

University of Windsor

Scholarship at UWindor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

2007

Robust design of collaborative information system in a prototype assembly facility

Bjorn Gunnar Franzon
University of Windsor

Follow this and additional works at: <https://scholar.uwindsor.ca/etd>

Recommended Citation

Franzon, Bjorn Gunnar, "Robust design of collaborative information system in a prototype assembly facility" (2007). *Electronic Theses and Dissertations*. 4685.
<https://scholar.uwindsor.ca/etd/4685>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

**ROBUST DESIGN OF COLLABORATIVE INFORMATION SYSTEM
IN A PROTOTYPE ASSEMBLY FACILITY**

by

Bjorn Gunnar Franzon

**A Thesis
Submitted to the Faculty of Graduate Studies
through Industrial Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science at the
University of Windsor**

Windsor, Ontario, Canada

2007

© 2007 Bjorn Franzon



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*

ISBN: 978-0-494-34906-9

Our file *Notre référence*

ISBN: 978-0-494-34906-9

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

ABSTRACT

The intent of this research was to engineer a collaborative data management system for an assembly facility supplying prototype products to a development and test centre. During the assembly of prototypes, multiple parts and subcomponents are exposed to engineering design changes, necessitating meticulous documentation and archiving of the bills of material. The findings from this research suggest, by using sound engineering methodologies and by extending the research to multiple fields of science, it is possible to design a data management system that has analytical proof for being robust and simple in its design. Robustness of the design is proven using a novel approach combining Axiomatic Design with the House of Quality. Improvements are mathematically evaluated by calculating the complexity of the designs using an innovative complexity formula originating from this research. A step-by-step approach is also developed, lending this research to be a framework for future design improvements.

DEDICATION

To my lovely wife Susan

for always generously sharing the best moments in life with me

ACKNOWLEDGEMENTS

I want to start by thanking Dr. Leo Oriet for giving me the opportunity to further my academics at the Master's level and for the opportunity of conducting this research which gave me valued experience from the industry. I also thank Dr. Oriet for his kind support, allowing me insight to documents and processes and giving me open doors to access real-life data to base my research on.

I extend a sincere thank you to all graduate faculty members of the Industrial Engineering department of the University of Windsor for their acceptance and trust in my work. I specially thank the professors on my committee: Dr. Francisco Moro, Dr. Zbigniew Pasek, and Dr. Waguih ElMaraghy for their continuous support and guidance on the research I conducted. I thank Dr. Moro for sharing with me his knowledge in Information Systems Management and Computer Science; which without, this thesis would not have been possible. I thank Dr. Pasek for his directions that kept my research on track, and for teaching me Organizational Behaviour and Management. A special thank you goes to Dr. ElMaraghy for his expertise in the area of Engineering Design Methodologies and Product Development. Dr. ElMaraghy's knowledge greatly helped this thesis to be brought into new dimensions: engineering design robustness and complexity. I also thank Ms. Mummery for always keeping me ahead of schedules and at all times putting the students in front of everything.

I thank the company at which this research was conducted. From my heart, thanks go especially to Helen, Jamie, and Tom (in no particular order) for being the most generous and kind individuals. I thank Jamie for early on letting me master the bills of material and engineering changes. I thank Helen for supporting my every thought and

idea. And I thank Tom for from the get-go trusting my ability by giving me expert knowledge of the MRP system.

I thank Jim for fully supporting my research and for giving it a high priority during times of change. I thank Mike for extending my knowledge of the MRP system and showing me every trick there was to know. I thank Julie and Dave for always generously answering all my questions. I thank Steve and Jeff for their continuous support and guidance, which through meetings and phone conferences brought their expertise to my attention and extended the deliverables of this thesis. I thank Angela for helping me to clarify the cost justifications and for always guiding me to the right resources in the corporation. And I thank all the employees at the company for supporting my research with ideas and opinions.

I also thank my parents Berith and Bernt, my brother Joakim, and my sister Karin for unconditionally having stood by my side throughout my education.

Lastly, a special thank you goes to my wife Susan whom I know has sacrificed a lot for the cause of my education. I thank you from the bottom of my heart for always being understanding and supportive and for bringing laughter and joy to my long hours in front of the computer.

TABLE OF CONTENTS

ABSTRACT	i
DEDICATION	ii
ACKNOWLEDGEMENTS	iii
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS AND TERMS	xi
CHAPTER	
I. INTRODUCTION	
Bill of Material Fundamentals	1
Corporate Structure.....	3
Product Order Process	3
Four Different Bills of Material.....	5
Creation of Build BOM	7
Initial Opportunity – Dual BOMs.....	8
Build Event – Pre-Build.....	9
Build Event – Active Build.....	9
Build Event – Post Build	11
II. PROBLEM STATEMENT	
Problem Identification – Case Introduction.....	12
Case 1 – Multiple Files and Folders	13
Case 2 – Redundancy and Repetitiveness.....	14
Case 3 – Unused Information	15
Case 4 – Weak Data Integrity	16
Case 5 – Process Improvements	17
Case 6 – Reliability and Validity.....	17
Case 7 – Fragile Media	18
Case 8 – Increasing File Size.....	19
Case 9 – Security Breach.....	21
Case 10 – Disconnected Integration	22
Case 11 – Inaccuracy and Inconsistency	23
Case 12 – Complexity of Information Flow	23

	Case Summary	24
	Research Objectives.....	26
	Literature Support.....	26
	Literature Support – Transaction Error.....	27
	Literature Support – Technology Improvements.....	28
	Literature Support – Information and Data Quality.....	29
III.	REVIEW OF LITERATURE	
	Rationale for Literature Review	33
	Quality Function Deployment	34
	Robust Design.....	36
	Complexity.....	37
	Information Systems Design.....	40
	Summary of Literature Review.....	42
IV.	DESIGN AND METHODOLOGY	
	Design Framework.....	44
	Engineering Design Methodologies.....	45
	Information Flow Mapping.....	47
	Document Categorization	50
	Feasibility Study	51
	Quality Function Deployment	52
	Voice of Customer and Experts	54
	Functional Requirements	56
	House of Quality.....	57
	Axiomatic House of Quality.....	62
	Theory of Inventive Problem Solving	71
	Uncoupled Design Matrix.....	74
	Detailed Engineering Model.....	76
	Improved Engineering Design.....	78
	Data Flow Diagram.....	80
	Discourse on Evaluation of Designs.....	81
V.	ANALYSIS OF RESULTS	
	Matrix Tree Theorem.....	83
	Improved Complexity Formula	84
	Complexity Matrices.....	90
	Complexity Calculation	95
	Enhanced Engineering Design.....	96

