)} \tag{5.22}$$

The MI values are utilized to compute weights as,

$$w_i = \frac{I_i}{\sum_{j=1}^{ay} I_j} \tag{5.23}$$

5.5 Case study

Figure 5.3 illustrates an example proposed EC from dataset # 8. Consider the task of determining the similarity between this proposed EC and following three past ECs from dataset # 8: $EC-1$, $EC-2$ and $EC-6$, shown in Figures A.2, A.3 and A.7, respectively, of Appendix A.2. The example proposed EC is same as $EC-4$, shown in Figure A.5 of Appendix A.2, from our example database. The quantitative attributes in the database are discretized using a popular discretization approach called the proportional k-interval discretization [19].

Table B.21 in Appendix B.5 illustrates the eight important attributes based on which the similarity between a proposed EC and a past EC from dataset # 8 should be computed. This set of important attributes are identified using the approach discussed in chapter IV. Out of these eight attributes, one, namely *new_ configuration.part. surface_finish.value*, is of ordinal type, three, namely *new_ configuration.part. production_ rate.unit_ quantity*, *new_ configuration.part. tolerance_ range.upper_ limit*

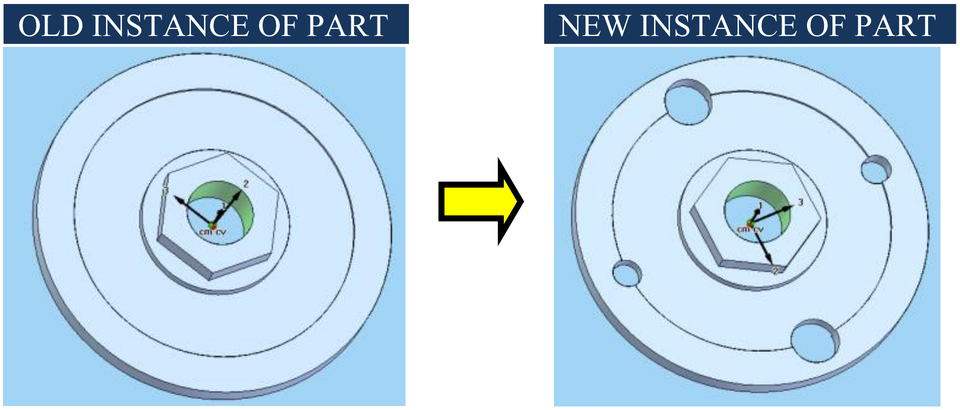


Figure 5.3: Example proposed EC from dataset # 8. This proposed change is same as $EC - 4$ from our example database

and *new_ configuration.part. tolerance_ range.lower_ limit*, are of quantitative type and the remaining four, namely *old_ configuration.part. shape.features*, *old_ configuration.part. process.name*, *new_ configuration.part. shape.features* and *old_ configuration.part. process.required_ machine.type* are of categorical aggregate type. The probability values required for the similarity computation are determined based on the 17 training instances in the dataset # 8.

5.5.1 Attribute values similarity

This section discusses the application of our approach for computing similarity between values of an attribute of each type.

Categorical values

Consider an categorical aggregate attribute *old_ configuration.part.process.name*. Each value of this attribute contains a combination of values from the following list of primitive values: *{molding, casting, drilling, milling, planing, turning}*. These primitive set of values are related to each other according to the taxonomy illustrated in Figure 5.2.

The values of *old_ configuration.part.process.name* in the proposed change, $EC-$

1 and $EC-6$ is $\{casting\}$, $\{casting, milling\}$ and $\{drilling\}$, respectively. The probability of occurrence of these aggregate values in the change database is 0.29, 0.06 and 0.12, respectively, and the conditional distribution of impact given these values is $0.8, 0.2$, $0.999, 0.001$ and $0.001, 0.999$, respectively. The most specific parent of these three values is *Shaping process*, whose probability of occurrence is 1.0. The probability of occurrence of primitive values are: $p(casting) = 0.35$, $p(milling) = 0.25$, $p(drilling) = 0.1$. Based on this information, the value of s_s, I_s^t, s_p and I_p^t between $\{casting\}$ and $\{casting, milling\}$ is 0.28, 3.27, 0.97 and 0.22, respectively, and the overall similarity between these values is 0.32. Similarly, the value of s_s, I_s^t, s_p and I_p^t between $\{casting\}$ and $\{drilling\}$ is 0.0, 2.92, 0.24 and 0.22, respectively, and the overall similarity between these values is 0.02. We can verify from observation that the value $\{casting\}$ is indeed more similar to the value $\{casting, milling\}$ as compared to the value $\{drilling\}$. Moreover, the manufacturing processes *drilling* and *casting* are unrelated in context of their semantics. Therefore, the similarity between the values $\{casting\}$ and $\{drilling\}$ should be close to zero.

Ordinal values

Consider the attribute *old_ configuration.part.surface_finish.value.* Figure 5.4 illustrates a graphical representation of adjacency relationship among the three values of this attribute. A link between two nodes in Figure 5.4 implies that the values are adjacent. The probability of occurrence of each value is also shown in Figure 5.4.

The value of *old_ configuration.part.surface_finish.value* in proposed change, $EC-1$ and $EC-2$ is *smooth*; whereas its value in $EC-6$ is *rough.* The probability of occurrence of values *smooth* and *rough* are 0.41 and 0.12, respectively. From the probability values, the information content I_s^t in the values *smooth* and *rough* is

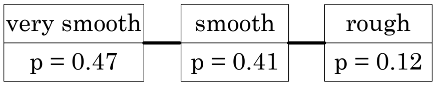


Figure 5.4: Graphical representation of the adjacency relationship among the values of attribute *old_ configuration.part.surface_finish.value*. A link between two nodes represents that the corresponding values are adjacent. The probability, p, of occurrence of each value is shown below it

1.31. As seen in Figure 5.4, apart from *smooth* and *rough* there are no additional nodes on the path from *smooth* to *rough*. Therefore, the information that is common to *smooth* and *rough* can be quantified as $I_s^c = -2 \times \log(0.41 + 0.12) = 0.55$, Based on the common and total information, the value of s_s between *smooth* and *rough* is 0.42. The similarity in context of impact between *smooth* and *rough* is determined as 0.99, and the overall similarity between these values is 0.6. Similarly, the similarity between values *smooth* and *very-smooth* is computed as 0.53; whereas the similarity between values *very-smooth* and *rough* is computed as 0.31. These results match well with the expectation that the similarity between two adjacent values, e.g., *very-smooth* and *smooth* or *smooth* and *rough*, must be greater than the similarity between two non-adjacent values, e.g., *very-smooth* and *rough*.

Quantitative values

To demonstrate the application of our approach for determining similarity between values of a quantitative attribute consider the attribute *old_ configuration.part. production_ rate.unit_ quantity*. Figure 5.5 illustrates a graphical representation of adjacency relationship among the four discretized labels in the domain of this attribute.

The value of *old_ configuration.part.production_ rate. unit_ quantity* in proposed EC is 27750, its value in $EC-1$ and $EC-6$ is 15250, and in $EC-2$ is 45000. Consider

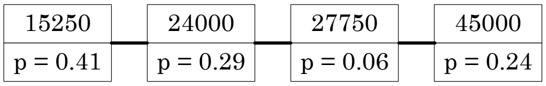


Figure 5.5: Graphical representation of the adjacency relationship among the values of attribute *old_ configuration.part.production_ rate.unit_ quantity*. A link between two nodes represents that the corresponding values are adjacent. The probability, p, of occurrence of each value is shown below it

the values 27750 and 15250. Based on the probability values, the information content I_s^t in values 27750 and 15250 is $I_s^t = 1.62$. From Figure 5.5, the various values that appear on the path from 15250 to 27750 are: 15250, 24000 and 27750. Thus, the information that is common to values 15250 and 27750 is quantified as $I_s^c = -2 \times \log \{p(15250) + p(24000) + p(27750)\} = 0.24$. From the total and the common information values, the similarity s_s between 15250 and 27750 is computed as 0.14. The similarity in context of impact between 27750 and 15250 is determined as 0.93, and the overall similarity between these values is 0.25. Similarly, the overall similarity between values 27750 and 45000 is computed as 0.48. These results match well with the expectation that the similarity between two adjacent values must be greater than the similarity between two non-adjacent values.

5.5.2 Engineering change similarity

Table 5.1 summarizes the results of applying our approach for computing similarity between attribute values in the proposed EC and the three past changes. It also shows the amount of information about impact in each attribute determined using equation (5.22).

Attribute	Similarity value between attribute-values in proposed change and $EC-1$	Similarity value between attribute-values in proposed change and $EC-2$	Similarity value between attribute-values in proposed change and $EC-6$	Amount of information about impact in attribute
old_ configuration.part. shape.features	1.0	1.0	0.79	0.06
old_ configuration.part. process.name	0.28	1.0	0.07	0.23
old_ configuration.part. process.tool.id	0.21	0.61	0.001	0.29
new_ configuration.part. shape.features	0.77	1.0	0.51	0.11
new_ configuration.part. production_ rate. unit_ quantity	0.25	0.48	0.25	0.13
new_ configuration.part. surface_ finish.value	1.0	1.0	0.6	0.004
new_ configuration.part. tolerance_ range. upper_ limit	1.0	1.0	0.6	0.14
new_ configuration.part. tolerance_ range. lower_ limit	1.0	1.0	0.6	0.14

Table 5.1: Similarity between attribute-values in proposed EC and three past ECs - $EC-1$, $EC-2$ and $EC-6$, and the information about impact in each attribute

Using equation (5.21) for aggregation, the similarity values are $S_2 = 0.84$, $S_2 = 0.53$, and $S_6 = 0.29$. From manual observation of attribute-values, it can be verified that the proposed change is more similar to $EC-2$ than the remaining two past changes from the perspective of atleast six attributes, namely *old_ configuration.part. shape.features*, *old_ configuration.part. process.name*, *new_ configuration.part. shape.features*, *old_ configuration.part. surface_ finish.value*, *old_ configuration.part. tolerance_ range.upper_ limit* and *old_ configuration.part. tolerance_ range.lower_ limit*. Similarly, the proposed change is more similar to $EC - 1$ than the remaining two past changes from the perspective of atleast four attributes, namely *old_ configuration.part. shape.features*, *old_ configuration.part. surface_ finish.value*, *old_ configuration.part. tolerance_ range.upper_ limit* and *old_ configuration.part. tolerance_ range.lower_ limit*. On the other hand, from the perspective of none of the nine attributes it can be inferred that the proposed change is more similar to $EC-6$ than the remaining two past changes. These manual observations confirm our results that the proposed change shown in Figure 5.3 is indeed more similar to the past change $EC - 2$ as compared to the changes $EC - 1$ and $EC - 6$. Similarly, the proposed change is more similar to the past change $EC - 1$ as compared to the change $EC - 6$.

5.6 Evaluation

Our information-based approach to compute similarity between ECs is evaluated against two state-of-the-art approaches, namely metric space [43] and probability-based [45]. Due to its applicability to mixed data types, generalized Minkowski metric [42] is utilized for computing distance required in the metric space approach. The metric space and probability-based approaches are selected among the relevant approaches reviewed in section 2.3, since they can be applied in their original form to compute similarity between instances of an object, such as EC, that is represented using a predefined list of disparate attributes. In addition to the two state-of-the-art approaches, our approach is compared against a statistical approach. In statistical approach, the similarity between attribute-values is not determined based on the semantics, but only based on equation (5.1). The goal of comparing our approach against a statistical approach is to evaluate an important assertion made in section 5.1 that an approach to compute similarity between attribute-values should utilize not only observed conditional distribution of impact, but also semantics associated with values.

Following section discusses the strategy followed for evaluation.

5.6.1 Strategy

10 datasets discussed in Appendix A.3 are utilized for evaluation. Each dataset has 17 training instances and 6 test instances. For each dataset, Appendix B.5 summarizes a set of important attributes based on which the similarity between its each proposed and past ECs should be computed. The probability values required for the similarity computation are determined based on all the changes in the training dataset. Figure C.1 in Appendix C.1 and Figure 5.2 illustrates example IS-A type taxonomies that relate the primitive values in the domain of attributes *old_ configuration.part.process.tool.id* and *old_ configuration.part.process.name*, respectively. These taxonomies are utilized for determining similarity between values of the corresponding attributes.

For each dataset, the similarity results obtained using various approaches are compared from two perspectives: (a) precision in retrieving similar ECs and (b) accuracy in predicting the impact of proposed EC effect.

Precision in retrieving similar ECs

A standard information retrieval metric, namely Mean Average Precision (MAP), is used to evaluate the similarity results in context of the precision in retrieving similar ECs. An average precision value approximates the average area under a precision-recall curve for a given instance, and the MAP is the mean value of average precision of various instances in a dataset [86]. Let MAP_{test} and $MAP_{training}$ represent the MAP values for a given test and training dataset, respectively. In 0.632 bootstrap, the overall value of MAP, denoted as MAP, for a given dataset is computed as,

$$MAP = 0.632 \times MAP_{test} + 0.368 \times MAP_{training} \tag{5.24}$$

For a given dataset, an approach with the highest value of MAP is superior to other approaches. Computing MAP requires the information about past ECs that are actually similar to each proposed change. This information is determined using manual observation.

Success in predicting impact

For each proposed change in a training dataset, top 3 most similar past changes are used to predict its impact based on the maximum likelihood scheme. For a given dataset, let $SR_{training}$ and SR_{test} represent the average success rate of all training instances and test instances, respectively. The overall success rate for a given dataset is computed as,

$$SR = 0.632 \times SR_{test} + 0.368 \times SR_{training} \tag{5.25}$$

The overall success rate values enable the comparison of our approach against the state-of-the-art approaches in context of fulfilling the overall goal. For a given dataset, an approach with the largest success rate is better than the other approaches.

5.6.2 Results

For all datasets, Appendix C.2 summarizes the similarity rankings of proposed ECs in various datasets obtained using all four approaches.

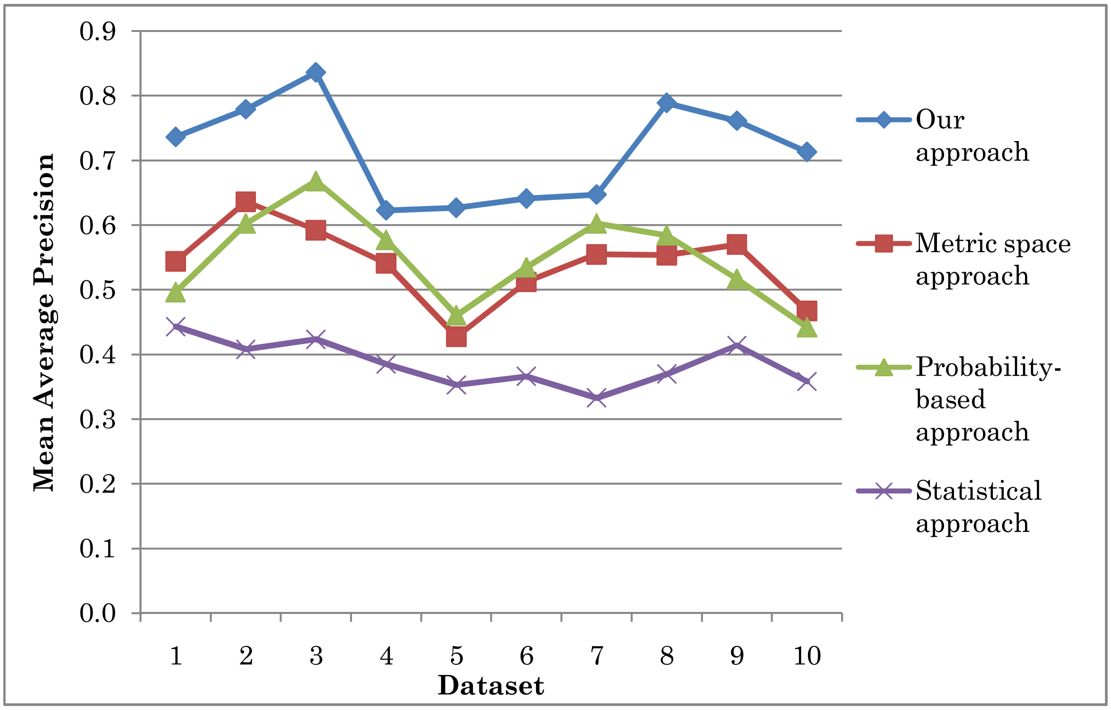


Figure 5.6: Overall MAP in retrieving past changes, which are similar to various proposed changes, determined using various approaches to compute similarity between ECs

Precision in retrieving similar ECs

Computing MAP requires information about past ECs that are actually similar to the proposed EC. As discussed earlier, this information is determined based on the manual observation. For all datasets, Appendix C.3 summarizes the past ECs that are determined as similar to the proposed ECs based on the manual observation.

For all 10 datasets, Figure 5.6 summarizes the overall MAP in retrieving similar past changes determined using our approach, a statistical approach, metric space approach with generalized Minkowski metric and probability-based approach. The MAP values for various test and training datasets are shown in Appendix C.4.

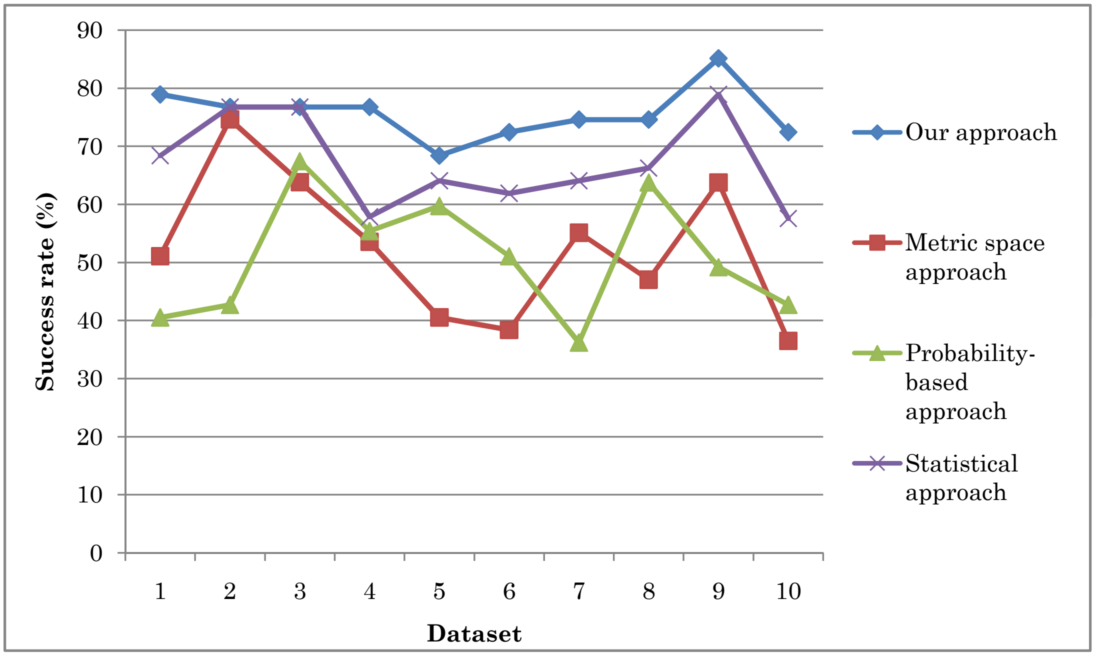


Figure 5.7: Overall success rate in predicting impact determined using various approaches to compute similarity between ECs

Success in predicting impact

For all 10 datasets, Figure 5.7 illustrates the values of success rate in predicting the impact using all four approaches. The success rate values for various test and training datasets are shown in Appendix C.5

5.6.3 Analysis

This section presents an analysis of evaluation results.

Precision in retrieving similar ECs

As seen from Figure 5.6, the MAP using our approach is greater than the MAP using a statistical, metric space or probability-based approach in each of the ten datasets. The average value of MAP using our approach, probability-based approach, metric space approach and statistical approach is 0.72, 0.55, 0.54 and 0.385, respectively.

Significance test

In order to determine whether the difference in the MAP values between our approach and other approaches are statistically significant, and not due to a chance effect in the estimation of MAP, a significance test was carried out using the results of 10 bootstrap runs. Since each of the 10 datasets are derived from a single database, the corrected resampled t-test [19] was chosen for the significance test. In this test, the t-statistic is determined using equation,

$$t = \frac{\bar{\delta}}{\sqrt{\left[\frac{1}{k} + \frac{n_2}{n_1}\right] \times \sigma_{\delta}^2}} \tag{5.26}$$

where, $\bar{\delta}$ is the mean difference in the MAP values from our approach and a state-of-the-art approach, σ_{δ}^2 is the variance of the difference in the MAP values, n_1 is the number of training instances, n_2 is the number of test instances, and k is the number of times the test is repeated. For our experiment, $k = 10$, $n_1 = 17$, $n_2 = 6$, $\bar{\delta}$ and σ_{δ}^2 are based on the 10 differences in the MAP values presented in Figure 5.6. Following the corrected resampled t-test, it is determined with 99.5% confidence that there is a statistically significant difference in the MAP values using our approach and metric space approach with generalized Minkowski metric. Similarly, it is determined with 98% confidence that there is a statistically significant difference in the MAP values using our approach and probability-based approach, and it is determind with 99.9% confidence that there is a statistically significant difference in the MAP values using our approach and statistical approach.

Effect size

The Cohen's d statistic, which accounts for the sample size based on the Hedge's adjustment [87], was utilized for determining how large is the difference in the expected MAP from our approach and each of the state-of-the-art approaches. The value of d statistic for difference in expected value of MAP between our approach and metric space approach is 2.4, our approach and probability-based approach is 2.17, and our approach and statistical approach is 5.44. Based on the d statistic values, it is inferred that the expected MAP from our approach is greater than the expected MAP from metric space approach by 2.4 times the pooled standard deviation in the MAP from the two approaches. Similarly, the expected MAP from our approach is greater than the expected MAP from probability-based approach by 2.17 times the pooled standard deviation in the MAP from the two methods, and the expected MAP from our approach is greater than the expected MAP from statistical approach by 5.44 times the pooled standard deviation in the MAP from the two methods.

Success in predicting impact

As seen in Figure 5.7, for each dataset the success rate using our approach is greater than or equal to success rate using remaining three approaches. The average value of success rate using our approach, metric space approach, probability-based approach and statistical approach is 75.7%, 52.4%, 50.9% and 67.26%, respectively. In case of our approach, for a very large number of proposed changes (i.e., $N > 100$) the true success rate shall lie between 56% and 76% with 90% confidence. In case of metric space approach the true success rate lies between 37% and 57% with 90% confidence, for probability-based approach the true success rate shall lie between 35%

and 55% with 90% confidence, and for statistical approach the true success rate shall lie between 43% and 63% with 90% confidence. Thus, for a very large number of proposed changes, it can be inferred with 90% confidence that the true success rate using our approach shall be greater than the true success rate using metric space and probability-based approaches.

Significance test

Following the corrected resampled t-test, it is determined with 99% confidence that there is a statistically significant difference in the success rate using our approach and metric space approach with Generalized Minkowski metric. Similarly, it is determined with 95% confidence that there is a statistically significant difference in the success rate using our approach and probability-based approach, and it is determined with 85% confidence that there is a statistically significant difference in the success rate using our approach and statistical approach.

Effect size

Based on the Cohen's d statistic values, it is inferred that the expected success rate from our approach is greater than the expected success rate from metric space approach by 2.14 times the pooled standard deviation in the success rate from the two approaches. Similarly, the expected success rate from our approach is greater than the expected success rate from probability-based approach by 2.65 times the pooled standard deviation in the success rate from the two methods, and the expected success rate from our approach is greater than the expected success rate from statistical approach by 1.08 times the pooled standard deviation in the success rate from the two approaches.

Discussion

With our approach, it is found that typically the impact is incorrectly predicted for a proposed change that is on the same part as one or more changes from the top 3 most similar past changes, but has different impact from those changes. To address this issue, we recommend that if the majority of top 3 retrieved past changes are on the same part as the proposed change, then that proposed change should be evaluated in detail.

Typically, for a given proposed EC, statistical approach retrieves past ECs that have same impact as the proposed EC; whereas it fails to retrieve past ECs that are actually similar to proposed EC. For example, consider the proposed change $EC-11$ from dataset # 2. Based on manual observation, two past ECs that are similar to $EC-11$ are $EC-12$ and $EC-15$. Using statistical approach, the top three most similar unique past ECs that are identified as similar to $EC-11$ are $EC-6$, $EC-8$ and $EC-16$. Each of these three retrieved ECs and $EC-11$ have high impact. However, from manual observation none of these retrieved ECs are most similar to $EC-11$, since a majority of the important attribute values in $EC-11$ are different from the values in $EC-6$, $EC-8$ and $EC-16$. For instance, value of $old\_configuration.part.process.name$ in $EC-11$ is *{casting}*; whereas the value of this attribute in $EC-6$, $EC-8$ and $EC-16$ is *{drilling}*, *{drilling}* and *{turning}*, respectively. Similarly, value of $old\_configuration.part.process.tool$ in $EC-11$ is *{cast-tool-5}*; whereas the value of this attribute in $EC-6$, $EC-8$ and $EC-16$ is *{drill-tool-3}*, *{drill-tool-4}* and *{turn-tool-1}*, respectively. As a result, the MAP using statistical approach is very low, while the success rate in predicting impact is not equally low. Utilizing semantics associated with attribute values enables overcoming such limitations associated with the statistical approach. Based on our approach, the top three unique past changes that are most similar to $EC - 11$ from dataset # 2 are $EC - 4$, $EC - 12$ and $EC - 15$.

In a few cases, none of the most similar past ECs retrieved by statistical approach are either actually similar to proposed EC or have same impact as the proposed EC. For example, consider the proposed change $EC - 15$ from dataset # 8. Based on the manual observation, two past ECs that are most similar to $EC - 15$ are $EC - 9$ and $EC - 11$. Using statistical approach, the top two most similar unique past ECs that are identified as similar to $EC - 15$ are $EC - 1$ and $EC - 5$, since the distribution of impact associated with a majority important attributes values in $EC - 15$ is same or very similar to that associated with corresponding attributes values in $EC - 1$ and $EC - 5$. However, both $EC - 1$ and $EC - 5$ have low impact; whereas impact of $EC - 15$ is high. Our approach overcomes such cases by utilizing the semantics associated with attribute values along with the conditional distribution of impact.

5.7 Summary and Future work

Summary

A knowledge-based system to predict the impact of a proposed Engineering Change (EC) effect relies on an approach for computing similarity between ECs. This chapter discussed an approach to compute similarity between ECs that are defined by a set of disparate attributes. Since the available information is probabilistic, the fundamental measures of information are utilized for defining measures to compute similarity between two attribute values or ECs. The semantics associated with EC attribute-values are identified and utilized to determine similarity between them. The similarity measure is defined differently for each attribute type, but has the same fundamental meaning, which allows aggregation of attribute-value similarities to determine the similarity between changes. The weights in the aggregation function are proportional to the amount of information about impact of proposed EC effect in an attribute, since the objective is to determine similarity in the context of predicting impact. This approach of utilizing the probabilistic information in an EC knowledge-base for computing similarity accounts for variations in the similarity perception among enterprises or within an enterprise over a period of time.

