Modeling Engineering Change Management Process in Virtual Collaborative Design Environments

Vildan Kocar

A Thesis

in

the Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Applied Science (Mechanical Engineering) at
Concordia University
Montréal, Québec, Canada

August 2006

Your file Votre référence
ISBN: 978-0-494-20760-4
Our file Notre référence
ISBN: 978-0-494-20760-4

ABSTRACT

Modeling Engineering Change Management Process in
Virtual Collaborative Design Environments
Vildan Kocar

The globalization of the business world results in geographical dispersion of parties involved in design. One of the techniques proposed for providing early involvement of dispersed parties is utilizing Virtual Collaborative Design Environments for supporting conceptual and embodiment design. However, design is not restricted to development stage. A product's configuration goes through engineering changes for improving and refining design throughout the product's life-cycle.

In this thesis, we focus on Engineering Change Management process and modeling this process within a Virtual Collaborative Design Environment. We propose an Active Distributed Virtual Change Environment named ADVICE for performing Engineering Change Management functions. This non-immersive environment offers a superior approach to the existing Engineering Change Management solutions by merging graphical and parametric data involved in the process into a virtual object, which improves comprehension of users and hence decreases the time required for review. ADVICE employs data mining techniques to process captured change history and provides user support with prioritization and change propagation mechanisms.

The proposed environment is demonstrated through a sample application. For verifying the prioritization and change propagation mechanisms, experiments involving synthetic data are conducted. The experiments presented the capability of ADVICE to facilitate Engineering Change Management.

ACKNOWLEDGEMENTS

I would like to take the opportunity to express my gratitude to people that have supported me for accomplishing this thesis.

I am grateful to my supervisor, Dr. Ali Akgunduz, for his support and guidance. He always encouraged me to see "the big picture" and think creatively rather than getting drowned in details. It has been a privilege to work with him and benefit from his ideas.

I would like to thank Dr. Chevy Chen, Dr. Kudret Demirli and Dr. Yong Zeng for taking part in my thesis committee and providing valuable comments for my research.

Greatest appreciation is reserved for my husband, Ilhan for his patience and encouragement. He always had confidence in me and motivated me through all the time we have been together. This has not changed during the time I worked on this thesis.

Finally, I would like to express special thanks to my family, my sisters Gul and Elif, my nephew Yigitalp and above all my parents Inci and Yilmaz for their endless love and support. This thesis is dedicated to them.

To my parents, Inci & Yilmaz SUR

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

ADVICE	Active Distributed Virtual Change Environment
BOM	Bill of Materials
CE	Concurrent Engineering
CM	Configuration Management
DSM	Decision Support Matrix
EC	Engineering Change
ECB	Engineering Change Board
ECM	Engineering Change Management
ECN / ECO	Engineering Change Notice (Notification) / Engineering Change Order
ECR / ECP	Engineering Change Request / Engineering Change Proposal
MRP / ERP	Manufacturing Resource Planning / Enterprise Resource Planning
ODBC	Open Database Connectivity
PDM	Product Data Management
R&D	Research and Development
STEP	Product Data Representation and Exchange Application Standard / Standard for the Exchange of Product Model Data
VCDE / CVDE	Virtual Collaborative Design Environments
VE	Virtual Environment
VP	Virtual Prototype
VR	Virtual Reality

CHAPTER 1

INTRODUCTION

1.1. PROBLEM AREA

In today's highly competitive environment, companies consider the effectiveness of design process as their primary concern for survival. A product launched to market should meet three essential requirements in order to cope with the competitors: lower cost, higher quality and less time-to-market. All of these requirements address one direction; the need for improving the design process.

Until late 70s, companies used to have functional departments responsible of each element of the product life cycle. The customer requirements and market data were collected by Sales & Marketing, they were transferred to R&D department, technical specifications and engineering drawings were prepared and the design was finally released for production [1]. In this traditional type of design, referred to as *Throw-over-the-wall strategy*, the information flow was one-way and deviations from the customer requirements were common. Moreover, as the functional departments were only concerned with their share of the process and do not see the "overall picture", it was probable to neglect some important aspects that could create contradictions with other functional elements [2].

The need for improving the design process gave rise to radical changes in organizational structure and communication methods. Concurrent Engineering (CE) teams, involving members of different functional departments were formed to carry out design process [1, 2]. For ensuring early involvement of the staff specialized in other processes (manufacturing, quality, sales, purchasing, etc.) in design, companies started adopting cross-functional or matrix organization structures. With the introduction of the Supply Chain concept, the customers, suppliers and the distributors also joined CE teams.

However, globalization of economy brought another aspect; geographical dispersion of the parties involved in design. Companies outsourcing and expanding their operations over various locations worldwide need a medium for assuring effective communication amongst team members. This issue, bringing together needs of the business world with novelties offered by research, resulted in utilization of emerging technology of Virtual Reality (VR) for supporting collaborative work.

Various research papers have been published on use of VR environments for concurrent design, referred to as Virtual Collaborative Design Environments (VCDE). Most of these studies, which will be discussed in detail in the Literature Review section, are concerned with conceptual and embodiment design and occur in development stage of the product life-cycle before design is released for production. (Figure 1 [3])

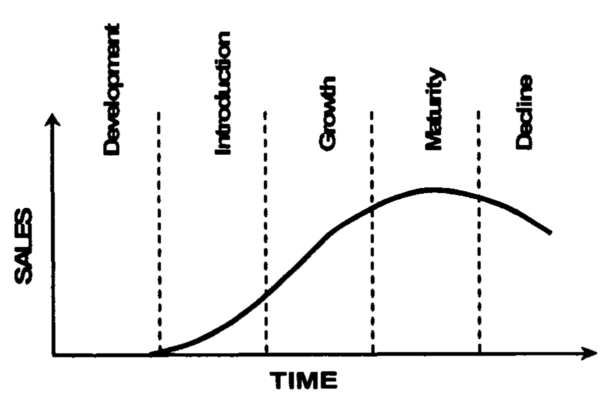


Figure 1 - Stages of the product life-cycle

However, design is a dynamic and iterative process which is not limited to the product development stage. Design of a product is revised and improved in a recursive manner throughout the life-cycle. The need for revision may be due to the correction and refinement of the initial design, as well as other factors such as feedback from the customer, unavailability of the raw materials, manufacturability issues or technological innovations.

Regardless of the reason, the changes made on design are unavoidable; moreover they are desirable, particularly in the early stages of the product's life until a maturity level is reached [4]. These changes, which are defined as modifications in forms, fits, functions, materials, dimensions, etc. in components constituting the design, are called "Engineering Changes" (ECs) [5].

Engineering Change Management (ECM) is the process of organizing, controlling and managing the workflow and information flow for Engineering Changes. In this study, we focus on ECM process and modeling this process in VCDE, which we believe, is a lacking approach in literature.

1.2. MOTIVATION

The ECM process is recognized as one of the most complicated and problematic processes in an organization. Main reasons may be summarized as follows:

a) The number of ECs may reach high levels, especially for high-tech and complex products involving large number of components. Some examples include automotive, aerospace, military, electronics and communication industries, which contain thousands to hundred thousands of components used in a diverse range of product families.

b) In order to reduce raw material and setup costs, it is a common approach to utilize the same sub-assemblies or components in different product families, if possible. This causes an EC to affect a variety of product families or different models of the same product, which complicates the management of the process further.

c) A component used in a product design may go through many ECs through its lifetime until reaching obsolescence. These ECs, denoted as revisions of the same component, should be recorded and tracked in a systematic manner which requires attention and diligence.

d) Yap et al. [6] states that design involves two basic types of information in different representation formats, addressed to left and right lobes of brain.

    - Parametric information: numerical data such as dimensions, mass, volume, tolerances, etc.; perceived better by the logic-oriented left-brain that focuses on details and specifics.
    - Graphical information: drawing or images perceived better by artistic-oriented right-brain that comprehends the overall picture and the relationships. (Figure 2 [7])

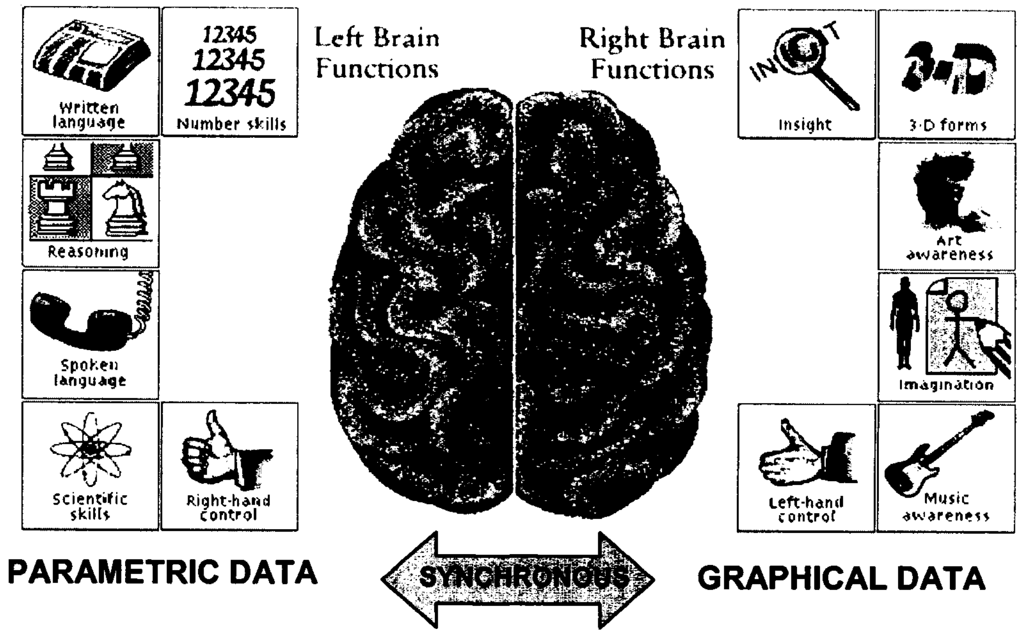


Figure 2 - Parametric vs. Graphical Data

The same idea applies to ECs which involve processing of two types of documents whether ECM system is paper-based or computer-aided. One of these documents involves parametric / numerical data explaining an engineering change and the other is the technical drawing of the change. During ECM process these two documents should be updated in a synchronous manner, which causes the same task to be duplicated.

e) Another challenge is controlling the interactions of the components with each other, with machinery, tools and processes in the organization. Before an EC is authorized,

- warehouse and work-in-process inventories of the component should be analyzed;
- scrap or rework decisions should be made;

- work orders and purchase orders influenced should be examined;
- tooling, fixtures and equipment requirements should be checked;
- effects on lead times, production schedules and project budget should be evaluated, etc.

All of these interactions demand contribution of various departments to approval of the EC, increasing the number of decision makers and hence, the total man-hours devoted for a specific change. It is reported that [8] Engineering Changes consume one-third to one-half of the engineering capacity, which extremely increases the R&D cost.

f) Some of the decision makers, whose responsibility is to analyze and evaluate the change according to their expertise, are non-technical people, such as staff of the purchasing, planning and marketing departments or external parties like customers and suppliers. These members of the design team usually have difficulty in comprehension of complicated technical drawings defining ECs, which may cause errors and time losses.

g) A major difficulty is the determination of the affected components and changing them accordingly. This concept of an EC impacting other components is referred to as "Change Propagation". Change propagation analysis is hard to accomplish and results in prolongation of the cycle time for an EC. In their work for analyzing climate control system development, Loch and Terwiesch [9] stated that cycle time for ECs exceeds actual problem solving time by a factor of 10 or more in some cases. In parallel to their findings, Blackburn [10] reports that the value-added time in ECM process is as low as 8.5 % due to snowballing effects of changes.

h) As the cycle times get longer, the number of simultaneous changes increases and a mechanism for coordinating these changes is required. Some changes may take longer time to be processed as they affect many components and product families. These changes should be processed prior to changes with little impact. Prioritization is usually managed by an Engineering Change Coordinator [4, 11, and 12] but involves consideration of many aspects and is not a straightforward process.

Although the pace of the EC authorization process is extremely crucial, EC cycle times get longer and longer owing to the factors stated above. The extended cycle times result in higher costs. Moreover, as a consequence of the complexity associated in ECM process, possibility of having errors or neglecting important aspects increases, causing the quality of the design to drop down dramatically. In the worst case, even product recalls might occur.

For providing an example for the consequences of a weakly managed ECM system, let us imagine a military communication systems manufacturer producing frequency hopping radios that should be durable in -40°/+60° range and in humid environment. We may assume that marketing and sales department transfers the customer requests demanding a reduction in the weight of some models. As changing the batteries, which are accountable for the majority of the weight is not possible; R&D department proposes a change in mechanical configurations of these radio models, slightly changing some stainless steel bolts and nuts by shaving a thin layer from them. The bolts and nuts are being used in hundreds of different configurations produced by the company, which demands a detailed analysis and review of the EC by various departments and evolution of the other connection items involved. Apart from prolonged cycle time for realizing the change, due to complexity and extensiveness of the EC's effects, it is probable that some important aspects might be neglected. Let us assume in one of the radio models the engineers fail to consider the fit with the other bolts involved in assembly, which causes a tiny gap in the mechanical structure. The items are produced and delivered to the customer, but during impermeability tests performed by the customer the inspected products fail due to this tiny gap. The customer returns the whole party delivered and the company faces problems involving disassembly and re-assembly costs as well as distrust of the customer. If the problem is noticed during use rather than inspection, it might create even more critical and nasty consequences.

In our view, the unpleasant consequences of this scenario could be eliminated by providing users a tool that assures fast and accurate comprehension of the change, aids organizing and accelerating the process and gives user support during review and evaluation. In order to speed up and facilitate comprehension, we may employ VR while for organizing the process and providing user support we may utilize engineering change history of similar changes to form a knowledge-base to direct and manage the process itself. With the use of such a tool, we can avoid overlooking the fit and aptness with other elements as in the scenario and decrease the cycle time of the EC.

1.3. OBJECTIVE

Based on the key problems identified in the previous section and in the light of the scenario presented, we may state that the difficulties associated with ECM may be defeated by:

- Providing the users a shared, real-time and simulated 3D representation of the EC that could be perceived rapidly and accurately by members of the CE team,
- Capturing the parametric EC data from this 3D graphical representation to be used for predicting change propagation information and ordering change requests.

In this study, our objective is to propose a smart Engineering Change Management system embedded in a distributed Virtual Collaborative Design Environment that provides user support by prioritizing the change requests and offering information about impact of the changes.

1.4. SOLUTION APPROACH

There are two basic questions that shape up our approach:

- *"Can we expedite and improve Engineering Change Management process by employing the Virtual Environments for presenting information more effectively?"*

- *"Is it possible to predict effects of changes requested and prioritize change requests based on the historical change-data?"*

For responding the first question, we analyzed the phases involved in ECM process and formed the corresponding database structure. This structure is linked to a distributed 3D-Virtual Environment for data entry and presentation. This environment, named as ADVICE (Active Distributed Virtual Change Environment), offers effective means of presenting data for increasing comprehension of users at each phase. For presenting the proposed model, we built a sample assembly structure. In this example, we limited the ECs to material, shape and dimension changes as the objective is showing how to implement the proposed framework rather than providing a complete solution.

For the second question, we analyzed the techniques for mining sequential change data and finding out the engineering change patterns. We applied two sequential pattern mining algorithms. The first one serves the purpose of finding frequent patterns among the changes made in a specific product family. If the change requested takes place in these patterns, we have a possible set of triggered changes to be offered to the user. The second algorithm searches patterns of changes frequent in most of the product families. If a change pattern exists in many product families, it is an important change that involves plenty of analysis and should be prioritized accordingly. For verifying the proposed procedure, we applied experiments using synthetic data and analyzed the results.

1.5. ROADMAP FOR THE THESIS

The thesis is organized as follows:

- Chapter 2 provides basic information and a review of previous works about three main concepts employed in this study: Engineering Change Management process, Virtual Collaborative Design Environments and Data Mining techniques for sequential data.
- Chapter 3 presents our approach for modeling ECM process in VCDE based on the proposed environment ADVICE and the implementation on a sample model.
- Chapter 4 involves the data mining techniques applied for offering smart means of handling ECs using historical data. The experiments made on synthetic data and the analysis also take part in this chapter.
- Chapter 5 refers to significance of this work and the future extensions are discussed.

CHAPTER 2

BACKGROUND AND RELATED WORK

To the best our knowledge, there is no application considering ECs associated with Virtual Reality. As discussed earlier, VCDE research so far, has dealt with the new product development (conceptual and embodiment design) and does not involve ECM process. Similarly we have not encountered any work on utilization of data mining techniques for capturing and retrieving ECM data from the history of changes. For this reason, we structured this chapter in three sections for providing background knowledge about each of these topics.

2.1. ENGINEERING CHANGE MANAGEMENT

2.1.1. Background

Before going through the literature on Engineering Change Management concept, we should give some brief knowledge about the ECM process. Whether text-based or visual-aided, Engineering Change Management process is composed of three main phases: Request, Approval and Notification. These phases are discussed in detail below.

Phase I: Engineering Change Request (ECR)

Sometimes referred to as Engineering Change Proposal (ECP), in this stage the need for a change emerges and placed in ECM system by an initiator with a manual or electronic form. The initiator may be anyone concerned with the component and thinks there is room for improvement. In most companies, there exists a person or a department working as a "coordinator", receiving the ECRs and filtering them before entering ECM system to avoid unnecessary changes or duplicates. The ECR is also checked in the history of ECs and rejected if it is already suggested but was not found feasible.

Apart from the initiator name and department, ECR contains information about which component to change; which attributes to change; the reason of the change; the time of the change and an attached technical drawing representing the change. The change may be adding or deleting a part from the product structure (BOM). In this case, change request is placed on the parent sub-assembly, not the component itself. Sometimes, ECRs may have supplementary documents for validating the reason of the change such as declaration of unavailability from the supplier, revised sales order involving new customer specifications or non-conformance reports. But the essential document attached to an ECR is just the technical drawing.

As discussed earlier, ECRs may reach high levels and prioritization may be needed. There is need for a methodology for assuring ECR prioritization. In most organizations, this is managed by the coordinator mentioned above, assigning arriving requests to groups by applying ABC analysis. For the most important ECRs, immediate action is taken, cancelling all work orders and purchase orders of the component and processing the change instantaneously, defined sometimes as "crash-basis" [13]. In some cases, companies apply batching for the less significant and less urgent change requests to manage similar types of ECRs together. Batching is considered to be improper by some researchers [9] and listed among the key contributors of long cycle times. However, Bhuiyan et al. [14] suggest that batching brings a significant reduction in time and effort during the product development stage. In our view, batching may be applied to some extent; it is more crucial to improve the subjective prioritization procedure applied for deciding whether requests should be batched or processed right away.

Phase II: Approval Cycle

Before ECR enters the approval cycle, the technical documents are flagged to be "on-hold" status until ECM process is finalized for preventing confusion and errors.

ECRs are released for approval to parties concerned with the component. People taking part in approval process are referred to as Engineering Change Board (ECB) and belong to different functional departments. The members of the ECB are not fixed. Each ECR demands participation of different people. This is usually managed by forwarding the ECR to relevant staff in each functional department. In companies utilizing MRP / ERP systems, roles defined in component part master records (planner, buyer, inspector, work center etc.) may be used for determining the evaluator.

Once an ECB member receives a change request for approval, his task is evaluating the change in terms of its feasibility and impacts on interacting processes, tools & equipment, procedures, routing and other components in product structure. For accomplishing this task, along with his expertise, the evaluator needs information about the component's current status in MRP/ ERP system (inventory level, open or scheduled work orders, purchase orders, lead time, cost, inspection results, approved vendors, sequence of operations and work centers it goes through, components on the same-level in BOM etc.). It is stated that a component's EC approval phase becomes harder to manage as the constructional complexity, number of variants and number of components interfaced grows [15]. In some cases, it is impossible to check all the interfacing components in product structure due to time limits, although finding out the change's impact on other components that needs to be changed, in other words, change propagation is a compulsory output of approval phase. If the component takes part in multiple product structures, determining the change propagation becomes a burden for the evaluators.

The approval cycle phase ends when all the members in ECB approve the change. In case of any rejection, the ECR is returned to the initiator with the reason for rejection and recorded for future reference.