VI.	CONCLUSIONS AND RECOMMENDATIONS	
	Implementation and Change Management	100
	System Specifications	105
	Integration with UGS Teamcenter	106
	Further Possibilities and Research	107
	Concluding Remarks.....	109
VII.	DELIVERABLES	
	Detailed Model	110
	Data Flow Diagram.....	111
	Entity-Relationship Diagram	112
	Database Table Structure	113
	Example of Table Data	114
	APPENDICES	120
	Substitution Statistics.....	121
	Microsoft Help and Support Article IDs.....	122
	Document Categorization Template	123
	TRIZ 40 Inventive Principles	124
	Step-By-Step Approach to Robust Design of IS	125
	REFERENCES.....	126
	Literature.....	126
	Periodical Documents on the Internet.....	129
	Nonperiodical Documents on the Internet	130
	VITA AUCTORIS	132

LIST OF TABLES

Table 1 – Excel 2002 File Sizes.....	20
Table 2 – Summary of Cases with Opportunity for Improvements.....	25
Table 3 – Data Quality Categories.....	29
Table 4 – Attributes of Information Quality	30

LIST OF FIGURES

Figure 1 – Bill of Material (Excel Format).....	2
Figure 2 – Product Hierarchy.....	5
Figure 3 – Bills of Material in the MRP database.....	6
Figure 4 – High-Level Build Event Processes.....	9
Figure 5 – Excel Warning Message.....	18
Figure 6 – Literature Survey with Opportunity Gaps.....	33
Figure 7 – Systems Development Life Cycle Steps.....	41
Figure 8 – IDEF ₀ Diagram of Project Plan.....	44
Figure 9 – Engineering Design Methodologies.....	46
Figure 10 – Current Information Flow Diagram.....	48
Figure 11 – Information Flow Entity Icons.....	49
Figure 12 – Document Categorization Legend.....	49
Figure 13 – Processes to Improve.....	52
Figure 14 – House of Quality Scheme.....	58
Figure 15 – House of Quality Matrix.....	59
Figure 16 – Coupling of Design Matrices.....	63
Figure 17 – Initial Axiomatic House of Quality Matrix.....	67
Figure 18 – DP-Reduced Axiomatic House of Quality Matrix.....	69
Figure 19 – Decoupled Axiomatic House of Quality Matrix.....	70
Figure 20 – Uncoupled Axiomatic House of Quality Matrix.....	75
Figure 21 – Detailed Engineering Model.....	76
Figure 22 – Improved Information Flow Diagram.....	79

Figure 23 – Data Flow Diagram of Database	80
Figure 24 – Illustration of the Matrix Tree Theorem.....	84
Figure 25 – Example (1) of Process Graphs	86
Figure 26 – Example (2) of Process Graphs	87
Figure 27 – Modified Complexity Formula.....	88
Figure 28 – Numbered Original Information Flow Diagram.....	91
Figure 29 – Numbered Improved Information Flow Diagram	92
Figure 30 – Complexity Matrix for Original System.....	93
Figure 31 – Complexity Matrix for Improved System	93
Figure 32 – Tree-Generating Determinant.....	94
Figure 33 – Enhanced Information Flow Diagram	97
Figure 34 – Numbered Enhanced Information Flow Diagram	98
Figure 35 – Complexity Matrix for Enhanced System	98
Figure 36 – System Installation Strategies.....	103
Figure 37 – UGS Teamcenter Modules	106
Figure 38 – Example of Weighted Graph.....	108
Figure 39 – Detailed Model of the Information System	110
Figure 40 – Data Flow Diagram of Database	111
Figure 41 – Entity-Relationship Diagram of Database.....	112
Figure 42 – Database Table Structure.....	113
Figure 43 – Bill of Material (Excel Spreadsheet)	114

LIST OF ABBREVIATIONS AND TERMS

Axiomatic Design (AD): A matrix-based engineering design methodology where the coupling between Functional Requirements and Design Parameters are resolved through axioms (see Coupling, Functional Requirements, and Design Parameters).

Bill of Material (BOM): A list of parts used in an assembly. Often, a BOM contains hierarchical information showing assemblies, sub-assemblies, and basic parts.

Coupling: The level of dependency between two objects, often in a table structure.

Customer Attribute (CA): A non-technical term describing how customers wish a product or process would perform in order to meet their needs, usually in the form of a full sentence (see Regulatory Attribute, Functional Requirement, and Design Parameter).

Data Flow Diagram (DFD): A common form of a process model that graphically illustrates the movement of data between external entities. It also shows the processes and how the data is stored in the system.

Data Management System (DMS): A system or software to manage data and run operations on the data requested by users.

Design Parameter (DP): A technical term describing an item, product, function, procedure, sequence, or parameter able to satisfy a Functional Requirement (see Customer Attribute, Regulatory Attribute, and Functional Requirement).

Entity Relationship (ER): The relationships among the entities in a business process.

Entity-Relationship Diagram (ER diagram): A detailed, logical, and graphical representation of the entity relationships (see Entity Relationship).

Feature Assembly Variation (FAV): A Number sequence indicating the relationship between a feature (component), subassembly (parent) and its parts (children).

Functional Requirement (FR): A technical term describing the desired performance of a product or process, usually in the form of a verb followed by a noun (see Customer Attribute, Regulatory Attribute, and Design Parameter).

House of Quality (HOQ): A matrix based decision method in the form of Quality Function Deployment (see Quality Function Deployment).

Intranet: An Internet-like network within the digital boundaries of an organization.

Material Requirements Planning (MRP): Software based production planning and inventory control system used to manage manufacturing processes.