A case-study is presented to demonstrate the application of our approach to an example EC scenario. The results of case-study verify the correctness of our approach to compute similarity between attribute values as well as ECs. The example EC datasets discussed in chapter I are utilized for evaluating our approach against a statistical approach and the two state-of-the-art approaches, namely metric space with generalized Minkowski metric and probability-based. The evaluation is done from two perspectives: (a) MAP in retrieving similar ECs and (b) accuracy in predicting the impact of proposed EC effect. The results show that there is statistically significant improvement in MAP and accuracy value obtained using our approach as compared to those obtained using remaining three approaches. Furthermore, it is determined that the expected value of MAP from our approach is greater than the expected MAP from metric space approach by 2.4 times the pooled standard deviation in the MAP from two approaches. Similarly, the expected MAP from our approach is greater than the expected MAP from probability-based approach by 2.17 times the pooled standard deviation in the MAP from two methods, and the expected MAP from our approach is greater than the expected MAP from a statistical approach by 5.44 times the pooled standard deviation in the MAP from two methods. The results demonstrate the limitations of statistical approach in retrieving similar ECs. Based on the results, it can also be inferred with 90% confidence that for a very large number (i.e., $N > 100$) of changes, the accuracy in predicting impact of proposed EC effect obtained using our approach shall be greater than that obtained using metric space and probability-based approaches.

Future work

The EC attributes can be grouped into three categories: attributes that are associated with state of the product before change, attributes that are associated with state of the product after change, and non product attributes. Future work on computing similarity between ECs should study the influence of this categorization on the approach for aggregating attribute values similarities.

CHAPTER VI

Predicting impact of proposed engineering change effect

This chapter presents an approach to predict the impact of proposed EC effect based on the similar past ECs.

6.1 Motivation

As discussed in section 1.5, two changes that have high value of similarity between them might not have same impact due to differences in context of impact between some of its attribute values. The nature of relationship between attribute-value differences and differences in impact is unknown. In addition, there is no formal approach to predetermine such relationship.

In the area of ECM, so far, a simple majority voting method [11] has been used to predict impact of a proposed EC effect based on a set of similar past ECs. The problem of predicting impact of proposed EC effect based on similar past ECs is similar to the problem of similarity-based classification in Computer Science domain and the problem of case adaptation in the area of CBR. There exists several approaches to the problem of similarity-based classification [47] and case adaptation [54]. The existing approaches, within and outside the domain of ECM, do not account for differences in context of impact between attributes values in the process of prediction/classification.

6.2 Objective

The objective of this research phase is to develop an approach, which

For: a proposed EC

Given: (a) database of past ECs, (b) similarity values between proposed and past ECs, and (c) pairwise similarity values between past ECs

Determines: whether the expected cost impact of proposed EC effect is significant (high) or insignificant (low)

In addition to the assumptions discussed in section 4.2 about the nature of knowledge-base that stores past ECs, following assumptions are made in solving the aforementioned problem:

- EC knowledge is captured using a large number (in hundreds) of disparate and interdependent attributes. As discussed in chapter I, only some of these attributes should be used for similarity computation. It is assumed that a set of EC attributes based on which the similarity between ECs is computed is known. Chapter IV presents an approach to determine important attributes of an EC which should be used to determine similar ECs.

- The similarity between ECs is computed in context of predicting impact. Chapter V presents an approach to compute similarity between changes in context of predicting impact.

6.3 Our approach

Figure 6.1 illustrates our overall approach to predict impact which addresses the challenges discussed in section 1.5.3. In the first step, a value of k is determined based on the training data such that classification success rate is maximized. Since there are differences between attribute values, the impact of proposed EC might not be same as that of each of the k most similar past ECs. To accommodate this, second step in our approach is to consider each of the k most similar past ECs and determine the probability of event that the impact of proposed EC effect is same as its impact. In the last step, k probability values determined in second step are aggregated to estimate a probability distribution of impact values of the proposed change effect. Once the probability distribution of impact values is determined, a value that has maximum probability is selected as the value of proposed EC effect. Following section discuss our approach to estimate a probability that the impact of proposed EC effect is same as the impact of a past EC that is similar to it.

6.3.1 Estimating probability of same impact

Consider following notations:

- c_0: a proposed EC
- $\{c_i : 1 \leq i \leq k, k \in \mathbb{N}\}$: set of k past changes that are most similar to c_0
- S_{0i}: similarity value between c_0 and c_i, where $1 \leq i \leq k$
- $Y = \{y_i : 1 \leq i \leq n, i \in \mathbb{N}\}$: set of n important attributes
- Z: impact attribute with domain values $\{h, l\}$, where h denotes *high* and l denotes *low*

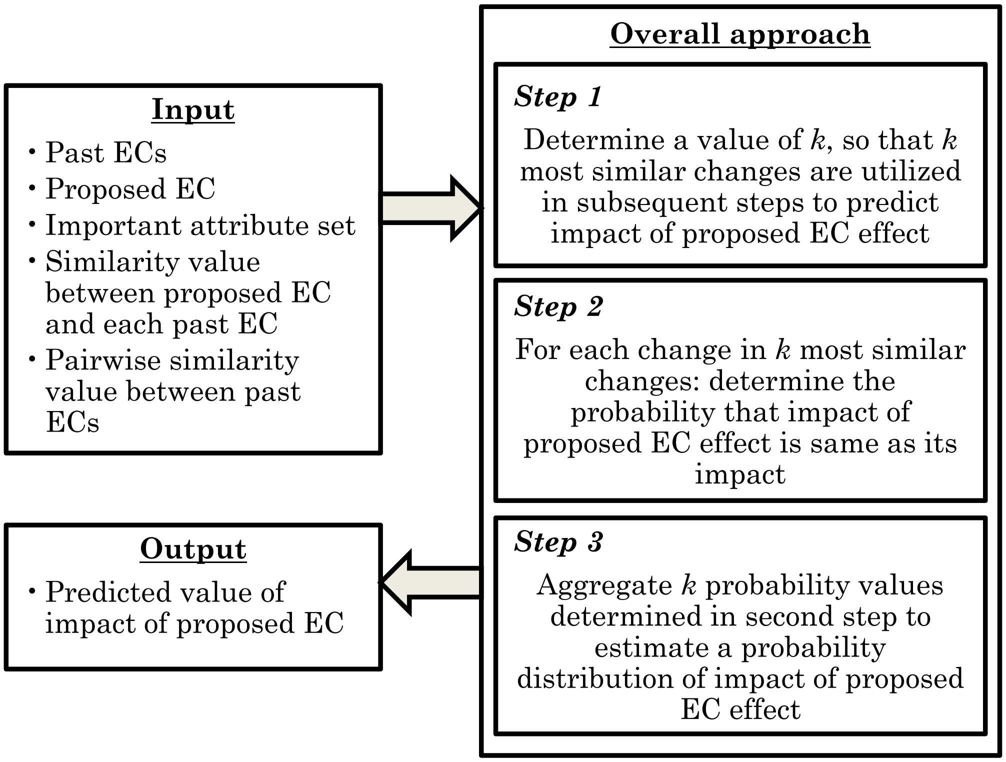

Input
• Past ECs
• Proposed EC
• Important attribute set
• Similarity value between proposed EC and each past EC
• Pairwise similarity value between past ECs
Overall approach
Step 1
Determine a value of k, so that k most similar changes are utilized in subsequent steps to predict impact of proposed EC effect
Step 2
For each change in k most similar changes: determine the probability that impact of proposed EC effect is same as its impact
Step 3
Aggregate k probability values determined in second step to estimate a probability distribution of impact of proposed EC effect
Output
• Predicted value of impact of proposed EC

Figure 6.1: Our approach of predicting impact of proposed EC effect

- z_i: realization of Z in a change c_i, where $0 \leq i \leq k$

Consider a variable $v_{ij} \in \{0, 1\}$ that is defined as,

$$v_{ij} = \begin{cases} 1 & \text{if } z_i = z_j; \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

Consider a past change c_j, which is one of the k past ECs that are most similar to proposed EC. As discussed earlier, two changes that have a high value of similarity between them might not have same impact value due to differences in context of impact between its attribute values. To address this challenge, a probability that the impact of effect of c_0 is same as the impact of corresponding effect of c_j is determined in following two steps:

1. *Compute differences in context of impact between attribute values:* In the first

step, for each important attribute the difference in context of impact between its values in c_0 and c_j is determined.

2. *Aggregate the attribute-value differences:* In the second step, the attribute-value differences determined in earlier step are aggregated to compute a probability that the impacts of two changes are same.

Following sections present a detailed discussion on each of the two aforementioned steps.

Differences in context of impact between attribute values

Consider a generic EC attribute, X, which can be of simple/aggregate qualitative or quantitative type. The information available in change database to quantify difference between two values, say A and B, of X is the distribution of impact associated with these values. Let $P_{Z|A} = \{p(Z = h|A), p(Z = l|A)\}$ and $P_{Z|B} = \{p(Z = h|B), p(Z = l|B)\}$ represent the conditional probability distribution of impact given the values A and B, respectively. The difference, denoted as d_{AB}, between values A and B is quantified based on the distance between the distributions $P_{Z|A}$ and $P_{Z|B}$. The desirable properties of a function used to compute d_{AB} are:

1. If $P_{Z|A} = P_{Z|B}$, then $d_{AB} = 0$; since if two probability distributions are exactly same, then the distance between them is 0.

2. $d_{AB} = d_{BA}$, since distance between $P_{Z|A}$ and $P_{Z|B}$ is same as the distance between $P_{Z|B}$ and $P_{Z|A}$.

3. The distance between $P_{Z|A}$ and $P_{Z|B}$ is maximum for following two cases: (1) $P_{Z|A} = \{1, 0\}$ and $P_{Z|B} = \{0, 1\}$ or (2) $P_{Z|B} = \{1, 0\}$ and $P_{Z|A} = \{0, 1\}$. For these two case, $d_{AB} = 1$, i.e., difference is maximum.

A function that satisfies all the three aforementioned properties is in the form,

$$d_{AB} = 1 - cos(\theta_{AB}) \tag{6.2}$$

where, $cos(\theta_{AB})$ is a cosine similarity metric defined as [84],

$$cos(\theta_{AB}) = \frac{p(Z=h|A) \times p(Z=h|B) + p(Z=l|A) \times p(Z=l|B)}{\sqrt{(p(Z=h|A))^2 + (p(Z=l|A))^2 \times (p(Z=h|B))^2 + (p(Z=l|B))^2}} \tag{6.3}$$

Aggregating differences between attribute values

This section presents our approach of aggregating attribute values differences to compute a probability that the impacts of two changes are same. Let d_l^{0j} denote the difference between values of important attribute y_l in changes c_0 and c_j, and $D^{0j} = \left\{d_1^{0j}, \ldots, d_n^{0j}\right\}$ represent the vector of differences between values of important attributes in changes c_0 and c_j. For simplicity, it is assumed that each difference variable is discretized such that there are three values in its domain. For example, three values in domain of d_l^{0j} are $\{a_l, b_l, c_l\}$, where $a_l > b_l > c_l$ and $a_l, b_l, c_l \in [0,1]$. Let $p(v_{0j} = 1|D^{0j})$ denote the probability that the impact of change c_0 is same as the impact of its nearest neighbor c_j given D^{0j}. As discussed earlier, a major challenge in aggregating attribute values differences is that the nature of relationship between D^{0j} and v_{0j} is unknown. To address this challenge, we utilize the pairs of all most similar past changes, since these pairs together implicitly capture the nature of relationship between D^{0j} and v_{0j}. The pairs of all most similar past ECs are determined by identifying k past EC that are most similar to each past EC. The pairs of all most similar past changes can be partitioned into two groups. Let V_1 denote pairs of all most similar past changes that have same value of impact, and V_0 denote pairs of all most similar past changes that have different value of impact. The probability

$p(v_{0j} = 1|D^{0j})$ is computed using an alternate form of Bayes rule [88],

$$p(v_{0j} = 1|D^{0j}) = \frac{p(D^{0j}|V_1) \times p(V_1)}{p(D^{0j}|V_0) \times p(V_0) + p(D^{0j}|V_1) \times p(V_1)} \tag{6.4}$$

where, $p(V_1)$ is the probability that a pair of most similar past ECs has same value of impact, $p(V_0)$ is the probability that a pair of most similar past ECs has different value of impact, $p(D^{0j}|V_1)$ is the probability of value D^{0j} among pairs of all most similar past ECs that have same value of impact, and $p(D^{0j}|V_0)$ is the probability of value D^{0j} among pairs of all most similar past ECs that have different value of impact. The values $p(V_0)$ and $p(V_1)$ can be determined from the available pairs of most similar past changes. The probabilities $p(D^{0j}|V_1)$ and $p(D^{0j}|V_0)$ can be determined if the corresponding probability distributions can be estimated. Following section presents our approach to estimate the two unknown distributions.

Estimating unknown probability distributions

Consider following notation:

- $D = \{d_1, \ldots, d_n\}$: a set of n discrete variables corresponding to n attributes. An l^{th} variable, d_l, in D measures the degree to which two values of l^{th} attribute are different in context of impact. Thus, this variable is similar to d_l^{0j}, except that we now drop the superscript $0j$ to indicate that the quantity d_l is not specific to a particular pair of changes. As discussed earlier, it is assumed that each difference term is discretized such that there are three values in its domain, i.e., $d_l \in \{a_l, b_l, c_l\}$, where $a_l > b_l > c_l$ and $a_l, b_l, c_l \in [0, 1]$. Since each variable in D can take one of the three values, the domain of D is of size 3^n. Let $D = \{D_1, \ldots, D_{3^n}\}$ represent 3^n values in the domain of D.

- $Q(D|V_1) = \left\{q_{i|V_1} : 1 \leq i \leq 3^n, i \in \mathbb{N}\right\}$: observed probability distribution of D among pair of all most similar past changes that have same value of impact.
- $P(D|V_1) = \left\{p_{i|V_1} : 1 \leq i \leq 3^n, i \in \mathbb{N}\right\}$: true probability distribution of D among pair of most similar past changes that have same value of impact.
- $Q(D|V_0) = \left\{q_{i|V_0} : 1 \leq i \leq 3^n, i \in \mathbb{N}\right\}$: observed probability distribution of D among pair of all most similar past changes that have different value of impact.
- $P(D|V_0) = \left\{p_{i|V_0} : 1 \leq i \leq 3^n, i \in \mathbb{N}\right\}$: true probability distribution of D among pair of most similar past changes that have different value of impact.
- $n_{V_1|d_l=a}$: number of pairs of most similar changes in V_1 that have a value of d_l as a
- $n_{V_0|d_l=a}$: number of pairs of most similar changes in V_0 that have a value of d_l as a

This section presents our approach to estimate $P(D|V_1)$. As discussed later in this section, the same approach with a little modification will be applicable to estimate $P(D|V_0)$. Once the two distributions are known, the values $p(D^{0j}|V_1)$ and $p(D^{0j}|V_0)$ can be determined for equation (6.4).

Lower the differences in context of impact between values of important attributes in two changes, greater is the probability that the impact of two changes will be same. This premise along with the observed data is utilized to estimate $P(D|V_1)$. To obtain a reasonable estimate of $P(D|V_1)$, the expectation of each d_l value is constrained with respect to $P(D|V_1)$ such that,

$$E_{P(D|V_1)}[d_l] = \mu_l; \forall l = 1, 2, \ldots, n \tag{6.5}$$

where, μ_l is an average value of d_l among all the changes that have same value of impact. A value of μ_l is determined using weighted sum as,

$$\mu_l = w_a^l \times a_l + w_b^l \times b_l + w_c^l \times c_l; \forall l = 1, 2, \ldots, n \tag{6.6}$$

where, w_a^l, w_b^l and w_c^l denote the weights associated with difference values $d_l = a_l$, $d_l = b_l$ and $d_l = c_l$, respectively. As mentioned earlier, lower the attribute values differences between two changes, greater is the probability that the impact of those two changes will be same. Thus, each of the difference values $d_l = a_l$, $d_l = b_l$ and $d_l = c_l$ are not equally weighted in equation (6.6); lower difference values are assigned larger weights as compared to higher difference values. This is accomplished by taking weights as inversely proportional to the magnitude of associated value. That is,

$$w_a^l = \frac{n_{V_1|d_l=a_l} \times (1 - a_l)}{n_{V_1|d_l=a_l} \times (1 - a_l) + n_{V_1|d_l=b_l} \times (1 - b_l) + n_{V_1|d_l=c_l} \times (1 - c_l)} \tag{6.7}$$

Equations similar to (6.7) follow for the weights w_b^l and w_c^l.

Example VI.1. To understand how equations (6.6) and (6.7) are used, consider the case in which there are two important attributes: $\{y_1, y_2\}$. Difference in values of y_1 in context of impact are discretized into three intervals, such that $a = 0.2$, $b = 0.5$ and $c = 0.8$. Since there are two important attributes and three possible difference values, the domain of D has 9 values, i.e., $D_1 = \{0.2, 0.2\}$, $D_2 = \{0.2, 0.5\}$, $D_3 = \{0.2, 0.8\}$, and so on. Let there be 3 pairs of most similar past changes that have same value of impact. Two of these changes have a difference vector $\{0.2, 0.2\}$ and one has a difference vector $\{0.5, 0.2\}$. That is, $n_{V_1|d_1=0.2} = 2$, $n_{V_1|d_1=0.5} = 1$ and $n_{V_1|d_1=0.8} = 0$. Based on equation (6.7), the three weights are $w_a^1 = 0.89$, $w_b^1 = 0.11$ and $w_c^1 = 0.0$, and using equation (6.6) $\mu_1 = 0.233$. An arithmetic mean would give $\mu_1 = 0.3$, which is an overestimation as compared to our approach, since arithmetic mean does not account for the magnitude of difference values.

In addition to satisfying the n constraints specified by (6.5), probability values in distribution $P(D|V_1)$ should sum to 1, i.e.,

$$\sum_{i=1}^{3^n} p_{i|V_1} = 1 \tag{6.8}$$

There might be several values of $P(D|V_1)$ that satisfy the constraints in equations (6.5) and (6.8). Among these several values, the best possible values are determined based on the principle of minimum cross entropy, which states that among all the distributions that satisfy the constraints choose one that is closest to the observed distribution [73]. In the principle of minimum cross entropy, the total distance between unknown and observed distribution is computed using the KL-divergence as,

$$KL(P(D|V_1)||Q(D|V_1)) = \sum_{i=1}^{3^n} p_{i|V_1} \times \log \frac{p_{i|V_1}}{q_{i|V_1}} \tag{6.9}$$

Thus the distribution $P(D|V_1)$ is determined such that the equation (6.9) is minimized subject to the constraints (6.5) and (6.8). Some of the $q_{i|V_1}$ values might be zero. This presents an issue in estimating a minimization of equation (6.9). To address this issue, a technique called Laplace estimator is utilized to convert zero observed probability value to a very small non-zero value [19].

The approach discussed in this section is also applicable to estimate the distribution $P(D|V_0)$ with two modifications. The first modification is to utilize pairs of all most similar past changes in V_0 instead of those in V_1. The second modification is in the equations used to compute weights. Since higher the attribute values differences, greater is the probability that the impact of two changes is different, equation (6.7) is modified to following,

$$w_a^l = \frac{n_{V0|d_l=a_l} \times a_l}{n_{V0|d_l=a_l} \times a_l + n_{V0|d_l=b_l} \times b_l + n_{V0|d_l=c_l} \times c_l} \tag{6.10}$$

6.3.2 Aggregating k probability values

The k probability values, $\{p(v_{0i} = 1|D^{0i}) : 1 \leq i \leq k, k \in \mathbb{N}\}$, determined in previous step are aggregated using weighted sum to determine the probability of the event that the impact of proposed EC effect is high,

$$p(z_0 = h) = \sum_{i:z_i=h} r_i \times p(v_{0i} = 1|D^{0i}) + \sum_{i:z_i=l} r_i \times (1 - p(v_{0i} = 1|D^{0i})) \tag{6.11}$$

where, r_i is the weight associated with probability $p(v_{0i} = 1|D^{0i})$. Since it is assumed that similarity value between changes is computed in context of predicting impact, higher the value of S_{0i}, greater is the confidence in value $p(v_{0i} = 1|D^{0i})$. Thus, weight r_i is taken as proportional to similarity value S_{0i}. The value $p(z_0 = l)$ is determined as $p(z_0 = l) = 1 - p(z_0 = h)$. Once the probability distribution of impact of proposed EC effect is determined, the proposed EC effect is assigned an impact value that has maximum probability.

6.4 Case study

Figure 6.2 illustrates an example proposed EC from dataset # 8. Consider the task of predicting the impact of this proposed change on *Process.* The example proposed EC is same as $EC-4$, shown in Figure A.5 of Appendix A.2, from our example database. Table B.21 in Appendix B.5 illustrates the eight important attributes that should be used for evaluating the impact of effect of proposed ECs from dataset # 8. This set of important attributes are identified using the approach discussed in chapter IV.

6.4.1 Determining a value of k

The first step of our overall approach is to determine a minimum value of k based on the training data such that classification success rate is maximized. For first four

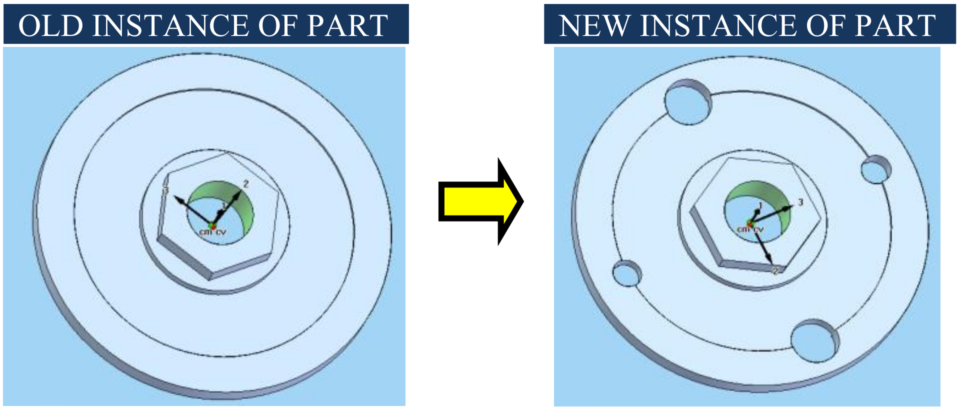


Figure 6.2: Example proposed EC from dataset # 8. This proposed change is same as $EC - 4$ from our example database

values of k, Table 6.1 illustrates success rate in classifying 17 training instances from dataset # 8. As seen in Table 6.1, success rate is maximum for $k = 3$. Therefore, top three most similar ECs will be utilized in subsequent steps to predict the impact of proposed EC effect.

k	Success rate in classifying training instances (%)
1	82.35
2	82.35
3	88.23
4	82.35

Table 6.1: Success rate in classifying training instances for various values of k

Based on our approach to compute similarity, it is determined that the 3 training instances from dataset # 8 which are most similar to proposed EC are $EC - 2$, $EC - 9$ and $EC - 11$, shown in Figures A.3, A.10 and A.12, respectively. The impact on *Process* due to these three changes are *low*, *high* and *high*, respectively.

6.4.2 Estimating probability of same impact of two similar changes

The second step in our approach is to consider each most similar EC in sequence and determine a probability of the event that impact of proposed EC effect is same as impact of that most similar EC effect. The remaining portion of this section demonstrates our approach to determine a probability of the event that impact of effect of proposed EC is same as that of $EC - 11$. The results for remaining two similar changes follow accordingly.

Differences in context of impact

Our approach to determine a probability that impact of two changes are same requires the differences in context of impact between values of important attribute in these two changes. Equation (6.2) is utilized to compute differences between values of important attributes. The difference values are discretized into three intervals using equal-frequency binning approach [19] with three bins. Table 6.2 illustrates discretized labels of differences in context of impact between values of important attributes in proposed EC and $EC - 11$.

Important attribute	Difference in context of impact between values in proposed EC and $EC - 11$
old_configuration.part. process.name	0.15
old_configuration.part. process.tool.id	0.85
old_configuration.part. shape.features	0.05
new_configuration.part. production_rate. unit_quantity	0.6
new_configuration.part. surface_finish.value	0.01
new_configuration.part. tolerance_range.lower_limit	0.1
new_configuration.part. tolerance_range.upper_limit	0.1
new_configuration.part. shape.features	0.1

Table 6.2: Differences in context of impact between values of important attributes in proposed EC and $EC - 11$

Aggregating differences between attribute values

For the differences shown in Table 6.2, Table 6.3 summarizes the relevant probability values determined based on the minimum cross entropy formulation discussed in section 6.3.1. Based on equation (6.11) and the probability values discussed in Table 6.3 it is determined that the probability of the event that the impact of proposed EC effect is same as that of $EC - 11$ is 0.06.

Probability term	Value
$p(D^{11}\|V = 1)$	0.00001
$p(D^{11}\|V = 0)$	0.00029
$p(V = 0)$	0.34
$p(V = 1)$	0.66
$p(D^{11})$	0.00011

Table 6.3: Relevant probability values required to determine a probability of the event that the impact of proposed EC effect is same as that of $EC - 11$. D^{11} denotes the differences in context of impact between values of important attributes in proposed EC effect and $EC - 11$.

Following a similar approach it is determined that probability of the event that the impact of proposed EC effect is same as that of $EC - 9$ is 0.06, and probability of the event that the impact of proposed EC effect is same as that of $EC - 2$ is 1.0.

6.4.3 Aggregating k probability values

The last step of our approach is to aggregate the k probability values determined in second step to estimate a probability distribution of impact of the proposed EC effect. Using equation (6.11), it is determined that the probability of impact of proposed EC effect being low is 0.96, and the probability of impact of proposed EC effect being high is 0.04. Based on these probability values, it is inferred that the impact of proposed EC effect is *low*. This result matches with the known information that the actual impact of proposed EC effect is *low*.

6.5 Evaluation

The approach discussed in this chapter is evaluated against two state-of-the-art approaches, namely k-Nearest Neighbor (NN) and a generative similarity based classifier called regularized local Similarity Discriminant Analysis (SDA). The k-NN is selected among the relevant approaches reviewed in section 2.4, since it is one of the simplest approaches and yet often works very well [48, 19]. The regularized local SDA is selected because it has superior performance as compared to other popular similarity-based classifiers, such as local SDA or SVM [53]. Following sections present the evaluation strategy, followed with evaluation results and analysis.

6.5.1 Strategy

10 datasets discussed in Appendix A.3 are utilized for evaluation. Each dataset has 17 training instances and 6 test instances. For each dataset, Appendix B.5 and Appendix C.2 summarizes a set of important attributes and similarity results, respectively, that are utilized for predicting the impact of effect of proposed ECs in various datasets. The probability values required for the prediction are determined based on all the changes in the training dataset. The difference values determined using equation (6.2) are discretized using equal-frequency binning approach [19] with three bins.

For each dataset, the prediction results obtained using our approach and two state-of-the-art approaches are compared from perspective of success rate in predicting the impact of proposed EC effect. For a given dataset, let $SR_{training}$ and SR_{test} represent the success rate in predicting impact for all training instances and test instances, respectively. In 0.632 bootstrap, the overall success rate, denoted as SR, for a given dataset is computed as,

$$SR = 0.632 \times SR_{test} + 0.368 \times SR_{training} \tag{6.12}$$

Overall success rate is the measure for comparing various approaches in our evaluation. An approach with the largest success rate is better than other approaches.

6.5.2 Implementation and results

The minimum cross entropy formulation discussed in section 6.3.1 is encoded in Matlab. The minimum cross entropy formulation requires solving a constrained non-linear optimization problem. The Matlab's implementation of interior-point algorithm is selected to solve our constrained non-linear optimization problem [79]. The remaining steps in the overall approach are implemented using a combination of standalone programs in Visual C++ and MS Excel.