Phase III: Engineering Change Notice / Notification (ECN)

This phase is named after a document (Engineering Change Notice, ECN) used for distributing the change to all related parties by the coordinator department. This document, referred to as Engineering Change Order (ECO) in some organizations, is designed to respond the following questions [13]:

- WHAT is being changed? The component changed is defined.
- HOW is it being changed? The technical drawing showing the change is attached.
- WHEN will the change be effective? The effective date until which the old version of the technical documents will be used, is stated.

- WHO is responsible for performing the actions to be taken due to the change? An action plan defining the tasks for each recipient of the ECN is given (rework for old-versioned inventory, change of inspection criteria, price updates, etc.).
- WHERE does the change demand updates? The documents that need revision such as, technical drawings, user manuals, operation manuals are listed.
- WHY is the change made? The reason of the change is given.

ECM process ends with the distribution of ECN together with the new revision of the technical document. Next, triggered changes resulting from change propagation enter the ECM process. Figure 3 shows the overall flowchart for ECM process.

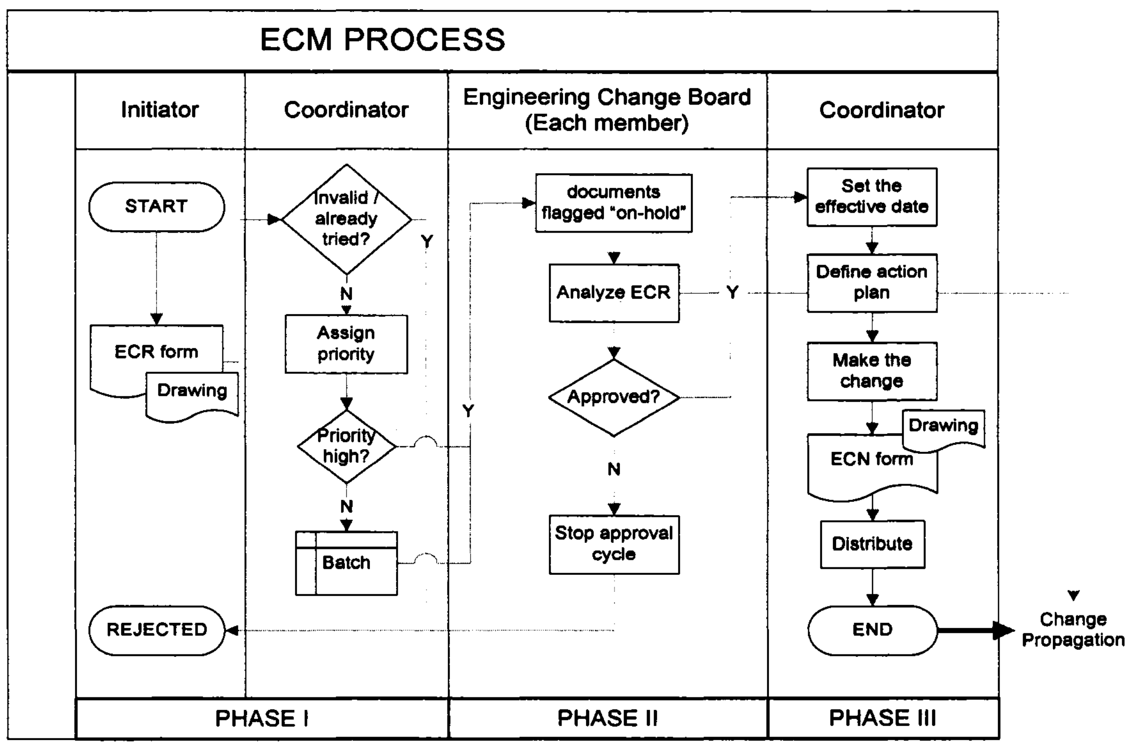


Figure 3 - Flowchart defining Engineering Change Management process

2.1.2. Related Literature

When we go through literature about Engineering Change Management, we see that most of the work on this area involves survey research and industrial case studies conducted in companies. There are some studies proposing tools and solutions but most of them are product or company specific. This is not surprising as the ECM concept is a business-driven research area.

In some studies, ECM is discussed within Configuration Management (CM) concept, which is a more generalized framework than ECM and a common term used especially in defence and software industries. CM involves the management of a product's documentation through the entire life cycle until the disposal by assuring all the sub-components and processes conform to this documentation [16].

The common background for the survey studies is their emphasis on the significance of improving ECM process and attempts to identify the problems associated with changes. Among them, Huang and Mak [12] offer the most comprehensive survey analyzing 100 UK manufacturing companies. The study revealed that a high percentage of the companies recognize the importance of an effective ECM system and have formal procedures for handling it. Most of the companies investigated in the survey compromised on the need for a coordinator and utilized prioritization of ECRs. An interesting result of the study is the fact that companies usually employ paper-based systems despite presence of computer-aids. A further study analyzing computer aids and searching the reasons for poor utilization of computer-based ECM systems reveals that although they provide simultaneous and easy access to EC data, computer-based systems are not preferred because they are not customizable and they demand intensive effort and commitment while providing little help [17].

Industrial case studies are the widest area among previous work. In general, they aim to analyze ECM system in a specific company and propose solutions based on their analysis and observations. Wright [18], who offers a detailed review of the research in ECM area between years 1980 and 1995, mentions studies dealing with VLSI design, discrete electronic circuit design and naval vessels.

An important work which focuses on climate control system in automobile development [9], lists the key contributors to long lead times in ECM process as; complex approval process; capacity and congestion; batching; snowball effects of changes; and organizational issues. In this article, the change propagation is defined as snowballing effect of changes and the interactions are referred to as couplings. Based on their observations, the authors defined these interactions as; couplings between a component and its corresponding manufacturing process (product-process); couplings between a component and other components within the same subsystem (intra unit product-product); and couplings between a component and other components in different subsystems (inter unit product-product). This definition, although being narrower than our definition of interactions, is consistent with our idea of considering different product families in change propagation. The approach proposed for handling snowball effects of changes is rating the strength of couplings in a scale of 1 to 5 and using this information to trigger changes. The authors also analyzed capacity of the workforce and congestion effects in another study [19] and proposed some qualitative solutions.

Another research paper [4] involves analysis of ECM system in British Aerospace Military Aircraft Division, which involves processing of customer-driven change requests that are sponsored and prioritized directly by the customer. The system is a computer-based application developed in-house as the commercially available ECM packages need to be customized and refined. In a comparative study of three Swedish companies, it is stated that the company-specific characteristics of ECM process decreases adoption of standardized Product Data Management (PDM) products [20]. Similarly, Tavcar and Duhovnik [15] analyze and compare three examples of different production types in terms of ECM; individual production; serial production of modules; and production of household appliances and specify that each have specific requirements.

A complete and generalized tool for processing ECs is offered by Huang et al. [5] inspired by web technologies and the results of their survey on computer-aids in ECM [17]. The aim is providing a solution that is more user-friendly and simple compared to commercial PDM systems and allow simultaneous access and functionality unlike in-house developed, standalone ECM systems. The structure is based on a relational SQL database, offering web forms for user interface. According to them a change might have 3 types of impacts, on processes on machines and on fixtures. These impacts are recorded during evaluation stage and rated in 5 categories of severity. Developed ECN form contains effective date and action plan information, in conformance with ECN definition in Phase III but technical document attachment is not included in the proposed web solution.

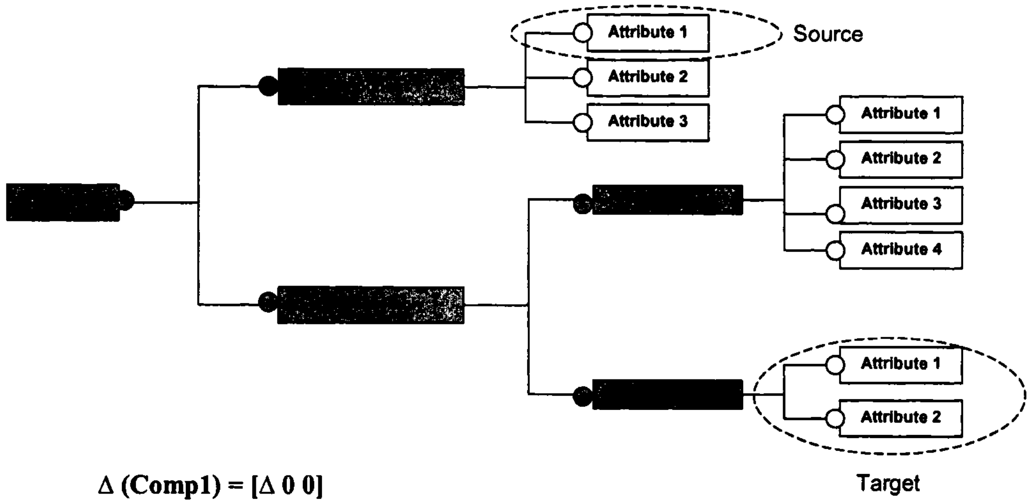
The other tools offered to aid ECM process involve, applications focusing on a specific aspect of ECM rather than complete solutions. An example is [11] a system tracking percentage of job completed and deadlines for approval, warning the member of ECB delaying the work. There are some studies which focus on aiding the user for performing change propagation. A parameter-based approach is used in some of them [21, 22] which involves; defining engineering parameters during initial design, discussing their dependencies on each other one-by-one in formal meetings and forming a parameter network defining relations. This parameter network, created by a small team during initial design stage, is used for triggering changes throughout the life-cycle. The methodology is subjective and creates extensive workload. A similar study [23] uses Design Structure Matrices (DSM) concept for defining dependency relationships between components in terms of probabilities during initial design. An interesting idea in the article is dividing change propagation to two composites, *likelihood* of occurrence due to another change and the *impact* defining average proportion of design that needs to be redone if the change propagates, as a measure of significance of the change.

There are numerous studies analyzing change propagations in relation with STEP – ISO 10303 (Standard for the Exchange of Product Model Data) [24-27]. The aim of STEP is providing a generalized framework among companies collaborating by standardizing the data presentation. STEP uses a specific data definition language called

EXPRESS for defining entities, attributes and schemas representing the relations between entities. Everything can be "expressed" using EXPRESS; likewise ECM process. Peng and Trappey provide a good practice of designing EXPRESS model of the ECM process [28]. For broader information about STEP, ISO web site may be consulted [29].

Among the studies employing STEP for offering a change propagation mechanism, the work of Cohen [26, 27] has been particularly inspiring for our study. In his work, instead of components, attributes that define the components are propagated through a recursive formulation. The attributes of components used in design are linked through C-FAR matrices, which provide a new format for design representation. These matrices contain qualitative values for defining linkages, such as Low, Medium and High. For finding out how a change to an attribute in one component influence attributes of another component;

- Entities (components) and attributes are defined;
- Relations between components and attributes are represented on an EXPRESS schema;
- Source and target components are determined;
- Influence path connecting source and target is specified;
- For expressing the influence path mathematically a recursive formulation is employed. A sample EXPRESS graph and formulization is given below.



Δ **(Comp1) = [**Δ **0 0]**
Δ **(Product) =** Δ **(Comp1) * CFAR(Comp1, Product)**
Δ **(Subassembly) =** Δ **(Product) * CFAR (Product, Subassembly)**
Δ **(Comp3) =** Δ **(Subassembly) * CFAR (Subassembly, Comp3)**

Figure 4 - Representation of EXPRESS schema for Change propagation

However, the approach assumes dependency relationships between attributes, i.e. values in C-FAR matrices, are pre-defined and given as an input. In this way, the study shares the same deficiency with parametric-based and DSM-based approaches [21-23]; definition of the dependencies is based on subjective decisions made during initial product development; they do not change through time. On the other hand, we know most of the designs in key industries such as automotive, aerospace, defence, etc. have long life-cycles and go through thousands of changes triggering others, some of which do not correspond to dependencies defined by initial design team.

We believe, it is essential to update the dependency information retrieving history of change sequences in order to provide effective means of change propagation. Capturing and storing history of changes may be managed by employing VCDE concept, which will be studied in the following section.

2.2. VIRTUAL COLLABORATIVE DESIGN ENVIRONMENTS

A Virtual Environment (VE) is a three-dimensional computer-simulated environment allowing user interaction with virtual copies of the real-world objects. In literature, the term is sometimes used interchangeably with Virtual Reality. According to our understanding, Virtual Reality defines the general framework and techniques for user interaction with virtual objects; our attention is rather on Virtual Environments.

Unlike ECM concept, the literature on VE is extremely rich including manufacturing, micro and nano-technology, aerospace, ergonomics, defence (avionics, submarine and mine detection training), medicine and surgery, heritage, retail, furniture, architecture, education and entertainment applications. As we are dealing with Engineering Change Management, which is a design activity demanding high level of collaboration we have narrowed down our focus to Virtual Collaborative Design Environments (VCDE).

In his comprehensive work, Horvath [30] has enumerated 6 major components of VCDE presented in a kernel-like structure (Figure 5). Based on these components we may define Virtual Collaborative Design Environments as shared, simulated 3D environments facilitating creative collaboration of the design team;

- by supplying real-time virtual visualization and manipulation of the model allowing discussions of geographically distributed parties;

- by supporting the designers with enhanced CAD functionality and knowledge management tools to capture and present design data.

Figure 5 - Kernel-like structure showing 6 ingredients of VCDE

As it may be noticed, this definition does not necessarily impose a VCDE to employ technological devices for assuring an immersive atmosphere. In fact, virtual environments range from elementary (non-immersive monitor-based VE offering mouse and keyboard interaction) to highly complex systems (VE offering immersive display and Human-machine interaction technologies making use of special devices or systems). Due to our circumstances, the proposed model in this thesis involves an elementary non-immersive desktop application supporting 3D visualization, but it may be updated to offer an immersive environment with visual, auditory and haptic feedback by employing innovative interaction technologies. (Figure 6)

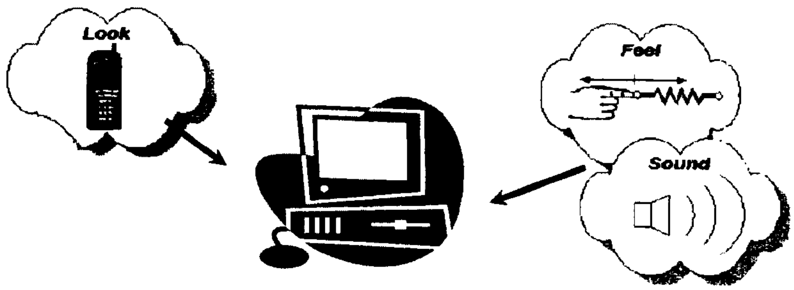


Figure 6 - Visual, Auditory and Haptic interaction in immersive VE

There is a vast amount of literature concerning the development of VR interaction devices. These devices, some of which are commercially available, include Head Mounted Displays (HMD), Virtual Retinal Displays, EyePhone, Shutter Glasses, Binocular Omni-Orientation Monitor for visual interaction, Head Related Transfer Function (HRTF) based systems for localizing sounds, Haptic and Tactile feedback devices, Gyroscopic orientation tracker, 3D scanner, 3D mouse, Data Gloves (pinch glove), etc. Apart from these technologies there are second generation VR systems such as CAVE and Workwall, which are cubicle interfaces and desktop stereo displays such as Responsive Workbench. Brief information about descriptions and working principles of these technologies is beyond our scope and can be obtained referring to [31-33].

Throughout the literature, we encounter many approaches utilizing either these technologies or elementary 3D systems to develop VCDE. These studies search an answer to the following research question: "*Can VCDE revolutionize the presentation and utilization of the design knowledge?*" Yap et al. [6] states that the most important feature offered by 3D Virtual Environments is the ability to articulate difficult-to-describe objects and structures, because design involves representation, interpretation and communication of implicit knowledge. He mentions an interesting analogy for the design process; people can recognize a face they have met but are not usually able to describe it verbally. Likewise, most of the time designers have difficulty in expressing the concept in their minds with words.

Apart from the extended capabilities to express design knowledge, Virtual Prototypes (VP) in 3D design environments considerably improve the user comprehension and aid the CE process by offering characteristics like ability to navigate and fly through and by supplying inside-out, outside-in and 360° views. In addition to visualization and presentation abilities, in a comprehensive review about Virtual 3D prototypes [34], brief knowledge about features such as checking for fit and interference with other components; testing and verification of functions based on kinematics, computational fluid dynamics, finite element analysis, etc.; supplying manufacturability simulations; and applying human factors analysis are given.

Among the studies in literature, although none matches our purpose of modeling ECM process, we may mention the following that provided insight.

- DDRIVE (Distributed Design Review in Virtual Environments) [35], which aims to provide Virtual Meetings between users with 2D workstations and users equipped with 3D displays and haptic devices;

- MDE (Multimodal Design Environment) [36], focusing on actions like sketching, cutting, drilling, bending, etc. involved in industrial design and proposing tools like Virtual Cursor, Interaction Ball, Cutting Plane in VE;
- Manufactur [37], developed for collaboration of architecture projects which offers documents and drawings to be attached to 3D boxes in the workspace, linking the content of the document to this 3D object by OLE database;
- Provit [38], providing an immersive conference management system for design reviews working on optical networks;
- Multi-agent environment for supporting steel girder bridge design [39], that warns the user checking dimension constraints;
- A system for transferring configuration involved in CAD models to a virtual environment via XML and sharing them on the web using VRML [40];
- A web-environment that allows design team to communicate with text messages that are recorded in log files forming history of design [41];
- A VE for design reviews [42], presenting data based on Gestalt principles of psychology, arranging alternatives of the selected component in a virtual conveyor belt, highlighting the alternatives using colors and even providing presentation of the BOM tree with virtual objects.

Although Kan et al. [41] mentioned recording text messages of the design team for forming history; neither his study nor any other study offered analyzing design logs based on data mining techniques to retrieve useful information in VCDE. Beyond the scope of design process, there are some approaches for knowledge extraction in Collaborative VE.

One of these solutions applied clustering to find proportional distributions in a VE serving as a workspace for student assignments [43]. Another proposed utilizing Exploratory Data Analysis to find out participations of students based on the logs of communication in a Virtual Campus [44]. Finally, Dolunay and Akgunduz [45] developed an automated end-user behaviour assessment tool for remote design and testing through analyzing the logs of VR simulations.

2.3. DATA MINING TECHNIQUES FOR SEQUENTIAL DATA

Data Mining, as defined by Wang and Yang [46], is the process of extracting implicit knowledge and discovery of interesting characteristics that are not explicitly represented in the databases. Sequential pattern mining is a subheading in this area analyzing sequences of data to come up with meaningful patterns showing behaviour of the dataset. It has been an active research area in a wide range of applications such as analyzing customer behaviour, stocks and markets, DNA sequences, performance analysis, crime solving, web navigation analysis, natural disaster prediction and scientific analysis.

In sequential pattern mining, the problem may be generalized as finding the complete set of frequent sub-sequences (or patterns) in a given set of data sequence. This frequency is named as "*Support*" of the sub-sequence. We say that a sequence S supports a sub-sequence s_i if $s_i \subseteq S$.

The data in a sequence is in the form of transactions occurring at given points in time. These transactions may involve only one event or concurrent or parallel events occurring at the same time. Each of these transactions are referred to as "*itemsets*". A sequence involving *k* itemsets (transactions) is denoted as "*k-sequence*".

To provide an example for illustrating the definitions above, let us consider a sequence *S* = < (A) (BC) (D) (AE) (C) >. In this sequence there are five transactions; among them B and C are concurrent events, while A and C are serial events. The sequence S is called 5-sequence, denoting the number of itemsets it involves. The support of the sub-sequence s_i= < (A) (C) >, denoted by $supp(s_i)$ is 2, meaning that the sub-sequence s_i exists twice in *S*= < (**A**) (BC) (D) (**A**E) (**C**) >.

The patterns whose support is larger than a minimum threshold are frequent patterns. The ultimate goal of mining sequential patterns is finding all frequent patterns given pre-defined minimum threshold value, which is named as minimum support.

The earliest approach to sequential pattern mining was by Agrawal and Srikant [47], whose work constituted a framework for other researchers. Their aim was analyzing customer transactions in retail industry to come up with possible patterns of products purchased in a sequence by most of the customers in order to develop marketing strategies. Their work is based on the Apriori property that was first proposed by themselves in another work [48] and has guided most of the studies in the area. The Apriori property is formally defined as the following [46]:

Let Π be a set of sequences, P be a sequential pattern and supp(P) be the support of P in Π. Then, $supp(P) \leq min\ (supp(P'))\ \forall P' \subseteq P$.