Quality Function Deployment (QFD): Engineering design tool that evaluates the importance of customers' needs, the company's future activities and the competitors (see House of Quality).

Random Access Memory (RAM): A computer's main memory, or primary storage, used for displaying and manipulating data.

Regulatory Attribute (RA): A non-technical term described by experts how a product or process must function in order to meet their needs, usually in the form of a full sentence (see Customer Attribute, Functional Requirement, and Design Parameter).

Structured English Query Language (SQL): A database query language using basic English phrases to structure the queries, such as SELECT..., FROM..., WHERE...

Theory of Inventive Problem Solving (TRIZ): Russian creative design methodology based upon the study of thousands of patents, using contradictions to solve problems.

CHAPTER I

INTRODUCTION

Bill of Material Fundamentals

Many manufacturing facilities have processes that involve assembling parts according to a predetermined sequence while using a specific material list. In these facilities, a Bill of Material (BOM) is used to reflect the parts needed for the assembly of the products. The BOM is also used to group the parts into work stations, or man assignments, as well as sub-assemblies. In a production assembly plant, the BOM is a dynamic document that is automatically updated to reflect the latest changes and releases of assignments, parts, assemblies, and drawings. This allows the products to be built according to the latest set of instructions.

In a prototype [assembly] facility, with the main purpose of evaluating the assembly processes before product launch, the BOM is required to be a static document to allow the original contents to be validated for part and assembly accuracy. Wherever there are quantity discrepancies or the man assignments, parts, assemblies, or processes are not as desired, the BOM must allow for changes to be made to its contents accordingly. The discrepancies and the resulting changes, communicated through deviations and substitutions, need to be contained in documents for approval. In addition, the inconsistencies and workarounds also need to be controlled through documents that allow the results from the validation to be filtered, searched for, and revisited when needed. Moreover, when the process involves evaluating multiple product configurations, each product will require a BOM of its own. The purpose of the prototype assembly is to combine as many configurations as possible, through as few

products as practical; therefore, each product will have a set of unique part numbers, thus a unique BOM. The amount of information in the subsidiary documents related to changes and evaluations will also increase with the BOMs, as many parts are still common on the products, thus may appear in multiple documents. Even though some changes affect multiple products, the reference and background information must be tied to the individual BOMs for containment of the changes made.

Figure 1 shows a small portion of a prototype assembly BOM as an Excel spreadsheet. Some engineering changes have been made to the BOM. Here, the first strikethrough-lines (double) indicate an obsolete group-assembly on the BOM. Similarly, the next strikethrough-line (single) shows a part number that has been removed, although replaced with the part number directly below it, shown in italic font. Lastly, a work assignment, or Work in Progress (WIP), has also been rerouted to a new location.

Installation	Asm Var	Item	Part#	WIP#	New WIP#	Description	Qty	JoM
N010053477	0300					ASSEMBLY A		
N010053477	0300	1	30021R1	120		PART 1	2	PC
N010053477	0300	1	30754R1	110		PART 2	2	PC
N010053477	0300	2	507419C1	120	110	PART 3	1	PC
N010053477	0300	3	3661320C1	120		PART 4	1	PC
N010053477	0300	4	503505C1	120		PART 5	2	PC
N010053477	0300	5	194046H1	840		PART 6	2	PC
N080056049	0104					ASSEMBLY B		
N080056049	0104	3	306132C1	740		PART 7	4	PC
N080056049	0102					ASSEMBLY C		
N080056049	0102	18	306132C1	830		PART 7	1	PC
N080056049	0103					ASSEMBLY D		
N080056049	0103	2	3536111C1	710		PART 8	4	PC
N080056049	0103	2	3544557C1	740		PART 9	4	PC
<i>N080056049</i>	<i>0103</i>	2	<i>306132C1</i>	710		<i>PART 7</i>	4	<i>PC</i>
N090053326	0100					ASSEMBLY E		
N090053326	0100	4	3557745C3	110		PART 10	1	PC
N090053326	0100	4	3571199C1	710		PART 11	2	PC
N090053326	0100	5	3557746C3	110		PART 12	1	PC
N090053326	0100	5	3571199C1	710		PART 13	2	PC
N090053326	0100	6	3531907C2	520		PART 14	2	PC

Figure 1 – Bill of Material (Excel Format)

Corporate Structure

The manufacturing unit of the company is made up of two groups; a production group and a prototype group. The production group assembles the products that will be sent to distribution centres for sale to customers. However, prior to the introduction of a product at the production facilities, the prototype group is responsible for conducting extensive evaluation and validation of the assemblies and processes involved with the design of the new product. The prototype group is operational through the Prototype Centre. Products are assembled at the Prototype Centre during the [prototype] build event with the aid of the designers from the Development Centre and with the insight on processes at the assembly plants. The Development Centre is responsible for the design of parts and assemblies, as well as for the testing and evaluation of major assembly components and completed products. The Development Centre, in conjunction with different product centres, is also responsible for new product launches as well as component changes due to government liabilities and changes of regulations and classifications. The assembly sequences and processes, part designs and interactions, and contents of BOMs are all validated at the Prototype Centre. Any design errors, assembly miscalculations, or prototype part shortages that emerge during the build event are resolved at the Prototype Centre through consultation with the Development Centre.

Product Order Process

The build event is based on the build schedule, which is determined by the program launch managers. The schedule is usually planned around important milestones and test dates. The Development Centre decides the number of products to build at the

Prototype Centre, as well as the configurations, or features, of the products. The decision is based upon which components and product features need to be tested. Usually, the products consist of as many different features as possible to allow thorough testing of all component configurations. Once the features are selected, orders for the products are placed at the Prototype Centre through the Product Order System. However, before an order can be placed, drawings must be made for the parts used on the features available for the product. Therefore, the order process starts with the creation of drawings by engineers at the Development Centre.