For all datasets, Appendix D.1 summarizes the results of predicting the impact of effect of proposed ECs using our approach and the two state-of-the-art approaches. Figure 6.3 illustrates the values of success rate in predicting the impact using all three approaches. The success rate values for various test and training datasets are shown in Appendix D.2

6.5.3 Analysis

As seen in Figure 6.3, success rate using our approach is greater than that using two state-of-the-art approaches for all datasets. The expected value of overall success rate using our approach, k-NN approach and regularized local SDA is 90.27%, 76.96% and 70.64, respectively. The average value of success rate using our approach, k-NN and regularized local SDA on various test datasets is 86.7%, 68.34% and 58.33%, respectively. In case of using our approach for a very large number of proposed

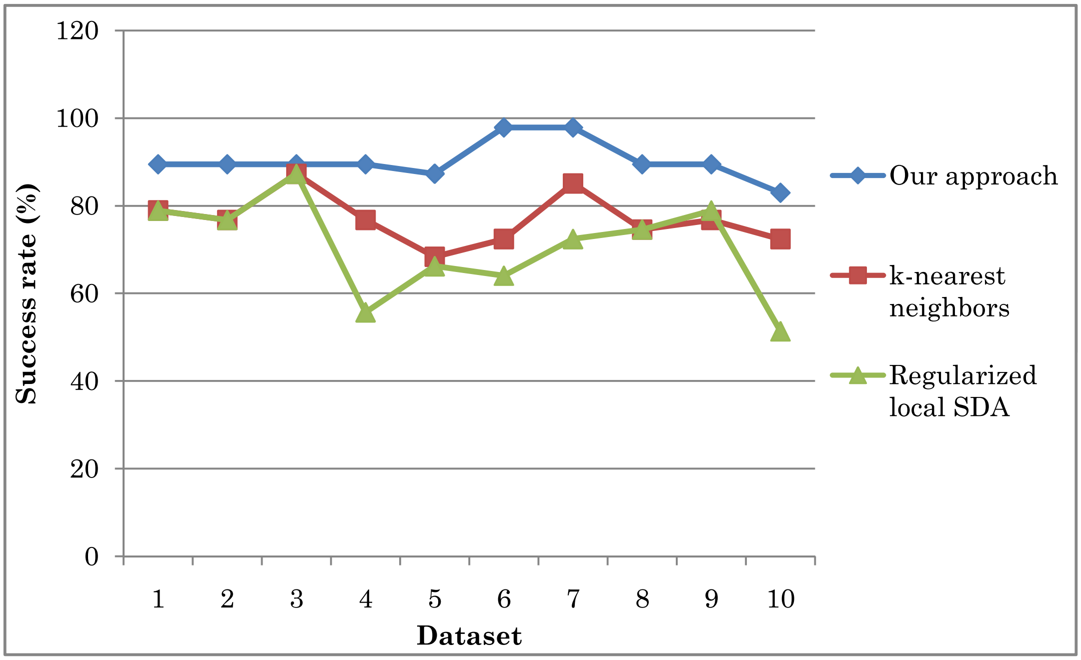


Figure 6.3: Overall success rate in impact prediction based on our approach, k-NN and regularized local SDA approach

changes, i.e., $N > 100$, the true success rate shall lie between 77.94% and 92.28% with 90% confidence. In case of k-NN the true success rate shall lie between 57.95% and 77.17% with 90% confidence; whereas in case of regularized local SDA the true success rate shall lie between 47.8% and 68.14% with 90% confidence. Thus, for a very large number of proposed ECs, it can be inferred with 90% confidence that the true success rate using our approach shall be greater than the true success rate using the two state-of-the-art approaches.

Significance test

In order to determine whether the difference in overall success rate using our approach and state-of-the-art approaches are statistically significant, and not due to a chance effect in the estimation of success rate, a corrected resampled t-test was carried out using the results of the 10 datasets. Following this test, it is determined with 99% confidence that there is a statistically significant difference in the overall success rate using our approach and k-NN approach. Similarity, it is determined with 95% confidence that there is a statisticaly significant difference in the overall success rate using our approach and regularized local SDA.

Effect size

The Cohen's d statistic, which accounts for the sample size based on the Hedge's adjustment [87], was utilized for determining how large is the difference in the expected success rate from our approach and each of the two state-of-the-art approaches. The value of d statistic for difference in expected value of overall success rate between our approach and k-NN is 2.47, and its value for difference in expected value of overall success rate from our approach and regularized local SDA is 2.2. Based on the d statistic values, it is inferred that the expected success rate from our approach is greater than the expected success rate from k-NN approach by 2.47 times the pooled standard deviation in the success rate from the two approaches. Similarly, the expected success rate from our approach is greater than the expected success rate from regularized local SDA by 2.2 times the pooled standard deviation in the success rate from the two approaches.

Discussion

Our approach performs better than the two state-of-the-art approaches, since in addition to all information utilized by these approaches, our approach utilizes the information about: (a) attribute value differences in context of impact and (b) relationship between attribute-value differences and differences in impact values.

With all the three approaches it is found that impact is incorrectly predicted each time a proposed EC requires changing the base feature of a part that is procured from outside an enterprise, while none of the retrieved *k* similar past ECs require changing base feature of a part that is procured from outside an enterprise. To address this issue, we recommend that such proposed ECs should be evaluated in detail.

6.6 Summary and Future work

Summary

Determining whether a proposed EC should undergo a fast-track evaluation or a detailed evaluation requires an approach to predict impact of proposed EC effect. This chapter present an approach to predict impact of proposed EC effect based on the similar past ECs. Two changes that have a high value of similarity between them might not have same impacts due to differences in context of impact between its attribute-values. To address this challenge, our approach quantifies the differences in context of impact between important attribute values in two changes. Since the nature of relationship between attribute-value differences and differences in impact is unknown, the Bayes' rule is utilized to predict the differences in impact based on the differences between attribute values. The probability estimates required in the Bayes' rule are determined based on the principle of minimum cross entropy.

An example EC knowledge-base is utilized for evaluating our approach against the two state-of-the-art approaches, namely k-Nearest Neighbor (NN) and regularized local Similarity Discriminant Analysis (SDA). The evaluation is done from perspective of success rate in predicting the impact of proposed EC effect. The results show that there is statistically significant improvement in success rate obtained using our approach as compared to that obtained using two state-of-the-art approaches. Furthermore, it is determined that the expected value of success rate from our approach is greater than the expected value from k-NN approach by 2.47 times the pooled standard deviation in the success rate from two approaches. Similarly, the expected MAP from our approach is greater than the expected success rate from probability-based approach by 2.2 times the pooled standard deviation in the success rate from two methods. Based on the results, it can also be inferred with 90% confidence that for a very large number of proposed ECs the true success rate using our approach shall be greater than the true success rate using the two state-of-the-art approaches.

Future work

The cost impact of a proposed EC depends on the current cost, state and usage of resources within an enterprise which are required to implement proposed EC. Our approach for predicting impact of proposed EC effect does not account for the temporal changes in the cost, state and usage of resources within an enterprise. For example, cost of purchasing a new equipment at the time of implementing proposed EC might be significantly different from that at the time of implementing a past EC that is most similar to the proposed EC. Such changes can affect the prediction of impact of proposed EC effect. Future work on this research can extend the current approach to account for such temporal changes in the cost, state and usage of resources. Understanding such temporal changes can also provide better estimates of interrelationships between differences in the attribute values and differences in impact of a change. These estimates might be useful in enhancing our existing statistical approach to determine differences in context of impact between attribute values.

Another interesting avenue of future work is to study the effect of difference between expected cost of false positive and false negative cases on predicting impact of proposed EC effect. In context of problem addressed in this dissertation, false negative case is incorrectly classifying an effect with high impact as not-high impact; whereas false positive case is incorrectly classifying an effect with not-high impact as high impact effect. A false positive case results in additional unwanted expenditure on detailed evaluation; whereas a false negative case might have undesirable downstream effects, such as high expenditure on correcting the effects of an EC that should not have been accepted in first place.

CHAPTER VII

Evaluation of approach to dissertation problem

Previous three chapters together presents our overall approach to classify impact of proposed EC effect into significant/insignificant. As discussed in chapter I, the problem of classifying impact of proposed EC effect into significant/insignificant is similar to a typical classification problem in the fields of Data mining, Machine learning and Pattern recognition. There exists several state-of-the-art supervised learning approaches to solve the classification problem. This chapter presents an evaluation of our approach against a few approaches that are widely used in practice. The chapter begins by briefly describing the state-of-the-art approaches against which our approach is compared. Later, the evaluation strategy, results and analysis are presented.

7.1 State-of-the-art approaches

Our overall approach will be compared against following five state-of-the-art supervised learning approaches to the classification problem,

1. Naïve Bayes classifier
2. C4.5 decision tree classifier
3. k-Nearest Neighbor (NN) classifier

4. A SVM classifier with Radial basis function (RBF) kernel

5. A feedforward neural network classifier called multilayer percepton

The aforementioned five state-of-the-art approaches are selected, since they are widely used in practice. A detailed discussion of these approaches appear in a standard data mining and pattern recognition textbooks, e.g., [19, 20]. Following paragraphs briefly discuss these approaches.

Naïve Bayes is a simple approach of classification based on the Bayes's rule. It assumes that all the attributes are completely independent of each other. The probability values required in the Bayes's rule are obtained from training instances. The Naïve Bayes approach typically handles the quantitative attributes by assuming that they have a normal probability distribution.

Decision tree is one of the widely used approaches to represent structural patterns in the data. Given a decision tree created from training instances, a new instance is classified based on the values of its attributes. C4.5 is a program for creating a decision tree based on the information gain criterion and the divide-and-conquer algorithm. It can handle both quantitative and qualitative attributes.

k-NN is a lazy classifier which utilizes k nearest neighbors to classify a new instance. A suitable distance function is utilized to determine the proximity between new instance and training instances. Most implementations of k-NN utilize Euclidean distance to quantify the difference between values of quantitative attributes. For qualitative attributes, typically, the distance is taken as one if the values are same and zero if the values are different.

A SVM classifier utilizes nonlinear mapping functions, which are commonly referred to as kernels, to transform the space containing training instances into a new space. It then fits a maximum margin linear model on the instances in the transformed space. The linear model in the transformed space is utilized to classify a new instance. Two commonly used kernels with SVM classifier are RBF (or Gaussian) and sigmoid kernels.

A percepton or neuron represents a hyperplane, i.e., linear model, in a space containing training instances. In multilayer percepton, backpropagation algorithm is utilized to interconnect several simple percepton-like models in a hierarchical structure with three or more layers. One of these layers is input layer, the second is output layer and the remaining are "hidden layers". The network of perceptons represents a nonlinear classifier, which is used to classify a new instance. It is interesting to note that a multilayer percepton with one hidden layer is same as a SVM classifier with sigmoid kernel.

7.2 Strategy

WEKA [21], which implements various machine learning algorithms, will be utilized to obtain prediction results using five state-of-the-art approaches. For Naïve Bayes classifier, C4.5 decision tree classifier, SVM classifier with RBF kernel and multilayer percepton, default parameter values suggested by WEKA are used. In k-NN, predictions using multiple neighbors can be weighted either equally or according to the inverse distance from test instance. As compared to weighting equally, weighting according to the inverse distance is found to have a better performance for majority of the datasets. Therefore, k-NN approach with predictions weighted according to the inverse distance from test instance is used for our evaluation. The best value of k for k-NN is identified based on the hold-one-out cross-validation approach.

10 datasets discussed in Appendix A.3 are utilized for evaluation. For each

dataset, the prediction results obtained using our approach and state-of-the-art approaches are compared from perspective of success rate in predicting the impact of effect of training/test instances. For a given dataset, let $SR_{training}$ and SR_{test} represent the success rate in predicting impact for all training instances and test instances, respectively. In 0.632 bootstrap, the overall success rate, denoted as SR, for a given dataset is computed as,

$$SR = 0.632 \times SR_{test} + 0.368 \times SR_{training} \tag{7.1}$$

Overall success rate is the measure for comparing various approaches in our evaluation. An approach with the largest success rate is better than other approaches.

7.3 Results

Figure 7.1 illustrates the overall success rate in predicting impact for various datasets using various approaches. Success rates for each test and training datasets are summarized in the Appendix E.1.

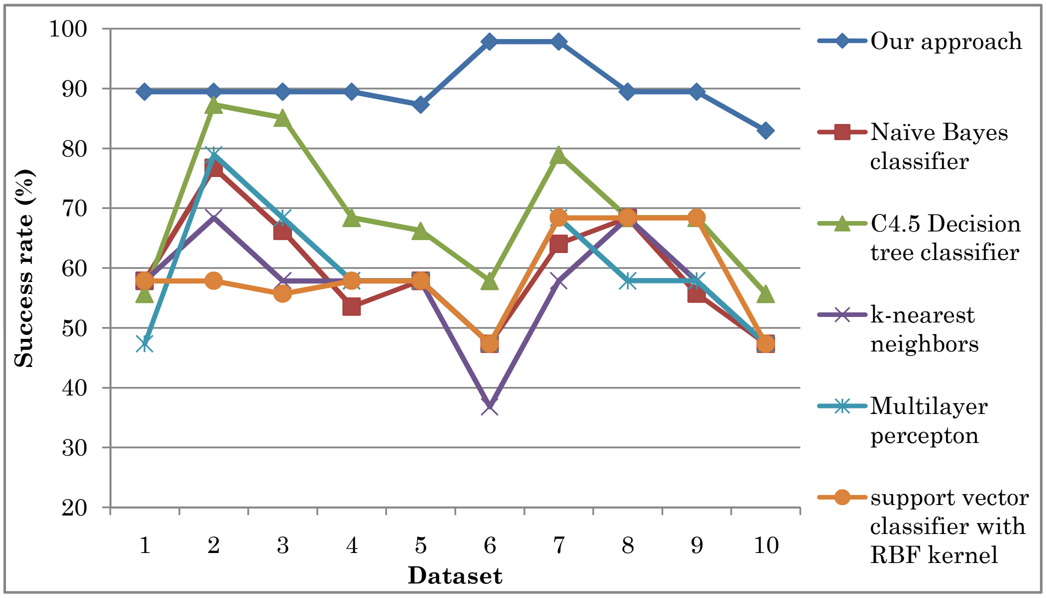


Figure 7.1: Overall success rate in impact prediction based on our overall approach and five state-of-the-art approaches. The five state-of-the-art approaches are Naïve Bayes classifier, C4.5 decision tree classifier, k-nearest neighbors, multilayer percepton and support vector classifier with RBF kernel

7.4 Analysis

As seen in Figure 7.1, success rate using our approach is greater than that using five state-of-the-art approaches for all datasets. The expected value of overall success rate using our approach, Naïve Bayes, C4.5 decision tree classifier, k-nearest neighbors, multilayer percepton and support vector classifier with RBF kernel is 90.27%, 59.5%, 69.21, 56.81, 58.92 and 58.7, respectively.

Significance test

In order to determine whether the difference in overall success rate using our approach and state-of-the-art approaches are statistically significant, and not due to a chance effect in the estimation of success rate, a corrected resampled t-test was carried out using the results of the 10 datasets. Following this test, it is determined with 99.8% confidence that there is a statistically significant difference in the overall success rate using our approach and Naïve Bayes, support vector classifier with RBF kernel, k-NN or multilayer percepton. Similarly, it is determined with 95% confidence that there is a statisticaly significant difference in the overall success rate using our approach and C4.5 decision tree classifier.

Effect size

The Cohen's d statistic was utilized for determining how large is the difference in the expected success rate from our approach and state-of-the-art approaches. The value of d statistic for difference in expected value of overall success rate between: (a) our approach and Naïve Bayes is 4.0, (b) our approach and support vector classifier with RBF kernel is 4.75, (c) our approach and k-NN is 4.42, (d) our approach and multilayer percepton is 3.73, and (e) our approach and C4.5 decision tree classifier is 2.32. Based on the d statistic values, it is inferred that the expected success rate from our approach is greater than the expected success rate from: (a) Naïve Bayes approach by 4.0 times the pooled standard deviation in the success rate from the two approaches, (b) support vector classifier with RBF kernel by 4.75 times the pooled standard deviation in the success rate from the two approaches, (c) k-NN by 4.42 times the pooled standard deviation in the success rate from the two approaches, (d) multilayer percepton by 3.73 times the pooled standard deviation from two approaches and (e) C4.5 decision tree classifier by 2.32 times the pooled standard deviation in the success rate from the two approaches.

7.5 Summary

This chapter presented an evaluation of our overall approach to classify impact of proposed EC effect against five state-of-the-art supervised learning approaches to the classification problem. 10 datasets discussed in Appendix A.3 are utilized for evaluation. Each dataset has 17 training instances and 6 test instances. The evaluation is done from the perspective of success rate in predicting impact of proposed EC effect. The results show that there is a statistically significant improvement in the success rate obtained using our approach as compared to that obtained using each of the five state-of-the-art approaches. Furthermore, it is determined that the expected success rate from our approach is greater than the expected success rate from: (a) Naïve Bayes approach by 4.0 times the pooled standard deviation in the success rate from the two approaches, (b) support vector classifier with RBF kernel by 4.75 times the pooled standard deviation in the success rate from the two approaches, (c) k-NN by 4.42 times the pooled standard deviation in the success rate from the two approaches, (d) multilayer percepton by 3.73 times the pooled standard deviation from two approaches and (e) C4.5 decision tree classifier by 2.32 times the pooled standard deviation in the success rate from the two approaches.

CHAPTER VIII

Conclusion

This chapter provides a summary of the key contributions of the work presented in this dissertation. It highlights the applications in other areas that can benefit from this work. It also discusses directions for future research.

8.1 Research summary

This research has developed methods to enable a knowledge-based approach to predict the expected cost impact of a proposed Engineering Change (EC) effect by integrating the relevant concepts from information theory and data mining along with the knowledge specific to the domain of manufacturing. The overall goal is to enable manufacturing enterprises to make quick decisions about which effects of proposed EC should undergo a detailed evaluation process.

Only some of the large number of EC attributes are important for retrieving past ECs that can be used to evaluate the impact of a proposed EC effect. The problem of determining important EC attributes has not been addressed earlier. This research formulates the problem of determining important attributes as a multi-objective optimization problem. Measures are defined to quantify importance of an attribute set for two interrelated target tasks, namely retrieving similar ECs and predicting impact of proposed EC effect. An ACO-based search procedure is used for efficiently locating the important set of attributes.

Utilizing past EC knowledge to predict the impact of proposed EC effect requires an approach to compute similarity between ECs. We are not aware of an approach to determine similarity between ECs in context of predicting impact of proposed EC effect. The approach to compute similarity between ECs developed in this research fills this gap. Since the available information is probabilistic, the measures of information are used for defining measures to compute similarity between two attribute values or ECs. The semantics associated with attribute values are utilized to determine similarity between two attribute values.

Finally, this research focuses on the problem of predicting impact of proposed EC effect based on the similar past ECs. This problem has not been addressed earlier. Our approach to address this problem incorporates a technique to quantify the differences, in context of predicting impact, between important attribute values in two changes. Since the nature of relationship between attribute value differences and differences in impact is unknown, the BayesŠ rule is utilized to predict the difference in impact based on the differences between attribute values.

8.2 Specific contributions

The primary contribution of this research is the application and enhancement of techniques from data mining and machine learning using domain-specific knowledge to address the problem of EC evaluation. This includes:

- Measures to quantify importance of an attribute set for two interrelated target tasks, namely retrieving similar ECs and predicting impact of proposed EC effect.

- Procedure to estimate true form of probability distribution of attribute values by combining observed distributions and the information obtained from domain rules.

- Information-theoretic similarity measures to compute similarity between two ECs or attribute values. The measures are defined such that the similarity values are determined using the available statistical knowledge (observed probabilities) as well as domain-specific knowledge (taxonomic definitions).

- A procedure based on the principle of minimum cross entropy to estimate the probabilities required in the BayesŠ rule, which is used in the process of predicting the impact of proposed EC effect from the similar past engineering changes.

8.3 Application to other problems

This section highlights the problems in other application areas which can benefit from the work presented in this dissertation.

1. The microarray gene expression data consist of an array of tissue samples that are classified into different classes of phenotypes, e.g., cancerous or normal. The tissue samples are represented using a large number, i.e., in range of 2000 to 30000, of genes (or features). In bioinformatics, the problem of phenotype classification is to classify a new tissue sample based on the values (or expression level) of genes measured in it [89]. The two values of genes are typically interrelated by a taxonomy of IS-A type [90, 91]. The system developed in this dissertation can be applied to address the problem of phenotype classification.

2. The problem of text categorization is to classify a document, which is defined

using a set of terms, into a predefined category [92]. This problem has several applications, e.g., categorization of medical records in the area of health care informatics [93]. The overall approach developed in this dissertation can be suitably applied to address the problem of text categorization.

3. The feature-based shape similarity assessments require an approach to compute similarity between two values of a product feature [94]. The approach to compute attribute values similarity proposed in this research can be applied to address this problem.

8.4 Future research

Limitations of methods discussed in this dissertation are discussed in sections 4.8, 5.7 and 6.6, respectively. These sections also summarize the future work to address some of these limitations and to extend the methods developed in this dissertation.

In addition, following paragraphs discuss a few interesting and major directions for future research that are related to the problems discussed in this dissertation.

1. **Accounting for temporal changes:** Currently our problem and approach does not account for any temporal changes. The temporal changes can be at an attribute level. For instance, the cost or specifications of a material might have changed over the period of time or the list of attributes that capture an EC might be altered over a period of time. Similarly, there can be temporal changes in the cost, state and usage of resources within an enterprise. For example, cost of purchasing a new equipment at the time of implementing proposed EC might be significantly different from that at the time of implementing a past EC that is most similar to the proposed EC. Such changes can affect the decision about the important attributes as well as the prediction of impact of proposed

EC effect. Accounting for such temporal changes in the process of predicting impact of proposed EC effect presents an interesting avenue of future research.

2. **Evaluating impact of process changes:** The scope of our current methods is limited to product changes. There can also be changes at the process-level. For example, changes in manufacturing process technique that is applicable to multiple products. Such changes are bound to have some impact on the other elements of an enterprise. For example, change in manufacturing process sequence of a product might have an impact on the control and usage of manufacturing resources within an enterprise. The ability to evaluate the impact of process changes within an enterprise will be extremely useful.

3. **Accounting for interrelationships among past changes:** Our current approach assumes that past engineering changes in the database are independent of each other. However, a database might contain two or more changes that are interrelated. It will be interesting to extend our current approach to account for interrelationships among changes in the database.

4. **Managing scalability:** As discussed in chapter I, a typical enterprise handles large number of ECs each year. As a result, an EC database might contain several thousand ECs. An important topic of future research should analyze and modify the developed methods to manage its scalability. This shall require algorithms to use resources, such as parallel computing and the ability to store large datasets.

5. **Development of a repository:** As discussed chapter I, there lacks benchmark datasets that can be utilized to validate the procedures developed in this area. This research created a small database of ECs to evaluate various ap-

proaches developed in this research. It will be extremely useful to extend this database to create a large repository of benchmark EC datasets, which can be utilized to validate the procedures developed by researchers in this field.

APPENDICES

APPENDIX A

Example engineering change database

A.1 Example STEP-compliant data model for EC

Figure A.1 illustrates an example STEP-compliant data model for capturing the knowledge associated with a change. There are 100 attributes, out of which 62 are of categorical type and the remaining are of quantitative type. Several elements of this data model are derived from the STEP manufacturing APs: AP 224 [58] and AP 240 [59]. For the simplicity of explanation, figure utilizes a terminology that is different from that used for various elements in the STEP APs.

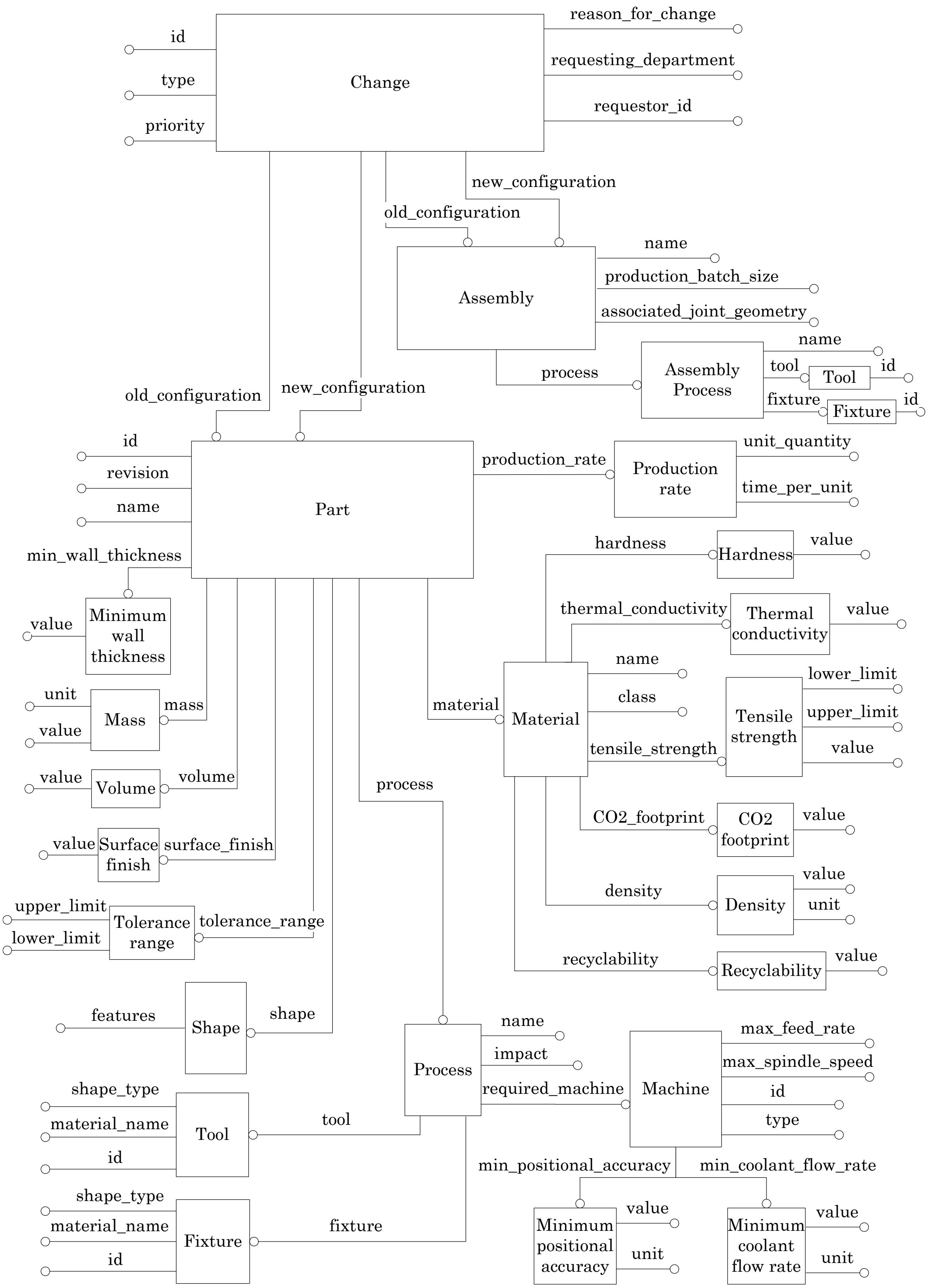


Figure A.1: Example STEP-compliant data model for capturing the knowledge associated with an EC

A.2 Example ECs

Figures A.2 , . . . , A.18 illustrate the 17 changes in the example EC knowledgebase which are used for evaluation of various approaches developed in this dissertation.