This idea may be interpreted simply as follows, if a longer pattern is frequent; all its subsets are frequent; likewise if a pattern is not frequent then none of its super-sequences is frequent. The 5-step procedure named as "*AprioriAll*" [47], basically determines the frequent events or event groups in sorted customer sequences (called "*Litemsets*"); transforms the sequences to be expressed in terms of these Litemsets; apply AprioriAll algorithm in a level-wise manner to find the set of frequent sub-sequences and eliminate the sub-sequences already present in a larger sequence to find maximal sequences. The main principle of level-wise AprioriAll algorithm is based on Apriori property; the algorithm proposes starting 1^{st} level (or iteration) with 1-sequences and generating level *i*, based on the frequent sub-sequences of level *(i-1)*.

AprioriAll algorithm is aimed at finding common patterns among customers. We think we can apply this algorithm to our case for finding patterns of changes among product families or models. However, for searching patterns within a product family, we need another approach. Mannila et al. [49] offers two approaches based on telecommunication alarm management utilizing the concept of "*episodes*". An episode is partially ordered collection of events occurring together. The main idea is describing temporal relationships between events and checking the sequence of events according to the time frame they belong. If we do not include time frame concept our results might be inaccurate and improper as we may try to build sequential relationship between irrelevant events. On the other hand, if we just control consecutive events we would overlook most of the sequential relationships.

The first algorithm offered by Mannila et al. [49], WINEPI uses a sliding time-window through the sequence where the width of window is pre-defined. An episode is defined as frequent if the support (frequency) of the episode exceeds minimum frequency threshold where the support of the episode is defined as the number of time-windows it occurs. The Apriori property holds as all sub-episodes β are frequent if an episode α is frequent, where $\beta \leq \alpha$.

The problem with the WINEPI algorithm arises when an event occurs more than one time within the same time window. For this reason, a second approach called MINEPI is developed by the same authors [49]. This procedure is based on recording the beginning and end times for each potential episode and eliminating them from the potential set if they are beyond the window size limits.

There are other studies associated with finding sequential patterns, most of which represent solutions to more complex sequential pattern mining problems with constraints. Among them, GSP (Generalized Sequential Pattern) [50] incorporates taxonomies or hierarchies of events, minimum and maximum time-gap constraints and sliding time windows into AprioriAll algorithm. FreeSpan [51] and PrefixSpan [52] propose partitioning of the database to smaller databases rather than employing a Apriori-based candidate generation procedure. Among them, PrefixSpan aims to improve FreeSpan by providing prefix-based projection using the frequent itemsets at each step as prefixes. PrefixSpan offers a more effective approach than Apriori-based algorithms for longer sequential patterns in terms of run-time performance. However, for relatively short sequences where the run-time performance is not very critical, AprioriAll is more preferable as PrefixSpan involves complexity associated with construction of the projected databases. The other studies in literature are more specific and constrained approaches using fundamental principles of the algorithms mentioned above. To list a few, we may mention ISP (Interval Sequential Patterns) Algorithm [53], dealing with intervals between customer transactions; Data Mining Language [54], mining just the user-required patterns; approaches for handling multi-dimensional datasets [55]; episode-based algorithm defining events as a function of attributes [56]; procedures for introducing constraints to PrefixSpan algorithm [57].

To sum up, as most of the literature in Sequential Data Mining is based on introductions of constraints and more complex structures, we find it more reasonable to study the most fundamental approaches AprioriAll and MINEPI which are appropriate for our data set. These algorithms will be adapted to change data to find the patterns of changes in Chapter 4.

CHAPTER 3

MODELING ECM PROCESS IN VCDE: ADVICE

As outlined earlier in Chapter 1, the objective of this thesis is "*developing a smart Engineering Change Management system embedded in a Distributed Virtual Collaborative Design Environment that provides user support by prioritizing the change requests and offering information about change's impact on other components*". This chapter presents our methodology for reaching this objective and defines the proposed environment ADVICE (Active Distributed Virtual Change Environment).

ECM process, as described briefly in Section 2.1.1 and illustrated in Figure 3, is composed of 3 basic phases, namely Engineering Change Request, Approval Cycle and Engineering Change Notice. For modeling this process on VCDE, we first need to determine data requirements for each phase based on the sequence of actions and actors carrying out these actions. We, then establish the corresponding database structure. The next step is to link this database structure to the interactive Virtual Environment, ADVICE offering a user interface for 3D presentation of data to users taking part in the process. Finally, we develop a Decision Support System assuring textual and graphical data flow throughout ECM process and providing user support based on the historical change-data.

After describing the proposed environment ADVICE in detail with all aspects, later in the thesis, we demonstrate the characteristics of the proposed environment ADVICE with a sample application.

3.1. DATABASE STRUCTURE

3.1.1. Overview of actions and usage requirements in an ECM system

There are basically four user groups (actors) involved in an ECM process. All of these actors realize some portion of the process by performing actions via interaction with other actors.

These actors (or user groups) are:

- Initiator
- Coordinator
- Engineering Change Board Members
- Departments to be informed of the change

Determining the actions carried out by these actors throughout the process is essential not only for identifying data requirements but also for designing the user interface of ADVICE in the following sections.

A Use-Case Diagram describing actions performed by the above mentioned actors is given in Figure 7.

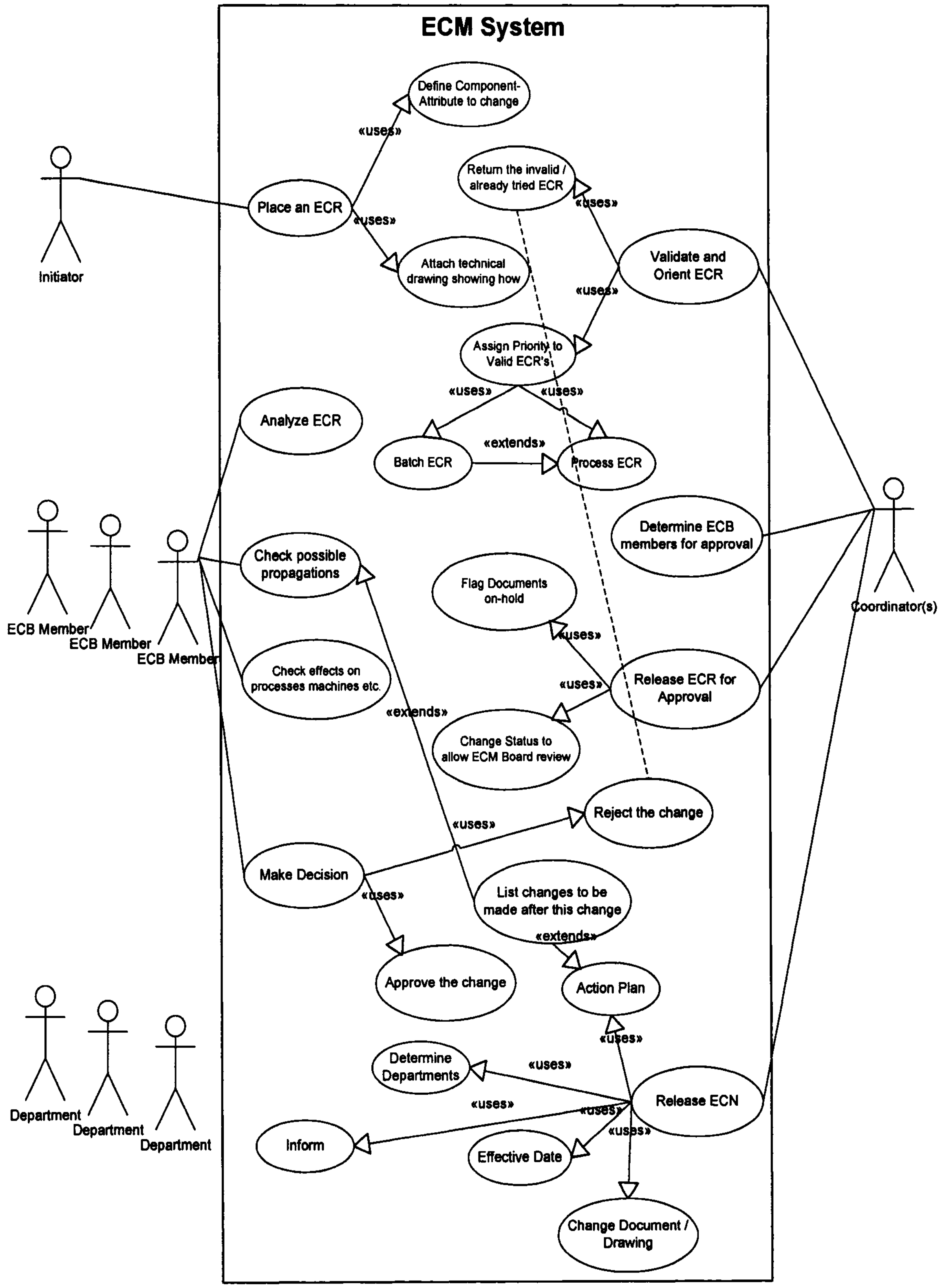


Figure 7 - Use-Case Diagram for ECM process

3.1.2. Data Requirements for each phase in ECM

Each phase of ECM demands storing, processing and presenting a variety of information about the change for the actors to perform the actions presented in Figure 7.

Phase I: Engineering Change Request (ECR)

In a traditional ECM system, ECR phase starts with an ECR form involving textual and graphical information (technical drawing). Most of this information is based on BOM tables in MRP/ERP and component specifications in CAD systems that are already available in the company's data warehouse. For this reason, before passing to the data requirements related to ECR itself, we should design the supplementary data tables containing BOM and CAD data.

Generally, BOM tables in MRP / ERP system involve the following fields for each component in each product family:

- Product Model / Assembly
- Component No
- Component Description
- Level on BOM
- Quantity per assembly
- Unit of Measure
- Revision
- Price (or unit cost if the component is a "make" component rather than "buy")

CAD files containing technical drawings involve component specifications which are expressed in terms of "*attributes*" of the component. Attributes define the component; some examples may be material code, height, width, depth, radius, shape code, weight, density, colour, maximum energy, minimum temperature, maximum load, resistance etc. Each component has different attributes and each initial design is actually based on assigning the "best" values to these attributes so that the overall assembly functions "as it should". Engineering Changes are suggested to change the values of these attributes on a specific product model. In the light of these, we may say the CAD structure already involves:

- Product Model / Assembly
- Component No
- Attribute Name
- Current Value of the attribute
- Position vector of the component, i.e. location in the overall assembly
- Rotation vector of the component, i.e. roll, pitch, yaw

CAD structure for a product model/assembly matches with BOM structure defined in MRP/ ERP system for that product model. As we have mentioned in Chapter 1, a component may exist in more than one product models offered by a company. An ECR is given to change a component's attribute(s) in a specific product model; it may or may not be required for the other models containing that product in their structure. For this reason, ECR information should be stored including the product model, component and attribute(s) changed. Figure 8 represents sample product structure for two different product models of fishing reels having 7 common components [58]. In case of an EC given to one of these components in one assembly, it is obvious that the necessity of changing the same component on the other assembly might arrive.

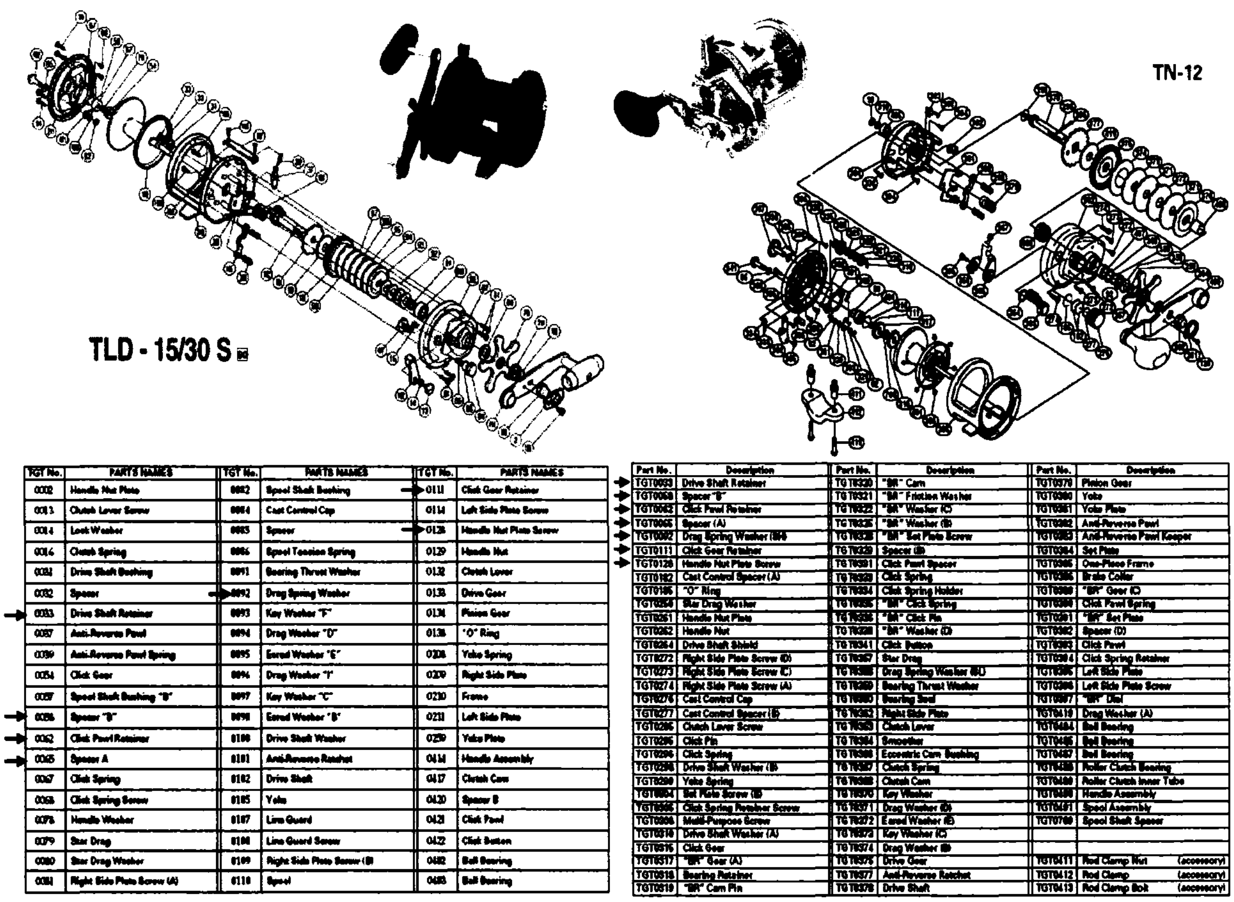


Figure 8 - Comparative product structures for two product models [58]

We may also add a Human Resource or Personnel table to supplementary tables containing BOM and CAD structure. This table already present in personnel records involves:

- Person's Name
- Login ID used in computer network
- Department

Having defined Supplementary tables, we may list the data requirements for each phase of ECM. Table 1 lists these data requirements and possible sources for collecting them in Phase I, ECR phase.

Table 1 - Data requirements for Phase I

Phase I :ECR - Data Requirements	***Suggested Source***
EC No, a unique number assigned to EC when ECR is entered	Auto Number
Product Family / Assembly change is made	User entry
Component No	User entry
Component Description	BOM table
Level on BOM	BOM table
Quantity per assembly	BOM table
Attribute Changed	User Entry
Previous value of the attribute	CAD table
New Value of the attribute	User Entry
Reason Code (i)	User Entry
Reason Explanation	Reasons table
Reason Comment	User Entry
Time of ECR, recorded as "Now" when ECR is entered	Auto Date
Initiator, recorded as Network login (ii)	Personnel table
Technical Drawing showing the current values of attributes, location and orientation of components on the assembly	CAD table
Status, assigned to R (Request) at the time of entry	Auto Text
Priority, at first assigned to value proposed by the ECM system and revised by the coordinator (iii)	User Entry

In this table three aspects should be noted:

(i) Reason codes are mapped to corresponding textual explanations in another table (*Reasons* table) for statistical purposes and involve reasons such as, "Failure to meet customer requirements", "Correction of a drawing error", "Non-conformance to inspection standards", etc. Using the reason codes, it is possible to standardize and statistically analyze the ECs and take corrective / preventive action for improving the processes in an organization. For instance, high levels of ECs due to "Failure to meet customer requirements" imply that there could be some problems in communication and transfer of customer requirements to design; and consequently the procedure for carrying out this process should be re-examined and improved.

(ii) Initiator, like all the other actors (user groups) taking part in ECM process, gets access to ECM system with a login ID. For this reason, capturing this information and connecting with personnel records to retrieve name and department of the initiator is not a problem. Presenting users information in different levels of detail using the same environment could be possible by defining their roles and corresponding login and password information.

(iii) ADVICE provides user support by proposing an initial value A, B or C for priority, "A" being the most important. This is done by applying a smart mechanism for mining change sequences in the history. This mechanism will be explained in detail in Chapter 4. For now, it is sufficient to know that the output of this mechanism is a *Priority* table containing A, B and C values for each component and attribute. This table provides the initial value for priority which is revised and changed by the coordinator if needed. For instance, the coordinator might decide a "C" priority change request should be processed earlier due to high costs associated with the component changed and hence, change the priority level to "A" or "B". The *Priority* table does not contain a default priority value for all possible changes; it just contains the changes that have frequently created an impact among product families and hence, changes that have considerably longer lead times due to their impact. For this reason, the default priority values are supplied only for some changes; the importance of the other changes should be evaluated and assigned by the coordinator himself. Priority levels assigned by coordinator or proposed by ADVICE stick to the EC and aid all the actors taking part throughout ECM to organize their tasks in order to accelerate and improve the process.

As it may be noticed the number of User Entries in Table 1 is quite low compared to overall data requirements. This is managed by providing connections to supplementary tables mentioned. Figure 9 represents these relationships and connections for data requirements.

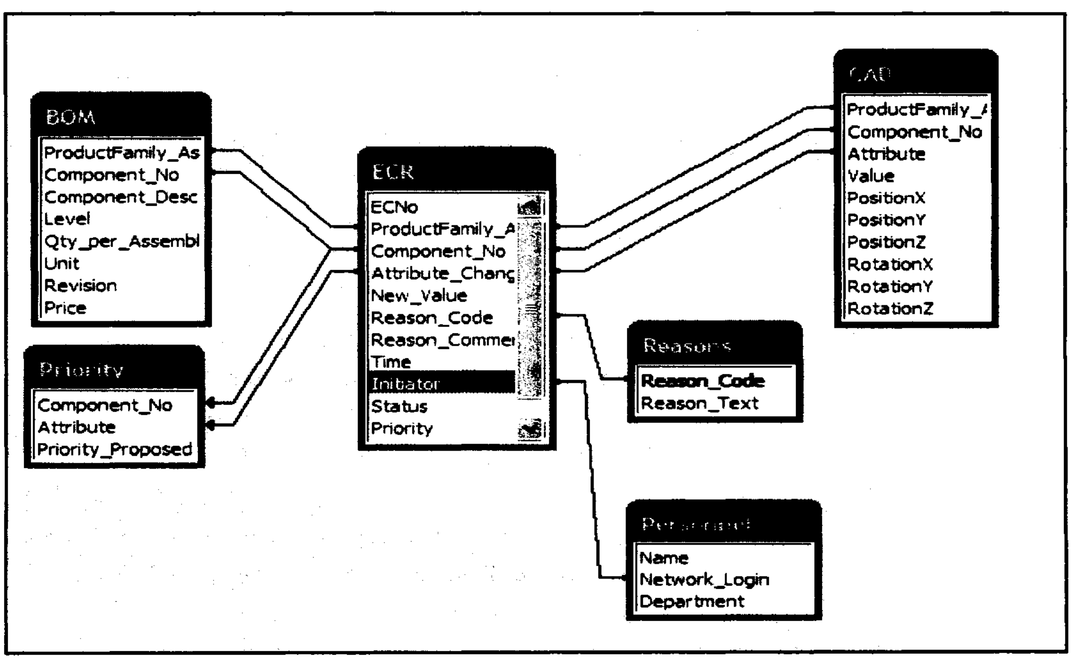


Figure 9 - Entity Relationship diagram for ECR phase

There are two data entry actions during ECR Phase matching to actions in Use-case diagram presented in Figure 7. The first one, realized by the Initiator, is obligatory for starting the ECM process while the other is realized by Coordinator for revising priority if needed and optional. It is the Coordinator that releases the ECR to enter the next phase and changes its status from "R-Request" to "A-Approval". However, the system may also be designed for automatic release to approval according to priorities.