Once the drawings are uploaded into the Product Order System, information about the parts and assemblies are sent to the Material Resource Planning (MRP) system, which then sends back an “effectivity date” (break date) for the same. The effectivity date keeps track of when old parts become obsolete and new parts come into effect at the assembly plants. The system is now ready to receive orders for products containing the new parts. Once an order is received in the Product Order System, the system checks the configuration of the product order to ensure that it is possible to build the product as specified. If cleared, the order is uploaded into a system where all features are associated with an installation, an assembly, and a variation, giving it a unique number: Feature Assembly Variation (FAV) number. This process basically links a feature number together with all installations and parts used in that feature, creating a relational hierarchy with parts assigned to a specific installation as shown in Figure 2. The hierarchy starts with a group, containing a set of features. The features contain installations (sub-assemblies in combination with other components), which contain parts that make up that

installation. As such, contrary to a BOM which contains both installations and parts, a product order contains a list of features selected by a customer.

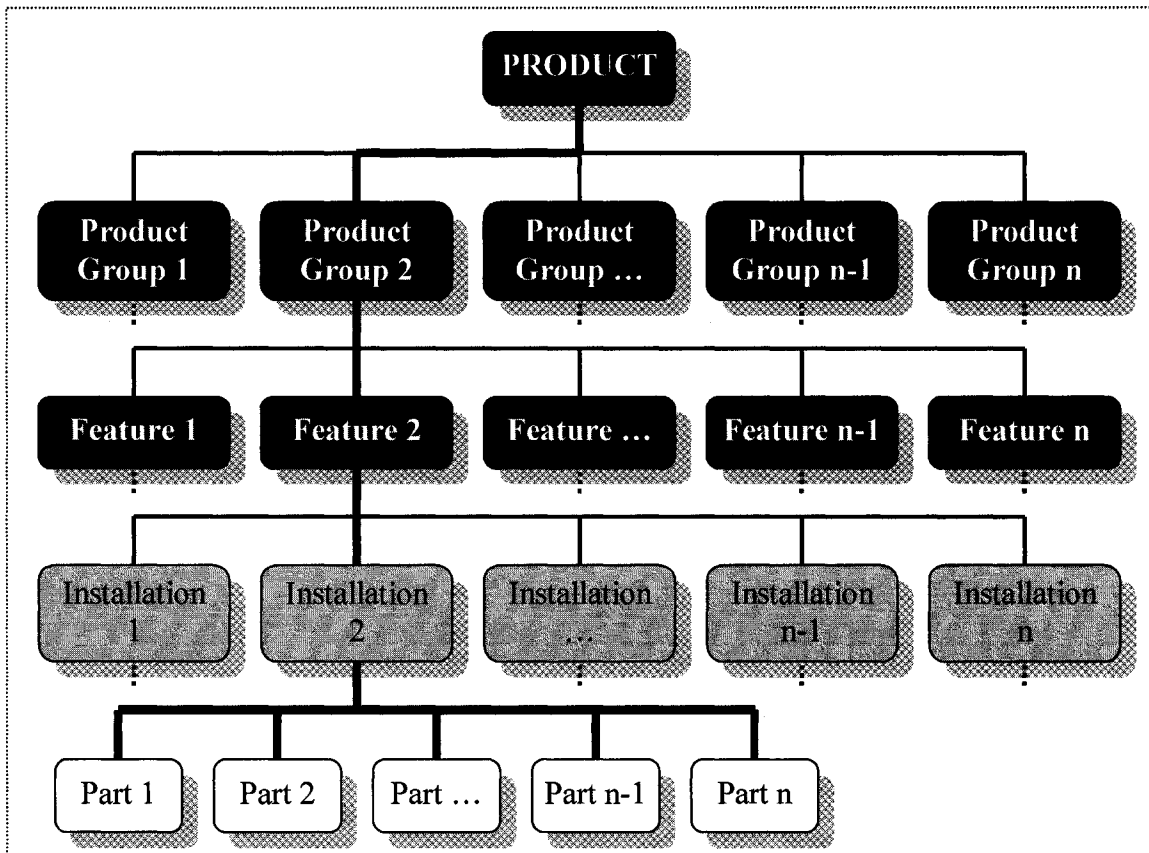


Figure 2 – Product Hierarchy

Four Different Bills of Material

The [prototype] assembly facility normally uses three BOMs: the Production Bill of Material (PBOM), the Estimated Bill of Material (EBOM), and the Build BOM. Once an order for a product is uploaded into the Product Order System, the accompanying BOMs are then created in the MRP system. Both the PBOM and the EBOM are physically stored in the MRP database, whereas the Build BOM is an Excel file stored on a server at the assembly facility. However, aside from the above mentioned PBOM and EBOM, there is actually one more BOM structure in the MRP database: a Customized

Bill of Material (CBOM). Nonetheless, The BOMs serve different purposes and are kept at various levels within the database structure of the MRP system. Figure 3 shows the three different BOMs in the MRP system as well as the Build BOM.