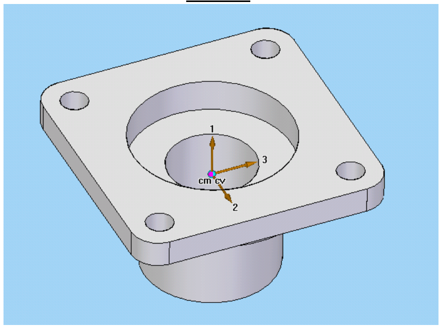
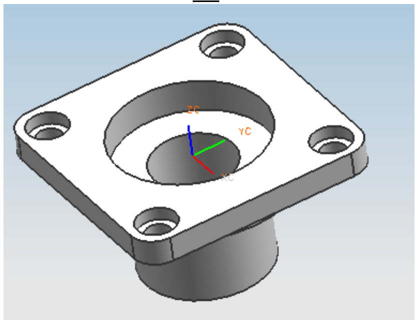
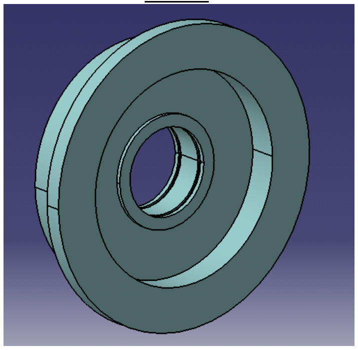
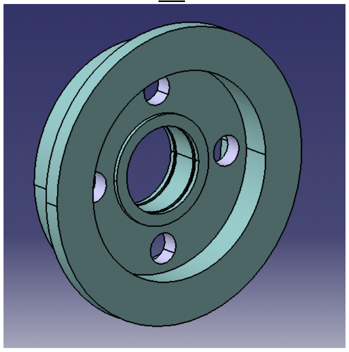
Change ID	*EC-1*	
Change in shape of part	from  	to 
A few attribute values	A) Values of a few attributes related to the old_configuration of part which are typically found to be important: *1)* part.shape.feature_existence = *{protrusion, pocket, simple-hole}* 2) part.process.name = *{casting, milling}* 3) part.process.tool.id = *{cast-tool-1, mill-tool-2}* 4) part.process.required_machine.type = *{casting, mill}* B) Values of a few attributes related to the new_configuration of part which are typically found to be important: 1) part.shape.feature_existence = *{ protrusion, pocket, simple-hole, counterbore-hole}* 2) part.process.name = *{casting, milling}* 3) part.material.name = *CS-1030* 4) part.production_rate.unit_quantity = *15250* 5) part.surface_finish.value = *smooth* 6) part.tolerance_range.upper_limit = *0.03125* 7) part.tolerance_range.lower_limit = *0.00475* 8) part.volume.value = *0.48* C) Values of a few attributes that are unrelated to any specific configuration of parts and are typically found to be important: 1) reason_for_change = *technical improvement* 2) requesting_department = *design*	
Impact	Overall impact of change on Process = *low* The overall impact is determined based on the following information about this change, 1) relative cost of new tool/equipment/technology= *1* (on scale of 5) 2) relative cost of increase in production time = *1* (on scale of 5) 3) relative cost of disruption in manufacturing = *1* (on scale of 3) 4) relative cost of redesign = *2* (on scale of 3) 5) relative cost of training employees = *0* (on scale of 3)	

Figure A.2: Example change $EC-1$ of type change in shape. Values of a few attributes and impact of $EC-1$ on Process is shown

<table>
<tr><td><b>Change ID</b></td><td colspan="2"><i>EC-2</i></td></tr>
<tr><td><b>Change in shape of part</b></td><td>from
</td><td>to
</td></tr>
<tr><td><b>A few attribute values</b></td><td colspan="2">A) Values of a few attributes related to the old_configuration of part which are typically found to be important:
<i>1)</i> part.shape.feature_existence = <i>{protrusion, pocket, simple-hole}</i>
2) part.process.name = <i>{casting}</i>
3) part.process.tool.id = <i>{cast-tool-3}</i>
4) part.process.required_machine.type = <i>{casting}</i>

B) Values of a few attributes related to the new_configuration of part which are typically found to be important:
1) part.shape.feature_existence = <i>{ protrusion, pocket, simple-hole}</i>
2) part.process.name = <i>{casting, drilling}</i>
3) part.material.name = <i>CI-100</i>
4) part.production_rate.unit_quantity = <i>45000</i>
5) part.surface_finish.value = <i>smooth</i>
6) part.tolerance_range.upper_limit = <i>0.03125</i>
7) part.tolerance_range.lower_limit = <i>0.00475</i>
8) part.volume.value = <i>20.63</i>

C) Values of a few attributes that are unrelated to any specific configuration of parts and are typically found to be important:
1) reason_for_change = <i>technical improvement</i>
2) requesting_department = <i>design</i></td></tr>
<tr><td><b>Impact</b></td><td colspan="2">Overall impact of change on Process = <i>low</i>

The overall impact is determined based on the following information about this change,
1) relative cost of new tool/equipment/technology = <i>1</i>(on scale of 5)
2) relative cost of increase in production time = <i>4</i> (on scale of 5)
3) relative cost of disruption in manufacturing = <i>1</i> (on scale of 3)
4) relative cost of redesign = <i>1</i> (on scale of 3)
5) relative cost of training employees = <i>0</i> (on scale of 3)</td></tr>
</table>

Figure A.3: Example change $EC-2$ of type change in shape. Values of a few attributes and impact of $EC-2$ on Process is shown

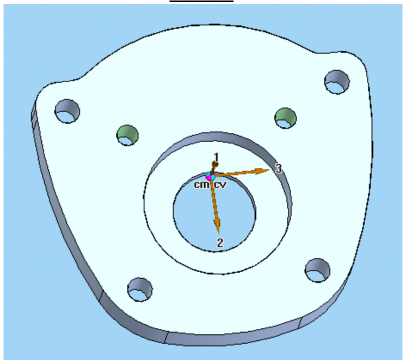
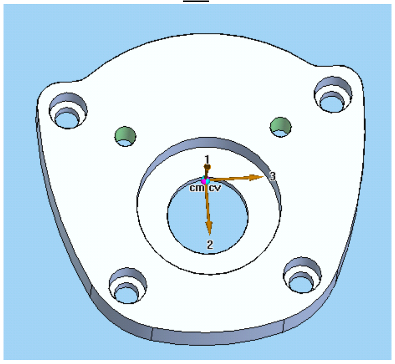
Change ID	*EC-3*	
Change in shape of part	from  	to 
A few attribute values	A) Values of a few attributes related to the old_configuration of part which are typically found to be important: *1)* part.shape.feature_existence = *{protrusion, pocket, simple-hole}* 2) part.process.name = *{milling}* 3) part.process.tool.id = *{mill-tool-2}* 4) part.process.required_machine.type = *{mill}* B) Values of a few attributes related to the new_configuration of part which are typically found to be important: 1) part.shape.feature_existence = *{ protrusion, pocket, simple-hole, counterbore-hole}* 2) part.process.name = *{milling}* 3) part.material.name = *CS-1080* 4) part.production_rate.unit_quantity = *27750* 5) part.surface_finish.value = *very-smooth* 6) part.tolerance_range.upper_limit = *0.0135* 7) part.tolerance_range.lower_limit = *0.000775* 8) part.volume.value = *0.75* C) Values of a few attributes that are unrelated to any specific configuration of parts and are typically found to be important: 1) reason_for_change = *customer request* 2) requesting_department = *sales*	
Impact	Overall impact of change on Process = *low* The overall impact is determined based on the following information about this change, 1) relative cost of new tool/equipment/technology = *1*(on scale of 5) 2) relative cost of increase in production time = *4* (on scale of 5) 3) relative cost of disruption in manufacturing = *1* (on scale of 3) 4) relative cost of redesign = *1* (on scale of 3) 5) relative cost of training employees = *0* (on scale of 3)	

Figure A.4: Example change $EC - 3$ of type change in shape. Values of a few attributes and impact of $EC - 3$ on Process is shown

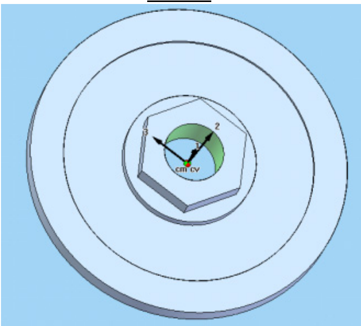
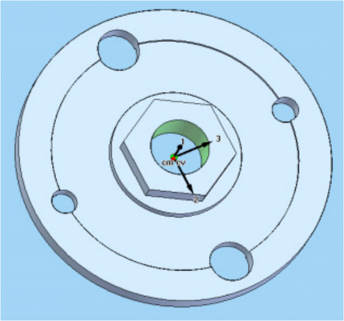
Change ID	*EC-4*	
Change in shape of part	from 	to 
A few attribute values	A) Values of a few attributes related to the old_configuration of part which are typically found to be important: *1)* part.shape.feature_existence = *{protrusion, pocket, simple-hole}* 2) part.process.name = *{casting}* 3) part.process.tool.id = *{cast-tool-1}* 4) part.process.required_machine.type = *{casting}* B) Values of a few attributes related to the new_configuration of part which are typically found to be important: 1) part.shape.feature_existence = *{ protrusion, pocket, simple-hole}* 2) part.process.name = *{casting, drilling}* 3) part.material.name = *AL-105* 4) part.production_rate.unit_quantity = *27750* 5) part.surface_finish.value = *smooth* 6) part.tolerance_range.upper_limit = *0.03125* 7) part.tolerance_range.lower_limit = *0.00475* 8) part.volume.value = *0.3* C) Values of a few attributes that are unrelated to any specific configuration of parts and are typically found to be important: 1) reason_for_change = *technical improvement* 2) requesting_department = *design*	
Impact	Overall impact of change on Process = *low* The overall impact is determined based on the following information about this change, 1) relative cost of new tool/equipment/technology = *1*(on scale of 5) 2) relative cost of increase in production time = *4* (on scale of 5) 3) relative cost of disruption in manufacturing = *1* (on scale of 3) 4) relative cost of redesign = *1* (on scale of 3) 5) relative cost of training employees = *0* (on scale of 3)	

Figure A.5: Example change $EC - 4$ of type change in shape. Values of a few attributes and impact of $EC - 4$ on Process is shown

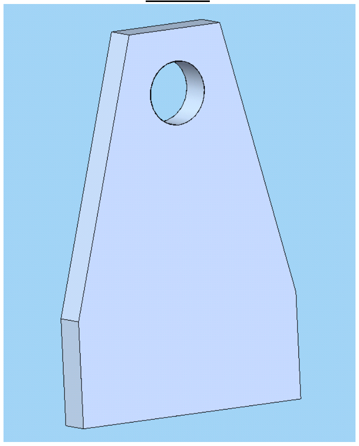
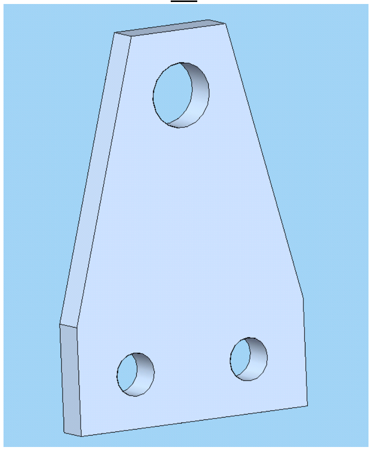
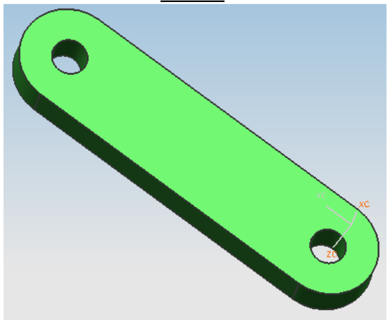
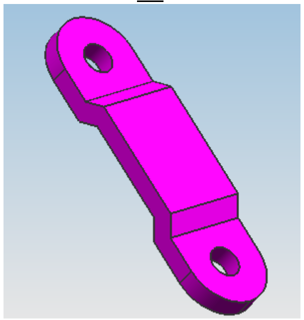
Change ID	*EC-5*	
Change in shape of part	<u>from</u> 	<u>to</u> 
A few attribute values	A) Values of a few attributes related to the old_configuration of part which are typically found to be important: *1)* part.shape.feature_existence = *{protrusion, simple-hole}* 2) part.process.name = *{planing}* 3) part.process.tool.id = *{planer-tool-1}* 4) part.process.required_machine.type = *{planer}* B) Values of a few attributes related to the new_configuration of part which are typically found to be important: 1) part.shape.feature_existence = *{ protrusion, simple-hole}* 2) part.process.name = *{planing, drilling}* 3) part.material.name = *CI-220* 4) part.production_rate.unit_quantity = *15250* 5) part.surface_finish.value = *very-smooth* 6) part.tolerance_range.upper_limit = *0.0055* 7) part.tolerance_range.lower_limit = *0.000325* 8) part.volume.value = *4.4* C) Values of a few attributes that are unrelated to any specific configuration of parts and are typically found to be important: 1) reason_for_change = *corrective action* 2) requesting_department = *manufacturing*	
Impact	Overall impact of change on Process = *low* The overall impact is determined based on the following information about this change, 1) relative cost of new tool/equipment/technology = *1*(on scale of 5) 2) relative cost of increase in production time = *3* (on scale of 5) 3) relative cost of disruption in manufacturing = *1* (on scale of 3) 4) relative cost of redesign = *2* (on scale of 3) 5) relative cost of training employees = *0* (on scale of 3)	

Figure A.6: Example change $EC-5$ of type change in shape. Values of a few attributes and impact of $EC-5$ on Process is shown

<table>
<tr><td><b>Change ID</b></td><td colspan="2"><i>EC-6</i></td></tr>
<tr><td><b>Change in shape of part</b></td><td>from
</td><td>to
</td></tr>
<tr><td><b>A few attribute values</b></td><td colspan="2">A) Values of a few attributes related to the old_configuration of part which are typically found to be important:
<i>1)</i> part.shape.feature_existence = <i>{protrusion, simple-hole}</i>
2) part.process.name = <i>{drilling}</i>
3) part.process.tool.id = <i>{drill-tool-3}</i>
4) part.process.required_machine.type = <i>{drill}</i>

B) Values of a few attributes related to the new_configuration of part which are typically found to be important:
1) part.shape.feature_existence = <i>{ protrusion, edge-blend, simple-hole}</i>
2) part.process.name = <i>{planing, drilling}</i>
3) part.material.name = <i>AL-105</i>
4) part.production_rate.unit_quantity = <i>15250</i>
5) part.surface_finish.value = <i>rough</i>
6) part.tolerance_range.upper_limit = <i>0.02175</i>
7) part.tolerance_range.lower_limit = <i>0.0022</i>
8) part.volume.value = <i>0.3</i>

C) Values of a few attributes that are unrelated to any specific configuration of parts and are typically found to be important:
1) reason_for_change = <i>corrective action</i>
2) requesting_department = <i>manufacturing</i></td></tr>
<tr><td><b>Impact</b></td><td colspan="2">Overall impact of change on Process = <i>high</i>

The overall impact is determined based on the following information about this change,
1) relative cost of new tool/equipment/technology = <i>2</i>(on scale of 5)
2) relative cost of increase in production time = <i>4</i> (on scale of 5)
3) relative cost of disruption in manufacturing = <i>2</i> (on scale of 3)
4) relative cost of redesign = <i>1</i> (on scale of 3)
5) relative cost of training employees = <i>0</i> (on scale of 3)</td></tr>
</table>

Figure A.7: Example change $EC-6$ of type change in shape. Values of a few attributes and impact of $EC-6$ on Process is shown

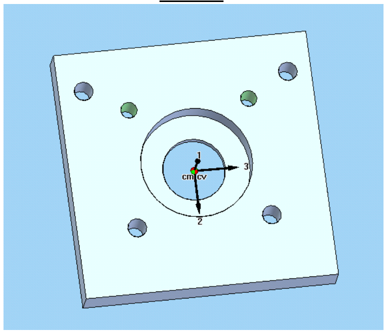
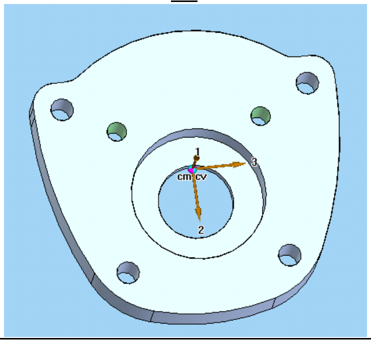
Change ID	*EC-7*	
Change in shape of part	from 	to 
A few attribute values	A) Values of a few attributes related to the old_configuration of part which are typically found to be important: *1)* part.shape.feature_existence = *{protrusion, pocket, simple-hole}* 2) part.process.name = *{milling}* 3) part.process.tool.id = *{mill-tool-2}* 4) part.process.required_machine.type = *{mill}* B) Values of a few attributes related to the new_configuration of part which are typically found to be important: 1) part.shape.feature_existence = *{ protrusion, pocket, simple-hole, counterbore-hole}* 2) part.process.name = *{milling}* 3) part.material.name = *CS-1080* 4) part.production_rate.unit_quantity = *24000* 5) part.surface_finish.value = *very-smooth* 6) part.tolerance_range.upper_limit = *0.0135* 7) part.tolerance_range.lower_limit = *0.000775* 8) part.volume.value = *0.75* C) Values of a few attributes that are unrelated to any specific configuration of parts and are typically found to be important: 1) reason_for_change = *technical improvement* 2) requesting_department = *sales*	
Impact	Overall impact of change on Process = *high* The overall impact is determined based on the following information about this change, 1) relative cost of new tool/equipment/technology = *1*(on scale of 5) 2) relative cost of increase in production time = *4* (on scale of 5) 3) relative cost of disruption in manufacturing = *2* (on scale of 3) 4) relative cost of redesign = *1* (on scale of 3) 5) relative cost of training employees = *0* (on scale of 3)	

Figure A.8: Example change $EC - 7$ of type change in shape. Values of a few attributes and impact of $EC - 7$ on Process is shown

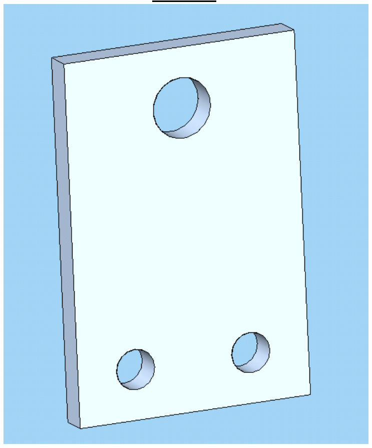
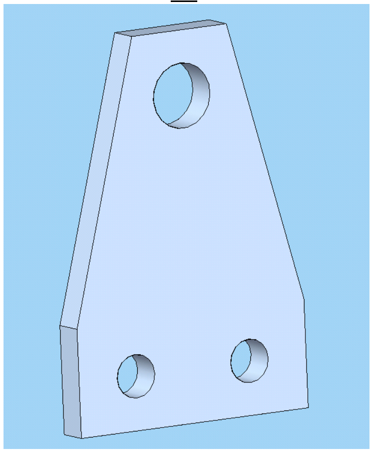
Change ID	*EC-8*	
Change in shape of part	from 	to 
A few attribute values	A) Values of a few attributes related to the old_configuration of part which are typically found to be important: *1)* part.shape.feature_existence = *{protrusion, simple-hole}* 2) part.process.name = *{drilling}* 3) part.process.tool.id = *{drill-tool-4}* 4) part.process.required_machine.type = *{drill}* B) Values of a few attributes related to the new_configuration of part which are typically found to be important: 1) part.shape.feature_existence = *{ protrusion, simple-hole}* 2) part.process.name = *{drilling, planing}* 3) part.material.name = *CI-220* 4) part.production_rate.unit_quantity = *15250* 5) part.surface_finish.value = *rough* 6) part.tolerance_range.upper_limit = *0.02175* 7) part.tolerance_range.lower_limit = *0.0022* 8) part.volume.value = *4.4* C) Values of a few attributes that are unrelated to any specific configuration of parts and are typically found to be important: 1) reason_for_change = *customer request* 2) requesting_department = *marketing*	
Impact	Overall impact of change on Process = *high* The overall impact is determined based on the following information about this change, 1) relative cost of new tool/equipment/technology = *2*(on scale of 5) 2) relative cost of increase in production time = *4* (on scale of 5) 3) relative cost of disruption in manufacturing = *1* (on scale of 3) 4) relative cost of redesign = *1* (on scale of 3) 5) relative cost of training employees = *0* (on scale of 3)	

Figure A.9: Example change $EC - 8$ of type change in shape. Values of a few attributes and impact of $EC - 8$ on Process is shown

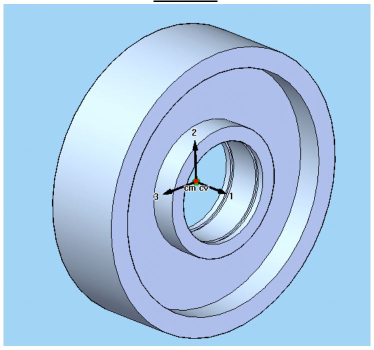
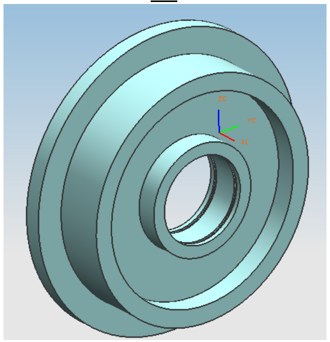
Change ID	*EC-9*
Change in shape of part	from  to 
A few attribute values	A) Values of a few attributes related to the old_configuration of part which are typically found to be important: *1)* part.shape.feature_existence = *{protrusion, pocket, simple-hole}* 2) part.process.name = *{casting}* 3) part.process.tool.id = *{cast-tool-4}* 4) part.process.required_machine.type = *{casting}* B) Values of a few attributes related to the new_configuration of part which are typically found to be important: 1) part.shape.feature_existence = *{ protrusion, pocket, simple-hole}* 2) part.process.name = *{casting}* 3) part.material.name = *CI-100* 4) part.production_rate.unit_quantity = *45000* 5) part.surface_finish.value = *smooth* 6) part.tolerance_range.upper_limit = *0.03125* 7) part.tolerance_range.lower_limit = *0.00475* 8) part.volume.value = *5.22* C) Values of a few attributes that are unrelated to any specific configuration of parts and are typically found to be important: 1) reason_for_change = *problem prevention* 2) requesting_department = *manufacturing*
Impact	Overall impact of change on Process = *high* The overall impact is determined based on the following information about this change, 1) relative cost of new tool/equipment/technology = *4*(on scale of 5) 2) relative cost of increase in production time = *0* (on scale of 5) 3) relative cost of disruption in manufacturing = *2* (on scale of 3) 4) relative cost of redesign = *2* (on scale of 3) 5) relative cost of training employees = *0* (on scale of 3)

Figure A.10: Example change $EC-9$ of type change in shape. Values of a few attributes and impact of $EC-9$ on Process is shown

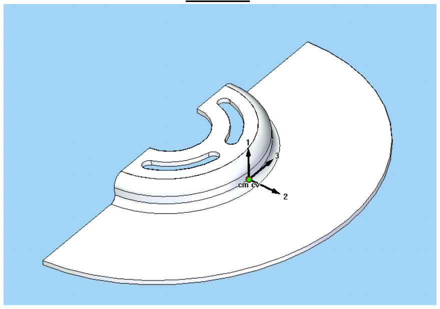
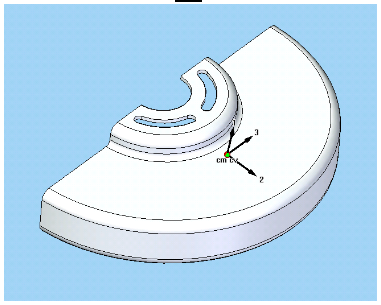
Change ID	*EC-10*	
Change in shape of part	<u>from</u> 	<u>to</u> 
A few attribute values	A) Values of a few attributes related to the old_configuration of part which are typically found to be important: *1)* part.shape.feature_existence = *{protrusion, pocket}* 2) part.process.name = *{molding}* 3) part.process.tool.id = *{mold-tool-3}* 4) part.process.required_machine.type = *{injection molding}* B) Values of a few attributes related to the new_configuration of part which are typically found to be important: 1) part.shape.feature_existence = *{ protrusion, simple-hole}* 2) part.process.name = *{molding}* 3) part.material.name = *EPOXY* 4) part.production_rate.unit_quantity = *45000* 5) part.surface_finish.value = *very-smooth* 6) part.tolerance_range.upper_limit = *0.0135* 7) part.tolerance_range.lower_limit = *0.000775* 8) part.volume.value = *0.88* C) Values of a few attributes that are unrelated to any specific configuration of parts and are typically found to be important: 1) reason_for_change = *problem prevention* 2) requesting_department = *sales*	
Impact	Overall impact of change on Process = *high* The overall impact is determined based on the following information about this change, 1) relative cost of new tool/equipment/technology = *5*(on scale of 5) 2) relative cost of increase in production time = *0* (on scale of 5) 3) relative cost of disruption in manufacturing = *3* (on scale of 3) 4) relative cost of redesign = *1* (on scale of 3) 5) relative cost of training employees = *1* (on scale of 3)	

Figure A.11: Example change $EC - 10$ of type change in shape. Values of a few attributes and impact of $EC - 10$ on Process is shown

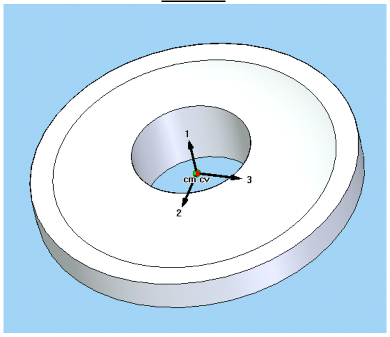
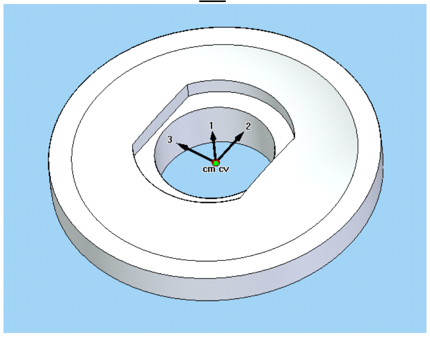
Change ID	*EC-11*	
Change in shape of part	from  	to 
A few attribute values	A) Values of a few attributes related to the old_configuration of part which are typically found to be important: *1)* part.shape.feature_existence = *{protrusion, simple-hole}* 2) part.process.name = *{casting}* 3) part.process.tool.id = *{cast-tool-5}* 4) part.process.required_machine.type = *{casting}* B) Values of a few attributes related to the new_configuration of part which are typically found to be important: 1) part.shape.feature_existence = *{ protrusion, pocket, simple-hole}* 2) part.process.name = *{casting, milling}* 3) part.material.name = *AL-105* 4) part.production_rate.unit_quantity = *24000* 5) part.surface_finish.value = *smooth* 6) part.tolerance_range.upper_limit = *0.03125* 7) part.tolerance_range.lower_limit = *0.00475* 8) part.volume.value = *0.44* C) Values of a few attributes that are unrelated to any specific configuration of parts and are typically found to be important: 1) reason_for_change = *technical improvement* 2) requesting_department = *design*	
Impact	Overall impact of change on Process = *high* The overall impact is determined based on the following information about this change, 1) relative cost of new tool/equipment/technology = *2*(on scale of 5) 2) relative cost of increase in production time = *4* (on scale of 5) 3) relative cost of disruption in manufacturing = *1* (on scale of 3) 4) relative cost of redesign = *2* (on scale of 3) 5) relative cost of training employees = *0* (on scale of 3)	

Figure A.12: Example change $EC - 11$ of type change in shape. Values of a few attributes and impact of $EC - 11$ on Process is shown