Phase II: Approval Cycle

The data requirements and sources associated with this phase are given below.

Table 2 - Additional Data Requirements for Phase II

Phase II : Approval Cycle – Data Requirements	***Suggested Source***
All data collected in ECR phase (Table 1)	Previous phase
EC No	User Entry
Date released to approval, recorded as "Now" when ECR is released by coordinator	Auto Date
ECB Members, selected by the coordinator	User Entry
Approval / Rejection (Disapproval), by ECB members	User Entry (Yes/No)
Date approved / rejected, specific to each member record	Auto Date
Comment	User Entry
Status on ECR table which has been assigned to A when the EC is released to approval is updated to: o D if any member rejects (disapproves) ; o and to N when there are no members left to review the change	ECR table
Propagated Component, Propagated Attribute, Probability proposed	Change Propagation Table

Similar to priority, ADVICE offers user support by providing the possible change sequences and their associated probabilities by utilizing a data mining mechanism. This mechanism is explained later in Chapter 4. A *Change Propagation* table is added to database structure and connected to EC records to be presented to user during the approval cycle as this may affect their decision. This table includes the following fields: Product model / assembly, Component changed, Attribute changed, Propagated Component, Propagated Attribute, Probability of change propagation. Probability information will be converted to color codes (R G B) for effective presentation with the use of another table (*Probability* table).

It is not certain that all the changes proposed by ADVICE need to be effective subsequently. These suggestions just provide a basis for decision making process. During approval cycle ECB evaluates the propagated changes proposed by the ECM system and enters the changes they expect to be triggered next in Change-next Component and Change-next Attribute fields. As there are a number of ECB members taking part in the approval process simultaneously through distributed workstations, rather than updating the Propagated Component and Propagated Attribute fields directly, we need to record these values in another table (*Change-next*) leaving the original values proposed by ADVICE for revision of other members. These records will be used in the next step to determine the Action Plan. Figure 10 represents the additional requirements of Phase II presented on Entity Relationship diagram.

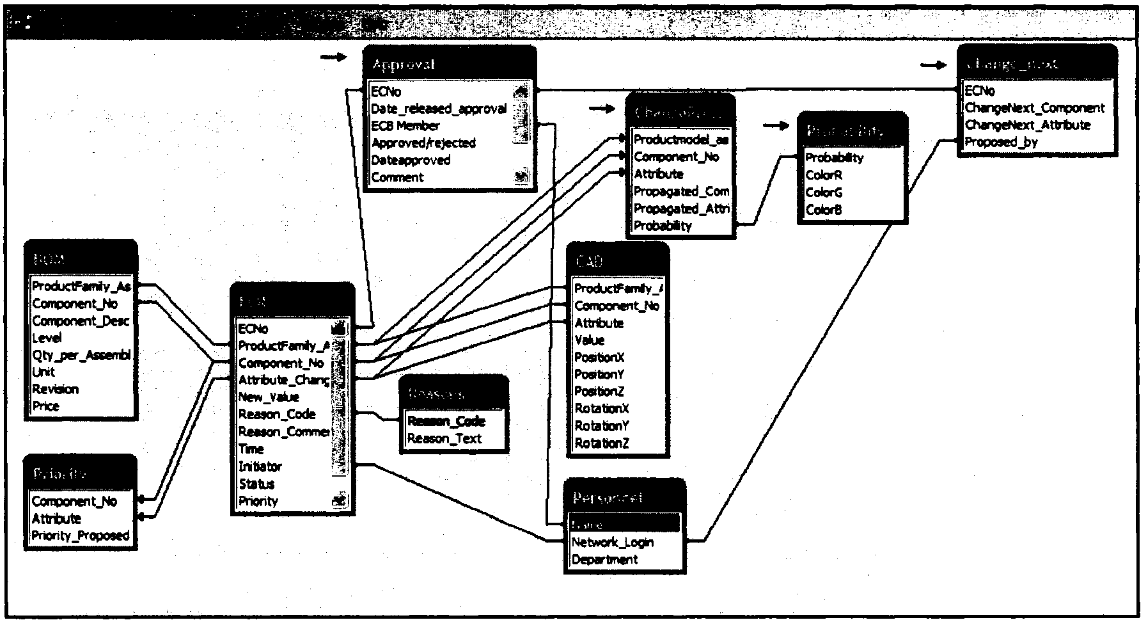


Figure 10 - Entity Relationship diagram for Approval Cycle phase

This phase includes three basic data entry actions. The first one is by coordinator selecting the ECB members, which results in the ECB members to get access to approval interface with their Network Login ID's. The others are performed by ECB members for approving / rejecting the ECR and determining changes to be made next.

Phase III: Engineering Change Notice (ECN)

ECN phase, managed by the coordinator, includes the distribution of the change to related parties. The coordinator basically:

- determines the departments to be informed;
- decides on the date change will be effective;
- prepares the action plan containing changes to be triggered next, as defined by the ECB members in the previous phase and stored in Change-next Component and Change-next Attribute fields;
- releases the ECN by changing the status of the ECR to "O-order"

Table 3 shows the data requirements for this phase.

Table 3 - Additional Data Requirements for Phase III

Phase III : ECN - Data Requirements	***Suggested Source***
All data collected in ECR and Approval cycle (Table 2)	Previous phases
EC No	User Entry
Date released to distribution, recorded automatically	Auto Date
Status on ECR table, set to "O-order" by coordinator	ECR table
Effective Date	User Entry
Departments to be informed	User Entry
Action Plan, containing a sequence of actions to be performed by responsible departments	User Entry

After ECN phase the entity relationships are defined in the following way.

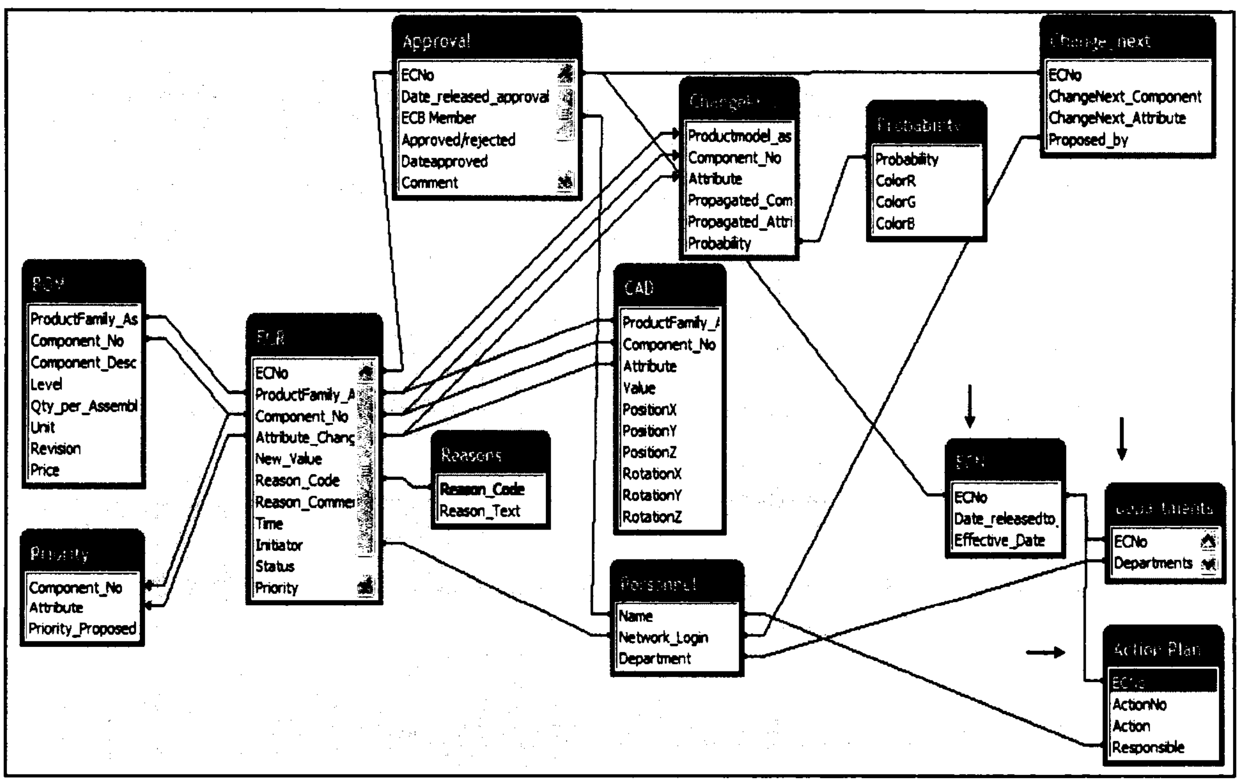


Figure 11 - Database structure after 3 phases

It should be noted that the action plan is not restricted to changes to be triggered next. Each EC may demand a variety of actions assigned to responsible people or departments in respect to its content. These actions are based on comments of the ECB during approval process. An example scenario involving possible actions following an EC is given below with associated departments.

Table 4 - A sample Action Plan

No	*Action*	*Responsible*
1	Overall inventory of the previous version of the item is determined	Warehouse
2	20% of this inventory is released to a rework order to be used in another assembly	Production Planning
3	Rework order processed	Production
4	The rest of the inventory is returned to supplier	Purchasing
5	Purchase Orders and Work Orders in process are cancelled	Production Planning, Purchasing, Production
6	Tooling, equipment and fixtures compatible with the new version are added to production manuals, the incompatible tools, etc. are removed	Production
7	Lead time of the new version is incorporated to ERP system and production schedules are revised accordingly	Production Planning
8	As the production is scheduled-out (overall lead time gets longer), sales orders are postponed in collaboration with customer	Sales
9	Testing and inspection manuals are revised according to new version of the item	Quality
10	The cost of the item overall sales price of the product is revised	Accounting
11	Triggered change requests are placed on ECM system	Related member of ECB
	....	

This example exhibits two basic concepts. First, it illustrates the heavy workload and immense sea of details involved in ECM evidently and clarifies the reason why the process is seen as a burden by most companies and defined as "evil" in literature [18]. Secondly, the content of the Action Plan presents diverse concerns in different industries, different production types, organization structures, etc. Although these actions involve close relationships with already present ERP/MRP structures, it is not easy to incorporate them in a generic ECM model. Besides, it is beyond our research scope to go in such detail as we have defined our focus on supporting the ECM users about prioritizing and triggering changes based on history which already represents an outstanding approach for facilitating the process.

For this reason, we assume action plan in ECN only contains changes to be triggered next for our research purposes. This results in elimination of the *Action Plan* table as it is exactly same as *Change-next* table now. In this way, ECN phase includes querying and reporting the contents of the *ECN*, *Departments*, *Change-next* tables and announcement of this report on the user network. The ECN phase data requirements are limited to determination of effective date and departments by coordinator as the changes triggered are already present from the previous phase.

3.2. MODELING ECM PROCESS IN VIRTUAL ENVIRONMENT

3.2.1. Linking Database Structure to Distributed Virtual Environment

Having established the database structure covering the entire ECM process, the next step is modeling this process in a Virtual Collaborative Design Environment. This model assures synchronization of parametric and graphical data involved in ECs and hence, eliminates the need for duplicate tasks on technical drawing and ECM forms for proposing, evaluating and distributing change knowledge. It provides a medium for generating the corresponding 3D assemblies through inputs from the database structure, as well as storing the keyboard and mouse actions performed on 3D assembly to the database. This is managed by linking the database structure proposed in the previous section to interactive real-time 3D graphics interface modeled in Visual C++ using Open Inventor library. The connection is made through OLEDB Data Provider. The system architecture of ADVICE is presented in Figure 12.

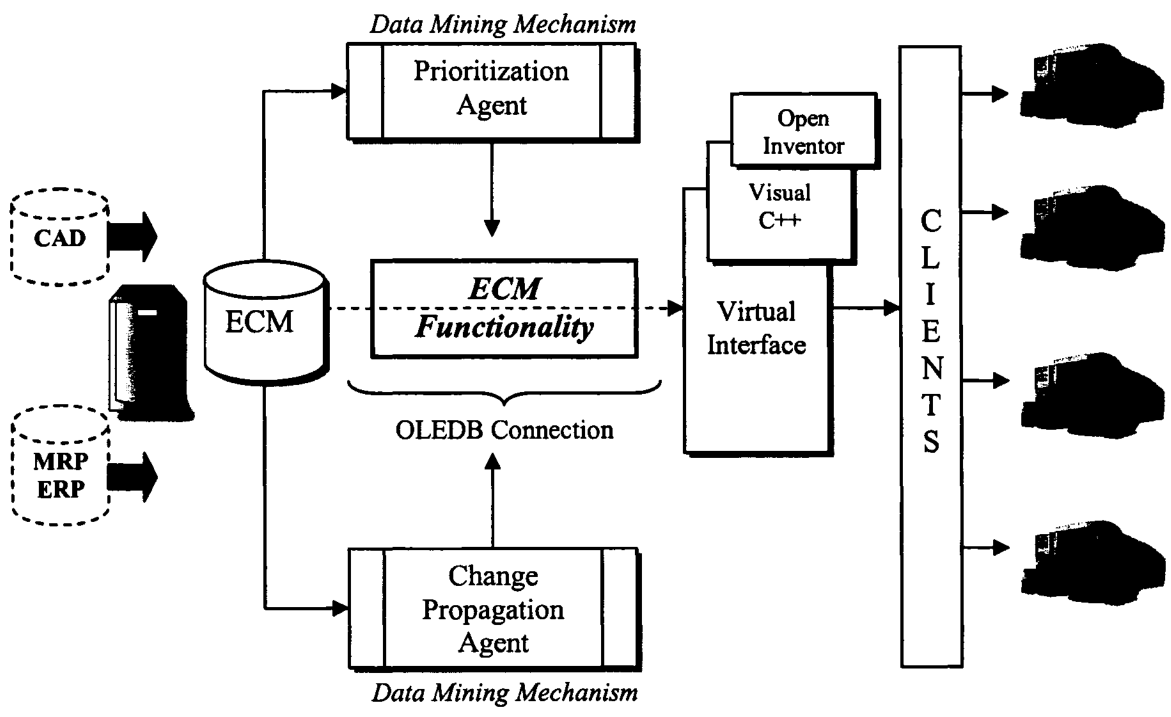


Figure 12 - System Architecture

3.2.2. Processing an Engineering Change in ADVICE

For describing the management of engineering changes in ADVICE, let us consider a product model / assembly X involving *n* components and *m* levels. Each component *i* in product structure is defined with *k* attributes. At the beginning of the ECM process, the components forming the overall assembly have pre-defined values for each attribute showing current revision of the product model. Accordingly, we may express graphical representation of *i* as a function of these attributes as well as translation and rotation vector showing its location and orientation in product structure. In this graphical representation, the attributes such as height, width, depth, radius, shape, material used are expressed explicitly while attributes such as weight, density, tolerances, maximum load, etc. are implicit and stored at the background in relation with the component.

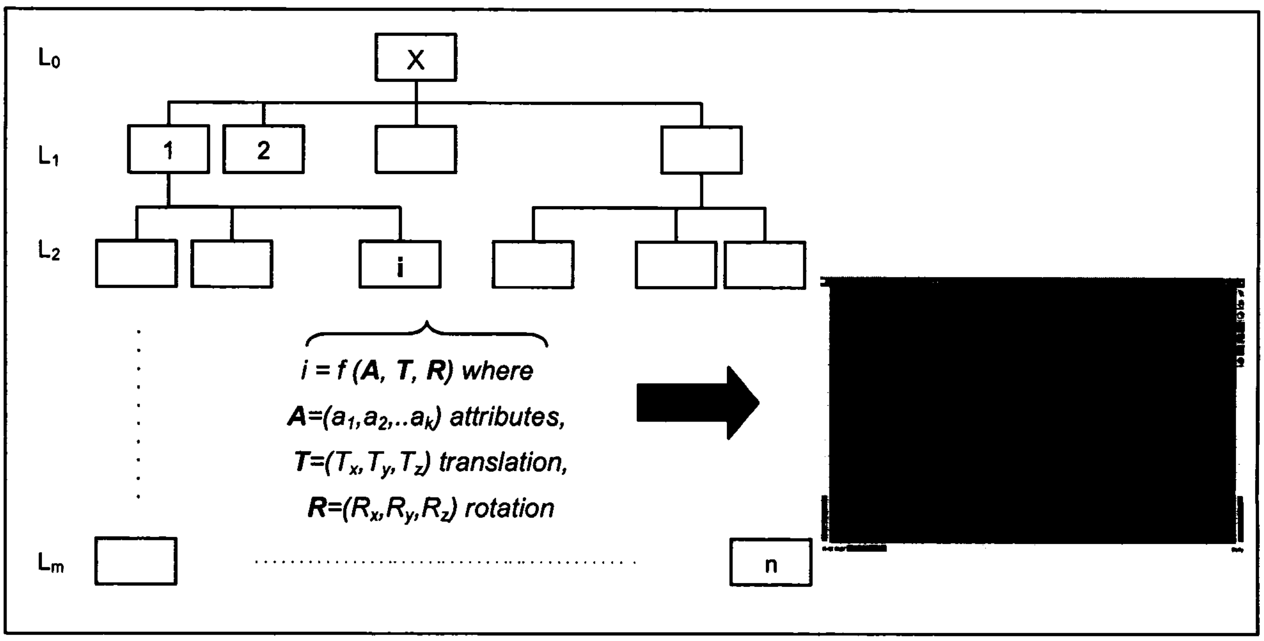


Figure 13 - Product structure and the corresponding graphical representation

For placing an ECR, instead of filling in an ECR form and attaching the current drawing of the component changed, the initiator selects the component directly from the

3D graphical representation supplied by ADVICE using mouse controls and make a choice of attribute to be changed using keyboard. The initiator may demand one or more attribute changes concerning the same component in one transaction. These changes are stored in different records but share the same ECR number.

Having selected the set of attributes to be changed, the initiator proposes new values for these attributes using keyboard control. If the attributes changed are among the explicit attributes represented in 3D model, such as material code, shape or dimensions, the model is updated in real-time to reflect the change proposed. In this way, the initiator might analyze the outcome of the change in ADVICE environment, which provides 360° perspectives improving cognition. ADVICE also offers the capability to try different scenarios about the attribute's value as the database is not updated until confirmation of the change.

The initiator has to state a reason for his request, which is assured by entering the reason code supplied by ADVICE. These reason codes are mapped to textual explanations in *Reasons* table on the database working in the background. In addition to specifying a reason code, initiator may sometimes clarify the reason further or state a comment about the ECR. This is managed by entering textual information that would be stored on the Reason comment field on the ECR table in the database.

When the ECR is confirmed, the *ECR* table is checked automatically to find out if a change on the same component and attribute already exists in the database. There are two types of warning messages supplied to user at this step if there is a match in the *ECR* table.

1. If the status of the matching-change is R-Request, A-Approval or N-Notice, the change is already in-process and the ECR is rejected automatically without being recorded on the database.
2. If the status of the change is O-Order or D-Disapproval, a change on the same attribute has already been offered in the history, but has been re-changed or disapproved. In this case, the initiator is provided the details of the change such as the value assigned to the attribute at the time of the change, time & initiator of the change and comments during Approval / Disapproval process. As the circumstances might have changed and there is possibility that the reasons for disapproval or re-change might no longer be present, there is no automatic action. The initiator uses his/her judgement to re-confirm the change and it is stored in *ECR* table.

Next step is the prioritization of the ECRs confirmed and stored in the database. This is managed by coordinator using "Release-to-Approval" form, which contains ECRs in descending order of date and the corresponding priorities retrieved from *Priority* table filled by the data mining mechanism that will be explained in Chapter 4. If the *Priority* table does not include record about the component and attribute in question, the coordinator assigns a value A, B or C; otherwise he/she may revise the value proposed by the mechanism. Using the same form the coordinator selects and releases the ECRs to approval choosing the members of the ECB in a sub-form.

Each member of the ECB gets access to ECRs in his/her responsibility listed in ascending order of priority and time. Selecting the change to be approved, ECB members are provided the graphical representation of the change in ADVICE, containing two 3D models for current and proposed configurations respectively.