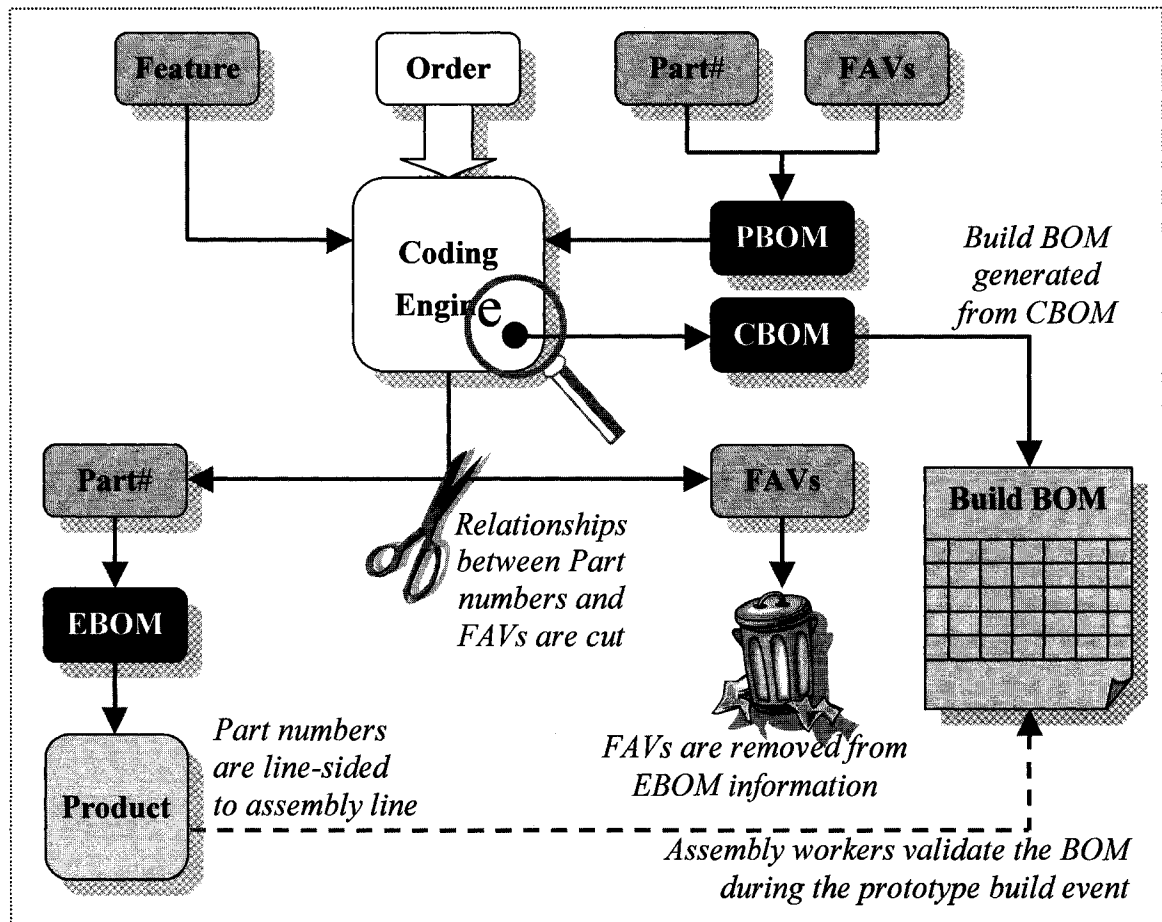


Figure 3 – Bills of Material in the MRP database

The PBOM is similar to a library, consisting of all FAVs, as well as sub-assemblies with their related part numbers. It is from this BOM that the Product Order System creates the relationship between a feature and an installation when an order is coded. The PBOM is updated daily with the latest releases of features, installations, and part numbers, as well as the break date when the new releases will come into effect. The CBOM, however, shows a temporary BOM of a complete order of a product as it would be coded by the Product Order System at that given moment. The CBOM will show all FAVs, and their

part numbers, associated with that order. Nevertheless, the CBOM is not a real BOM, but just a snapshot of how the product would be coded. The EBOM on the other hand, shows how the actual order is coded for the product. The EBOM is created from the PBOM through the coding engine. However, the EBOM is used by the MRP system to disperse material to the assembly line. Therefore, all FAVs are lost in the EBOM, and instead all similar part numbers are grouped together under the same Work in Progress (WIP) location number with the part number quantities added to a total count.

Creation of Build BOM

As mentioned earlier, a production facility uses the automatically updated CBOM to assemble their products by, and the EBOM to allocate the material to the product. However, this creates a problem at the prototype facility: The BOM needs to be a static document, such as the EBOM, but also needs to show the FAV-part number relationship, such as the CBOM. Furthermore, changes to the BOM must be made possible, and comments and validated assemblies and quantities need to be indicated on the BOM. Due to these constraints, a rather awkward procedure is necessary to produce the desired result: when an order is uploaded into the Product Order System and made ready for production at the Prototype Centre (i.e. released into the EBOM) the CBOM for that order must be extracted at the same moment, and then be separately maintained in a document. At the Prototype Centre, this document is called a Build BOM. The Build BOM is used for validation purposes as well as assembly instructions. As previously mentioned, the CBOM will automatically change through time, thus making it important to generate the Build BOM as soon as the order is released into the EBOM. If this is not

properly executed, there will be discrepancies between the EBOM and the Build BOM, meaning the two BOMs will not match. Hence, the parts listed to be used in the assembly may not be the ones that would be delivered to the line-side, or the parts needed may never arrive as no demand for them is shown. Furthermore, this also creates the cumbersome effort to maintain two different BOMs for the same product: one BOM for the MRP purposes, and one for the actual assembly and validation purposes. To ensure that the two BOMs are kept identical, changes must always be done in both.

Initial Opportunity – Dual BOMs

At this point, one could assume that a solution to the BOM discrepancies and duplication of efforts would be to modify the EBOM to include the FAVs. As easy as this might seem, its implementation is far from simple. The MRP system is used throughout the entire corporation. Any changes made to the structure of the EBOM at the Prototype Centre would also affect the other assembly plants. Recent statistics show that whatever quantity the Prototype Centre produces over the course of a year, the assembly plants produces, per day, more than two and a half times that quantity. Therefore, it is not feasible to justify a change of the MRP system to meet the needs of the Prototype Centre. Consequently, a solution must be found which allows two separate BOMs, while still accomplishing the intended tasks.

Build Event – Pre-Build

When the Development Centre has decided upon which product combinations to build at the Prototype Centre, the build event begins. A build event consists of three phases: Pre-Build, Active Build, and Post Build, as shown in Figure 4. During the Pre-Build event, the orders are defined in a build schedule and Process Flow and Sequence Charts are developed. A Potential Failure Mode and Effects Analysis (pFMEA) is also conducted on new design components.