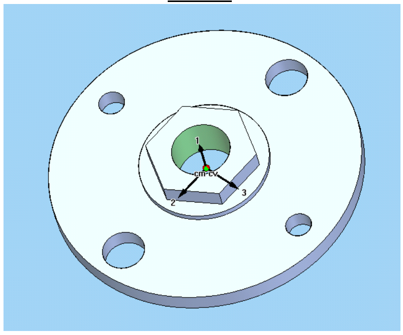
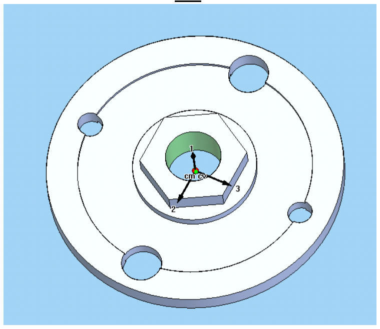
Change ID	*EC-12*	
Change in shape of part	<u>from</u> 	<u>to</u> 
A few attribute values	A) Values of a few attributes related to the old_configuration of part which are typically found to be important: *1)* part.shape.feature_existence = *{protrusion, simple-hole}* 2) part.process.name = *{casting, drilling}* 3) part.process.tool.id = *{cast-tool-2, drill-tool-3}* 4) part.process.required_machine.type = *{casting, drill}* B) Values of a few attributes related to the new_configuration of part which are typically found to be important: 1) part.shape.feature_existence = *{ protrusion, pocket, simple-hole}* 2) part.process.name = *{casting, milling}* 3) part.material.name = *AL-105* 4) part.production_rate.unit_quantity = *45000* 5) part.surface_finish.value = *smooth* 6) part.tolerance_range.upper_limit = *0.03125* 7) part.tolerance_range.lower_limit = *0.00475* 8) part.volume.value = *0.3* C) Values of a few attributes that are unrelated to any specific configuration of parts and are typically found to be important: 1) reason_for_change = *technical improvement* 2) requesting_department = *design*	
Impact	Overall impact of change on Process = *high* The overall impact is determined based on the following information about this change, 1) relative cost of new tool/equipment/technology = *1*(on scale of 5) 2) relative cost of increase in production time = *5* (on scale of 5) 3) relative cost of disruption in manufacturing = *1* (on scale of 3) 4) relative cost of redesign = *1* (on scale of 3) 5) relative cost of training employees = *0* (on scale of 3)	

Figure A.13: Example change $EC - 12$ of type change in shape. Values of a few attributes and impact of $EC - 12$ on Process is shown

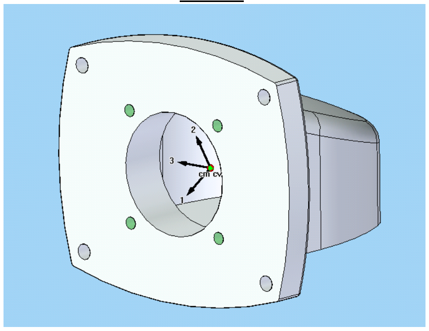
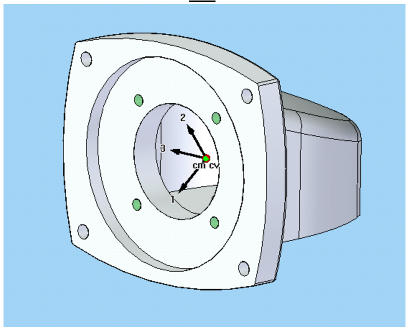
Change ID	*EC-13*	
Change in shape of part	from  	to 
A few attribute values	A) Values of a few attributes related to the old_configuration of part which are typically found to be important: *1)* part.shape.feature_existence = *{protrusion, simple-hole}* 2) part.process.name = *{molding}* 3) part.process.tool.id = *{mold-tool-3}* 4) part.process.required_machine.type = *{injection molding}* B) Values of a few attributes related to the new_configuration of part which are typically found to be important: 1) part.shape.feature_existence = *{ protrusion, pocket, simple-hole}* 2) part.process.name = *{molding}* 3) part.material.name = *EPOXY* 4) part.production_rate.unit_quantity = *45000* 5) part.surface_finish.value = *very-smooth* 6) part.tolerance_range.upper_limit = *0.0135* 7) part.tolerance_range.lower_limit = *0.000775* 8) part.volume.value = *2.97* C) Values of a few attributes that are unrelated to any specific configuration of parts and are typically found to be important: 1) reason_for_change = *corrective action* 2) requesting_department = *manufacturing*	
Impact	Overall impact of change on Process = *high* The overall impact is determined based on the following information about this change, 1) relative cost of new tool/equipment/technology = *5*(on scale of 5) 2) relative cost of increase in production time = *0* (on scale of 5) 3) relative cost of disruption in manufacturing = *3* (on scale of 3) 4) relative cost of redesign = *2* (on scale of 3) 5) relative cost of training employees = *1* (on scale of 3)	

Figure A.14: Example change $EC - 13$ of type change in shape. Values of a few attributes and impact of $EC - 13$ on Process is shown

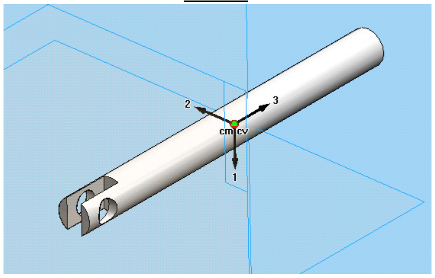
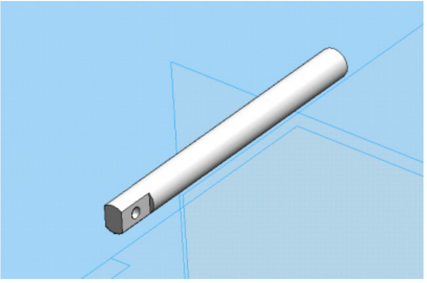
Change ID	*EC-14*	
Change in shape of part	from 	to 
A few attribute values	A) Values of a few attributes related to the old_configuration of part which are typically found to be important: *1)* part.shape.feature_existence = *{protrusion, pocket, simple-hole}* 2) part.process.name = *{drilling, milling}* 3) part.process.tool.id = *{drill-tool-4, mill-tool-2}* 4) part.process.required_machine.type = *{drill, mill}* B) Values of a few attributes related to the new_configuration of part which are typically found to be important: 1) part.shape.feature_existence = *{ protrusion, pocket, simple-hole}* 2) part.process.name= *{drilling, milling}* 3) part.material.name = *CU-C85700* 4) part.production_rate.unit_quantity = *24000* 5) part.surface_finish.value = *rough* 6) part.tolerance_range.upper_limit = *0.02175* 7) part.tolerance_range.lower_limit = *0.0022* 8) part.volume.value = *0.43* C) Values of a few attributes that are unrelated to any specific configuration of parts and are typically found to be important: 1) reason_for_change = *technical improvement* 2) requesting_department = *design*	
Impact	Overall impact of change on Process = *low* The overall impact is determined based on the following information about this change, 1) relative cost of new tool/equipment/technology = *1*(on scale of 5) 2) relative cost of increase in production time = *1* (on scale of 5) 3) relative cost of disruption in manufacturing = *1* (on scale of 3) 4) relative cost of redesign = *3* (on scale of 3) 5) relative cost of training employees = *0* (on scale of 3)	

Figure A.15: Example change $EC-14$ of type change in shape. Values of a few attributes and impact of $EC-14$ on Process is shown

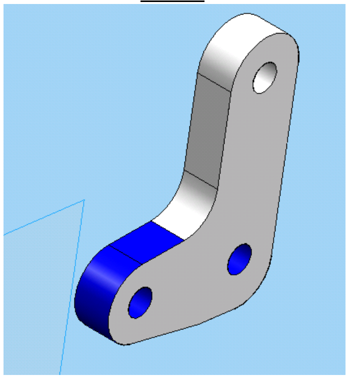
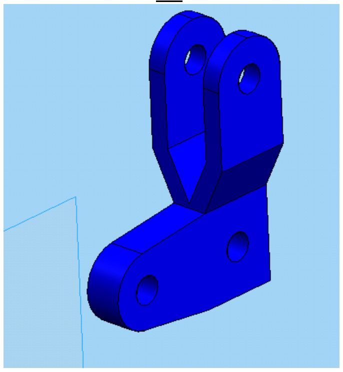
Change ID	*EC-15*
Change in shape of part	from  to 
A few attribute values	A) Values of a few attributes related to the old_configuration of part which are typically found to be important: *1)* part.shape.feature_existence = *{protrusion, edge-blend, simple-hole}* 2) part.process.name = *{casting }* 3) part.process.tool.id = *{cast-tool-4}* 4) part.process.required_machine.type = *{casting}* B) Values of a few attributes related to the new_configuration of part which are typically found to be important: 1) part.shape.feature_existence = *{ protrusion, pocket, simple-hole}* 2) part.process.name = *{casting}* 3) part.material.name = *CS-1080* 4) part.production_rate.unit_quantity = *45000* 5) part.surface_finish.value = *smooth* 6) part.tolerance_range.upper_limit = *0.03125* 7) part.tolerance_range.lower_limit = *0.00475* 8) part.volume.value = *0.16* C) Values of a few attributes that are unrelated to any specific configuration of parts and are typically found to be important: 1) reason_for_change = *technical improvement* 2) requesting_department = *design*
Impact	Overall impact of change on Process = *high* The overall impact is determined based on the following information about this change, 1) relative cost of new tool/equipment/technology = *4*(on scale of 5) 2) relative cost of increase in production time = *0* (on scale of 5) 3) relative cost of disruption in manufacturing = *2* (on scale of 3) 4) relative cost of redesign = *3* (on scale of 3) 5) relative cost of training employees = *0* (on scale of 3)

Figure A.16: Example change $EC - 15$ of type change in shape. Values of a few attributes and impact of $EC - 15$ on Process is shown

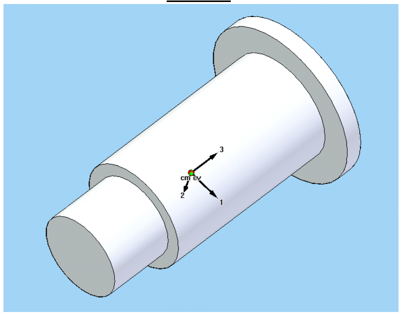
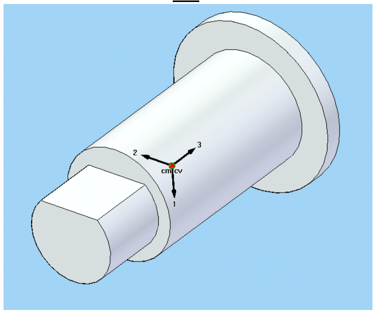
Change ID	*EC-16*	
Change in shape of part	from 	to 
A few attribute values	A) Values of a few attributes related to the old_configuration of part which are typically found to be important: *1)* part.shape.feature_existence = *{protrusion}* 2) part.process.name = *{turning}* 3) part.process.tool.id = *{turn-tool-2}* 4) part.process.required_machine.type = *{lathe}* B) Values of a few attributes related to the new_configuration of part which are typically found to be important: 1) part.shape.feature_existence = *{ protrusion, pocket}* 2) part.process.name = *{milling, turning}* 3) part.material.name = *CU-C85700* 4) part.production_rate.unit_quantity = *15250* 5) part.surface_finish.value = *very-smooth* 6) part.tolerance_range.upper_limit = *0.0055* 7) part.tolerance_range.lower_limit = *0.000325* 8) part.volume.value = *0.19* C) Values of a few attributes that are unrelated to any specific configuration of parts and are typically found to be important: 1) reason_for_change = *corrective action* 2) requesting_department = *manufacturing*	
Impact	Overall impact of change on Process = *high* The overall impact is determined based on the following information about this change, 1) relative cost of new tool/equipment/technology = *2*(on scale of 5) 2) relative cost of increase in production time = *3* (on scale of 5) 3) relative cost of disruption in manufacturing = *1* (on scale of 3) 4) relative cost of redesign = *2* (on scale of 3) 5) relative cost of training employees = *1* (on scale of 3)	

Figure A.17: Example change $EC - 16$ of type change in shape. Values of a few attributes and impact of $EC - 16$ on Process is shown

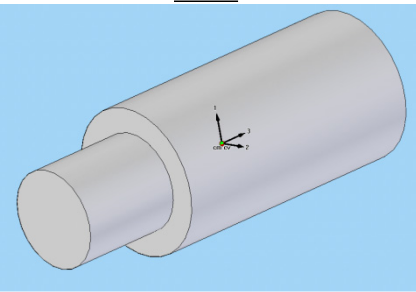
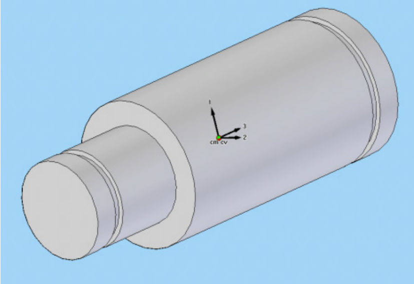
Change ID	*EC-17*	
Change in shape of part	from 	to 
A few attribute values	A) Values of a few attributes related to the old_configuration of part which are typically found to be important: *1)* part.shape.feature_existence = *{protrusion}* 2) part.process.name = *{turning}* 3) part.process.tool.id = *{turn-tool-2}* 4) part.process.required_machine.type = *{lathe}* B) Values of a few attributes related to the new_configuration of part which are typically found to be important: 1) part.shape.feature_existence = *{ protrusion, pocket}* 2) part.process.name = *{turning}* 3) part.material.name = *CU-C85700* 4) part.production_rate.unit_quantity = *15250* 5) part.surface_finish.value = *very-smooth* 6) part.tolerance_range.upper_limit = *0.0055* 7) part.tolerance_range.lower_limit = *0.000325* 8) part.volume.value = *2.9* C) Values of a few attributes that are unrelated to any specific configuration of parts and are typically found to be important: 1) reason_for_change = *technical improvement* 2) requesting_department = *design*	
Impact	Overall impact of change on Process = *low* The overall impact is determined based on the following information about this change, 1) relative cost of new tool/equipment/technology = *1*(on scale of 5) 2) relative cost of increase in production time = *1* (on scale of 5) 3) relative cost of disruption in manufacturing = *1* (on scale of 3) 4) relative cost of redesign = *2* (on scale of 3) 5) relative cost of training employees = *1* (on scale of 3)	

Figure A.18: Example change $EC - 17$ of type change in shape. Values of a few attributes and impact of $EC - 17$ on Process is shown

A.3 Datasets for evaluation

Table A.1 summarizes the training and test instances in various datasets created from 17 changes discussed in Appendix A.2 using 0.632 bootstrap procedure. These datasets are utilized for evaluation of various approaches developed in this dissertation.

Dataset #	17 training instances	6 test instances
1	EC-1, EC-2, EC-2, EC-2, EC-3, EC-3, EC-5, EC-6, EC-11, EC-12, EC-12, EC-13, EC-13, EC-14, EC-15 EC-15, EC-16	EC-4, EC-7, EC-8, EC-9, EC-10, EC-17
2	EC-3, EC-3, EC-4, EC-5, EC-5, EC-6, EC-8, EC-8, EC-12, EC-12, EC-13, EC-13, EC-14, EC-15, EC-15, EC-16, EC-17	EC-1, EC-2, EC-7, EC-9, EC-10, EC-11
3	EC-2, EC-3, EC-3, EC-5, EC-5, EC-6, EC-8, EC-8, EC-9, EC-9, EC-10, EC-10, EC-12, EC-14, EC-16, EC-16, EC-17	EC-1, EC-4, EC-7, EC-11, EC-13, EC-15
4	EC-1, EC-3, EC-5, EC-5, EC-5, EC-6, EC-8, EC-8, EC-9, EC-10, EC-10, EC-12, EC-12, EC-12, EC-14, EC-15, EC-17	EC-2, EC-4, EC-7, EC-11, EC-13, EC-16
5	EC-1, EC-1, EC-2, EC-4, EC-4, EC-6, EC-8, EC-8, EC-12, EC-12, EC-13, EC-13, EC-14, EC-15, EC-15, EC-16, EC-17	EC-3, EC-5, EC-7, EC-9, EC-10, EC-11
6	EC-1, EC-1, EC-4, EC-4, EC-5, EC-6, EC-7, EC-7, EC-9, EC-9, EC-10, EC-13, EC-14, EC-16, EC-16, EC-16, EC-17	EC-2, EC-3, EC-8, EC-11, EC-12, EC-15
7	EC-3, EC-3, EC-4, EC-4, EC-5, EC-6, EC-7, EC-7, EC-9, EC-9, EC-12, EC-12, EC-13, EC-13, EC-14, EC-16, EC-17	EC-1, EC-2, EC-8, EC-10, EC-11, EC-15
8	EC-1, EC-1, EC-2, EC-5, EC-5, EC-6, EC-7, EC-7, EC-9, EC-10, EC-10, EC-11, EC-11, EC-14, EC-16, EC-16, EC-17	EC-3, EC-4, EC-8, EC-12, EC-13, EC-15
9	EC-2, EC-3, EC-3, EC-4, EC-4, EC-6, EC-8, EC-8, EC-9, EC-9, EC-11, EC-12, EC-12, EC-13, EC-13, EC-14, EC-17	EC-1, EC-5, EC-7, EC-10, EC-15, EC-16
10	EC-1, EC-2, EC-2, EC-3, EC-4, EC-6, EC-8, EC-8, EC-9, EC-9, EC-13, EC-13, EC-14, EC-16, EC-16, EC-16, EC-17	EC-5, EC-7, EC-10, EC-11, EC-12, EC-15

Table A.1: 10 training and test datasets created from 17 example changes

APPENDIX B

Knowledge-base and evaluation results for the problem of determining important attributes

B.1 Example domain knowledge

This section presents the domain knowledge, i.e., *if-then* rules among attributes values, utilized for the case study and the evaluation of our approach to determine important attributes. Following 10 rules are identified from [69] and CES [68],

- ***if*** new_configuration.surface_finish.value = *A* AND
  new_configuration.tolerance_range.lower_limit = *0.0065* AND
  new_configuration.tolerance_range.upper_limit = *0.0275* AND
  new_configuration.min_wall_thickness.value = *0.09* AND
  new_configuration.production_rate.unit_quantity = *50000* AND
  old_configuration.material.class = *AL*
  then new_configuration.process.name = *casting*

- ***if*** new_configuration.surface_finish.value = *C* AND
  new_configuration.tolerance_range.lower_limit = *0.0035* AND
  new_configuration.tolerance_range.upper_limit = *0.016* AND

new_configuration.production_rate.unit_quantity = *5000*

then new_configuration.process.name = *{drilling, planing}*

- ***if*** new_configuration.surface_finish.value = *B* AND

  new_configuration.tolerance_range.lower_limit = *0.0008* AND

  new_configuration.tolerance_range.upper_limit = *0.0275* AND

  new_configuration.production_rate.unit_quantity = *50000*

  then new_configuration.process.name = *milling*

- ***if*** new_configuration.surface_finish.value = *A* AND

  new_configuration.tolerance_range.lower_limit = *0.0035* AND

  new_configuration.tolerance_range.upper_limit = *0.04* AND

  new_configuration.min_wall_thickness.value = *0.016* AND

  new_configuration.production_rate.unit_quantity = *50000* AND

  old_configuration.material.class = *PL*

  then new_configuration.process.name = *molding*

- ***if*** new_configuration.surface_finish.value = *A* AND

  new_configuration.tolerance_range.lower_limit = *0.00065* AND

  new_configuration.tolerance_range.upper_limit = *0.0275* AND

  new_configuration.min_wall_thickness.value = *0.6* AND

  new_configuration.production_rate.unit_quantity = *5000*

  then new_configuration.process.name = *planing*

- ***if*** new_configuration.surface_finish.value = *B* AND

  new_configuration.tolerance_range.lower_limit = *0.00065* AND

  new_configuration.tolerance_range.upper_limit = *0.016* AND

  new_configuration.production_rate.unit_quantity = *50000*

then new_configuration.process.name = *turning*

- ***if*** old_configuration.part.process.required_machine.type = *casting* AND old_configuration.part.shape.features = *{protrusion, pocket, simple-hole}* AND new_configuration.part.shape.features = *{protrusion, pocket, simple-hole, counterbore-hole}*

  then new_configuration.part.process.impact = *low*

- ***if*** old_configuration.part.process.required_machine.type = *casting* AND old_configuration.part.shape.feature_existence = *{protrusion, pocket, simple-hole}* AND new_configuration.part.shape.feature_existence = *{protrusion, pocket, simple-hole}*

  then impact = *low*

- ***if*** old_configuration.part.production_rate.unit_quantity = *50000* AND old_configuration.part.process.required_machine.type = *molding*

  then impact = *high*

- ***if*** old_configuration.part.process.required_machine.type = *molding* AND old_configuration.part.process.tool.id = *mold-too1-2* AND old_configuration.part.process.fixture.id = *mold-fixture-1*

  then impact = *high*

B.2 List of attributes that are candidates for being important

Tables B.1, B.2 and B.3 present 82 attributes that are candidates for being important. Each attribute is given an unique integer id, which is shown to its left.

Integer id	Candidate attribute
1	id
2	type
3	priority
4	reason_for_change
5	requesting_department
6	requestor_id
7	old_configuration.part.id
8	old_configuration.part.revision
9	old_configuration.part.name
10	old_configuration.part.production_rate.unit_quantity
11	old_configuration.part.production_rate.time_per_unit
12	old_configuration.part.mass.value
13	old_configuration.part.mass.unit
14	old_configuration.part.volume.value
15	old_configuration.part.surface_finish.value
16	old_configuration.part.min_wall_thickness.value
17	old_configuration.part.tolerance_range.lower_limit
18	old_configuration.part.tolerance_range.upper_limit
19	old_configuration.part.material.class
20	old_configuration.part.material.name
21	old_configuration.part.material.thermal_conductivity.value
22	old_configuration.part.material.CO2_footprint.value
23	old_configuration.part.material.tensile_strength.value
24	old_configuration.part.material.tensile_strength.lower_limit
25	old_configuration.part.material.tensile_strength.upper_limit
26	old_configuration.part.material.recyclability.value
27	old_configuration.part.material.hardness.value
28	old_configuration.part.material.density.value
29	old_configuration.part.material.density.unit
30	old_configuration.part.process.name
31	old_configuration.part.process.max_feed_rate
32	old_configuration.part.process.max_spindle_speed
33	old_configuration.part.process.required_machine.id
34	old_configuration.part.process.required_machine.type
35	old_configuration.part.process.required_machine. min_positional_accuracy.value

Table B.1: List of first 35 attributes that are candidates for being important

Integer id	Candidate attribute
36	old_configuration.part.process.required_machine. min_positional_accuracy.unit
37	old_configuration.part.process.required_machine. min_coolant_flow_rate.value
38	old_configuration.part.process.required_machine. min_coolant_flow_rate.unit
39	old_configuration.part.process.tool.id
40	old_configuration.part.process.tool.material.name
41	old_configuration.part.process.tool.shape.type
42	old_configuration.part.process.fixture.id
43	old_configuration.part.process.fixture.material.name
44	old_configuration.part.process.fixture.shape_type
45	old_configuration.part.shape.features
46	old_configuration.assembly.name
47	old_configuration.assembly.process.name
48	old_configuration.assembly.process.tool.id
49	old_configuration.assembly.process.fixture.id
50	old_configuration.assembly.associated_joint_geometry
51	old_configuration.assembly.production_batch_size
52	new_configuration.part.id
53	new_configuration.part.revision
54	new_configuration.part.name
55	new_configuration.part.production_rate.unit_quantity
56	new_configuration.part.volume.value
57	new_configuration.part.surface_finish.value
58	new_configuration.part.min_wall_thickness.value
59	new_configuration.part.tolerance_range.lower_limit
60	new_configuration.part.tolerance_range.upper_limit
61	new_configuration.part.material.class
62	new_configuration.part.material.name
63	new_configuration.part.material.thermal_conductivity.value
64	new_configuration.part.material.CO2_footprint.value
65	new_configuration.part.material.tensile_strength.value
66	new_configuration.part.material.tensile_strength.lower_limit
67	new_configuration.part.material.tensile_strength.upper_limit
68	new_configuration.part.material.recyclability.value
69	new_configuration.part.material.hardness.value
70	new_configuration.part.material.density.value

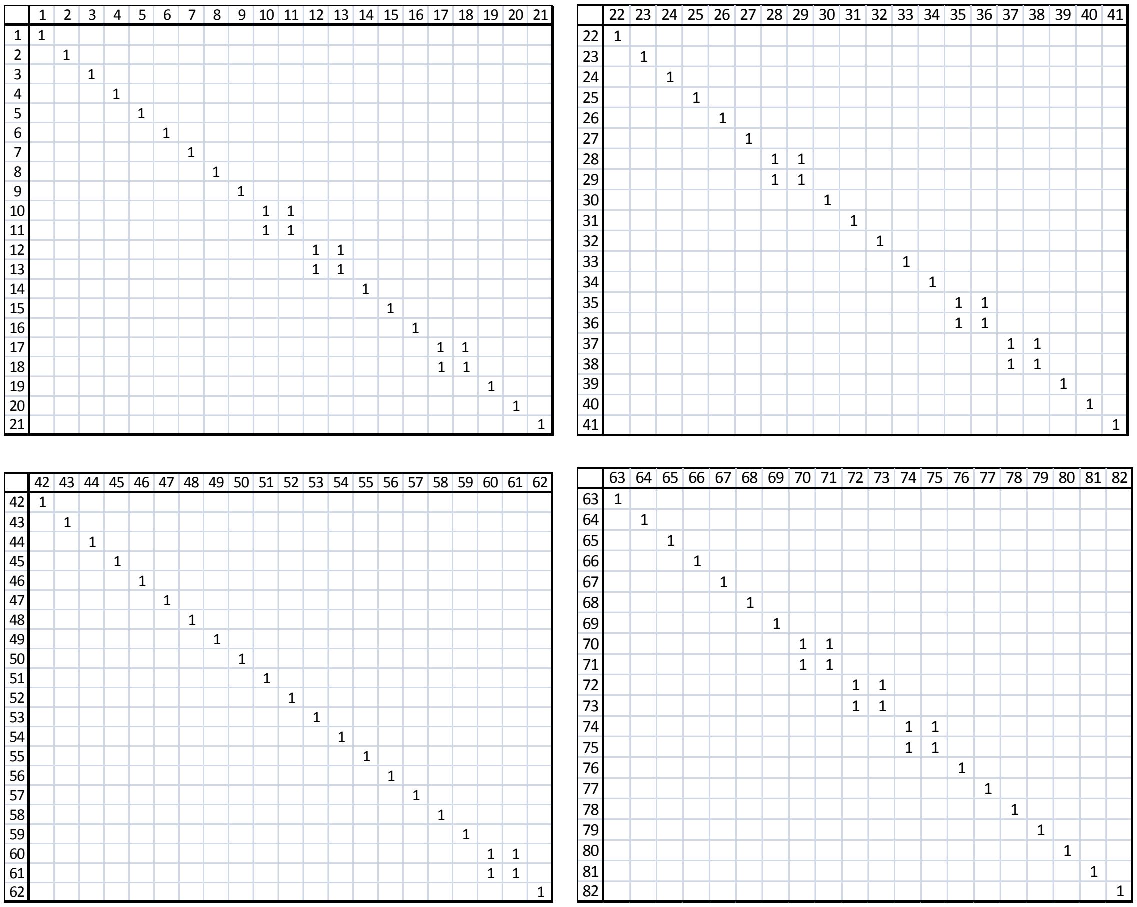
Table B.2: List of attributes 36 to 70 which are candidates for being important.

Integer id	Candidate attribute
71	new_configuration.part.material.density.unit
72	new_configuration.part.process.required_machine. min_positional_accuracy.value
73	new_configuration.part.process.required_machine. min_positional_accuracy.unit
74	new_configuration.part.process.required_machine. min_coolant_flow_rate.value
75	new_configuration.part.process.required_machine. min_coolant_flow_rate.unit
76	new_configuration.part.shape.features
77	new_configuration.assembly.name
78	new_configuration.assembly.process.name
79	new_configuration.assembly.process.tool.id
80	new_configuration.assembly.process.fixture.id
81	new_configuration.assembly.associated_joint_geometry_vector
82	new_configuration.assembly.production_batch_size

Table B.3: List of last 12 attributes that are candidates for being important.