- On the current configuration, component to be changed is highlighted ensuring that non-technical members can easily comprehend the scope of the ECR at first sight. The users may zoom in, out and rotate the 3D model for analyzing it in detail.
- The proposed configuration on the same window offers the change propagation information, which is one of the most distinguishing features of ADVICE. This information reflects the outcome of the data mining procedure (Chapter 4) on change history; the possible changes that could be triggered following the proposed change and the corresponding probabilities. For supplying this information to users in an effective manner, the probabilities are mapped to color codes defined in the database. The components having highest probability to be changed due to ECR in process are coloured "red" getting quick attention; the components with moderate probability to be changed are coloured "yellow" and components with the least probability of being changed are coloured "green". The rest of the components stay in original their colors. The colors Red, Yellow and Green are selected in analogy with traffic lights. Obviously, the basic assumption in this selection is that there are no components whose original color is red, yellow or green. Other colours may be used for varying needs and characteristics of different product structures. Another possibility is utilizing highlighting and

boxing the components effected with different colors preserving the original color. In short, the method adopted in this work, i.e. colouring the affected components in red-yellow-green is not the only way of presenting change propagation information; for different product structures, different methods could be applied.

Selecting the coloured components, the users may reach detailed information about the change proposed by Change Propagation mechanism including the probability level and the attribute(s) propagated in history. If there are more than one change sequences concerning different attributes of the same component, the one with the highest probability is reflected on ADVICE.

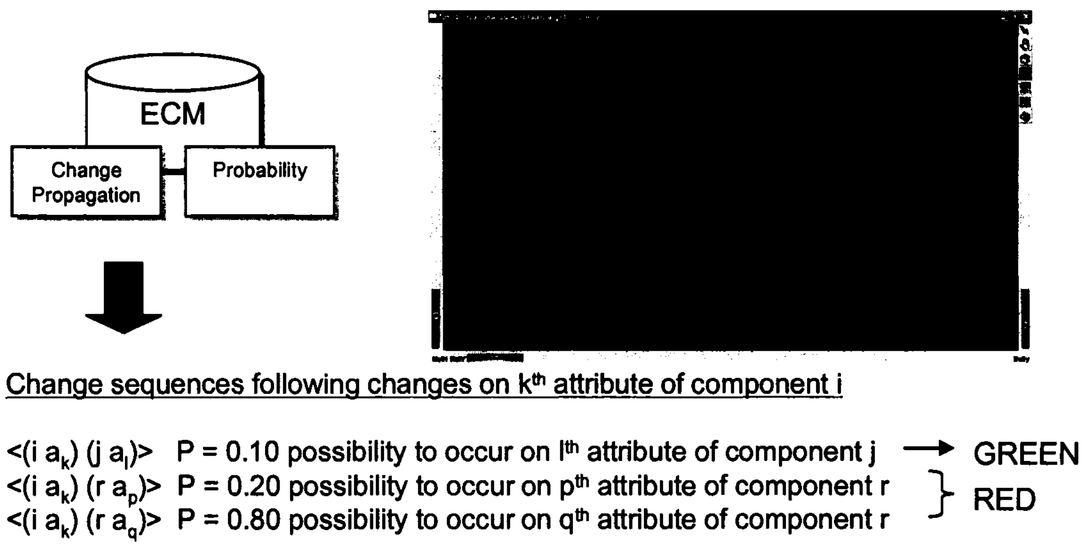


Figure 14 - Use of colors and highlighting in ADVICE

ECB members take the effects of changes on the other components into account while evaluating the change request. However, there is another aspect that should be considered. The fact that a change has triggered some others frequently in the history of changes and proposed by the system does not necessarily mean they should be included in the action plan and performed after the ECR. Moreover, there could be effects on components that are not proposed by the Change Propagation mechanism. ECB members should use their experience and judgement to decide on the changes that should be included in the action plan. This is ensured by selecting the components in the proposed configuration and choosing the attribute that needs to be included in the action plan by storing information on Change-next Component and Change-next Attribute fields in *Change-next* table in the database. As the approval process is a simultaneous process carried out by all ECB members concurrently, this should be done in a controlled manner, checking the values on the fields above and recording only the values that haven't been offered by any other member.

During this concurrent process, if any member rejects the change stating a comment, the status is changed to D-Disapproval and the change is returned to coordinator. Otherwise, the status is updated to N-Notice when the ECR is approved by all members. For these ECs the coordinator fills in the ECN form determining departments to be informed and the effective date. The entries in the form forms the basis for a report shared on the network with the personnel in the departments selected. An e-mail notification system might also be established for sharing ECN with related parties. The coordinator updates the BOM, CAD records when the ECN is distributed to departments.

To summarize, ADVICE brings an exceptional approach to Engineering Change Management process by offering decision support features that have not been offered by any other system in literature or by any commercial products. These features are, in general, based on two characteristics of the system:

- merging parametric & graphical data representations in one product model and presenting them to user in an effective manner to ensure fast and accurate comprehension throughout ECM;
- proposing estimates on important decision variables such as priority and change propagation based on historical data to facilitate decision making involved in ECM process

3.3. A SAMPLE APPLICATION ON ADVICE

In this thesis we did not have chance to work with an industrial partner. For this reason, database records and assembly structures used for explaining the proposed environment's working principles and characteristics, are not based on existing products; rather they are designed by us for demonstration purposes. However, as we have built a generic model for the Virtual Environment, the system may be adapted to involve a real-life application with some changes.

The assembly structure used for demonstrating features of the proposed system ADVICE is an office table with two shelves having 24 components in two BOM levels (Figure 15). For this imaginary product, the supplementary *BOM*, *CAD* and *Personnel* tables are created manually and the database structure is established. In general, these tables could be imported from existing records on CAD and MRP/ERP packages of a company.

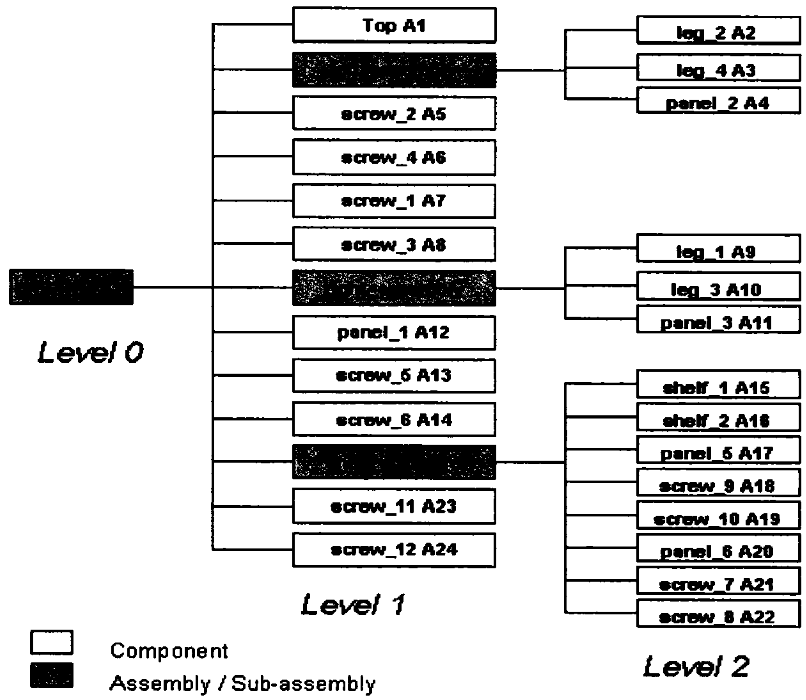


Figure 15 - Sample assembly product structure

For modeling the database structure represented in system architecture (Figure 12), we used Microsoft Access to build a desktop database, which was sufficient for the small-scaled sample we designed. However, increasing record size and the need for sharing data with clients simultaneously in a distributed environment necessitates using SQL server, Oracle, IBM-DB2, MySQL or any other ODBC-compliant data container rather than a desktop application. Fortunately enough MS Access allows connection with these data hosts for data storage and retrieval, which means the interface we designed, may be used even if storage medium is changed. Another method could be using ASP (Active Server Pages) and VRML (Virtual Reality Modeling Language) to provide data flow through internet server and design an interface to present information on web pages rather than queries, forms and reports offered by MS Access.

In the prototype involving the table assembly, we limited the attributes to be changed to material, shape and dimension changes (height, width, depth and radius). We selected these attributes because they can be expressed explicitly on the interface, allowing the user to get a quick glance on what is being changed. Among them, material is expressed in terms of spectral (color) properties. Although there could be a variety of material codes defining an office table, for demonstration purposes, three basic materials and the corresponding color codes are included in the prototype, namely "wood", "metal" and "plastic". Similarly, the shapes are limited to cubic or cylindrical shapes, as the sample assembly can be expressed in terms of these two shapes.

When the initiator enters the Virtual Environment ADVICE connected to ECM database to place an ECR, the system offers a start-up menu as in Figure 16. The menu explains the instructions for placing an Engineering Change Request.

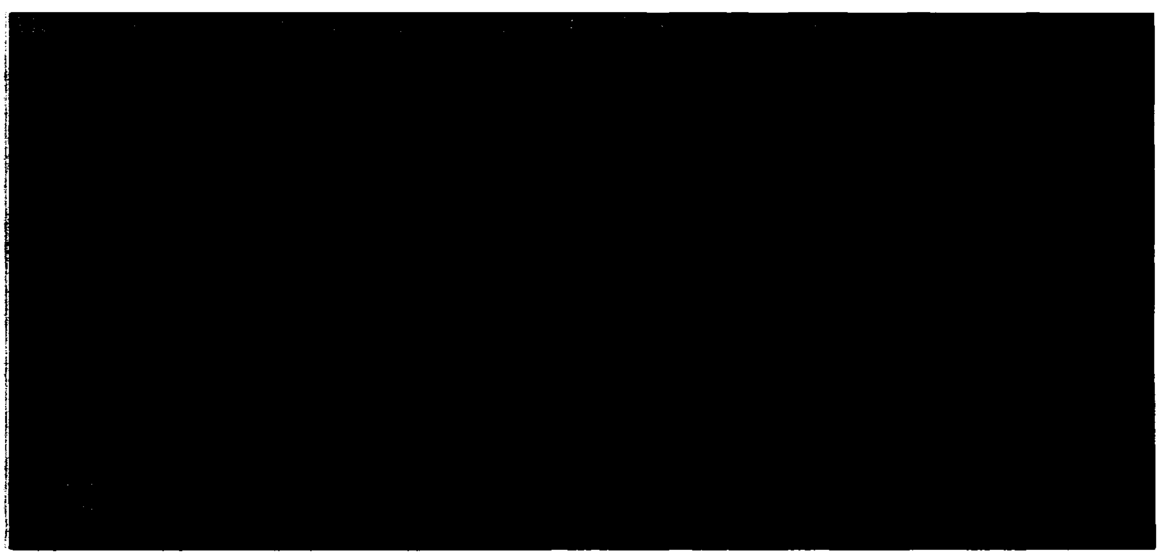


Figure 16 - Start-up menu in ADVICE

According to these instructions, the first step is selecting the component to be changed with the mouse; when this is done the selected component is highlighted in 3D environment. Next, attribute to be changed and the new value of the attribute is entered to ADVICE using keyboard controls mapped to different actions on the background. If the selected attribute is material or shape, pre-defined material or shape codes are used. For dimension changes, the initiator enters a value, which is recorded by the system.

The initiator may demand one or more attribute changes on the same component. ADVICE automatically updates and renders the 3D image when the changes are made. Figure 17 illustrates the interface for a sample change of height for the top part of the assembly.

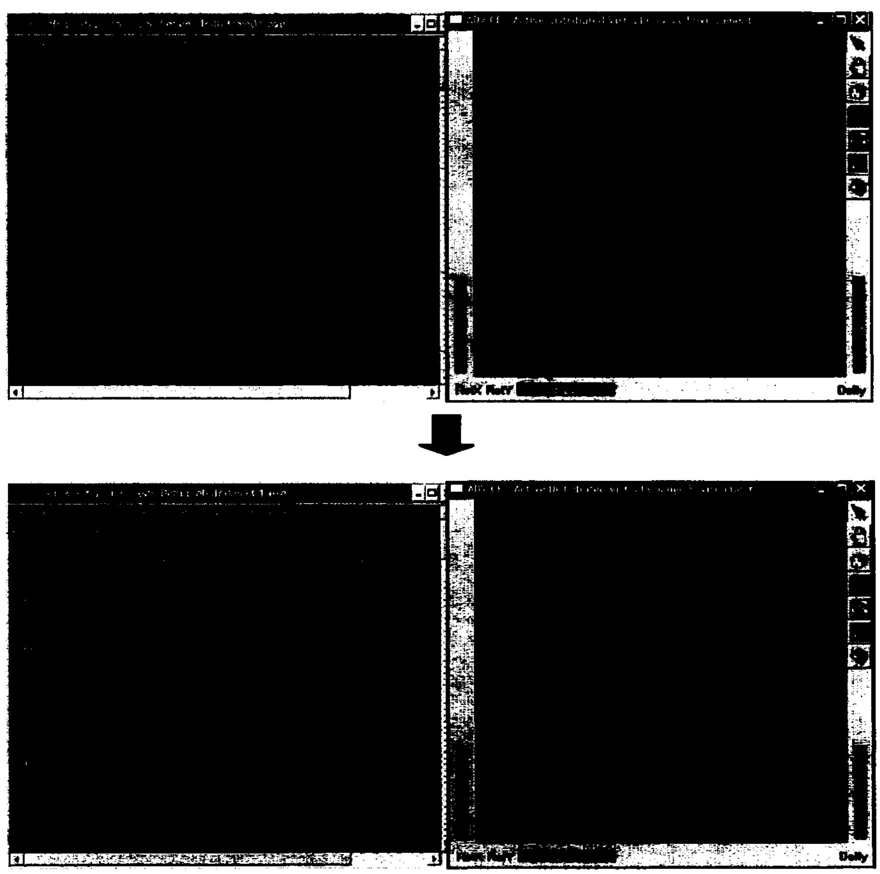


Figure 17 - Illustration of ECR on ADVICE

Next, the initiator states a reason code for requesting the change. The reason codes, which are mapped to textual explanations in the database, may differ across companies. For our sample case, we specified more general reasons as given in the table below [59].

Table 5 - Reasons Table

Reason_Code	Reason_Text
01	Correct drawing or engineering document error
02	Correct a usability, reliability or safety problem
03	Fix a bug or product defect
04	Improve performance and/or functionality
05	Improve manufacturability
06	Lower cost
07	Incorporate new customer requirements
08	Specify a new supplier or supplier part/material
09	Enhance installation, service, or maintenance
10	Respond to regulatory requirements
11	Non-conformance to inspection standards
12	Other (Please specify)

Record: 12 of 12

The ECR is checked in the database to find out if it has been suggested before. In case of "NO MATCH", the initiator is requested to confirm the change request (Figure 18).

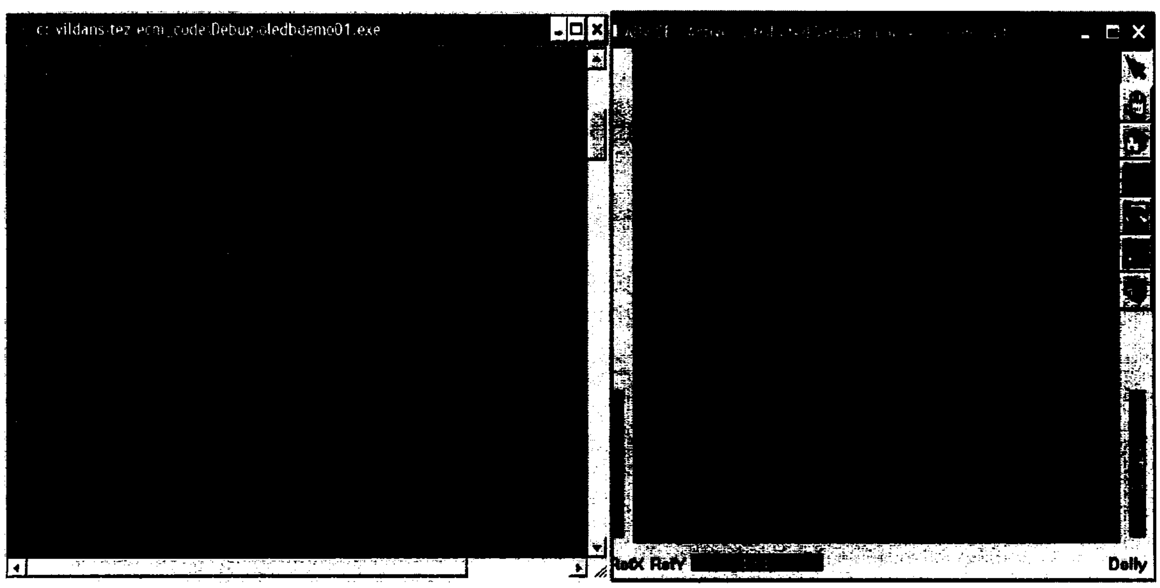


Figure 18 - Statement of reason and confirmation of ECR

As soon as the ECR is confirmed, it is stored on the database through the OLEDB connection. The coordinator uses an Access form to list, prioritize and release these

ECRs (Figure 19). This form involves the priority values suggested by ADVICE's Prioritization Agent as default. These values are based on frequency of change sequences among product families in history. If a change event triggers others frequently in many product families, then that change's impact is defined to be superior to others as the lead time required for performing the change increases concerning the triggered changes. However, the coordinator may use his experience and common sense to revise the priority. The same form is used to select the ECR, define ECB and release the ECR to approval.

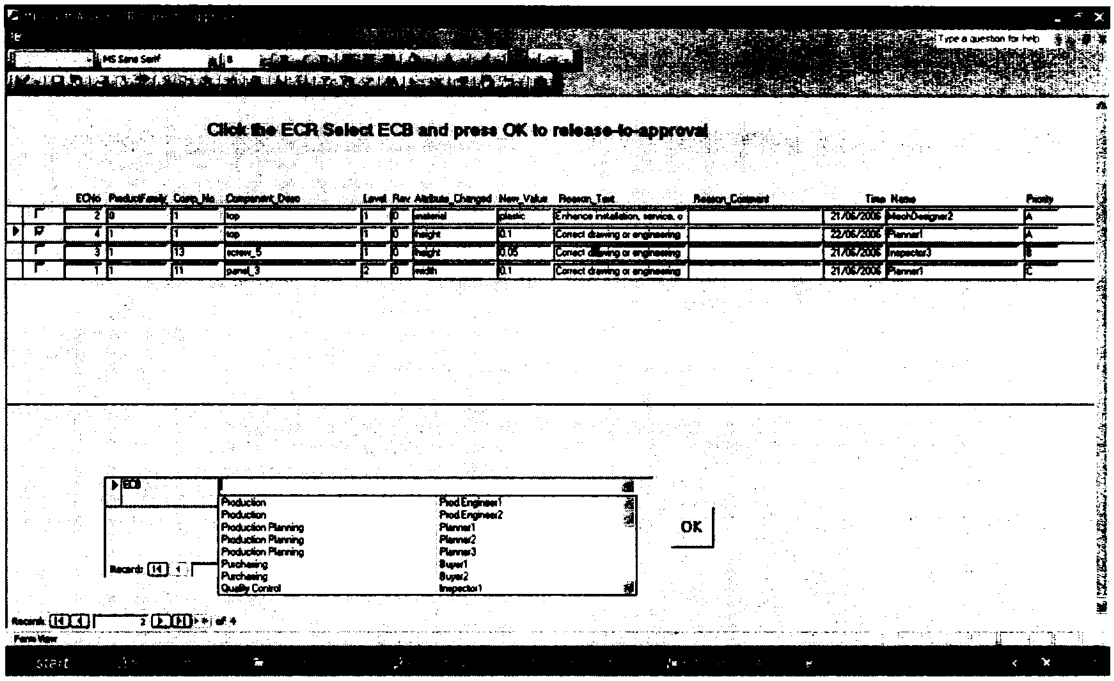


Figure 19 - Release-to-approval form

ECB members view the changes to be approved by themselves on an Access form connected to 3D interface (Figure 20). They have to consider a number of aspects before approving the change as stated before; inventory levels, scrap-rework decisions, compatibility with machines and testing equipment, lead-time and cost of the revised component, etc. The information needed by ECB could be incorporated to ADVICE with links to ERP system. However, we left them out of scope for our sample application and concentrated more on another important aspect that needs to be evaluated by the ECB members: the propagation of the change to other components.

ADVICE provides user support by presenting the change's possible effects in 3D model with colors. For finding out possible effects, the frequency of patterns including the change is investigated and associated with a probability value. This value is converted to color codes and offered to user. As the interface involves two models showing current and proposed versions in the same environment at this step, losing the material information on this coloured model does not constitute a problem. The following table shows the color codes we have used in our application. The color codes may be diversified according to needs of different applications.