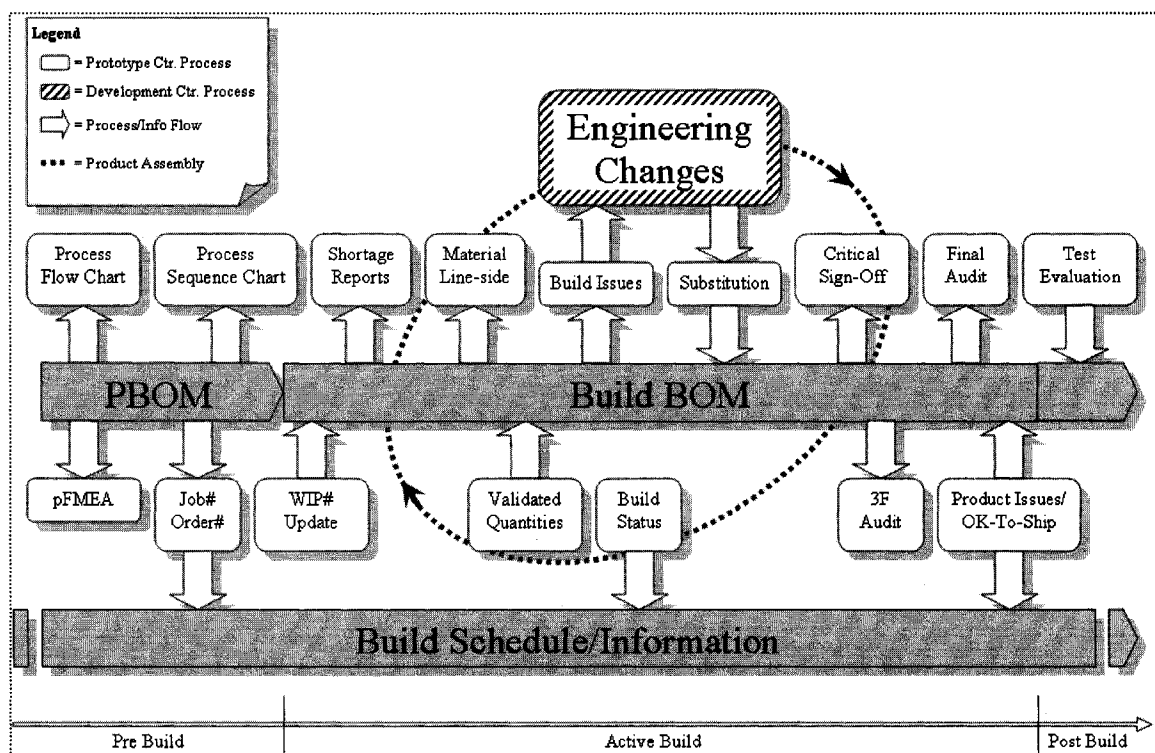


Figure 4 – High-Level Build Event Processes

Build Event – Active Build

During the initial stage of the Active Build event, the Build BOMs are reviewed to ensure all parts are assigned to the correct WIP location, or Operation Number (OP#). Any change to the part's WIP location is indicated on the Build BOM and also updated in

the EBOM, to allow the material to be correctly routed to the shop floor. If needed, part-shortage reports are generated through the MRP system to get an understanding of the maturity level of the build event. Once the prototype assembly starts, parts are line-sided, or kitted, to each WIP. Throughout the build, in real-time, the assembly workers indicate on the Build BOM the actual quantity of each part number used, for validation of the BOM contents. The completeness of all major assembly processes are also indicated with a percentage, to show the build status of each product. As mentioned earlier, the design engineers and program launch managers at the Development Centre are consulted to resolve any issues during the build event. Build issues are reported from the Prototype Centre as an Assembly Concern in the Engineering Change system, which is an SQL database. The design engineers at the Development Centre then resolve the issue by submitting a resolution in the form of a workaround, or if necessary, a substitution of part numbers and installations.

At the end of the Active Build event, there are two audits done on the product. Some components are also signed-off by the assembly workers to comply with legislations and regulations. The two audits performed on the products are a Form/Fit/Function (3F) Audit and a Final Audit. During the 3F Audit component alignments, gap between edges, and surface overlaps are measured. The Final Audit focuses on the overall product, from a customer's point of view once, assembled.

Before the product is shipped to the Development Centre for testing, the build issues are reviewed for completeness to ensure nothing is outstanding on the products. However, even with issues still open, an agreement can give the Prototype Centre the okay to ship the product as-built. At this point, the Build BOM and the EBOM are

compared. Any discrepancies between them are resolved in the EBOM, as the Build BOM is validated and seen as the Master BOM. Once the EBOM is corrected, the order for the product is closed in the MRP system. In addition, a copy of the product assembly issues document is attached to the Build BOM as a new spreadsheet. The password on the Build BOM is thereafter changed and the file is moved to a subfolder for archiving. Finally, a copy of the Build BOM is sent to the Development Centre by e-mail.

Build Event – Post Build

During the Post Build event, the prototype product and its Build BOM are now in the hands of the Development Centre, where testing and evaluation will be performed on the products. It is therefore extremely important to the engineers at the Development Centre that the product to be tested is built exactly the way the Build BOM indicates. If there are any cases of uncertainties, the parts need to be checked and verified for their accuracy, with respect to the Build BOM. Furthermore, if any parts are not as stated in the Build BOM they will be removed and substituted for the correct ones. The engineers at the Development Centre might also continue to substitute parts and assemblies on the products to further test other components and combinations. Thus, the Development Centre continues to maintain the Build BOM to reflect the current product composition.

Once the product has been thoroughly tested, it might be rebuilt to be as close as possible to production standards, to be sold as a “used product”. Therefore, the Build BOM is again used to record what has been added and/or removed from the product, so liability and warranty records can be kept for the product.

CHAPTER II

PROBLEM STATEMENT

Problem Identification – Case Introduction

The need for maintaining BOMs outside the MRP system and for developing product assembly issue tracking documents has until mid 2006 been limited to the Prototype Centre, responsible for the assembly of the products as well as for the validation of the assemblies, and the Development Centre, which facilitates the testing and evaluation of the product components. Nonetheless, future demands will require the prototype build events to also take place at the assembly plants, expanding the boundaries of the prototype build data management to include these facilities as well. Thus, an effective and simple data management system is imperative.