B.3 Constraint matrices

The matrices t^{in} and t^{ex} capture the information about attributes that are mutually inclusive and exclusive, respectively. For case study and evaluation, each of these matrices is of size 82×82, since there are 82 candidate attributes. Figures B.1 and B.2 illustrates the non-zero portions of matrices t^{in} and t^{ex}, respectively, for the case study and evaluation. The first row and column indicates the attribute integer id. All the empty cells in the figures B.1 and B.2 have a value of zero.



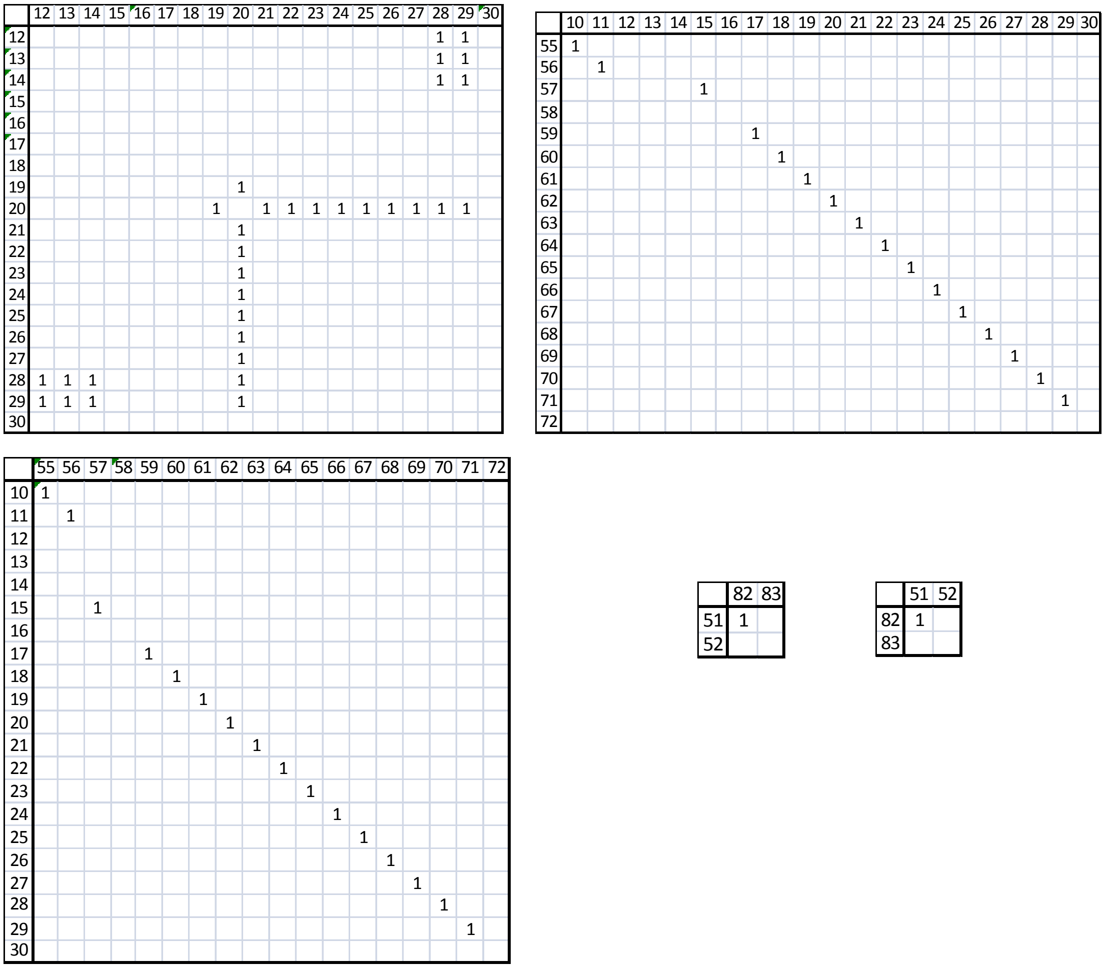
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	1																				
2		1																			
3			1																		
4				1																	
5					1																
6						1															
7							1														
8								1													
9									1												
10										1	1										
11										1	1										
12												1	1								
13												1	1								
14														1							
15															1						
16																1					
17																	1	1			
18																	1	1			
19																			1		
20																				1	
21																					1

	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
22	1																			
23		1																		
24			1																	
25				1																
26					1															
27						1														
28							1	1												
29							1	1												
30									1											
31										1										
32											1									
33												1								
34													1							
35														1	1					
36														1	1					
37																1	1			
38																1	1			
39																		1		
40																			1	
41																				1

	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62
42	1																				
43		1																			
44			1																		
45				1																	
46					1																
47						1															
48							1														
49								1													
50									1												
51										1											
52											1										
53												1									
54													1								
55														1							
56															1						
57																1					
58																	1				
59																		1			
60																			1	1	
61																			1	1	
62																					1

	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82
63	1																			
64		1																		
65			1																	
66				1																
67					1															
68						1														
69							1													
70								1	1											
71								1	1											
72										1	1									
73										1	1									
74												1	1							
75												1	1							
76														1						
77															1					
78																1				
79																	1			
80																		1		
81																			1	
82																				1

Figure B.1: Non-zero portions of matrix t^{in} for the case study and evaluation. The matrix t^{in} captures the information about attributes that are mutually inclusive. The first row and column indicates the attribute integer id. All the empty cells have zero values



	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12																	1	1	
13																	1	1	
14																	1	1	
15																			
16																			
17																			
18																			
19									1										
20								1		1	1	1	1	1	1	1	1	1	
21									1										
22									1										
23									1										
24									1										
25									1										
26									1										
27									1										
28	1	1	1						1										
29	1	1	1						1										
30																			

	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
55	1																				
56		1																			
57						1															
58																					
59								1													
60									1												
61										1											
62											1										
63												1									
64													1								
65														1							
66															1						
67																1					
68																	1				
69																		1			
70																			1		
71																				1	
72																					

	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
10	1																	
11		1																
12																		
13																		
14																		
15			1															
16																		
17					1													
18						1												
19							1											
20								1										
21									1									
22										1								
23											1							
24												1						
25													1					
26														1				
27															1			
28																1		
29																	1	
30																		

	82	83
51	1	
52		

	51	52
82	1	
83		

Figure B.2: Non-zero portions of matrix t^{ex} for the case study and evaluation. The matrix t^{ex} captures the information about attributes that are mutually exclusive. The first row and column of each matrix indicates the attribute integer id. All the empty cells have zero values

B.4 Important attributes for all datasets determined using subset evaluation criteria

Tables B.4 , . . . , B.13 summarize the important attributes for dataset # 1 , . . . , # 10, respectively, determined using our measures and the state-of-the-art evaluation criteria.

Approaches	*Important Attribute Set*
Decision tree classifier	old_configuration.part.process.tool.id
Naive Bayes classifier	old_configuration.part.process.tool.id
Information gain ratio	old_configuration.part.process.name
Our measure	old_configuration.part.process.tool.id, old_configuration.part.process.required_machine.type, new_configuration.part.shape.feature_existence, new_configuration.part.production_rate.unit_quantity, new_configuration.part.tolerance_range.upper_limit

Table B.4: Important attributes for dataset # 1 determined using our measures and the state-of-the-art evaluation criteria

Approaches	*Important Attribute Set*
Decision tree classifier	old_configuration.part.process.name
Naive Bayes classifier	old_configuration.part.process.name
Information gain ratio	new_configuration.part.surface_finish.value, new_configuration.part.tolerance_range.upper_limit
Our measure	old_configuration.part.process.required_machine.type, old_configuration.part.process.tool.id, new_configuration.part.production_rate.unit_quantity, new_configuration.part.tolerance_range.upper_limit

Table B.5: Important attributes for dataset # 2 determined using our measures and the state-of-the-art evaluation criteria

Approaches	*Important Attribute Set*
Decision tree classifier	old_configuration.part.process.name
Naive Bayes classifier	old_configuration.part.process.name, priority
Information gain ratio	old_configuration.part.process.name, old_configuration.part.process.required_machine.type, new_configuration.part.production_rate.unit_quantity
Our measure	old_configuration.part.process.required_machine.type, old_configuration.part.process.tool.id, new_configuration.part.production_rate.unit_quantity, new_configuration.part.surface_finish.value

Table B.6: Important attributes for dataset # 3 determined using our measures and the state-of-the-art evaluation criteria

Approaches	*Important Attribute Set*
Decision tree classifier	old_configuration.part.process.name
Naive Bayes classifier	old_configuration.part.process.name
Information gain ratio	old_configuration.part.process.name, old_configuration.part.process.required_machine.type, new_configuration.part.surface_finish.value, new_configuration.part.tolerance_range.upper_limit
Our measure	old_configuration.part.process.name, new_configuration.part.surface_finish.value, new_configuration.part.tolerance_range.upper_limit, new_configuration.part.tolerance_range.lower_limit

Table B.7: Important attributes for dataset # 4 determined using our measures and the state-of-the-art evaluation criteria

Approaches	*Important Attribute Set*
Decision tree classifier	old_configuration.part.process.name, old_configuration.part.shape.feature_existence
Naive Bayes classifier	old_configuration.part.process.name, old_configuration.part.shape.feature_existence, new_configuration.part.material.class
Information gain ratio	new_configuration.part.surface_finish.value, new_configuration.part.production_rate.unit_quantity, new_configuration.part.shape.feature_existence
Our measure	old_configuration.part.process.name, new_configuration.part.production_rate.unit_quantity, new_configuration.part.surface_finish.value, new_configuration.part.volume.value

Table B.8: Important attributes for dataset # 5 determined using our measures and the state-of-the-art evaluation criteria

Approaches	*Important Attribute Set*
Decision tree classifier	new_configuration.part.material.name
Naive Bayes classifier	new_configuration.part.material.name, reason_for_change
Information gain ratio	old_configuration.part.shape.features, new_configuration.part.material.class, requesting_department, new_configuration.part.material.recyclability.value
Our measure	old_configuration.part.process.name, old_configuration.part.process.required_machine.type, new_configuration.part.production_rate.unit_quantity, new_configuration.part.tolerance_range.upper_limit, new_configuration.part.tolerance_range.lower_limit

Table B.9: Important attributes for dataset # 6 determined using our measures and the state-of-the-art evaluation criteria

Approaches	*Important Attribute Set*
Decision tree classifier	reason_for_change, old_configuration.part.process.name, old_configuration.part.wall_thickness.value, old_configuration.part.process.max_feed_rate
Naive Bayes classifier	reason_for_change, old_configuration.part.process.name
Information gain ratio	new_configuration.part.shape.feature_existence, new_configuration.part.material.recyclability.value
Our measure	old_configuration.part.process.name, old_configuration.part.process.required_machine.type, new_configuration.part.production_rate.unit_quantity, new_configuration.part.tolerance_range.upper_limit, new_configuration.part.tolerance_range.lower_limit

Table B.10: Important attributes for dataset # 7 determined using our measures and the state-of-the-art evaluation criteria

Approaches	*Important Attribute Set*
Decision tree classifier	new_configuration.part.material.name
Naive Bayes classifier	reason_for_change, new_configuration.part. material.name
Information gain ratio	new_configuration.part. material.tensile_strength.upper_limit, new_configuration.part.material.hardness.value
Our measure	old_configuration.part.process.tool.id, old_configuration.part.surface_finish.value, new_configuration.part.production_rate.unit_quantity, new_configuration.part.tolerance_range.lower_limit

Table B.11: Important attributes for dataset # 8 determined using our measures and the state-of-the-art evaluation criteria

Approaches	*Important Attribute Set*
Decision tree classifier	reason_for_change, old_configuration.part.process.name, new_configuration.part. material.thermal_conductivity.value
Naive Bayes classifier	requesting_department, old_configuration.part. volume.value, old_configuration.part.mass.value, old_configuration.part.production_rate.time_per_unit, old_configuration.part.process.name
Information gain ratio	old_configuration.part.process.name, new_configuration.part.production_rate.unit_quantity, new_configuration.part.material.recyclability.value
Our measure	old_configuration.part.process.required_machine.type, old_configuration.part.process.tool.id, new_configuration.part.production_rate.unit_quantity, new_configuration.part.tolerance_range.lower_limit

Table B.12: Important attributes for dataset # 9 determined using our measures and the state-of-the-art evaluation criteria

Approach	*Important Attribute Set*
Decision tree classifier	requesting_department, new_configuration.part.tolerance_range.lower_limit
Naive Bayes classifier	priority, requesting_department, old_configuration.part.production_rate.time_per_unit, new_configuration.part.surface_finish.value
Information gain ratio	old_configuration.part.process.required_machine.type, new_configuration.part.surface_finish.value
Our measure	new_configuration.part.production_rate.unit_quantity, new_configuration.part.surface_finish.value, new_configuration.part.tolerance_range.upper_limit, new_configuration.part.tolerance_range.lower_limit

Table B.13: Important attributes for dataset # 10 determined using our measures and the state-of-the-art evaluation criteria

B.5 Important attributes for all datasets determined using overall approach

Tables B.14 , . . . , B.23 summarize the important attributes for dataset # 1 , . . . , # 10, respectively, determined using our overall approach.

Important Attribute Set
old_configuration.part.shape.feature_existence, old_configuration.part.process.name, old_configuration.part.process.tool.id, old_configuration.part.process.required_machine.type, new_configuration.part.shape.feature_existence, new_configuration.part.production_rate.unit_quantity, new_configuration.part.surface_finish.value, new_configuration.part.tolerance_range.upper_limit, new_configuration.part.tolerance_range.lower_limit

Table B.14: Important attribute set for dataset # 1 determined using our overall approach

Important Attribute Set
old_configuration.part.shape.feature_existence, old_configuration.part.process.name, old_configuration.part.process.tool.id, old_configuration.part.process.required_machine.type, new_configuration.part.shape.feature_existence, new_configuration.part.production_rate.unit_quantity, new_configuration.part.tolerance_range.upper_limit, new_configuration.part.tolerance_range.lower_limit

Table B.15: Important attribute set for dataset # 2 determined using our overall approach

Important Attribute Set
old_configuration.part.shape.feature_existence, old_configuration.part.process.name, old_configuration.part.process.tool.id, old_configuration.part.process.required_machine.type, new_configuration.part.shape.feature_existence, new_configuration.part.production_rate.unit_quantity, new_configuration.part.surface_finish.value

Table B.16: Important attribute set for dataset # 3 determined using our overall approach

Important Attribute Set
old_configuration.part.shape.feature_existence, old_configuration.part.process.name, new_configuration.part.shape.feature_existence, new_configuration.part.surface_finish.value, new_configuration.part.tolerance_range.upper_limit, new_configuration.part.tolerance_range.lower_limit

Table B.17: Important attribute set for dataset # 4 determined using our overall approach

Important Attribute Set
old_configuration.part.shape.feature_existence,
old_configuration.part.process.name,
new_configuration.part.shape.feature_existence,
new_configuration.part.production_rate.unit_quantity,
new_configuration.part.surface_finish.value,
new_configuration.part.volume.value

Table B.18: Important attribute set for dataset # 5 determined using our overall approach

Important Attribute Set
old_configuration.part.shape.feature_existence,
old_configuration.part.process.name,
old_configuration.part.process.required_machine.type,
new_configuration.part.shape.feature_existence,
new_configuration.part.production_rate.unit_quantity,
new_configuration.part.tolerance_range.upper_limit,
new_configuration.part.tolerance_range.lower_limit

Table B.19: Important attribute set for dataset # 6 determined using our overall approach

Important Attribute Set
old_configuration.part.shape.feature_existence,
old_configuration.part.process.name,
old_configuration.part.process.required_machine.type,
new_configuration.part.shape.feature_existence,
new_configuration.part.production_rate.unit_quantity,
new_configuration.part.tolerance_range.upper_limit,
new_configuration.part.tolerance_range.lower_limit

Table B.20: Important attribute set for dataset # 7 determined using our overall approach

Important Attribute Set
old_configuration.part.shape.feature_existence, old_configuration.part.process.name, old_configuration.part.process.tool.id, new_configuration.part.shape.feature_existence, new_configuration.part.production_rate.unit_quantity, new_configuration.part.surface_finish.value, new_configuration.part.tolerance_range.upper_limit, new_configuration.part.tolerance_range.lower_limit

Table B.21: Important attribute set for dataset # 8 determined using our overall approach

Important Attribute Set
old_configuration.part.shape.feature_existence, old_configuration.part.process.name, old_configuration.part.process.required_machine.type, new_configuration.part.shape.feature_existence, new_configuration.part.production_rate.unit_quantity, new_configuration.part.tolerance_range.upper_limit, new_configuration.part.tolerance_range.lower_limit

Table B.22: Important attribute set for dataset # 9 determined using our overall approach

Important Attribute Set
old_configuration.part.shape.feature_existence, old_configuration.part.process.name, new_configuration.part.shape.feature_existence, new_configuration.part.production_rate.unit_quantity, new_configuration.part.surface_finish.value, new_configuration.part.tolerance_range.upper_limit, new_configuration.part.tolerance_range.lower_limit

Table B.23: Important attribute set for dataset # 10 determined using our overall approach

APPENDIX C

Evaluation results for the problem of computing similarity

C.1 IS-A type taxonomies

Figure C.1 illustrates an example IS-A type taxonomy that relates the primitives values in the domain of categorical aggregate attribute - *old_ configuration.part. process.tool.id*. This taxonomy is used for computing similarity between values of attribute

old_ configuration.part.process.tool.id in various datasets.

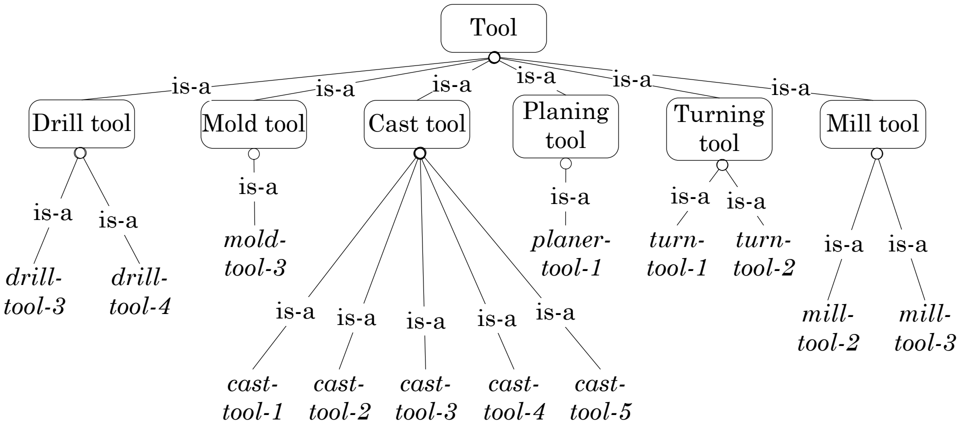


Figure C.1: Example taxonomy of IS-A type for the primitive values in the domain of categorical aggregate attribute *old_ configuration.part.process.tool.id*

C.2 Similarity results for all test datasets

Tables C.1 , . . . , C.10 summarize the sorted list of unique past ECs that are similar to proposed ECs in test dataset # 1 , . . . , # 10, respectively, determined using our approach, metric space approach, probability-based approach and a statistical approach. Each list is sorted in order of decreasing similarity value.

C.3 Similar changes identified based on manual observation

Tables C.11 , . . . , C.20 present two unique past ECs that are identified, based on manual observation, as most similar to various proposed ECs in test datasets $1, \ldots, 10$, respectively. This information is useful in computing the MAP values using various approaches.

Proposed EC	Two most similar unique training change instances
$EC-4$	$EC-1$, $EC-2$
$EC-7$	$EC-11$, $EC-16$
$EC-8$	$EC-6$, $EC-12$
$EC-9$	$EC-11$, $EC-15$
$EC-10$	$EC-13$, $EC-16$
$EC-17$	$EC-5$, $EC-14$

Table C.11: Two unique past ECs from training datasets # 1 which are most similar to various proposed ECs in test dataset # 1

Proposed EC	Two most similar unique training change instances
$EC-1$	$EC-3$, $EC-4$
$EC-2$	$EC-4$, $EC-3$
$EC-7$	$EC-16$, $EC-6$
$EC-9$	$EC-12$, $EC-15$
$EC-10$	$EC-13$, $EC-16$
$EC-11$	$EC-12$, $EC-15$

Table C.12: Two unique past ECs from training datasets # 2 which are most similar to various proposed ECs in test dataset # 2

Proposed EC	Two most similar unique training change instances
$EC-1$	$EC-3$, $EC-5$
$EC-4$	$EC-2$, $EC-3$
$EC-7$	$EC-16$, $EC-6$
$EC-11$	$EC-12$, $EC-9$
$EC-13$	$EC-10$, $EC-16$
$EC-15$	$EC-9$, $EC-12$

Table C.13: Two unique past ECs from training datasets # 3 which are most similar to various proposed ECs in test dataset # 3

Proposed EC	Two most similar unique training change instances
$EC-2$	$EC-1$, $EC-3$
$EC-4$	$EC-1$, $EC-3$
$EC-7$	$EC-6$, $EC-10$
$EC-11$	$EC-12$, $EC-15$
$EC-13$	$EC-10$, $EC-12$
$EC-16$	$EC-9$, $EC-10$

Table C.14: Two unique past ECs from training datasets # 4 which are most similar to various proposed ECs in test dataset # 4

Proposed EC	Two most similar unique training change instances
$EC-3$	$EC-1$, $EC-4$
$EC-5$	$EC-4$, $EC-2$
$EC-7$	$EC-16$, $EC-6$
$EC-9$	$EC-12$, $EC-15$
$EC-10$	$EC-13$, $EC-16$
$EC-11$	$EC-12$, $EC-15$

Table C.15: Two unique past ECs from training datasets # 5 which are most similar to various proposed ECs in test dataset # 5

Proposed EC	Two most similar unique training change instances
$EC-2$	$EC-1$, $EC-4$
$EC-3$	$EC-1$, $EC-4$
$EC-8$	$EC-6$, $EC-7$
$EC-11$	$EC-7$, $EC-9$
$EC-12$	$EC-13$, $EC-6$
$EC-15$	$EC-9$, $EC-10$

Table C.16: Two unique past ECs from training datasets # 6 which are most similar to various proposed ECs in test dataset # 6

Proposed EC	Two most similar unique training change instances
$EC-1$	$EC-3$, $EC-4$
$EC-2$	$EC-3$, $EC-4$
$EC-8$	$EC-6$, $EC-7$
$EC-10$	$EC-16$, $EC-13$
$EC-11$	$EC-12$, $EC-9$
$EC-15$	$EC-9$, $EC-12$

Table C.17: Two unique past ECs from training datasets # 7 which are most similar to various proposed ECs in test dataset # 7

Proposed EC	Two most similar unique training change instances
$EC-3$	$EC-1$, $EC-5$
$EC-4$	$EC-1$, $EC-2$
$EC-8$	$EC-6$, $EC-7$
$EC-12$	$EC-6$, $EC-9$
$EC-13$	$EC-10$, $EC-7$
$EC-15$	$EC-9$, $EC-11$

Table C.18: Two unique past ECs from training datasets # 8 which are most similar to various proposed ECs in test dataset # 8

Proposed EC	Two most similar unique training change instances
$EC-1$	$EC-3$, $EC-4$
$EC-5$	$EC-4$, $EC-3$
$EC-7$	$EC-11$, $EC-9$
$EC-10$	$EC-9$, $EC-13$
$EC-15$	$EC-9$, $EC-11$
$EC-16$	$EC-9$, $EC-12$

Table C.19: Two unique past ECs from training datasets # 9 which are most similar to various proposed ECs in test dataset # 9

Proposed EC	Two most similar unique training change instances
$EC-5$	$EC-3$, $EC-4$
$EC-7$	$EC-9$, $EC-16$
$EC-10$	$EC-13$, $EC-9$
$EC-11$	$EC-9$, $EC-13$
$EC-12$	$EC-13$, $EC-6$
$EC-15$	$EC-9$, $EC-13$

Table C.20: Two unique past ECs from training datasets # 10 which are most similar to various proposed ECs in test dataset # 10

C.4 MAP for all datasets

For all 10 datasets, Tables C.21 and C.22 summarize the MAP in retrieving changes using four different approaches which are similar to instances in test dataset and training dataset, respectively.

Test dataset #	Our approach	Metric space approach	Probability based approach	Statistical approach
1	0.67	0.5	0.41	0.2
2	0.73	0.56	0.54	0.15
3	0.81	0.53	0.66	0.22
4	0.53	0.36	0.46	0.13
5	0.49	0.34	0.31	0.11
6	0.59	0.48	0.49	0.20
7	0.62	0.54	0.61	0.13
8	0.81	0.54	0.61	0.17
9	0.7	0.51	0.38	0.17
10	0.66	0.4	0.39	0.14

Table C.21: MAP in retrieving past ECs, which are similar to proposed ECs in various test datasets, determined using various approaches to compute similarity between ECs

Training dataset #	Our approach	Metric space approach	Probability based approach	Statistical approach
1	0.85	0.62	0.65	0.86
2	0.87	0.77	0.71	0.85
3	0.89	0.7	0.69	0.78
4	0.78	0.86	0.78	0.83
5	0.87	0.57	0.72	0.77
6	0.73	0.56	0.62	0.65
7	0.69	0.58	0.59	0.69
8	0.76	0.57	0.54	0.72
9	0.87	0.68	0.75	0.84
10	0.81	0.59	0.54	0.74

Table C.22: MAP in retrieving past ECs, which are similar to changes in various training datasets, based on various approaches to compute similarity between ECs

C.5 Success rate for all datasets

For all 10 datasets, Tables C.23 and C.24 summarize the success rate in predicting the impact of effect of instances in test dataset and training dataset, respectively, using four different approaches.