Table 6 - Probability - Color Mapping

PROBABILITY	***COLOR***
$P<0.15$	Green (0,255,0)
$0.15 \leq P<0.60$	Yellow (255,255,0)
$P \geq 0.60$	Red (255,0,0)

Apart from approving or rejecting the ECR, ECB members have to state the components to be changed taking the suggestions of ADVICE into consideration. They simply select the components and include them to action plan with attributes expected to be changed according to their experience (Figure 20).

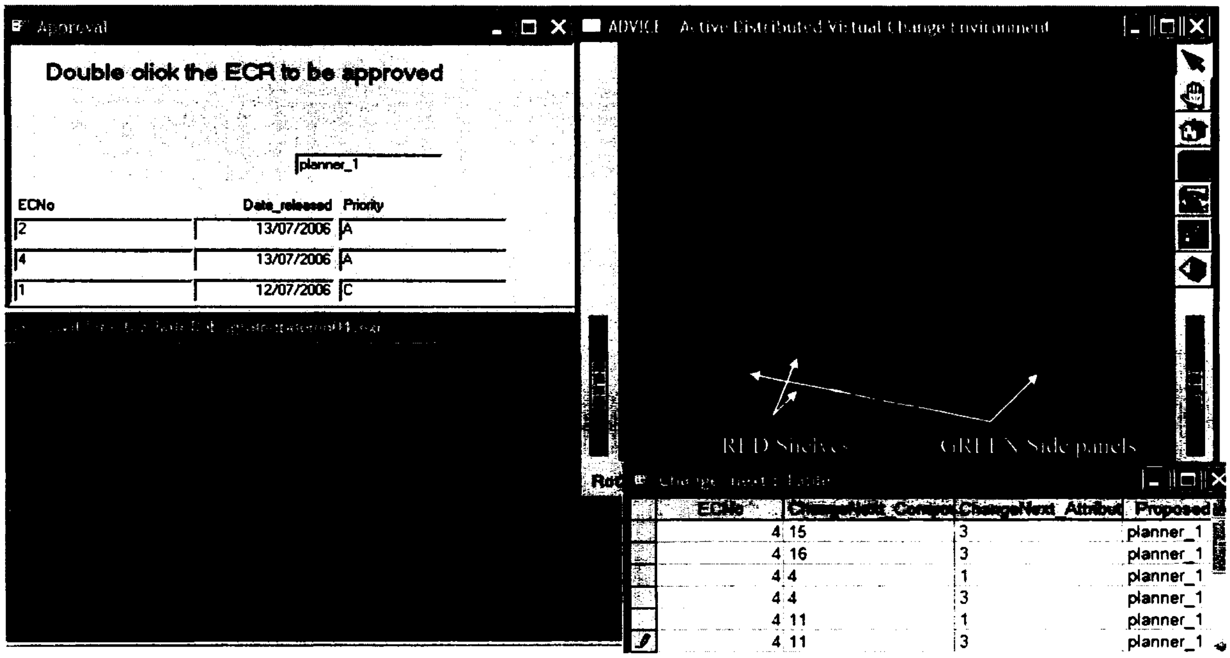


Figure 20 - Approval interface showing propagated changes as offered by ADVICE

Even though the implementation of ADVICE is demonstrated on a simple assembly containing a relatively small BOM, it is not hard to imagine how approval interface might facilitate and accelerate the process in a complex assembly structure by pointing out the affected components at first sight using colors. As there exists inner components in assembly structures in general, features such as transparency might be included to allow the coloured components to be apparent. If there is a need to diversify the probability ranges a color spectrum with increasing levels of intensity might be employed.

The change might have caused propagation in other product families as well, which has been revealed by the data mining mechanism used in Prioritization. Although we have not included the propagation across product families in our sample interface, it is an important part of ECM to make changes on other product families / assemblies.

For informing departments about the change in ECN step, ADVICE offers an ECN form filled in by the coordinator. In this form the coordinator enters the EC No and the action plan data is retrieved and reflected on the form. Accordingly he selects the departments to be informed and sets an effective date (Figure 21).

Announcement may be paper-based or electronic. But to enhance the comprehension it is better if the departments informed get access to ADVICE interface and review the previous and current versions of 3D model. Among the people getting informed, the responsible staff included in action plan (members of ECB that have suggested inclusion of triggered changes to this plan) may be distinguished through their login ID's. For them ADVICE may offer an interface pretty like the one on approval cycle on Figure 20 and present the triggered changes they have to place in the system using highlighting or colors.

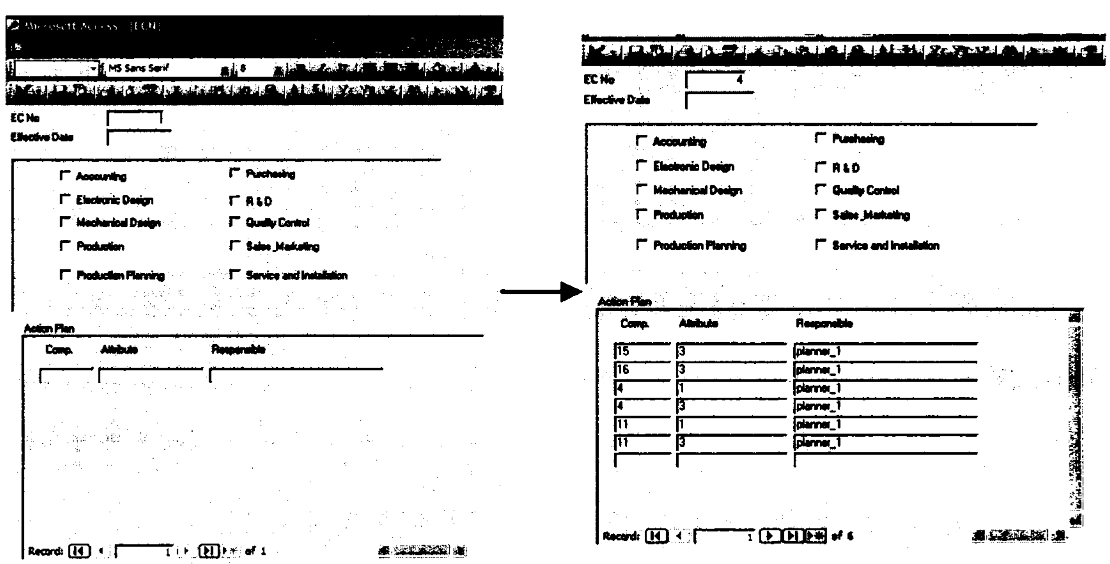


Figure 21 - ECN form filled in by coordinator

3.4. SUMMARY

In this chapter we searched answer for the first research question we have posed. *"Can we expedite and improve Engineering Change Management process by employing the Virtual Environments for presenting information more effectively?"*

We built a database model and corresponding Virtual Environment structure. This Virtual Environment named as ADVICE;

- combines graphical and parametric information involved in ECM process and represents them as a Virtual assembly model on its interface;
- proposes a tool for managing workflow and information flow in ECM process eliminating duplicated tasks;
- enhances comprehension of users with the use of interactive interface;
- provides active decision support for prioritization of requests and for determination of change propagation.

CHAPTER 4

MECHANISM FOR PROVIDING ACTIVE SUPPORT

Up to this point, we have explained the model proposed for processing engineering changes in a **D**istributed **V**irtual **C**hange **E**nvironment. In this chapter, we will give detailed explanation about the "Active" decision support and "*advice*"s offered to users by utilizing Prioritization and Change Propagation agents.

The chapter is organized in four sections; the first section presents our effort for describing a mechanism for assigning an initial priority to a change request. In the second section a methodology for predicting change propagation is explained. Finally, in Section 3 and 4, we conduct experiments to verify our methodology and discuss the results.

The main idea beneath the mechanism suggested in this chapter is utilizing the data captured and stored during ECM process to direct and facilitate the process itself. In this way, proposed environment ADVICE goes beyond the definition of a distributed virtual environment and acquires the ability to offer "smart" means of performing tasks involved in ECM process. This is accomplished by applying data mining techniques briefly explained in Literature Review (Section 2.3.) to obtain meaningful patterns showing behaviour of the change data.

4.1. ASSIGNING AN INITIAL PRIORITY FOR A CHANGE REQUEST

Priority of an EC is an indicator showing its importance among other changes in terms of its impact. The impact of an EC is a measure of the amount of design work that needs to be re-done if the change propagates [23]. An EC having a large impact covering most of the product models in a company's product range, should be processed in the first place. The cycle time for processing such an engineering change should be considered as a composite of the time required to evaluate and process the propagated changes in all product models concerning the changed component.

Consequently, we may define priority of a change as a function of its frequency of triggering change patterns in various product models.

4.1.1. Prioritization Agent

Prioritization Agent illustrated on System Architecture in Figure 12, takes in historical change data as input; applies data mining procedure to extract sequential patterns that are frequent among the product models in range and converts this result to "A", "B", "C" codes defining priority.

The inputs for this process are the change transactions recorded in ECM database containing product model, component and attribute(s) changed. We have already mentioned that each of these transactions may involve more than one attribute of the same component. It is possible that the user selects a component and changes a set of attributes regarding this component. Another important variable is the time the process is terminated by releasing ECN to departments. Although the transactions are actually recorded prior to that time during ECR phase, an ECR may be rejected at any point of approval cycle so there is no point in dealing with rejected change transactions.

As a result, the change transactions are expressed in the format < M_i $(C_{ij} \ldots C_{ik})$ t_i >

where M_i: model on which change transaction is performed,

C_{ij}: variable stating the change of j^{th} attribute of component i,

t_i: transaction time of change notice release for component i.

In this dataset, attribute changes that belong to the same item in each transaction are interchangeable. In other words, there is no difference between transactions (C_{12} C_{13}) and (C_{13} C_{12}) as they both occur at the same time. In the first transaction, the user selects component 1 and changes 2nd and 3rd attributes while on the second he selects component 1 and changes 3rd and 2nd attribute.

For solving the problem of extracting sequential patterns which are frequent among the product models, we established an analogy with AprioriAll algorithm [47]. As discussed in literature review section, the algorithm provides the most appropriate methodology for our purposes, presenting a more general and fundamental approach among other procedures for mining sequential patterns. The aim of AprioriAll is analyzing customer transactions in retail industry to come up with sequential patterns of purchase transactions which are frequent among customers. The next section involves implementation of this concept to our case of change sequences by replacing customers with product models and purchase transactions by engineering change transactions.

4.1.2. APRIORIALL Algorithm

Before passing to implementation of 5-step AprioriAll algorithm, let us review the terminology concerning Apriori-based approaches giving some basic definitions.

- **Itemset:** Collection of events occurring at the same time and in the same transaction. For our case, itemset $s_i = (C_{ij} \ldots C_{ik})$ is composed of engineering change events occurring at time t_i.

- **Sequence:** Collection of itemsets ordered in terms of transaction time. $S_i = \langle s_1 s_2 \ldots\ldots s_n \rangle$. A sequence involves all ordered transactions regardless of the product model.

- **Sub-sequence:** A sequence contained in another sequence without disrupting the order of events. $\alpha = \langle a_1 \quad a_2 \quad \cdots \quad a_n \rangle$ is defined as a sub-sequence of $\beta = \langle b_1 \quad b_2 \quad \cdots \quad b_m \rangle$, denoted as $\alpha \subseteq \beta$, if there exist integers $1 \le \theta_1 < \theta_2 < \cdots < \theta_n \le m$ such that $a_1 \subseteq b_{\theta_1}$, $a_2 \subseteq b_{\theta_2}, \ldots, a_n \subseteq b_{\theta_n}$.

- **Support of a sequence:** percentage of models including that sequence, denoted as *supp(S_i)*.

- **Large sequences:** Set of sub-sequences having support greater than a user-specified minimum support. Large sequences are denoted with L_k where k is the number of itemsets that the sequence contains. Similarly, a **k-sequence** stands for a sequence containing k itemsets.

- **Litemsets**: itemsets or subsets of itemsets having minimum support, in other words the most frequent changes among product models.

- **Maximal** large sequence: A large sequence which is not contained in another large sequence.

The objective of the AprioriAll algorithm is mining sequential patterns in a database $\mathcal{D}$ by finding the Maximal sequences among all the large sequences having a pre-defined minimum support. The algorithm involves 5 steps:

(i) SORT: All the transactions in database are sorted in the ascending order of model and time and model sequences composed of itemsets are obtained.

(ii) LITEMSET: Potential Litemsets are determined and searched in each model sequence to find out if they are above minimum support. These litemsets are used to create large sequences among which the resulting patterns are searched. In this step, the algorithm limits search space to sequences including the most frequent events.

(iii)TRANSFORM: Litemsets are mapped into consecutive integers and the original database $\mathcal{D}$ is transformed to $\mathcal{D}_T$ by expressing each model sequence in terms of litemsets.

(iv)SEQUENCE: Starting with 1-sequence, sub-sequences are created and searched within model sequences. For creating sub-sequences, a level-wise approach is used; at each level (iteration) k the candidates for next level (C_{k+1}) are formed by joining the members of the current level (L_k's).

The candidates generated are controlled in terms of two aspects. If the sub-sequences of candidates are not already selected as a large sequence in the previous step or if the candidate is below minimum support; then it is eliminated and removed from the candidate set. This step involves the core of the algorithm and results in a set containing all the large sequences generated in each step.

(v) MAXIMAL: The set of large sequences found in the previous step are analyzed and the sequences that are already contained in a larger sequence are eliminated to avoid redundancy. The elimination procedure starts from the largest sequence(s) in the set. In this way, patterns supported by sufficiently large number of models are found.

More information about the algorithm and pseudo-code could be obtained in [47]. A basic flowchart we have developed and used while coding the algorithm for conducting experiments is given in Figure 22.

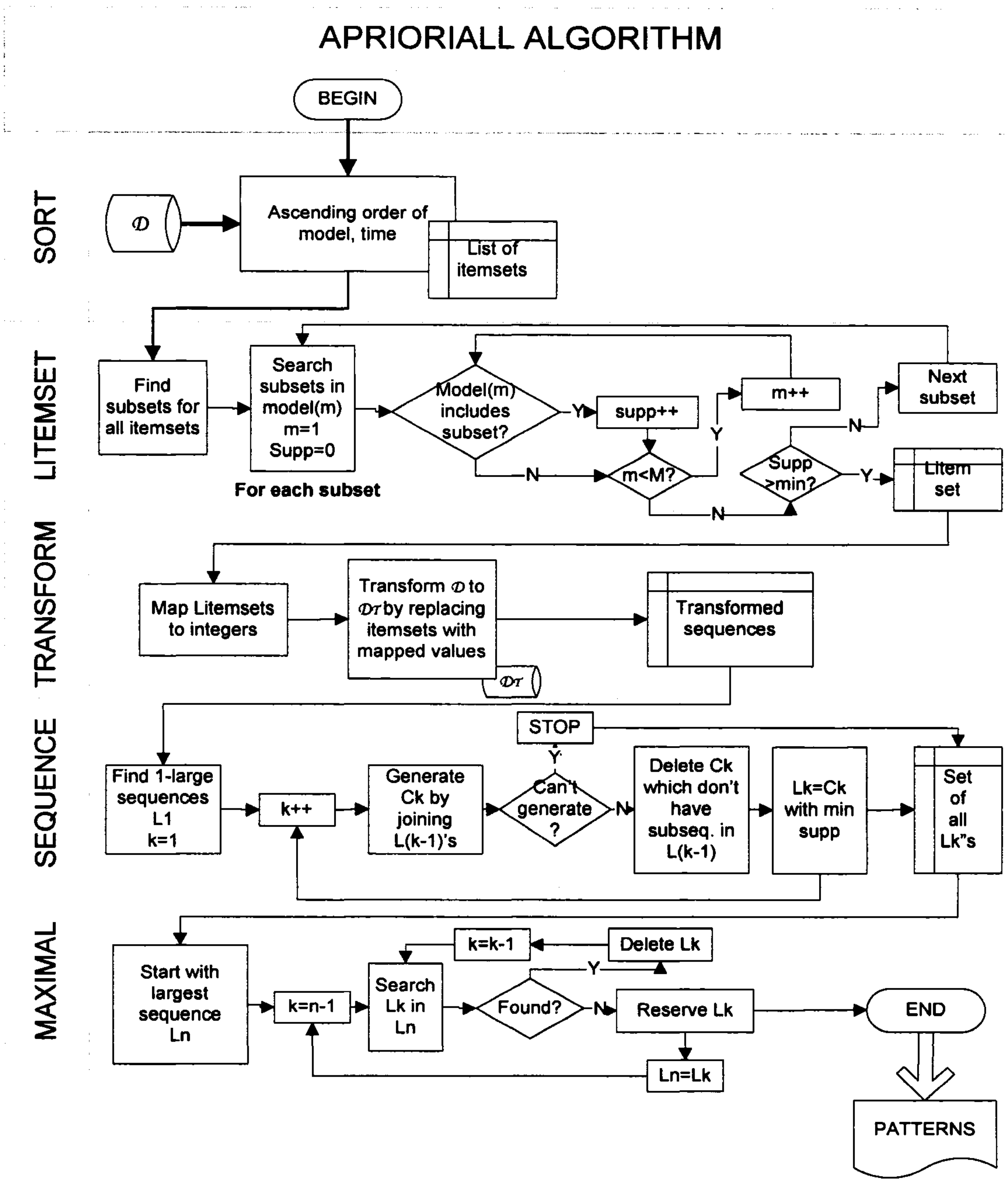


Figure 22 - Flowchart for implementing AprioriAll algorithm

4.1.3. Converting resulting sequences to priority codes

The output of AprioriAll algorithm is change patterns that have been repeated a minimum number of times in all product models. For converting them to priority codes, the following procedure is proposed.

For each C_{ij} in the resulting patterns, we calculate number of unique events following C_{ij} that belong to components other than i and multiply this number with the support of each pattern. In other words, we assign more weight to engineering changes that serve as activators starting the "domino effect" of changes in most of the product models. Moreover, we have to normalize this weight using the support of resulting patterns already offered by AprioriAll algorithm.

$$f_P(C_{ij}) = \sum_p \lfloor \mathrm{NTr}(C_{ij}, p) * \mathrm{supp}(p) \rfloor$$

Where $f_P(C_{ij})$: Priority Index of C_{ij}

p: patterns involving C_{ij},

supp(p): support of pattern p,

NTr(C_{ij},p): number of unique events triggered by C_{ij} in pattern p

We use this resulting Priority Index, $f_P(C_{ij})$ for each engineering change C_{ij} to find out to which priority group the change belongs. As stated earlier, there are 3 priority groups:

- "A": The most important change requests that should be processed urgently;

- "B": less urgent yet important change requests that should be scheduled accordingly;
- "C": the change requests that could be batched

In order to classify the overall set of frequent events in resulting patterns to these 3 priority groups, we use two threshold values for separating the classes, namely Th_A and Th_B. Using these pre-defined threshold values and the priority index, we apply the following rules to determine priority.

IF $f_P(C_{ij}) > Th_A$ **THEN** Priority: "A"

IF $Th_B \leq f_P(C_{ij}) < Th_A$ **THEN** Priority: "B"

IF $f_P(C_{ij}) \leq Th_B$ **THEN** Priority: "C"

(i) If the Priority index is above Th_A, we say C_{ij} needs to be performed as fast as possible as it has impact on many components and various models. We update *Priority* table in ECM database appending a record for C_{ij} with priority "A" or changing the existing value.

(ii) If the number is between Th_A and Th_B, we give C_{ij} less priority and update the *Priority* table by assigning priority "B" to component i and attribute j couple.

(iii) For the rest, we assign priority "C" and append records to *Priority* table.

Similar classifications are common in inventory management for prioritizing items according to their cost or in project management for ordering activities in project workflows. In some cases ABC analysis is discussed in relation to Pareto analysis and the elements are classified by assigning 5% of ranked items to "A", 15% of them to "B" and the rest (80%) to "C". It is also possible to fit a normal distribution and decide on the threshold values accordingly. The determination of threshold values depends on the context. In this study, we used mean and half-mean for setting Th_A and Th_B. As a result, after finding the priority index for all component-attribute couples, we ranked them and assigned "A" to the couples larger than mean; "B" to couples larger than half of the mean. The rest of the changes are set to "C" priorities.