The system currently in use poses a problem for the design engineers at the Development Centre, as they do not have a direct or easy view of the present configuration of the products. As a result, substitutions with wrong part numbers and installation numbers are frequently received at the Prototype Centre to be processed in the BOMs. Furthermore, the tracking of substituted parts and installations is not mistake proof and shows inconsistency. This may hinder the ability to test important components due to uncertainty of the level of accuracy of the BOM. Moreover, the media used for the BOMs show poor data integrity and can easily be changed or deleted. Additionally, the flow of information is complex and highly coupled, thus susceptible to inaccuracies and errors. Lastly, and most importantly, the security level on the BOM is very low and might pose a threat to the corporate security, allowing competitors insight to its prototype business. Table 2 on page 25 shows a summary of the following twelve concerns:

Case 1 – Multiple Files and Folders

At the Prototype Centre, the Build BOMs are currently supported by a Microsoft Visual Basic macro that formats an ASCII text file of the extracted CBOM into a shared Microsoft Excel file. Each order for a product mandates a separate Build BOM, hence a separate Excel file. The Build BOMs are stored in a folder structure relating to each build program. A new build program requires an additional folder, thus different Build BOMs can currently be found at five locations at the Prototype Centre. This setup with many files and folders create redundancy of processes and duplication of documents and procedures. In addition, the current location of the files often needs to be communicated in meetings with engineers and managers, leading to confusion and wasted time. The potential cost associated with multiple files and folders is intangible, but would be associated with the time spent on identifying the correct folder where the current BOM is stored. As well, there is an intangible cost of repeatedly having to communicate the location of the BOMs every time the location either changes or is unknown to people. However, the greatest potential cost of the current system would be if managers, engineers, or assembly workers would be using the wrong BOM in their work. This would in that case lead to tedious maintenance of the BOMs or to redundant efforts by all stakeholders. The potential cost associated with multiple files and folders can be calculated as: Summation of all users $\{(X \text{ hours/year spent working with wrong BOM}) \times (Y \text{ dollars per hour})\} + (Z \text{ hours spent restoring BOM to original}) \times (\$26.00 \text{ per hour}) =$ **dollars spent/year**

Case 2 – Redundancy and Repetitiveness

The employees of the Prototype Centre and the Development Centre need to have access to all in-process BOMs, readily at hand. Some of the issues discovered during the build event may affect more than just one product, if the issue relates to a common product or a complete product family. At any given point in time, there might be up to twenty active Build BOMs during a build event. Thus, it is important for the assembly workers at the Prototype Centre to be able to cross-reference a part number in multiple BOMs. It is also important for the Development Centre to have easy access to the Build BOMs at the Prototype Centre, giving them insight on the parts needed on all products, rather than on just a specific product. However, an Excel document stored on a server is not easily accessible and does not allow for cross-reference through multiple files, thus preventing the Prototype Centre and the Development Centre to proactively resolve issues on products not yet assembled. This might therefore lead to issues being duplicated and cause redundant work for both the Prototype Centre and the Development Centre. Substitutions also cause redundancy, as the substitutions submitted by the Development Centre are manually processed in both the Build BOM and the EBOM. Statistics from the Prototype Centre show that throughout a year there have been 20,000 part number changes made to all the BOMs (see Appendix A). This tedious process requires each Excel file to be opened and modified in the same manner each time. In addition, 50% of the substitutions submitted affected multiple products, thus required to be repeatedly and equally processed in many Excel Build BOMs. The potential cost associated with the manual processing can be calculated as: (20,000 parts changed per year) x (5 minutes per processed change) x (\$26.00 per hour) = **\$43,300/year**

Case 3 – Unused Information

A new build program usually consists of a new group of engineers at the Development Centre, thus different users of the Build BOMs. In addition, the engineers are often given new assignments during the build event once it reaches a mature level. This together with the many BOMs and different storage locations make a standard operating procedure between the Prototype Centre and the Development Centre complex and cumbersome to maintain. Therefore, many engineers at the Development Centre are unaware of the Build BOMs and instead often reference the CBOM in the MRP database or early extracts of BOMs for part substitutions. As a result, engineers at the Development Centre are using a BOM that is not relevant to the products being assembled at the Prototype Centre. In addition, the current, as-built, status of the product is unknown to the Development Centre and as a consequence, substitutions are submitted for parts already assembled on the products. Statistics show that approximately 15% of the incoming substitutions have already been submitted (see Appendix A). This creates a problem, as the duplication is first realized when it is processed. Even so, substitutions are also seen for parts to be exchanged on products that have been completed and shipped to the Development Centre. The potential cost associated with the duplication of substitutions can be calculated as: (300 substitutions per year) x (10 rows per substitution) x (5 minutes per processed substitution) x (\$26.00 per hour) = **\$6,500/year**

Case 4 – Weak Data Integrity

The structure of Excel is made very flexible to allow for a variety of usages, as the software is intended for Finance and Statistics. Therefore, an Excel file has few restrictions to what type of information can be entered or which modifications are allowed to the original contents, thus the data integrity is almost nonexistent. Furthermore, the data types (or cell formats) can easily be overridden or changed in the documents, causing errors to appear when formulae are used, such as “=VLOOKUP(“x”,X:Y,n,m)” and “=IF(Xn=Ym,,)”. Turban et al. (2005) describes the integrity of data as “especially important” in a collaborative computer environment, in order to sustain a high level of data quality. Although cells can be locked and formulae can be hidden from view, thus protected from being changed, the protection function in Excel is dependent upon which version of the software is being used. Only Excel 2003, or newer, has the desired functionality. Even so, the protection cannot be turned off while a file is shared. Therefore, in order to add rows or columns, the document first needs to be unshared (and consequently cannot be in use) and thereafter unprotected. Once the changes have been made, the process needs to be reversed with protecting and then sharing the file. Comparisons between the original BOMs and the final BOMs show that on average 300 parts are substituted on a BOM: hence 300 rows are changed per BOM. Therefore, using cell protection will cause added processing and prevent access to the BOM when information to the documents need to be added or updated. The potential cost associated with the weak data integrity can be calculated as: (300 rows per BOM) x (70 BOMs) x (5 minutes) x (2 people) x (\$26.00 per hour) = **\$91,000/year** *Assumption: one person has to wait while one person works the substitutions (which takes 5 minutes).*

Case 5 – Process Improvements

Some build event processes and documents can be improved directly: The Build Status is redundant as the same information can be found in the BOMs, while the data entries to the Product Issues document can be reduced without loss of information.