Test dataset #	Our approach	Metric space approach	Probability based approach	Statistical approach
1	66.67	50.0	33.33	50.0
2	66.67	66.67	33.33	66.67
3	66.67	66.67	66.67	66.67
4	66.67	33.33	50.0	33.33
5	50.0	33.33	50.0	50.0
6	66.67	33.33	50.0	50.0
7	66.67	66.67	33.33	50.0
8	66.67	33.33	66.67	50.0
9	83.33	66.67	33.33	66.67
10	66.67	16.67	33.33	50.0

Table C.23: Success rate in predicting the impact of effect of instances in test datasets based on various approaches to compute similarity between ECs

Training dataset #	Our approach	Metric space approach	Probability based approach	Statistical approach
1	100	52.94	52.94	100
2	94.12	88.24	58.82	94.12
3	94.12	58.82	68.75	94.12
4	94.12	88.24	64.71	100.0
5	100.0	52.94	76.47	88.24
6	82.35	47.06	52.94	82.35
7	88.24	35.29	41.18	88.24
8	88.24	70.59	58.82	94.12
9	88.24	58.82	76.47	100.0
10	82.35	70.59	58.82	70.59

Table C.24: Success rate in predicting the impact of effect of instances in training datasets based on various approaches to compute similarity between ECs

Proposed EC	Our approach	Metric space approach	Probability-based approach	Statistical approach
EC-4	EC-2, EC-1, EC-3, EC-15, EC-12, EC-14, EC-16, EC-11, EC-5, EC-13, EC-6	EC-2, EC-11, EC-15, EC-1, EC-12, EC-14, EC-13, EC-6, EC-3, EC-16, EC-5	EC-2, EC-1, EC-12, EC-11, EC-15, EC-14, EC-3, EC-6, EC-13, EC-16, EC-5	EC-2, EC-3, EC-1, EC-14, EC-11, EC-15, EC-12, EC-6, EC-13, EC-5, EC-16
EC-7	EC-3, EC-1, EC-14, EC-5, EC-2, EC-11, EC-16, EC-12, EC-6, EC-13, EC-15	EC-3, EC-14, EC-16, EC-5, EC-6, EC-1, EC-12, EC-2, EC-11, EC-13, EC-15	EC-3, EC-1, EC-2, EC-13, EC-16, EC-12, EC-14, EC-11, EC-5, EC-15, EC-6	EC-14, EC-1, EC-3, EC-2, EC-5, EC-11, EC-15, EC-6, EC-16, EC-12, EC-13
EC-8	EC-6, EC-12, EC-14, EC-13, EC-16, EC-11, EC-5, EC-15, EC-2, EC-3, EC-1	EC-6, EC-14, EC-3, EC-1, EC-11, EC-5, EC-12, EC-16, EC-15, EC-2, EC-13	EC-6, EC-14, EC-1, EC-11, EC-12, EC-5, EC-15, EC-2, EC-3, EC-16, EC-13	EC-6, EC-16, EC-13, EC-12, EC-11, EC-15, EC-2, EC-5, EC-1, EC-14, EC-3
EC-9	EC-15, EC-2, EC-12, EC-11, EC-3, EC-16, EC-1, EC-13, EC-6, EC-14, EC-5	EC-2, EC-12, EC-11, EC-15, EC-1, EC-14, EC-6, EC-3, EC-13, EC-16, EC-5	EC-2, EC-11, EC-15, EC-1, EC-12, EC-13, EC-14, EC-6, EC-3, EC-16, EC-5	EC-15, EC-1, EC-2, EC-13, EC-12, EC-6, EC-16, EC-3, EC-11, EC-14, EC-5
EC-10	EC-13, EC-16, EC-12, EC-6, EC-11, EC-2, EC-15, EC-14, EC-3, EC-1, EC-5	EC-13, EC-12, EC-2, EC-11, EC-15, EC-1, EC-14, EC-3, EC-16, EC-5, EC-6	EC-3, EC-2, EC-16, EC-14, EC-1, EC-13, EC-12, EC-11, EC-15, EC-5, EC-6	EC-13, EC-12, EC-6, EC-16, EC-15, EC-11, EC-2, EC-5, EC-3, EC-1, EC-14
EC-17	EC-16, EC-12, EC-13, EC-6, EC-11, EC-15, EC-5, EC-14, EC-2, EC-1, EC-3	EC-16, EC-14, EC-5, EC-6, EC-3, EC-12, EC-1, EC-11, EC-13, EC-2, EC-15	EC-16, EC-3, EC-5, EC-14, EC-13, EC-2, EC-1, EC-11, EC-12, EC-6, EC-15	EC-16, EC-6, EC-13, EC-12, EC-2, EC-5, EC-11, EC-15, EC-1, EC-14, EC-3

Table C.1: List of unique past ECs, which are similar to proposed ECs from dataset # 1, sorted in order of decreasing similarity value determined using various approaches

Proposed EC	Our approach	Metric space approach	Probability-based approach	Statistical approach
EC-1	EC-3, EC-14, EC-4, EC-5, EC-17, EC-6, EC-8, EC-16, EC-12, EC-15, EC-13	EC-4, EC-14, EC-12, EC-15, EC-6, EC-8, EC-5, EC-13, EC-16, EC-17, EC-3	EC-4, EC-12, EC-14, EC-3, EC-15, EC-6, EC-8, EC-13, EC-5, EC-16, EC-17	EC-3, EC-14, EC-5, EC-17, EC-4, EC-16, EC-15, EC-12, EC-6, EC-8, EC-13
EC-2	EC-4, EC-15, EC-12, EC-3, EC-13, EC-14, EC-17, EC-5, EC-6, EC-8, EC-16	EC-15, EC-4, EC-12, EC-13, EC-14, EC-3, EC-6, EC-8, EC-5, EC-16, EC-17	EC-12, EC-4, EC-15, EC-14, EC-13, EC-6, EC-8, EC-3, EC-5, EC-16, EC-17	EC-16, EC-4, EC-12, EC-17, EC-15, EC-5, EC-14, EC-13, EC-6, EC-8, EC-3
EC-7	EC-3, EC-14, EC-6, EC-5, EC-16, EC-17, EC-4, EC-8, EC-13, EC-12, EC-15	EC-14, EC-3, EC-5, EC-6, EC-8, EC-16, EC-17, EC-4, EC-12, EC-13, EC-15	EC-3, EC-14, EC-5, EC-13, EC-4, EC-6, EC-8, EC-12, EC-16, EC-17, EC-15	EC-3, EC-14, EC-4, EC-5, EC-17, EC-16, EC-15, EC-6, EC-8, EC-12, EC-13
EC-9	EC-15, EC-12, EC-4, EC-13, EC-6, EC-8, EC-14, EC-16, EC-3, EC-17, EC-5	EC-15, EC-4, EC-12, EC-13, EC-14, EC-3, EC-6, EC-8, EC-5, EC-16, EC-17	EC-12, EC-4, EC-15, EC-14, EC-13, EC-6, EC-8, EC-3, EC-5, EC-16, EC-17	EC-16, EC-12, EC-15, EC-13, EC-6, EC-8, EC-17, EC-4, EC-5, EC-14, EC-3
EC-10	EC-13, EC-12, EC-16, EC-6, EC-8, EC-15, EC-17, EC-4, EC-3, EC-14, EC-5	EC-13, EC-12, EC-15, EC-4, EC-3, EC-14, EC-16, EC-17, EC-5, EC-6, EC-8	EC-3, EC-13, EC-14, EC-16, EC-17, EC-4, EC-15, EC-12, EC-5, EC-6, EC-8	EC-12, EC-13, EC-15, EC-6, EC-8, EC-16, EC-17, EC-5, EC-4, EC-14, EC-3
EC-11	EC-4, EC-15, EC-12, EC-6, EC-8, EC-14, EC-16, EC-13, EC-3, EC-17, EC-5	EC-4, EC-15, EC-14, EC-13, EC-12, EC-6, EC-8, EC-3, EC-5, EC-16, EC-17	EC-4, EC-12, EC-15, EC-14, EC-6, EC-8, EC-3, EC-13, EC-5, EC-16, EC-17	EC-6, EC-8, EC-16, EC-15, EC-12, EC-13, EC-4, EC-17, EC-14, EC-5, EC-3

Table C.2: List of unique past ECs, which are similar to proposed ECs from dataset # 2, sorted in order of decreasing similarity value determined using various approaches

Proposed EC	Our approach	Metric space approach	Probability-based approach	Statistical approach
EC-1	EC-3, EC-5, EC-14, EC-2, EC-9, EC-17, EC-12, EC-16, EC-6, EC-8, EC-10	EC-12, EC-14, EC-2, EC-9, EC-10, EC-5, EC-16, EC-17, EC-6, EC-8, EC-3	EC-3, EC-2, EC-9, EC-14, EC-12, EC-5, EC-6, EC-8, EC-10, EC-16, EC-17	EC-5, EC-3, EC-14, EC-2, EC-17, EC-9, EC-16, EC-12, EC-10, EC-6, EC-8
EC-4	EC-2, EC-9, EC-12, EC-14, EC-3, EC-17, EC-5, EC-16, EC-10, EC-6, EC-8	EC-2, EC-9, EC-12, EC-10, EC-14, EC-3, EC-5, EC-16, EC-17, EC-6, EC-8	EC-2, EC-9, EC-3, EC-12, EC-14, EC-5, EC-6, EC-8, EC-10, EC-16, EC-17	EC-9, EC-2, EC-12, EC-3, EC-5, EC-14, EC-16, EC-6, EC-8, EC-17, EC-10
EC-7	EC-3, EC-14, EC-16, EC-5, EC-17, EC-9, EC-6, EC-12, EC-10, EC-2, EC-8	EC-14, EC-3, EC-5, EC-16, EC-17, EC-6, EC-8, EC-12, EC-2, EC-9, EC-10	EC-3, EC-14, EC-2, EC-9, EC-5, EC-12, EC-10, EC-16, EC-17, EC-6, EC-8	EC-3, EC-14, EC-5, EC-2, EC-17, EC-9, EC-16, EC-12, EC-6, EC-8, EC-10
EC-11	EC-9, EC-12, EC-2, EC-16, EC-14, EC-6, EC-8, EC-10, EC-3, EC-17, EC-5	EC-2, EC-9, EC-12, EC-10, EC-14, EC-5, EC-3, EC-16, EC-17, EC-6, EC-8	EC-12, EC-2, EC-9, EC-14, EC-5, EC-6, EC-8, EC-3, EC-16, EC-17, EC-10	EC-16, EC-9, EC-6, EC-8, EC-12, EC-10, EC-17, EC-14, EC-2, EC-3, EC-5
EC-13	EC-10, EC-12, EC-16, EC-8, EC-9, EC-6, EC-2, EC-17, EC-14, EC-5, EC-3	EC-10, EC-12, EC-2, EC-9, EC-14, EC-3, EC-5, EC-16, EC-17, EC-6, EC-8	EC-12, EC-2, EC-9, EC-10, EC-5, EC-3, EC-14, EC-16, EC-17, EC-6, EC-8	EC-10, EC-6, EC-8, EC-12, EC-16, EC-9, EC-17, EC-2, EC-14, EC-5, EC-3
EC-15	EC-9, EC-12, EC-2, EC-10, EC-16, EC-6, EC-8, EC-5, EC-3, EC-14, EC-17	EC-9, EC-2, EC-12, EC-10, EC-14, EC-3, EC-5, EC-6, EC-8, EC-16, EC-17	EC-12, EC-9, EC-2, EC-5, EC-6, EC-8, EC-14, EC-3, EC-10, EC-16, EC-17	EC-9, EC-16, EC-12, EC-10, EC-6, EC-8, EC-2, EC-5, EC-17, EC-3, EC-14

Table C.3: List of unique past ECs, which are similar to proposed ECs from dataset # 3, sorted in order of decreasing similarity value determined using various approaches

Proposed EC	Our approach	Metric space approach	Probability-based approach	Statistical approach
EC-2	EC-9, EC-1, EC-3, EC-15, EC-6, EC-8, EC-10, EC-14, EC-12, EC-5, EC-17	EC-9, EC-12, EC-1, EC-15, EC-10, EC-14, EC-6, EC-8, EC-3, EC-5, EC-17	EC-9, EC-1, EC-12, EC-15, EC-14, EC-6, EC-8, EC-3, EC-10, EC-5, EC-17	EC-9, EC-12, EC-6, EC-8, EC-10, EC-15, EC-14, EC-1, EC-3, EC-17, EC-5
EC-4	EC-9, EC-1, EC-3, EC-15, EC-6, EC-8, EC-10, EC-14, EC-12, EC-5, EC-17	EC-9, EC-12, EC-1, EC-15, EC-10, EC-14, EC-6, EC-8, EC-3, EC-5, EC-17	EC-9, EC-1, EC-12, EC-15, EC-14, EC-6, EC-8, EC-3, EC-10, EC-5, EC-17	EC-9, EC-12, EC-6, EC-8, EC-10, EC-15, EC-14, EC-1, EC-3, EC-17, EC-5
EC-7	EC-3, EC-14, EC-1, EC-10, EC-5, EC-17, EC-9, EC-6, EC-8, EC-12, EC-15	EC-3, EC-5, EC-17, EC-14, EC-6, EC-8, EC-10, EC-1, EC-12, EC-9, EC-15	EC-3, EC-10, EC-5, EC-14, EC-1, EC-9, EC-12, EC-17, EC-6, EC-8, EC-15	EC-3, EC-1, EC-14, EC-5, EC-17, EC-15, EC-9, EC-10, EC-12, EC-6, EC-8
EC-11	EC-9, EC-12, EC-15, EC-1, EC-6, EC-8, EC-10, EC-14, EC-3, EC-5, EC-17	EC-9, EC-15, EC-12, EC-1, EC-10, EC-14, EC-6, EC-8, EC-3, EC-5, EC-17	EC-12, EC-9, EC-15, EC-1, EC-6, EC-8, EC-14, EC-3, EC-5, EC-10, EC-17	EC-12, EC-6, EC-8, EC-9, EC-10, EC-15, EC-14, EC-1, EC-3, EC-5, EC-17
EC-13	EC-10, EC-6, EC-8, EC-3, EC-12, EC-9, EC-5, EC-15, EC-17, EC-14, EC-1	EC-10, EC-9, EC-15, EC-12, EC-1, EC-14, EC-3, EC-5, EC-17, EC-6, EC-8	EC-3, EC-5, EC-10, EC-12, EC-17, EC-14, EC-9, EC-6, EC-8, EC-1, EC-15	EC-10, EC-6, EC-8, EC-12, EC-9, EC-15, EC-14, EC-3, EC-1, EC-17, EC-5
EC-16	EC-17, EC-9, EC-3, EC-10, EC-14, EC-1, EC-5, EC-6, EC-8, EC-12, EC-15	EC-17, EC-5, EC-3, EC-6, EC-8, EC-14, EC-10, EC-12, EC-1, EC-9, EC-15	EC-17, EC-10, EC-5, EC-3, EC-14, EC-12, EC-9, EC-1, EC-6, EC-8, EC-15	EC-17, EC-5, EC-14, EC-3, EC-1, EC-10, EC-9, EC-6, EC-8, EC-12, EC-15

Table C.4: List of unique past ECs, which are similar to proposed ECs from dataset # 4, sorted in order of decreasing similarity value determined using various approaches

Proposed EC	Our approach	Metric space approach	Probability-based approach	Statistical approach
EC-3	EC-14, EC-1, EC-4, EC-2, EC-17, EC-12, EC-16, EC-13, EC-8, EC-6, EC-15	EC-14, EC-6, EC-16, EC-17, EC-8, EC-12, EC-4, EC-1, EC-13, EC-2, EC-15	EC-2, EC-4, EC-12, EC-14, EC-13, EC-1, EC-15, EC-16, EC-17, EC-6, EC-8	EC-14, EC-4, EC-1, EC-2, EC-17, EC-16, EC-12, EC-8, EC-6, EC-15, EC-13
EC-5	EC-8, EC-6, EC-13, EC-17, EC-2, EC-16, EC-15, EC-4, EC-1, EC-12, EC-14	EC-8, EC-17, EC-14, EC-6, EC-16, EC-1, EC-2, EC-4, EC-13, EC-12, EC-15	EC-1, EC-8, EC-13, EC-4, EC-2, EC-15, EC-12, EC-17, EC-6, EC-14, EC-16	EC-15, EC-8, EC-6, EC-13, EC-17, EC-16, EC-12, EC-2, EC-4, EC-14, EC-1
EC-7	EC-14, EC-1, EC-4, EC-2, EC-17, EC-16, EC-12, EC-13, EC-8, EC-6, EC-15	EC-14, EC-6, EC-16, EC-17, EC-8, EC-1, EC-4, EC-12, EC-13, EC-2, EC-15	EC-1, EC-4, EC-14, EC-12, EC-13, EC-2, EC-16, EC-17, EC-15, EC-6, EC-8	EC-14, EC-4, EC-1, EC-2, EC-17, EC-16, EC-12, EC-8, EC-6, EC-15, EC-13
EC-9	EC-2, EC-4, EC-1, EC-14, EC-15, EC-12, EC-13, EC-17, EC-16, EC-8, EC-6	EC-2, EC-13, EC-12, EC-4, EC-15, EC-1, EC-14, EC-8, EC-17, EC-6, EC-16	EC-2, EC-13, EC-12, EC-4, EC-1, EC-8, EC-14, EC-15, EC-6, EC-17, EC-16	EC-2, EC-1, EC-4, EC-17, EC-16, EC-14, EC-12, EC-15, EC-13, EC-8, EC-6
EC-10	EC-13, EC-12, EC-17, EC-2, EC-15, EC-8, EC-16, EC-6, EC-4, EC-14, EC-1	EC-2, EC-13, EC-4, EC-12, EC-15, EC-1, EC-14, EC-17, EC-16, EC-8, EC-6	EC-2, EC-17, EC-13, EC-4, EC-1, EC-16, EC-14, EC-15, EC-12, EC-8, EC-6	EC-12, EC-13, EC-8, EC-6, EC-15, EC-17, EC-16, EC-2, EC-4, EC-1, EC-14
EC-11	EC-12, EC-15, EC-4, EC-13, EC-8, EC-6, EC-14, EC-2, EC-17, EC-16, EC-1	EC-1, EC-4, EC-12, EC-13, EC-15, EC-2, EC-14, EC-6, EC-16, EC-17, EC-8	EC-4, EC-1, EC-14, EC-12, EC-15, EC-6, EC-13, EC-2, EC-8, EC-16, EC-17	EC-17, EC-16, EC-12, EC-8, EC-15, EC-6, EC-13, EC-14, EC-4, EC-1, EC-2

Table C.5: List of unique past ECs, which are similar to proposed ECs from dataset # 5, sorted in order of decreasing similarity value determined using various approaches

Proposed EC	Our approach	Metric space approach	Probability-based approach	Statistical approach
EC-2	EC-9, EC-4, EC-1, EC-10, EC-13, EC-14, EC-6, EC-7, EC-16, EC-17, EC-5	EC-9, EC-4, EC-13, EC-10, EC-1, EC-14, EC-7, EC-6, EC-5, EC-16, EC-17	EC-9, EC-4, EC-1, EC-14, EC-13, EC-10, EC-7, EC-6, EC-5, EC-16, EC-17	EC-9, EC-1, EC-4, EC-6, EC-10, EC-13, EC-16, EC-17, EC-14, EC-5, EC-7
EC-3	EC-7, EC-1, EC-13, EC-4, EC-6, EC-14, EC-10, EC-16, EC-17, EC-9, EC-5	EC-7, EC-14, EC-5, EC-16, EC-17, EC-6, EC-10, EC-13, EC-1, EC-4, EC-9	EC-7, EC-14, EC-4, EC-1, EC-13, EC-10, EC-5, EC-9, EC-16, EC-17, EC-6	EC-7, EC-6, EC-16, EC-17, EC-4, EC-13, EC-10, EC-9, EC-14, EC-1, EC-5
EC-8	EC-6, EC-14, EC-13, EC-7, EC-10, EC-1, EC-16, EC-17, EC-5, EC-4, EC-9	EC-6, EC-14, EC-7, EC-5, EC-16, EC-17, EC-1, EC-4, EC-9, EC-13, EC-10	EC-6, EC-14, EC-5, EC-4, EC-9, EC-1, EC-7, EC-13, EC-16, EC-17, EC-10	EC-6, EC-13, EC-10, EC-16, EC-17, EC-7, EC-9, EC-4, EC-1, EC-14, EC-5
EC-11	EC-4, EC-9, EC-1, EC-14, EC-7, EC-6, EC-16, EC-17, EC-13, EC-10, EC-5	EC-4, EC-1, EC-9, EC-13, EC-10, EC-14, EC-6, EC-7, EC-5, EC-16, EC-17	EC-4, EC-1, EC-9, EC-14, EC-6, EC-7, EC-13, EC-5, EC-10, EC-16, EC-17	EC-4, EC-14, EC-7, EC-9, EC-1, EC-6, EC-16, EC-17, EC-5, EC-10, EC-13
EC-12	EC-9, EC-6, EC-4, EC-13, EC-10, EC-1, EC-7, EC-14, EC-16, EC-17, EC-5	EC-9, EC-4, EC-13, EC-10, EC-1, EC-14, EC-7, EC-6, EC-5, EC-16, EC-17	EC-9, EC-4, EC-1, EC-13, EC-14, EC-6, EC-10, EC-7, EC-5, EC-16, EC-17	EC-6, EC-10, EC-13, EC-9, EC-16, EC-17, EC-7, EC-4, EC-1, EC-5, EC-14
EC-15	EC-9, EC-4, EC-1, EC-10, EC-13, EC-14, EC-6, EC-7, EC-5, EC-16, EC-17	EC-9, EC-4, EC-13, EC-10, EC-1, EC-14, EC-7, EC-6, EC-5, EC-16, EC-17	EC-9, EC-4, EC-1, EC-13, EC-6, EC-14, EC-10, EC-5, EC-7, EC-16, EC-17	EC-9, EC-1, EC-4, EC-10, EC-6, EC-5, EC-13, EC-16, EC-17, EC-14, EC-7

Table C.6: List of unique past ECs, which are similar to proposed ECs from dataset # 6, sorted in order of decreasing similarity value determined using various approaches

Proposed EC	Our approach	Metric space approach	Probability-based approach	Statistical approach
EC-1	EC-4, EC-7, EC-5, EC-3, EC-14, EC-16, EC-17, EC-6, EC-9, EC-12, EC-13	EC-14, EC-6, EC-5, EC-16, EC-17, EC-4, EC-7, EC-3, EC-9, EC-12, EC-13	EC-4, EC-9, EC-14, EC-7, EC-3, EC-12, EC-6, EC-13, EC-5, EC-16, EC-17	EC-14, EC-5, EC-16, EC-17, EC-7, EC-3, EC-4, EC-9, EC-6, EC-12, EC-13
EC-2	EC-9, EC-12, EC-4, EC-13, EC-7, EC-14, EC-3, EC-16, EC-17, EC-5, EC-6	EC-9, EC-12, EC-13, EC-4, EC-3, EC-14, EC-7, EC-6, EC-5, EC-16, EC-17	EC-9, EC-12, EC-4, EC-14, EC-13, EC-3, EC-7, EC-6, EC-5, EC-16, EC-17	EC-9, EC-7, EC-16, EC-17, EC-12, EC-14, EC-13, EC-6, EC-5, EC-4, EC-3
EC-8	EC-6, EC-16, EC-17, EC-5, EC-7, EC-14, EC-12, EC-13, EC-3, EC-4, EC-9	EC-6, EC-5, EC-16, EC-17, EC-14, EC-7, EC-3, EC-4, EC-12, EC-9, EC-13	EC-6, EC-14, EC-5, EC-4, EC-12, EC-9, EC-7, EC-3, EC-13, EC-16, EC-17	EC-6, EC-13, EC-12, EC-16, EC-17, EC-7, EC-4, EC-9, EC-3, EC-14, EC-5
EC-10	EC-13, EC-12, EC-9, EC-7, EC-3, EC-6, EC-14, EC-4, EC-16, EC-17, EC-5	EC-13, EC-9, EC-12, EC-4, EC-3, EC-14, EC-7, EC-16, EC-17, EC-5, EC-6	EC-3, EC-13, EC-9, EC-7, EC-14, EC-17, EC-16, EC-4, EC-12, EC-5, EC-6	EC-12, EC-13, EC-9, EC-6, EC-7, EC-16, EC-17, EC-14, EC-5, EC-3, EC-4
EC-11	EC-4, EC-7, EC-9, EC-14, EC-12, EC-16, EC-17, EC-3, EC-5, EC-6, EC-13	EC-4, EC-14, EC-9, EC-12, EC-13, EC-7, EC-6, EC-3, EC-5, EC-16, EC-17	EC-4, EC-12, EC-9, EC-14, EC-6, EC-7, EC-3, EC-13, EC-5, EC-16, EC-17	EC-4, EC-16, EC-17, EC-9, EC-14, EC-12, EC-6, EC-13, EC-5, EC-7, EC-3
EC-15	EC-9, EC-12, EC-4, EC-13, EC-7, EC-14, EC-3, EC-5, EC-16, EC-17, EC-6	EC-9, EC-12, EC-13, EC-4, EC-3, EC-14, EC-7, EC-6, EC-5, EC-16, EC-17	EC-12, EC-9, EC-4, EC-13, EC-6, EC-14, EC-3, EC-7, EC-5, EC-16, EC-17	EC-17, EC-5, EC-12, EC-16, EC-9, EC-7, EC-13, EC-14, EC-6, EC-3, EC-4

Table C.7: List of unique past ECs, which are similar to proposed ECs from dataset # 7, sorted in order of decreasing similarity value determined using various approaches

Proposed EC	Our approach	Metric space approach	Probability-based approach	Statistical approach
EC-3	EC-7, EC-10, EC-5, EC-1, EC-14, EC-9, EC-16, EC-2, EC-17, EC-11, EC-6	EC-7, EC-14, EC-5, EC-16, EC-17, EC-6, EC-1, EC-10, EC-2, EC-9, EC-11	EC-7, EC-10, EC-1, EC-5, EC-11, EC-2, EC-9, EC-14, EC-16, EC-17, EC-6	EC-7, EC-6, EC-10, EC-9, EC-11, EC-16, EC-1, EC-2, EC-17, EC-14, EC-5
EC-4	EC-2, EC-11, EC-9, EC-1, EC-14, EC-17, EC-5, EC-7, EC-6, EC-10, EC-16	EC-2, EC-9, EC-11, EC-1, EC-10, EC-14, EC-6, EC-7, EC-5, EC-16, EC-17	EC-11, EC-2, EC-9, EC-1, EC-14, EC-7, EC-6, EC-10, EC-5, EC-16, EC-17	EC-7, EC-2, EC-6, EC-5, EC-14, EC-16, EC-9, EC-11, EC-17, EC-10, EC-1
EC-8	EC-6, EC-1, EC-7, EC-14, EC-16, EC-10, EC-9, EC-17, EC-5, EC-11, EC-2	EC-6, EC-14, EC-7, EC-5, EC-16, EC-17, EC-1, EC-11, EC-2, EC-9, EC-10	EC-6, EC-14, EC-11, EC-2, EC-9, EC-1, EC-5, EC-7, EC-16, EC-17, EC-10	EC-6, EC-11, EC-9, EC-16, EC-10, EC-7, EC-1, EC-2, EC-17, EC-14, EC-5
EC-12	EC-9, EC-2, EC-6, EC-1, EC-10, EC-11, EC-7, EC-14, EC-16, EC-5, EC-17	EC-14, EC-2, EC-9, EC-11, EC-1, EC-10, EC-6, EC-7, EC-5, EC-16, EC-17	EC-2, EC-9, EC-11, EC-1, EC-14, EC-6, EC-7, EC-10, EC-5, EC-16, EC-17	EC-11, EC-9, EC-7, EC-6, EC-10, EC-16, EC-1, EC-2, EC-17, EC-14, EC-5
EC-13	EC-10, EC-7, EC-6, EC-9, EC-16, EC-1, EC-2, EC-14, EC-11, EC-17, EC-5	EC-10, EC-2, EC-9, EC-11, EC-1, EC-14, EC-7, EC-5, EC-16, EC-17, EC-6	EC-10, EC-7, EC-2, EC-9, EC-5, EC-11, EC-16, EC-17, EC-14, EC-1, EC-6	EC-10, EC-7, EC-11, EC-9, EC-6, EC-16, EC-2, EC-1, EC-17, EC-14, EC-5
EC-15	EC-9, EC-11, EC-2, EC-1, EC-10, EC-7, EC-6, EC-14, EC-16, EC-5, EC-17	EC-1, EC-2, EC-11, EC-9, EC-10, EC-14, EC-6, EC-7, EC-5, EC-16, EC-17	EC-9, EC-2, EC-11, EC-1, EC-6, EC-14, EC-10, EC-5, EC-7, EC-16, EC-17	EC-1, EC-5, EC-7, EC-10, EC-6, EC-16, EC-11, EC-2, EC-17, EC-14, EC-9

Table C.8: List of unique past ECs, which are similar to proposed ECs from dataset # 8, sorted in order of decreasing similarity value determined using various approaches