There are two more issues which should be determined about Prioritization Agent. These are *when* to actuate this mechanism and *what* to include. Ideally, Prioritization Agent should function whenever a change notice record enters the database updating priority table based on all of the previous changes up to that point. However, this may not be practical in some cases considering compilation time. For providing a practical solution to this problem, prioritization agent may be scheduled to operate during the night hours every business day, when the user groups are absent and the database is idle. The records are analyzed and updated before the users start using ADVICE. Moreover, the changes taken into consideration may be limited setting a minimum time constraint. In this way, far history is not taken into account and the results are more reliable. These two issues apply to Change Propagation agent, which will be discussed in the next section.

4.2. PREDICTING TRIGGERED CHANGES

For predicting the triggered changes on the same product model a data mining approach based on history of changes is utilized again. This time rather than frequency of change patterns among the models, patterns repeating within the same model are investigated. This process is handled by Change Propagation agent defined in system architecture in Figure 12.

4.2.1. Change Propagation Agent

Similar to Prioritization Agent, Change Propagation agent uses the data collected and stored in ECM database as input; processes the data to find out change patterns within each product model and returns the resulting probability value for each attribute change to serve as a basis for triggered changes.

The changes do not need to be consecutive to form a sequence. On contrary, most of the time it is possible that they are separated by irrelevant transactions. For this reason, we decided to use an episode based approach. An episode is ordered collection of events occurring together within a time frame. The objective is investigating the events following each other within a pre-defined time limit, i.e. time window.

There are two approaches proposed for solving this problem [49], which were briefly explained in Literature review. Among these approaches MINEPI is selected to be more appropriate for our purposes, as WINEPI which suggests using a sliding time window may cause problems if our dataset involves a change twice or more within the pre-defined time window. The implementation of MINEPI algorithm to engineering change sequences, which was originally proposed for alarm sequences, is discussed in the next section.

4.2.2. MINEPI Algorithm

In MINEPI algorithm, an event sequence s is defined within an interval limited by start and end times, (s, T_S, T_E); where $s = \langle (A_1, t_1)(A_2, t_2)\ldots(A_n, t_n)\rangle$, A_i's defining the events and $T_S \le t_i < T_E$ for $\forall i$=1,…,n.

Patterns are defined as frequent episodes occurring sufficiently large number of times in the event sequence s defined within $[T_S, T_E)$ interval. For adopting the algorithm to our case, sequence s is restricted to a model sequence and each event A_i is defined to be an attribute change on a component. When there are more than one attribute changes occurring in a component at the same time they are described as concurrent events. In the example sequence shown in the figure below, C_{12} and C_{13} are concurrent (parallel) events while C_{12} and C_{23} are serial events. The episodes represented in figure show a pattern < (C_{12}) (C_{23}) (C_{41}) > within the pre-defined time window; our goal is revealing these patterns given the sequence and window size.

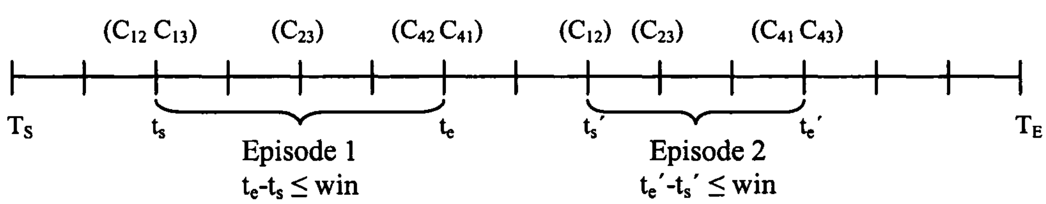


Figure 23 - Sample sequence demonstrating episodes

The algorithm functions by recording the start and end times for each candidate episode and eliminating them if they are beyond the pre-defined window size limits. The intervals limited by start and end times of a potential episode are called *minimal* *occurrences for* that episode. The set of all minimal occurrences for a given episode α is denoted with $mo(\alpha)$ and is composed of $[t_s\ t_e]$ intervals containing that episode.

$mo(\alpha)=\{[t_s\ t_e] \mid t_e - t_s \leq win\}$ where *win* is pre-defined window size

Similar to AprioriAll algorithm, an iteration-based approach is utilized for incrementing the size of the episodes gradually by generating candidate episodes at each level. The algorithm starts with singletons (episodes of size 1) and candidate episodes are generated by joining the episodes determined in the previous iteration. For joining the episodes, the minimal occurrence rule is used. If α and β are two episodes having occurrences in $[t_s, t_e]$ and $[t_s', t_e']$ intervals respectively, for joining α and β;

- $t_e' - t_s \leq win$ where $t_e \leq t_s'$
- $t_e - t_s' \leq win$ where $t_e' \leq t_s$

Minimal occurrence assures the pattern is searched within a pre-defined window. For filtering non-frequent episodes, we utilize minimum support threshold like AprioriAll algorithm. However, this time support threshold is defined as a frequency value rather than percentage.

Further information on general principles and pseudo codes of the algorithm could be reached referring to [49]. The following flowchart represents the procedure we used for implementing the algorithm.

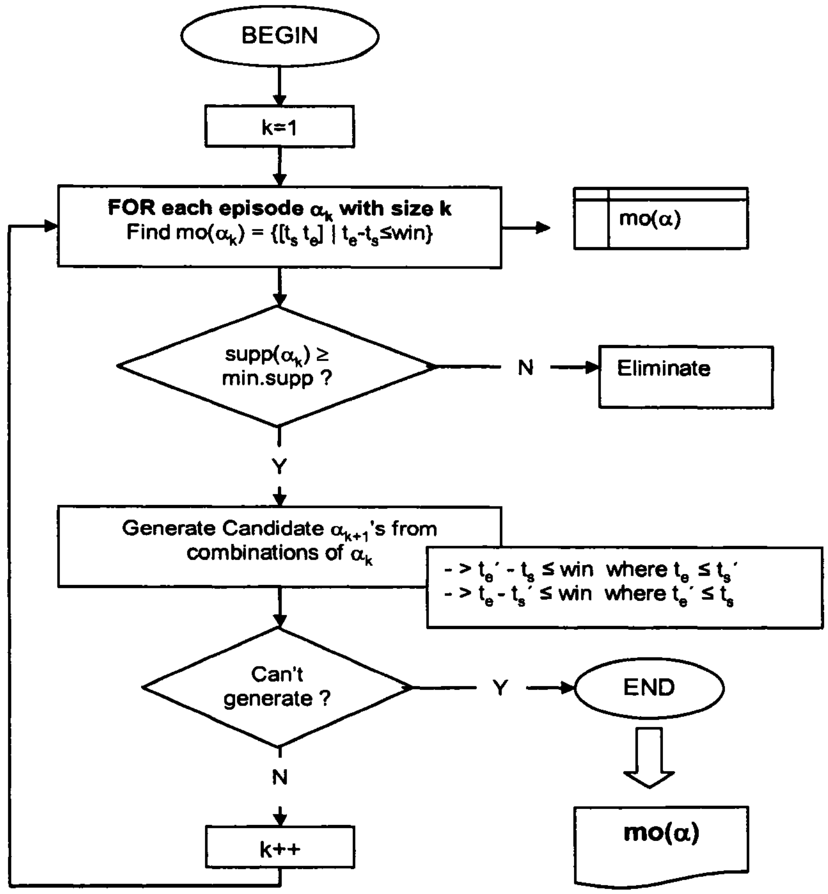


Figure 24 – Basic flowchart for implementation of MINEPI algorithm

By implementing MINEPI algorithm we find the patterns of changes frequently repeated in a model sequence. For discovering the probabilities of having these patterns, we have to compute the confidence of the episode rules associated with these patterns. The confidence of an episode rule $\alpha \rightarrow \beta$ is basically, $\frac{mo(\alpha \wedge \beta)}{mo(\alpha)}$. In other words, the probability is equal to the proportion of frequency of β following α to frequency of all occurrences of α within the given time window.

Using the procedure described in this section, Change Propagation Agent fills in the *Change Propagation* table and returns it to the ECM database. *Change Propagation* table defines the dependency of components on each other based on data history. This method could be combined with the parametric [21, 22] / DSM [23] / C-FAR [26, 27] based change propagation mechanisms proposed in literature. All of these mechanisms consider determining the dependency of components on each other during the initial product development stage and using matrices defining change propagation for triggering changes throughout the life cycle. Our methodology brings an opportunity to update these matrices by mining change data and finding out if the anticipated dependencies between components are accurate.

4.3. EXPERIMENTS

For verifying the mechanism concerning Prioritization and Change propagation agents, we generated 4 product models of office tables on ADVICE (Figure 25).

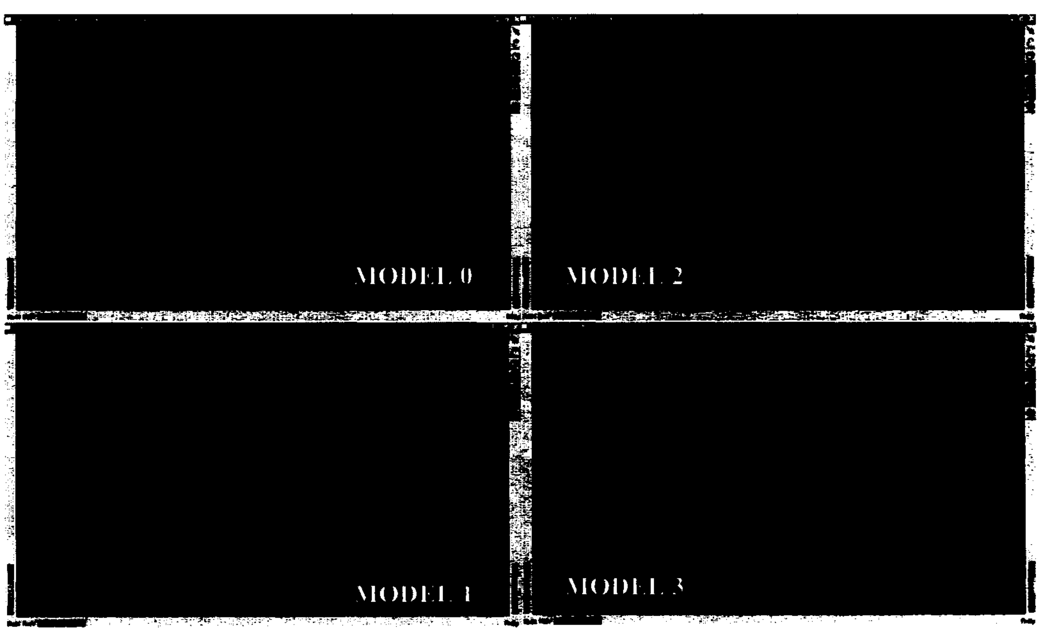


Figure 25 - Product models used in experiments

For these models a total of 43 components with their corresponding attribute values are created in the ECM database. The attributes are limited to material, shape and dimension as stated in the sample application in Chapter 3. The models have common components and sub-assemblies as well as unique components specific to each model. The number of components in each model's BOM is given below.

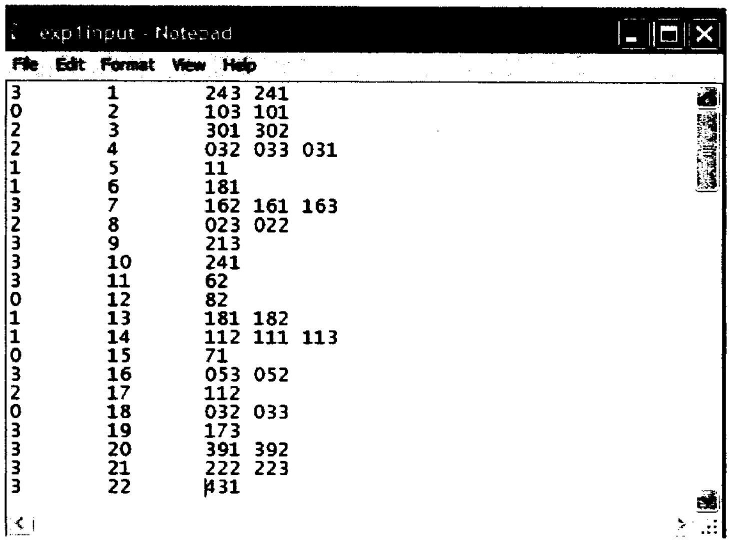
Table 7 - Component Size for each Model

Model No	*Number Of Components*
0	9 (all common with other models)
1	24 (all common with other models)
2	23 (14 of them common with other models)
3	26 (16 of them common with other models)

For experimentation, as it is not feasible to collect data with real subjects, we generated synthetic data. The data includes change transactions containing 1, 2 or 3 attribute changes. In this way, we created parallel and serial change events selected and ordered randomly.

For easily processing and analyzing data we mapped generated sequences to a three digit format for denoting each element of the change transactions. In this format XXY, first two digits denote the part number and the last digit denotes the attribute number, where 1 stands for material, 2 for shape and 3 for dimension change. This mapping could be changed when part number and attribute number increase.

The dataset that will be used as input for our experiments is in the format represented in Figure 26. The first column is for product model; the second shows transaction time and the third denotes the concurrent transactions. As an example, the first transaction represents component 24 is selected and 3^{rd} (dimension) and 1^{st} (material) attributes are changed on product model 3. As it can be recalled, there is no difference between (243 241) and (241 243), they are parallel events happening concurrently.



```
exp1input - Notepad
File Edit Format View Help
3   1   243 241
0   2   103 101
2   3   301 302
2   4   032 033 031
1   5   11
1   6   181
3   7   162 161 163
2   8   023 022
3   9   213
3   10  241
3   11  62
0   12  82
1   13  181 182
1   14  112 111 113
0   15  71
3   16  053 052
2   17  112
0   18  032 033
3   19  173
3   20  391 392
3   21  222 223
3   22  031
```

Figure 26 - Sample input format

4.4.1. Experiments with Prioritization Agent

We conducted 5 experiments each including computer generated 100 records based on the 4 models including 43 components presented on Figure 25. The distribution of records among models for each experiment data set is presented in Table 8.

Table 8 - Distribution of generated records among models in each experiment data set

Model	*Experiment 1*	*Experiment 2*	*Experiment 3*	*Experiment 4*	*Experiment 5*
0	16	7	10	7	9
1	30	28	39	26	37
2	21	29	25	28	30
3	33	36	26	39	24
Total	100	100	100	100	100

The minimum support threshold is selected as 50% in the experiments. As there are 4 models involved, 50% minimum support means the resulting patterns should be present in at least 2 models. The number of iterations for reaching these patterns changed between 7 and 14; reached 7 for Experiment 2 and 14 for Experiment 4. The following figure gives an example for the resulting patterns in Experiment 2. It may be noticed that for 3^{rd} iteration no patterns are listed. This is due to the fact that all large sequences with size 3 are eliminated in maximal phase of the algorithm.

Iteration	Supp	Patterns
7	2	(222) >> (051) >> (062 063) >> (191 193)
	2	(061) >> (071 072 073)
	2	(031) >> (051) >> (062 063) >> (191 193)
	2	(011) >> (051) >> (062 063) >> (191 193)
6	2	(222) >> (051) >> (062 063) >> (191) >> (011)
	2	(031) >> (051) >> (062 063) >> (191) >> (011)
5	2	(011) >> (213) >> (012) >> (191 193)
	2	(011) >> (191) >> (213) >> (012) >> (191 193)
	2	(011) >> (031) >> (032) >> (191 193)
	2	(011 012) >> (191 193)
4	2	(191) >> (011) >> (012) >> (191 193)
	2	(011) >> (191) >> (032) >> (191 193)
2	2	(222) >> (182) >> (011)
	2	(191) >> (213) >> (011)
	2	(191) >> (032) >> (011)
	2	(142) >> (103) >> (071)
	2	(032) >> (033) >> (013)
	2	(031) >> (182) >> (011)
	2	(031) >> (032) >> (011)
	2	(022) >> (033) >> (013)
1	2	(131) >> (071)
	2	(121) >> (071)
	2	(083) >> (071)
	2	(061) >> (092)
	2	(052) >> (071)
	2	(051) >> (071)
	2	(032) >> (071)
	2	(012) >> (071)
	2	(012) >> (051)
	2	(011) >> (182)

Figure 27 - Resulting patterns for Experiment 2

The patterns obtained as a result of running AprioriAll algorithm are converted to priority codes via computation of Priority Index. Threshold value for Priority A is selected as mean of the priority index; threshold value for Priority B is selected as half of the mean. As discussed earlier, these threshold values are pre-determined for classifying and ordering each component- attribute couple in terms of its importance.

Figure 28 shows the resulting priorities for each experiment. These priority values are transferred to Priority table to provide support via Prioritization Agent.

EXPERIMENT 1			EXPERIMENT 2			EXPERIMENT 3			EXPERIMENT 4			EXPERIMENT 5		
A threshold	2.17188		A threshold	1.675		A threshold	2.1875		A threshold	2.958333		A threshold	2.357143	
B threshold	1.08594		B threshold	0.8375		B threshold	1.09375		B threshold	1.479167		B threshold	1.178571	
031	6	A	011	4.5	A	032	4.5	A	023	7.5	A	091	7.5	A
032	6	A	031	3.5	A	042	4	A	162	6.5	A	093	5	A
062	4	A	222	3.5	A	232	3.5	A	241	6.5	A	053	5	A
161	3.5	A	051	3	A	041	3.5	A	222	6	A	132	4.5	A
112	3	A	032	2.5	A	102	3	A	101	4.5	A	022	2.5	A
033	3	A	061	2	A	133	3	A	052	3.5	A	241	2.5	A
052	3	A	213	2	A	132	3	A	221	3.5	A	223	2.5	A
111	3	A	191	2	A	131	3	A	193	3.5	A	243	2.5	A
243	3	A	012	2	A	183	2.5	A	192	3.5	A	113	2	B
113	3	A	063	1.5	B	181	2.5	A	191	3.5	A	112	2	B
241	3	A	062	1.5	B	123	2	B	053	3.5	A	111	2	B
222	2.5	A	142	1	B	121	2	B	043	2.5	B	051	2	B
181	2.5	A	022	1	B	233	2	B	041	2.5	B	063	2	B
202	2	B	083	0.5	C	141	2	B	042	2.5	B	023	1.5	B
082	2	B	103	0.5	C	143	2	B	102	2	B	101	1.5	B
021	1.5	B	052	0.5	C	231	2	B	021	1.5	B	061	1	C
023	1.5	B	121	0.5	C	103	2	B	232	1.5	B	052	1	C
042	1.5	B	033	0.5	C	092	1	C	133	1.5	B	242	1	C
201	1.5	B	131	0.5	C	091	1	C	051	1	C	011	0.5	C
121	1.5	B	182	0.5	C	073	1	C	061	1	C	103	0.5	C
122	1.5	B				051	1	C	092	1	C	102	0.5	C
123	1.5	B				163	1	C	063	1	C			
053	1.5	B				031	0.5	C	172	0.5	C			
071	1	C				033	0.5	C	022	0.5	C			
072	1	C												
043	1	C												
163	1	C												
041	1	C												
162	1	C												
223	1	C												
131	0.5	C												
012	0.5	C												

Figure 28 - Results for experiments with Prioritization Agent

The results exhibit the change actions that are frequently followed by changes in half of the product models ranked according to an index showing their impact. For increasing precision, the number of models and transactions could be increased.

4.4.2. Experiments with Change Propagation Agent

We conducted 4 experiments each including computer generated 250 records for validating the mechanism proposed for change propagation. Among the models presented in Figure 25, we focused on finding patterns in Model 3 utilizing a window size of 3 and a minimum support of 2 occurrences. We assumed the time interval between each transaction is 1 unit. Applying MINEPI algorithm for searching episodes containing the most frequent change patterns, we obtained the following results.

Model 3; Minimum occurrence threshold =2; Time Window=4							
EXPERIMENT 1		EXPERIMENT 2		EXPERIMENT 3		EXPERIMENT 4	
81 transactions; 133 changes		77 transactions; 130 changes		86 transactions; 140 changes		76 transactions; 118 changes	
Patterns	Probability	Patterns	Probability	Patterns	Probability	Patterns	Probability
021 >> 062	1.00	011 >> 412	1.00	013 >> 063	0.67	011 >> 352	0.40
041 >> 343	1.00	012 >> 411	0.50	022 >> 433	1.00	013 >> 201	0.50
162 >> 191	0.67	012 >> 412	0.50	032 >> 041	0.50	013 >> 202	0.50
181 >> 011	0.67	012 >> 413	0.50	041 >> 241	0.50	041 >> 011	0.67
183 >> 011	0.50	013 >> 411	0.67	041 >> 242	0.50	041 >> 341	0.67
193 >> 431	0.50	013 >> 412	0.67	041 >> 243	0.50	152 >> 421	1.00
202 >> 401	1.00	013 >> 413	0.67	041 >> 361	0.50	153 >> 421	1.00
202 >> 402	1.00	023 >> 431	1.00	053 >> 063	0.50	172 >> 202	0.67
221 >> 181	1.00	032 >> 411	1.00	053 >> 421	0.50	172 >> 233	0.67
221 >> 183	1.00	041 >> 012	0.50	062 >> 242	0.67	173 >> 202	1.00
231 >> 041	0.40	041 >> 013	0.50	063 >> 421	0.50	181 >> 152	1.00
231 >> 191	0.40	041 >> 353	0.50	173 >> 021	1.00	181 >> 153	1.00
231 >> 221	0.40	163 >> 222	0.67	193 >> 022	0.50	181 >> 421	1.00
231 >> 343	0.40	191 >> 221	1.00	241 >> 183	0.50	183 >> 152	0.50
231 >> 402	0.40	191 >> 222	1.00	241 >> 363	0.50	183 >> 153	0.75
232 >> 041	0.40	241 >> 052	0.67	242 >> 363	0.50	183 >> 421	0.75
232 >> 343	0.40	241 >> 053	0.67	243 >> 363	1.00	202 >> 182	0.50
233 >> 193	0.40	242 >> 393	0.67	361 >> 191	0.67	203 >> 161	1.00
233 >> 221	0.40	353 >> 012	0.50	361 >> 192	0.67	203 >> 162	1.00
233 >> 402	0.40	353 >> 013	0.50	361 >> 193	0.67	233 >> 041	0.67
243 >> 162	1.00	353 >> 412	0.50	362 >> 013	0.67	352 >> 041	1.00
361 >> 231	1.00	353 >> 413	0.50	391 >> 153	1.00	362 >> 203	0.40
361 >> 233	1.00	382 >> 243	0.50	412 >> 341	1.00	362 >> 212	0.40
363 >> 231	0.67	393 >> 031	0.67	433 >> 041	0.67	362 >> 213	0.40
363 >> 232	0.67	411 >> 041	0.50			382 >> 362	0.67
363 >> 233	0.67	411 >> 353	0.50			383 >> 362	0.67
363 >> 402	0.67	412 >> 041	0.50			393 >> 013	1.00
372 >> 183	1.00	412 >> 042	0.33			421 >> 362	0.67
411 >> 051	1.00	412 >> 241	0.33				
431 >> 232	0.50	412 >> 243	0.33				
431 >> 233	0.50	412 >> 353	0.33				
		413 >> 041	0.67				

Figure 29 – Results for experiments with Change Propagation agent

The algorithm resulted in episodes with size 3 and 2 as the window size is selected as 4. But the results above only show the episodes with size 2 as they are providing input for Change Propagation table in ECM database. The probabilities associated with each pattern represent the confidence of the rules involved in each pattern.

As an example, in the first pattern of the first experiment, we see that all of the material changes on component 2 are followed by shape changes on component 6. However, the last two patterns in the same experiment dataset reveal that 50% of the material changes in component 43 trigger shape and dimension changes in component 23. As it may be recalled these probabilities are presented to users using colors during approval process.

4.5. DISCUSSION OF RESULTS

The experiments demonstrated the capability of proposed Agents to facilitate two important processes in ECM life-cycle; prioritization of the changes and change propagation.

To the best of our knowledge, there are no previous works on how to prioritize the engineering change requests. In this way, the mechanism offered is characterized as a first approach to this problem. There are studies associated with Change Propagation; however they impose extensive time requirements for preparing the data that will serve as a basis for triggering the changes. For instance Cohen [27] states a domain expert worked about 40 hours to construct C-FAR matrices containing Low-Medium-High values for dependency of attributes in an injection moulding case study. Apart from time requirements the studies in literature depend on subjective evaluations of engineering staff. The agent proposed in this Chapter differentiates from these studies by the means of constituting data. Incorporating data mining to the process, the time required for preparing change propagation data has diminished to the level of seconds or minutes. Moreover the data relies on actual changes rather than anticipations of people.

The first group of experiments are based on synthetically created change data on 43 components in four models. The second group are restricted to 26 components in Model3. This is an extremely small data set when compared to real life cases with ten thousands of components and hundreds of model configurations. As the possible event types are scarce due to size of our data set, the events repeated more frequently than they would in an actual case when we generated random data. This caused the number of patterns and occurrences / frequency of patterns to increase.

However, as our purpose is demonstrating the applicability of the mechanisms proposed, we may conclude the results are satisfactory for validating the supporting agents since they have the following characteristics.

Prioritization and Change propagation agents;

- Depend on reality rather than expectations
- Provide objective evaluation
- No time required for preparing data as data is retrieved from database

- Easily updated if the parameters used in mining patterns need to be changed
- No need for update if the design is changed to include new components or attributes
- Guide the user for considering changes that are not expected but usually triggered
- Eliminate the need for controlling changes that are expected but never triggered

CHAPTER 5

CONCLUSION

5.1. CONTRIBUTION

Most people say design is 10% inspiration and 90% perspiration. Apart from the hard work and efforts needed for developing a new product, considerable endeavour is required for maintenance of the design, i.e. continuously improving and refining the design with Engineering Changes throughout the life-cycle of a product. Management of these Engineering Changes are considered to be one of the most problematic and complicated processes within an organization. If a company fails to plan and control the information and workflow involved in ECM properly, there is no doubt it will face time losses, increasing costs and decreasing quality level.

This thesis suggests a novel approach to ECM by incorporating two other domains into the process: Virtual Collaborative Design Environments and Sequential Pattern Mining. Using tools offered by these domains, we developed ADVICE, a distributed virtual environment that provides active user support for carrying out Engineering Change Management process more effectively.

Our contributions with this dissertation may be summarized as follows:

- Bringing a distinguished perspective to model ECM process in a Virtual Environment for providing a compact solution by combining parametric information addressing to logic-oriented left brain with graphical information addressing to artistic-oriented right brain

- Connecting geographically dispersed users in a shared environment for handling ECM functions in a collaborative manner
- Providing real-time manipulation of the shared 3D product model for performing ECM sub-processes
- Offering a database infrastructure for storing and tracking engineering change data involving a variety of components, attributes and product models in a systematic manner
- Proposing effective means of representing data on Virtual Environment so that Engineering Change could be perceived easily, rapidly and accurately even by the non-technical members of the Engineering Change Board
- Integrating with ERP and CAD systems for retrieving engineering change related data and eliminating the need for extensive data entry within ECM process
- Capturing the data involved in ECM, processing the data and returning to the ECM itself for offering "advice" on important decision parameters
- Providing a smart mechanism for assigning priorities to Engineering Change Requests
- Predicting change propagation data and offering this data to serve as a basis for actions to be performed after the completion of the Engineering Change life-cycle.
- Consequently, proposing a superior approach to solutions in literature for coping with problematic yet inevitable Engineering Change Management process

5.2. POTENTIAL ENHANCEMENTS

The environment ADVICE proposed in this dissertation has some limitations due to time considerations and inadequacy of technical infrastructure. With the inclusion of the concerns listed below, we may take a big step towards perfection.

- We did not have the chance to work with an industrial partner for this study. For this reason, Engineering Change Management model proposed in this thesis is based on the comprehensive literature review made on the subject and author's own professional experience. We believe that inclusion of industrial partners may provide new perspectives and ameliorate the model. Moreover, the suggested integration with ERP and CAD systems could be managed and more realistic products could be analyzed for verifying the model.
- Engineering Change Management process involves interactions with other components, processes, machinery, tools and equipment. Among these elements, we focused only on interactions with other components although we emphasized the significance of the other elements. For improving ADVICE, these elements should be integrated in the solution.
- Similarly, in Engineering Change Notice phase, we limited action plan to triggered changes as the actions to be taken are considered detailed and industry-specific. For having a complete model of ECM, we should express these actions in a generic way and incorporate them to the environment.
- The database structure which constitutes the foundation for ADVICE is a desktop application on Microsoft Access. We did not have the chance to transfer this application to SQL server or another ODBC-compliant data container. For

providing a fully collaborative environment this transfer should be assured. For allowing the suppliers, customers and dispersed departments in the process, internet server should be employed and a security mechanism should be offered. The 3D models may be transferred to VRML format and information carried over internet might be converted to XML.

- Menus in the proposed environment ADVICE might be created in the environment rather than on a separate command window.
- Apart from changing attributes of a component in a given product model, features for adding or removing a component from a sub-assembly might be provided.
- For prioritizing engineering change requests, there might be other factors determining impact such as cost of the changed component and lead time for producing or purchasing the component. These may be included to decision making process. Actually, impact of a change involves some ambiguity within the definition so it might be defined as a fuzzy number rather than a crisp value.
- The change propagation mechanism offered in ADVICE takes the changes triggered within the same product model into consideration. It may be enhanced to cover all product models.
- The cost of making a change could be defined as a function of costs associated with triggered changes.
- BOM structure could be expressed as a 3D virtual object on the environment. This representation might be connected to product model changed to increase comprehension.

- Each engineering change decision involves interaction with product dynamics and kinematics structure. Some parameters defining this structure are already involved in attributes defining the components. In case of changes on these attributes the system may provide simulations based on existing and proposed models.
- The environment could be transferred to an immersive environment including visual, audio and haptic devices.

5.3. FUTURE DIRECTIONS

Despite the limitations discussed in the previous section, our study has the particularity of being the first approach for integrating Engineering Change Management and Virtual Environments. This is surprising as the nature and needs of Engineering Change Management process is quite convenient to be treated within a Virtual Environment context.

As a process whose significance has been admitted by many researchers and a research topic that might be supported and financed by the industry, Engineering Change Management will continue to attract researchers' attention. We believe that this thesis presents a step towards improvement of this process.

REFERENCES

1. Ullman, David G., "*The Mechanical Design Process*", McGraw-Hill, Inc., 1997. ISBN 0-07-065756-4.
2. Bochenek, Grace M., Ragusa, James, M., "*Virtual Collaborative Design Environments: A Review, Issues, Some Research and the Future* ", Portland International Conference on Management of Engineering and Technology, Vol. Supplement, pp. 726-735, Portland, USA, (2001).
3. NetMBA Business Knowledge Center, Product Life-cycle Stages, URL: http://www.netmba.com/marketing/product/lifecycle/
4. Kidd, M.W., Thompson, G., "*Engineering Design Change Management*", Integrated Manufacturing Systems, vol. 11 (1), 2000, pp. 74-77.
5. Huang, G.Q., Yee, W.Y., Mak, K.L., "*Development of a web-based system for engineering change management*", Robotics and Computer Integrated Manufacturing 17 (June) (2001) 255-267.
6. Yap, A.Y., Ngwenyama, O., Osei-Bryson, K., "*Leveraging knowledge representation, usage, and interpretation to help reengineer the product development life cycle: visual computing and the tacit dimensions of product development*", Computers in Industry, v.51 n.1, p.89-110, May 2003.
7. MSN Encarta, "*Left and Right Brain Functions*", URL: http:// encarta. msn. com/ media_461516672/Left_and_Right_Brain_Functions.html
8. Soderberg, L.G., "*Facing up to the engineering gap*", McKinsey Quarterly Spring: 3-23 (1989).

9. Terwiesch, C., Loch, C.H., "*Managing the process of engineering change orders: the case of the climate control system in automobile development*", Journal of Product Innovation Management, Vol. 16, Issue 2 (1999) 160-172.

10. Blackburn, J.D., "*New product development: The new time wars*", ed. Time-based Competition Business One Irwin Publishers, Homewood IL, 1991.

11. Chinn, S.J., Madey, G.R., "*Temporal Representation and Reasoning for Workflow in Engineering Design Change Review,*" IEEE Transactions on Engineering Management, vol. 47, No. 4, pp. 485-492, Nov. 2000.

12. Huang, G.Q., Mak, K.L., "*Current practices of engineering change management in UK manufacturing industries*", International Journal of Operations and Production Management, 19, (1) (1999), 21-37.

13. Frank, D.N., "*Engineering Change: Too important to leave to the engineers*", APICS International Conference, 1980, 1st World Conference of Production and Inventory Control, Vienna, 1985.

14. Bhuiyan, N., Gatard, G., Thomson, V., "*Engineering Change Request Management in a New Product Development Process*", European Journal of Innovation Management, Vol. 9 No 1, (2006).

15. Tavčar, J., Duhovnik, J., "*Engineering Change Management in individual and mass production*", Robotics and Computer-Aided Manufacturing, Vol. 21, Issue 3 (2005) 205-215.

16. Crow, K., "*Configuration Management and Engineering Change Control*", DRM Associates, URL: http://www.npd-solutions.com/configmgt.html

17. Huang, G.Q., Mak, K.L., "*Computer aids for engineering change control*", Journal of Materials Processing Technology, 76, (1998), 19-187.

18. Wright, I.C., "*A review of research into engineering change management: implication for product design*", Design Studies 18, (January) (1997), 34-42.

19. Loch, C.H., Terwiesch, C., "*Accelerating the Process of Engineering Change Orders: Capacity and Congestion Effects*", Journal of Product Innovation Management, 16 (2), (1999), 145-159.

20. Pikosz, P., Malmqvist, J, "*A Comparative Study of Engineering Change Management in Three Swedish Engineering Companies*", Proceedings of DETC'98, Paper No DET98/EIM-5684, Atlanta, GA, USA, 1998.

21. Rouibah, K. and K Caskey. "*Change management in concurrent engineering from a parameter perspective*", *Computers in Industry*, 50 (2003) pp. 15-34.

22. Yang, J., Goltz, M., Han, S., "*Parameter-based Engineering Changes for a Distributed Engineering Environment*", Concurrent Engineering, Vol. 12, No. 4, 275-286 (2004).

23. Clarkson, P.J., Simons, C.S., Eckert, C.M., "*Predicting change propagation in complex design*", *ASME Journal of Mechanical Design,* 126 (5), 788-797, 2004.

24. You, C., Yeh, S., "*Engineering Change Propagation System using STEP*", Concurrent Engineering: Research and Applications, vol 10, No 4, pp 349-356, Dec. 2002.

25. Yang, C.O., Pei, H.N., "*Developing a STEP-Based Integration Environment to Evaluate the Impact of an Engineering Change on MRP*", International Journal of Advanced Manufacturing Technology, Vol.15, 11, October 1999, 769-779.

26. Cohen, T., Navathe, S.B., Fulton, R.E., "*C-FAR, Change Favorable Representation*", Computer-Aided Design, (2000), vol. 32, issue. 5-6, 321-338.

27. Cohen, T., "*A data approach to tracking and evaluating engineering changes*", PhD Dissertation, Georgia Institute of Technology, 1997.

28. Peng, T., Trappey, A.J.C., "*A step toward STEP-compatible engineering data management: the data models of product structure and engineering changes*", Robotics and Computer-Integrated Manufacturing Vol.14, p89--109 1998.

29. STEP 10303 standard, URL: http://www.tc184-sc4.org/ SC4_Open/ SC4_Work_ Products_Documents/STEP_(10303)/

30. Horvath, I., Vergeest, J.S.M., Rusak, Z., "*Six ingredients of collaborative virtual design environments*", Design 2002 - Proceedings of the 7th international design conference, (2002), pp. 67-74.

31. Stanney, K.M., "*Handbook of Virtual Environments: Design, Implementation, and Applications*", Lawrence Erlbaum Associates, Mahwah, NJ, 2002.

32. Kim, G., J., "*Designing Virtual Reality Systems*", Springer, 2005

33. Bochenek, G.M., Ragusa, J.M., "*Virtual (3D) Collaborative Environments: An improved environment for integrated product team interaction?*", Proceedings of the 36th Hawai International Conference on System Sciences (HICSS 03), 2002.

34. Zorriassatine, F., Wykes, C., Parkin, R., Gindy, N., "*A survey of virtual prototyping techniques for mechanical product development*", Proceedings of the Institute of Mechanical Engineers, Journal of Engineering Manufacture, Part B - Volume 217 - Number 4, April 2003.

35. Daily, M., Howard, M., Jerald, J., Lee, C., Martin, K., McInnes, D., Tinker, P., Smith, R.C., "*Distributed Design Review in Virtual Environments*", Proceedings of ACM Conference on Collaborative Virtual Environments 2000, ACM Press, New York, NY, September 2000, pp. 57-63.

36. Zabel A., Deisinger J., Weller F., Hamisch T., "*A multimodal design environment*", Proceedings ESS European simulation symposium and exhibition 1999, Erlangen-Nuremberg, Germany, 1999.

37. Bscher, M., Christensen, M., Gronbak, K., Krogh, P., Mogensen, P., Shapiro, D., Orbaek, P., "*Collaborative Augmented Reality Environments: Integrating VR, Working Materials, and Distributed Work Spaces*", Proceedings of Collaborative Virtual Environments 2000.

38. ProVit, Fraunhbolfer Institute, URL: http://www.provit.net/

39. Yabuki, N., Kotani, J., Shitani, T., "*A Cooperative Design Environment Using Multi-Agents and Virtual Reality*", Cooperative Design, Visualization, and Engineering: First International Conference, CDVE 2004 Palma de Mallorca, Spain, September 19-22, 2004.

40. Jezernik, A., Hren G., "A *Solution to Integrate Computer-aided Design (CAD) and Virtual Reality (VR) Databases in Design and Manufacturing Processes*", International Journal of Advanced Manufacturing Technology, 2003, 22, 768-774.

41. Kan, H.Y., Duffy, V.G., Su, C., "*An Internet virtual reality collaborative environment for effective product design*", Computers in Industry, Volume 45, Number 2, June 2001, pp. 197-213(17).

42. Kremer, K., "*A concept for virtual reality tools for design reviews*", Proceedings of the conference on Visualization '98, 205-210.

43. Biuk-Aghai, R.P., Simoff, S.J., "*An integrative framework for knowledge extraction in collaborative virtual environments*", Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work, (2001), 61-70.

44. Simoff, S.J., Maher, M.L., "*Analyzing participation in collaborative design environments*", Design Studies 21, (2000), 119-144.

45. Dolunay, B, Akgunduz, A, "*Virtual Reality Based End-User Assessment Tool for Remote Product and System Testing*", CDEN/ /RCCI Conference, 24-26 July 2006, Toronto, CANADA

46. Wang, Wei, Ynag, Jiong, "*Mining Sequential Patterns from Large Data Sets*", Springer, 2005.

47. Agrawal, R., Srikant, R., "*Mining Sequential Patterns*", Eleventh International Conference on Data Engineering, (ICDE'95), 1995.

48. Agrawal, R., Srikant, R., "*Fast Algorithms for Mining Association Rules in Large Databases*", Proceedings of the 20th International Conference on Very Large Data Bases, (1994), 487-499.

49. Mannila, H., Toivonen, H., Verkamo, A.I., "*Discovery of Frequent Episodes in Event Sequences*", Data Mining and Knowledge Discovery 1, 259-289 (1997).

50. Srikant, R., Agrawal, R., "*Mining Sequential Patterns: Generalizations and Performance Improvements*", Proceedings of 5th International Conference of Extending Database Technology, EDBT, Avignon, France, 1996.

51. Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., Hsu, M., "*FreeSpan: Frequent Pattern-Projected Sequential Pattern Mining*", Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining (KDD2000), 20-23, 2000.

52. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M., "*PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth*", Proceedings of the International Conference on Data Engineering (ICDE '01), 215-226, 2001.

53. Chiang, D., Lee, S., Chen, C., Wang, M., "*Mining Interval Sequential Patterns*", International Journal of Intelligent Systems, Vol. 20, Issue 3, March 2005, 359-373.

54. Yen, S., "*Mining Interesting sequential patterns for intelligent systems*", International Journal of Intelligent Systems, Vol. 20, No. 1, January 2005, 73-87.

55. Yu, C., "*Mining sequential patterns from multidimensional sequence data*", IEEE Transactions on Knowledge and Data Engineering, Vol. 17, no. 1, January 2005, 136-140.

56. Wojciechowski, M., "*Discovering frequent episodes in sequences of complex events*", ABDIS-DASFAA Symposium 2000, 205-214.

57. Pei, J., Han, J., Wang, W., "*Mining sequential patterns with constraints in large databases*", Proceedings of the 11th international conference on Information and Knowledge Management, 2002, 18-25.

58. Shimano Fishing, Product Catalog, Schematics, URL: http://fish.shimano.com/

59. Crow, K., "*Configuration Management and Engineering Change Control*", DRM Associates, URL: http://www.npd-solutions.com/configmgt.html