Proposed EC	Our approach	Metric space approach	Probability-based approach	Statistical approach
EC-1	EC-3, EC-14, EC-4, EC-2, EC-17, EC-9, EC-11, EC-12, EC-6, EC-8, EC-13	EC-4, EC-12, EC-14, EC-11, EC-2, EC-9, EC-13, EC-3, EC-6, EC-8, EC-17	EC-4, EC-11, EC-2, EC-9, EC-14, EC-3, EC-12, EC-6, EC-8, EC-13, EC-17	EC-3, EC-17, EC-14, EC-2, EC-4, EC-9, EC-11, EC-12, EC-6, EC-8, EC-13
EC-5	EC-4, EC-3, EC-17, EC-6, EC-8, EC-11, EC-14, EC-2, EC-12, EC-13, EC-9	EC-6, EC-8, EC-17, EC-3, EC-14, EC-12, EC-11, EC-13, EC-4, EC-2, EC-9	EC-6, EC-8, EC-3, EC-14, EC-13, EC-17, EC-11, EC-4, EC-12, EC-2, EC-9	EC-14, EC-3, EC-11, EC-2, EC-17, EC-4, EC-9, EC-12, EC-13, EC-6, EC-8
EC-7	EC-3, EC-14, EC-17, EC-4, EC-2, EC-9, EC-11, EC-12, EC-6, EC-8, EC-13	EC-3, EC-14, EC-6, EC-8, EC-17, EC-12, EC-4, EC-11, EC-2, EC-9, EC-13	EC-3, EC-14, EC-4, EC-11, EC-13, EC-6, EC-8, EC-2, EC-9, EC-12, EC-17	EC-3, EC-17, EC-14, EC-2, EC-4, EC-9, EC-11, EC-12, EC-13, EC-6, EC-8
EC-10	EC-13, EC-12, EC-9, EC-6, EC-8, EC-11, EC-2, EC-3, EC-4, EC-14, EC-17	EC-13, EC-2, EC-9, EC-4, EC-12, EC-11, EC-14, EC-3, EC-17, EC-6, EC-8	EC-3, EC-14, EC-13, EC-2, EC-9, EC-17, EC-4, EC-12, EC-11, EC-6, EC-8	EC-12, EC-13, EC-6, EC-8, EC-11, EC-9, EC-2, EC-4, EC-3, EC-14, EC-17
EC-15	EC-9, EC-11, EC-12, EC-2, EC-4, EC-13, EC-6, EC-8, EC-14, EC-3, EC-17	EC-11, EC-9, EC-2, EC-4, EC-12, EC-13, EC-14, EC-6, EC-8, EC-3, EC-17	EC-12, EC-11, EC-9, EC-2, EC-4, EC-6, EC-8, EC-13, EC-14, EC-3, EC-17	EC-11, EC-12, EC-13, EC-6, EC-8, EC-9, EC-2, EC-4, EC-14, EC-3, EC-17
EC-16	EC-17, EC-3, EC-14, EC-9, EC-11, EC-12, EC-6, EC-8, EC-4, EC-2, EC-13	EC-17, EC-6, EC-8, EC-3, EC-14, EC-12, EC-11, EC-13, EC-4, EC-2, EC-9	EC-17, EC-3, EC-14, EC-13, EC-6, EC-8, EC-11, EC-4, EC-12, EC-2, EC-9	EC-17, EC-9, EC-3, EC-14, EC-2, EC-4, EC-11, EC-12, EC-13, EC-6, EC-8

Table C.9: List of unique past ECs, which are similar to proposed ECs from dataset # 9, sorted in order of decreasing similarity value determined using various approaches

Proposed EC	Our approach	Metric space approach	Probability-based approach	Statistical approach
EC-5	EC-4, EC-16, EC-17, EC-3, EC-6, EC-8, EC-14, EC-13, EC-2, EC-9, EC-1	EC-16, EC-17, EC-6, EC-8, EC-14, EC-3, EC-1, EC-4, EC-13, EC-2, EC-9	EC-3, EC-13, EC-16, EC-17, EC-6, EC-8, EC-14, EC-4, EC-1, EC-2, EC-9	EC-1, EC-2, EC-9, EC-16, EC-17, EC-3, EC-14, EC-4, EC-6, EC-8, EC-13
EC-7	EC-3, EC-14, EC-1, EC-4, EC-2, EC-9, EC-13, EC-16, EC-17, EC-6, EC-8	EC-3, EC-14, EC-16, EC-17, EC-6, EC-8, EC-1, EC-4, EC-13, EC-2, EC-9	EC-3, EC-13, EC-14, EC-1, EC-4, EC-16, EC-17, EC-2, EC-9, EC-6, EC-8	EC-3, EC-14, EC-4, EC-1, EC-2, EC-9, EC-16, EC-17, EC-13, EC-6, EC-8
EC-10	EC-13, EC-2, EC-9, EC-16, EC-17, EC-3, EC-4, EC-14, EC-6, EC-8, EC-1	EC-13, EC-2, EC-9, EC-4, EC-1, EC-3, EC-14, EC-16, EC-17, EC-6, EC-8	EC-13, EC-3, EC-16, EC-17, EC-2, EC-9, EC-14, EC-4, EC-1, EC-6, EC-8	EC-13, EC-6, EC-8, EC-16, EC-17, EC-2, EC-9, EC-4, EC-14, EC-1, EC-3
EC-11	EC-4, EC-2, EC-9, EC-14, EC-1, EC-3, EC-13, EC-6, EC-8, EC-16, EC-17	EC-4, EC-1, EC-2, EC-9, EC-13, EC-14, EC-6, EC-8, EC-3, EC-16, EC-17	EC-4, EC-1, EC-2, EC-9, EC-14, EC-6, EC-8, EC-3, EC-13, EC-16, EC-17	EC-4, EC-14, EC-16, EC-17, EC-3, EC-2, EC-9, EC-13, EC-6, EC-8, EC-1
EC-12	EC-6, EC-9, EC-13, EC-4, EC-2, EC-8, EC-1, EC-14, EC-3, EC-16, EC-17	EC-2, EC-9, EC-4, EC-13, EC-1, EC-14, EC-3, EC-6, EC-8, EC-16, EC-17	EC-2, EC-9, EC-4, EC-1, EC-14, EC-6, EC-8, EC-13, EC-3, EC-16, EC-17	EC-16, EC-17, EC-13, EC-6, EC-8, EC-2, EC-9, EC-4, EC-1, EC-14, EC-3
EC-15	EC-13, EC-9, EC-4, EC-1, EC-2, EC-14, EC-6, EC-8, EC-3, EC-16, EC-17	EC-2, EC-9, EC-4, EC-13, EC-1, EC-14, EC-3, EC-6, EC-8, EC-16, EC-17	EC-2, EC-9, EC-4, EC-1, EC-6, EC-8, EC-14, EC-13, EC-3, EC-16, EC-17	EC-2, EC-9, EC-1, EC-16, EC-17, EC-4, EC-13, EC-6, EC-8, EC-3, EC-14

Table C.10: List of unique past ECs, which are similar to proposed ECs from dataset # 10, sorted in order of decreasing similarity value determined using various approaches

APPENDIX D

Evaluation results for the problem of predicting impact of proposed change effect

D.1 Prediction results for all test datasets

Tables D.1 , . . . , D.10 summarize the impact value of proposed ECs in test dataset # 1 , . . . , # 10, respectively, predicted using our approach, k-NN and regularized local SDA.

Proposed EC	Our approach	k-NN	Regularized local SDA
EC-4	low	low	low
EC-7	low	low	low
EC-8	high	high	high
EC-9	high	high	high
EC-10	high	high	high
EC-17	low	high	high

Table D.1: Impact values of proposed ECs in dataset # 1 predicted using our approach, k-NN and regularized local SDA

D.2 Success rate for all datasets

For all 10 datasets, Tables D.11 and D.12 summarize the success rate in predicting the impact of effect of instances in test dataset and training dataset, respectively, using three different approaches.

Proposed EC	Our approach	k-NN	Regularized local SDA
EC-1	low	low	low
EC-2	low	high	low
EC-7	low	low	low
EC-9	high	high	high
EC-10	high	high	high
EC-11	high	high	low

Table D.2: Impact values of proposed ECs in dataset # 2 predicted using our approach, k-NN and regularized local SDA

Proposed EC	Our approach	k-NN	Regularized local SDA
EC-1	low	low	low
EC-4	low	low	low
EC-7	low	low	low
EC-11	high	high	high
EC-13	high	high	high
EC-15	high	high	high

Table D.3: Impact values of proposed ECs in dataset # 3 predicted using our approach, k-NN and regularized local SDA

Proposed EC	Our approach	k-NN	Regularized local SDA
EC-2	low	low	high
EC-4	low	low	high
EC-7	low	low	low
EC-11	high	high	high
EC-13	high	high	high
EC-16	high	low	low

Table D.4: Impact values of proposed ECs in dataset # 4 predicted using our approach, k-NN and regularized local SDA

Proposed EC	Our approach	k-NN	Regularized local SDA
EC-3	low	low	low
EC-5	high	high	high
EC-7	high	low	low
EC-9	high	low	low
EC-10	high	high	high
EC-11	high	high	high

Table D.5: Impact values of proposed ECs in dataset # 5 predicted using our approach, k-NN and regularized local SDA

Proposed EC	Our approach	k-NN	Regularized local SDA
EC-2	low	low	high
EC-3	low	high	low
EC-8	high	high	high
EC-11	high	low	low
EC-12	high	high	high
EC-15	high	high	low

Table D.6: Impact values of proposed ECs in dataset # 6 predicted using our approach, k-NN and regularized local SDA

Proposed EC	Our approach	k-NN	Regularized local SDA
EC-1	low	low	low
EC-2	low	high	high
EC-8	high	high	high
EC-10	high	high	high
EC-11	high	high	low
EC-15	high	high	high

Table D.7: Impact values of proposed ECs in dataset # 7 predicted using our approach, k-NN and regularized local SDA

Proposed EC	Our approach	k-NN	Regularized local SDA
EC-3	high	high	high
EC-4	low	high	high
EC-8	high	high	high
EC-12	high	high	high
EC-13	high	high	high
EC-15	high	high	high

Table D.8: Impact values of proposed ECs in dataset # 8 predicted using our approach, k-NN and regularized local SDA

Proposed EC	Our approach	k-NN	Regularized local SDA
EC-1	low	low	low
EC-5	low	low	low
EC-7	low	low	low
EC-10	high	high	high
EC-15	high	high	high
EC-16	high	low	low

Table D.9: Impact values of proposed ECs in dataset # 9 predicted using our approach, k-NN and regularized local SDA

Proposed EC	Our approach	k-NN	Regularized local SDA
EC-5	low	low	low
EC-7	low	low	low
EC-10	high	high	high
EC-11	high	low	low
EC-12	high	high	low
EC-15	high	high	low

Table D.10: Impact values of proposed ECs in dataset # 10 predicted using our approach, k-NN and regularized local SDA

Test dataset #	Our approach	k-NN	Regularized local SDA
1	83.33	66.67	66.67
2	83.33	66.67	66.67
3	83.33	83.33	83.33
4	83.33	66.67	33.33
5	83.33	50.0	50.0
6	100.0	66.67	50.0
7	100.0	83.33	66.67
8	83.33	66.67	66.67
9	83.33	66.67	66.67
10	83.33	66.67	33.33

Table D.11: Success rate in predicting the impact of effect of instances in test datasets using various approaches to predict impact

Test dataset #	Our approach	k-NN	Regularized local SDA
1	100	100	100
2	100	94.11	94.11
3	100	94.11	94.11
4	100	94.11	94.11
5	94.11	100	94.11
6	94.11	82.35	88.24
7	94.11	88.24	82.35
8	100	88.24	88.24
9	100	94.11	100
10	82.35	82.35	82.35

Table D.12: Success rate in predicting the impact of effect of instances in training datasets using various approaches to predict impact

APPENDIX E

Evaluation results for overall approach

E.1 Success rate for all datasets

For all 10 datasets, Tables E.1 and E.2 summarize the success rate in predicting the impact of effect of instances in test dataset and training dataset, respectively, using our approach and five state-of-the-art approaches.

Test dataset #	Our approach	Naïve Bayes classifier	C4.5 decision tree classifier	k-NN	Multilayer percepton	Support vector classifier with RBF kernel
1	83.33	33.33	33.33	33.33	16.67	33.33
2	83.33	66.67	83.33	50.0	66.67	33.33
3	83.33	50	83.33	33.33	50.0	33.33
4	83.33	33.33	50.0	33.33	33.33	33.33
5	83.33	33.33	50.0	33.33	33.33	33.33
6	100	16.67	33.33	0.0	16.67	16.67
7	100	50	66.67	33.33	50.0	50.0
8	83.33	50	50.0	50.0	33.33	50.0
9	83.33	33.33	50.0	33.33	33.33	50.0
10	83.33	16.67	33.33	16.67	16.67	16.67

Table E.1: Success rate in predicting the impact of effect of instances in test datasets determined using our approach and five state-of-the-art approaches

Training dataset #	Our approach	Naïve Bayes classifier	C4.5 decision tree classifier	k-NN	Multilayer percepton	Support vector classifier with RBF kernel
1	100	100	94.11	100	100	100
2	100	94.11	94.11	100	100	100
3	100	94.11	88.24	100	100	94.11
4	100	88.23	100	100	100	100
5	94.11	100	94.11	100	100	100
6	94.11	100	100	100	100	100
7	94.11	88.23	100	100	100	100
8	100	100	100	100	100	100
9	100	94.11	100	100	100	100
10	82.35	100	94.11	100	100	100

Table E.2: Success rate in predicting the impact of effect of instances in training datasets determined using our approach and five state-of-the-art approaches

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] G. Q. Huang and K. L. Mak, "Computer aids for engineering change control," *Journal of Materials Processing Technology*, vol. 76, pp. 187–191, 1998.

[2] S. Gunther and N. Ramsey, "Managing obsolescence: Value Engineering Change Proposal proves its worth," in *Defense AT&L magazine*, vol. XXXIII, 2004, pp. 40–41.

[3] *VDA 4965 - Engineering Change Management (ECM)*, 2nd ed., Verband der Automobilindustrie (VDA), ProSTEP iViP Association and Strategic Automotive product data Standards Industry Group (SASIG), December 2006.

[4] R. Barzizza, M. Caridi, and R. Cigolini, "Engineering change: a theoretical assessment and a case study," *Production Planing and Control*, vol. 7, pp. 717–726, 2001.

[5] R. G. Boznak, *Competitive Product Development.* Business One Irwin/Quality Press, Milwaukee, WI, 1993.

[6] G. Q. Huang, W. Y. Lee, and K. L. Mak, "Current practice of engineering change management in hong kong manufacturing industries," *Journal of Materials and Processing Technology*, vol. 139, pp. 481–487, 2003.

[7] *VDA 4965-1 - Engineering Change Request (ECR)*, 3rd ed., Verband der Automobilindustrie (VDA), ProSTEP iViP Association and Strategic Automotive product data Standards Industry Group (SASIG), January 2010.

[8] G. Q. Huang and K. L. Mak, "Current practices of engineering change management in uk manufacturing industries," *International Journal of Operations and Production Management*, vol. 19, no. 1, pp. 21–37, 1999.

[9] T. A. W. Jarratt, "A model-based approach to support the management of engineering change," Ph.D. dissertation, Cambridge University Engineering Department, 2004.

[10] J. C. Giarratano and G. Riley, *Expert Systems: Principles and Programming*, 3rd ed. Course Technology, 1998.

[11] N. Joshi, "Methodologies for improving product development phases through PLM," Ph.D. dissertation, The University of Michigan, 2007.

[12] *ISO/WD 10303-11: Product data representation and exchange: Description methods: The EXPRESS Language Reference Manual*, ISO, 1998, iSO TC 184/SC4/WG11 N48.

[13] *ISO/IS 10303-1: Industrial automation systems and integration - Product data representation and exchange - Part 1: Overview and fundamental principles*, BS and ISO, 1994.

[14] Y. Yang, G. I. Webb, and X. Wu, "Discretization methods," in *The Data Mining and Knowledge Discovery Handbook*. Springer, 2005.

[15] *Teamcenter Change Management (CM) Viewer Help*, UGS Corporation, 2007.

[16] Rukta, Guenov, Lemmens, Schmidt-Schaffer, Coleman, and Riviere, "Methods for engineering change propagation analysis," in *25th International Congress of the Aeronautical Sciences*, 2006.

[17] P. J. Clarkson, C. Simons, and C. Eckert, "Predicting change propagation in complex design," *Journal of Mechanical Design*, vol. 126, no. 5, pp. 788–797, September 2004.

[18] C. Wanstrom and P. Jonsson, "The impact of engineering changes on materials planning," *Journal of Manufacturing Technology Management*, vol. 17, no. 5, pp. 561–584, 2006.

[19] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, J. Gray, Ed. Morgan Kaufmann Publishers, 2005.

[20] C. M. Bishop, *Pattern Recognition and Machine Learning*, M. Jordan, J. Kleinberg, and B. Scholkopf, Eds. Springer Science+Business Media, LLC, 2006.

[21] U. of Waikato, "Waikato Environment for Knowledge Analysis (WEKA) version 3.4.14," Software, 2008.

[22] H. J. Lee, H. J. Ahn, J. W. Kim, and S. J. Park, "Capturing and reusing knowledge in engineering change management: A case of automobile development," *Information Systems Frontiers*, vol. 8, no. 5, pp. 375–394, 2006.

[23] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, January 2000.

[24] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[25] S. K. M. Wong and Y. Y. Yao, "An information-theoretic measure of term specificity," *Journal of the American Society for Information Science*, vol. 43, pp. 54–61, 1992.

[26] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, April 2005.

[27] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Transactions on Computers*, vol. C-26, no. 9, pp. 917–922, September 1977.

[28] M. Dorigo and T. Stutzle, *Ant Colony Optimization.* MIT Press, 2004.

[29] J. Huang, Y. Cai, and X. Xu, "A hybrid genetic algorithm for feature selection wrapper based on mutual information," *Pattern Recognition Letters*, vol. 28, pp. 1825–1844, 2007.

[30] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[31] K. D. Meyer, S. J. Nasuto, and M. Bishop, *Stigmergic Optimization.* Springer Berlin / Heidelberg, 2006, ch. Stochastic Diffusion Search: Partial Function Evaluation In Swarm Intelligence Dynamic Optimisation, pp. 185–207.

[32] M. Dorigo and T. Stutzle, *Handbook of Metaheuristics.* Kluwer Academic Publishers, 2002, ch. The ant colony optimization metaheuristic: Algorithms, applications and advances, pp. 251–285.

[33] T. Stutzle and H. H. Hoos, "Max-min ant system," *Future Generation Computer Systems*, vol. 16, pp. 889–914, 2000.

[34] A. Al-Ani, "Feature subset selection using ant colony optimization," *International Journal of Computational Intelligence*, vol. 2, pp. 53–58, 2005.

[35] M. E. Basiri, N. Ghasem-Aghaee, and M. H. Aghdam, *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics.* Springer Berlin / Heidelberg, 2008, ch. Using Ant Colony Optimization-Based Selected Features for Predicting Post-synaptic Activity in Proteins, pp. 12–23.

[36] R. K. Sivagaminathan and S. Ramakrishnan, "A hybrid approach for feature subset selection using neural networks and ant colony optimization," *Expert Systems with Applications: An International Journal*, vol. 33, pp. 49–60, 2007.

[37] R. Jensen, *Swarm Intelligence in Data Mining.* Springer Berlin / Heidelberg, 2006, ch. Performing Feature Selection with ACO, pp. 45–73.

[38] Y. Y. Yao, *Entropy Measures, Maximum Entropy Principle and Emerging Applications*, ser. Studies in Fuzziness and Soft Computing. Springer, 2003, vol. 119, ch. Information-theoretic measures for knowledge discovery and data mining, pp. 115–136.

[39] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, 1948. [Online]. Available: http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html

[40] P. Resnik, "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language," *Journal of Artificial Intelligence Research*, vol. 11, pp. 95–130, 1999.

[41] S. Santini and R. Jain, "Similarity measures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 871–883, 1999.

[42] M. Ichino and H. Yaguchi, "Generalized Minkowski metrics for mixed feature-type data analysis," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 4, pp. 698–708, 1994.

[43] R. N. Shepard, "Attention and the metric structure of the stimulus space," *Journal of Mathematical Psychology*, vol. 1, pp. 54–87, 1964.

[44] A. Tversky, "Features of similarity," *Psychological Review*, vol. 84, pp. 327–352, 1977.

[45] S. Le and T. Ho, *Discovery Science*, 2004, ch. Measuring the Similarity for Heterogenous Data: An Ordered Probability-Based Approach, pp. 129–141.

[46] D. Lin, "An information-theoretic definition of similarity," in *Proceedings of the Fifteenth International Conference on Machine Learning*, J. W. Shavlik, Ed., 1998, pp. 296–304.

[47] Y. Chen, E. K. Garcia, M. R. Gupta, A. Rahimi, and L. Cazzanti, "Similarity-based classification: Concepts and algorithms," *Journal of Machine Learning Research*, vol. 10, pp. 747–776, 2009.

[48] S. Cost and S. Salzberg, "A weighted nearest neighbor algorithm for learning with symbolic features," *Machine Learning*, vol. 10, pp. 57–78, 1993.

[49] M. R. Gupta, R. M. Gray, and R. A. Olshen, "Nonparametric supervised learning by linear interpolation with maximum entropy," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, pp. 766–781, 2006.

[50] R. P. W. Duin, E. Pękalska, and D. de Ridder, "Relational discriminant analysis," *Pattern Recognition Letters*, vol. 20, pp. 1175–1181, 1999.

[51] E. Pękalska, P. Paclik, and R. P. W. Duin, "A generalized kernel approach to dissimilarity-based classification," *Journal of Machine Learning Research*, vol. 2, pp. 175–211, 2001.

[52] L. Cazzanti and M. R. Gupta, "Local similarity discriminant analysis," in *24th International Conference on Machine Learning*, 2007.

[53] L. Cazzanti and M. Gupta, "Regularizing the local similarity discriminant analysis classifier," in *ICMLA '09: Proceedings of the 2009 International Conference on Machine Learning and Applications.* IEEE Computer Society, 2009, pp. 184–189.

[54] R. Mitra and J. Basak, "Methods of case adaptation: A survey," *International Journal of Intelligent Systems*, vol. 20, no. 6, pp. 627–645, April 2005.

[55] W. Wilke and R. Bergmann, "Techniques and knowledge used for adaptation during case-based problem solving," in *11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA-98.* Springer, 1998, pp. 497–506.

[56] C. Mehta, L. Patil, and D. Dutta, *Advanced Design and Manufacturing Based on STEP*, ser. Springer Series in Advanced Manufacturing. Springer, August 2009, no. 978-1-84882-738-7, ch. STEP in the context of PLM.

[57] J. Owen, *STEP: An introduction.* Information Geometers, 1993.

[58] *ISO/IS 10303-224.2 Product data representation and exchange: Application protocol: Mechanical product definition for process planning using machining features*, ISO, December 2000, ISO TC 184/SC4/WG 3 N988.

[59] *ISO/IS 10303-240: Product data representation and exchange: Application protocol: Process plans for machined products*, ISO, April 2005, ISO TC 184/SC4/WG 3 N1461.

[60] *ISO/IS 10303-233: Product data representation and exchange: Application protocol: Systems engineering data representation*, ISO, 2003.

[61] *ISO/IS 10303-239 Product data representation and exchange: Application protocol: Product life cycle support*, ISO, 2005.

[62] *ISO/IS 10303-203: Product data representation and exchange: Application protocol: Configuration controlled design*, ISO, 1994.

[63] *ISO/IS 10303-212: Product data representation and exchange: Application protocol: Electrotechnical design and installation*, ISO, 2001.

[64] *ISO/IS 10303-214: Product data representation and exchange: Application protocol: Core data for automotive mechanical design processes*, ISO, August 2003.

[65] *ISO/IS 10303-232: Product data representation and exchange: Application protocol: Technical data packaging core information and exchange*, ISO, 2002.

[66] J. G. Inness, *Achieving successful product change.* Financial Times / Pitman Publishing, London, 1994.

[67] J. Brown and M. Boucher, "Engineering change management 2.0: Better business decisions from intelligent change management," Aberdeen Group: A Harte-Hanks Company, Tech. Rep., 2007.

[68] G. M. Intelligence, "Cambridge Engineering Selector (CES)," Software, 2008. [Online]. Available: http://www.grantadesign.com/products/ces

[69] J. G. Bralla, *Design for Manufacturability Handbook*, 2nd ed. McGraw-Hill, 1999.

[70] T.-C. Chang, *Expert process planning for manufacturing.* Addison-Wesley Pub. Co., 1990.

[71] P. Meseguer, F. Rossi, and T. Schiex, *Handbook of Constraint Programming.* Elsevier Inc., 2006, ch. Soft Constraints, pp. 281–328.

[72] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley and Sons, Inc., 2006.

[73] J. N. Kapur and H. K. Kesavan, *Entropy Optimization Principles with Applications.* Academic Press, Inc., 1992.

[74] B. Meyer, *Studies in Computational Intelligence.* Springer-Verlag Berlin Heidelberg, 2008, ch. Hybrids of constructive metaheuristics and constraint programming: A case study with ACO, pp. 151–183.

[75] K. Doerner, R. F. Hartl, and M. Reimann, "Competants for problem solving - the case of full truckload transportation," *Central European Journal of Operations Research*, vol. 11, no. 2, pp. 115–141, 2003.

[76] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[77] R. Giachetti, "A decision support system for material and manufacturing process selection," *Journal of Intelligent Manufacturing*, vol. 9, pp. 265–276, 1998.

[78] D. Cebon and M. F. Ashby, "The optimal selection of engineering entities," Cambridge University Engineering Department, Tech. Rep. CUED/CŨEDC/TR 59, November 1997.

[79] Mathworks, "Optimization toolbox user's guide," March 2010, available online at: http://www.mathworks.com/.

[80] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, Second, Ed. Lawrence Erlbaum Associates, Inc., 1988.

[81] R. Wilson, "On the theory of aggregation," *Journal of Economic Theory*, vol. 10, pp. 89–99, 1975.

[82] A. Rubinstein and P. Fishburn, "Algebraic aggregation theory," *Journal of Economic Theory*, vol. 38, pp. 63–77, 1986.

[83] A. M. Mathai and P. N. Rathie, *Basic concepts in information theory and statistics: axiomatic foundations and applications.* John Wiley and Sons, Inc., 1975.

[84] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining.* The MIT Press, 2001.

[85] G. Beliakov, A. Pradera, and T. Calvo., *Aggregation functions : a guide for practitioners.* Springer Berlin / Heidelberg, 2007.

[86] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval.* Cambridge University Press, 2008.

[87] S. Nakagawa and I. C. Cuthill, "Effect size, confidence interval and statistical significance: a practical guide for biologists," *Biological Reviews*, vol. 82, pp. 591–605, 2007.

[88] J. A. Gubner, *Probability and Random Processes for Electrical and Computer Engineers.* Cambridge University Press, 2006.

[89] Y. Wang, F. S. Makedon, J. C. Ford, and J. Pearlman, "Hykgene: a hybrid approach for selecting marker genes for phenotype classification using microarray gene expression data," *Bioinformatics*, vol. 21, pp. 1530–1537, 2005.

[90] G. Papachristoudis, S. Diplaris, and P. A. Mitkas, "Sofocles: Feature filtering for microarray classification based on gene ontology," *Journal of Biomedical Informatics*, vol. 43, pp. 1–14, 2010.

[91] J. Qi and J. Tang, "Gene ontology driven feature selection from microarray gene expression data," in *IEEE symposium on computational intelligence and bioinformatics and computational biology*, 2006, pp. 1–7.

[92] F. Sebastiani, *Text categorization.* Idea Group Publishing, 2005, ch. Text categorization, pp. 683–687.

[93] Q. Zhang, J. Tan, H. Zhou, W. Tao, and K. He, "Machine learning methods for medical text categorization," in *2009 Pacific-Asia Conference on Circuits, Communications and Systems*, 2009, pp. 494–497.

[94] A. Cardone, S. K. Gupta, and M. Karnik, "A survey of shape similarity assessment algorithms for product design and manufacturing applications," *Journal of Computing and Information Science in Engineering*, vol. 3, pp. 109–118, 2003.