The cost associated with the Build Status can be calculated as: (260 workdays) x (6 work areas) x (20 minutes per data entry per day) x (\$26.00 per hour) = **\$19,200/year**

The cost associated with the Product Issues can be calculated as: (260 workdays) x (5 minutes per issue) x (8 issues processed per day) x (\$26.00 per hour) = **\$4,500/year**

Case 6 – Reliability and Validity

During the testing of the products, it is extremely important to have a valid and accurate BOM. If the BOMs are not reflecting the true parts, the products will have to be rebuilt to assure the correct assembly. However, the Build BOMs are seldom completely accurate when shipped to the Development Centre. Therefore, the engineers at the Development Centre have to review the parts assembled to the products and make appropriate changes. As a result, the tests might have to be delayed or rescheduled, which could potentially postpone the launch schedule. The cost associated with reliability and validity can range from **\$10,000** if a test is not performed, to **\$50,000** which would be the average of a person's yearly salary with the job function of changing parts on the products to be tested.

Case 7 – Fragile Media

Besides the Excel documents being stored at multiple locations, the files are also very fragile. The documents can easily be deleted or moved by accident. In addition, a shared Excel file can also be made exclusive by anyone with access to the file. This will prevent any simultaneous changes to be saved to the file, as shown in Figure 5. Instead, the users will be prompted to save a copy of the file. If unnoticed, this will cause multiple copies of the files to be stored together with the original file, whereby the copies might be mistaken for the original file. To consolidate the copies and restore the original file is a tedious task, as there is no indication as to what was changed. The consolidation will also cause disruptions in the accessibility of the files, as the files need to be unshared and cannot be used while compared. Microsoft Help and Support (Article ID 130494, 214073, 271513, 814068, and 913770) describes the problem causing files to not be saved in more detail (see Appendix B).

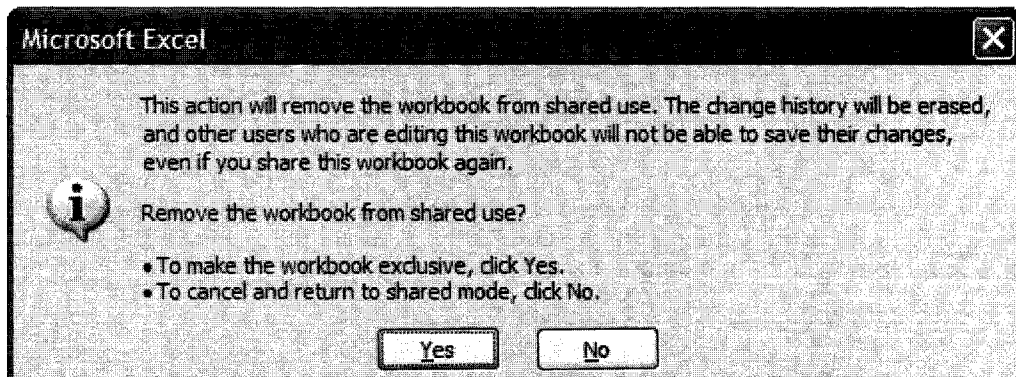


Figure 5 – Excel Warning Message

However, there is a tracking function in Excel that can be turned on to log all changes made to a shared document. Conversely, this is not a guaranteed source of information, as the change history is deleted as soon as the file is unshared as shown in Figure 5. The potential cost associated with fragile media is intangible, but can be identified as the

labour cost of having to restore one or many BOMs to its original contents. In the event of data accidentally being removed from the BOM, the cost can be calculated as: (X hours/year spent restoring BOMs) x (\$26.00 per hour) + (X hours/year spent restoring BOMs) x (Y #of workers waiting on BOMs during restoration) x (\$26.00 per hour) = **dollars spent/year**

Case 8 – Increasing File Size

A cause of concern is noticed when comparing the original file size with the same file at the end of the build event. The original Build BOM is approximately 800 kB when created and contains about 2,500 rows of data spread across 34 columns (16 columns empty at start). When the product is completed, the same file is usually 5 MB (although files of 40 MB are found), but only containing an additional 300 rows. Microsoft describes that the maximum, theoretical, size of an Excel file is based on the size of the computer's Random Access Memory (RAM). The RAM is the limit because the whole Excel file is loaded into the RAM when opened. Furthermore, a temporary copy of the file is created in the RAM when the file is saved, thus allocating twice the amount of memory. These allocations of memory quickly reduce the capacity of the RAM, hence slowing down all computer processes. However, the reason for the expanding file sizes can be explained by the fact that during the course of the build event there might have been 20,000 changes made by multiple users to the shared document. The experiment in Table 1 shows how quickly an Excel file increases in size when modified accordingly.

Document Action	Content Changes	File Size (kB)	
		Log Deactivated	Log Activated
Creation	Empty file	12	12
Saved	Empty file	14	14
Shared	Empty file	29	29
1 st Change	Letter “a” entered in cells A1 to A3000	243	343
2 nd Change	All cell contents in column A deleted	409	308
3 rd Change	Letter “a” entered in all cells in column A	7,139	4,912
4 th Change	All cell contents in column A deleted	8,469	6,331
Unshared	Empty file	18	18

Table 1 – Excel 2002 File Sizes

In addition, blank rows or columns are sometimes added through the use of “Copy and Paste”, which accidentally expands the Excel file to its limit of 65,536 rows and/or 256 columns. Depending on whether the columns, rows, or both have expanded, the file increases to contain, 700,000 active cells for the columns and 2.2 million active cells for the rows. Similarly, a total of 16.8 million active cells are created for changes to both columns and rows, which should be compared to the original document containing 85,000 active cells. Although blank cells in Excel do not allocate any memory, formatting of a cell will still cause an increase in file size. Microsoft Help and Support (Article ID 244435, 313275, and 816952) describes the problem in more detail (see Appendix B). The potential cost associated with increasing file sizes is intangible, but can be viewed as: (N hours/year spent waiting on computer processes due to open BOMs) x (\$26.00 per hour) = **dollars spent/year waiting**

Or (X \$ per RAM) x (Y #of computers) + (Z \$ per hour for IT personnel installing one RAM) x (Y #of computers) = **dollars spent/year on “unneeded RAM”**

Case 9 – Security Breach

The Excel Build BOMs are also a concern for a possible breach in the corporate security. Currently the BOMs are protected with a password, but only for modifying the files. As Excel is not integrated with the corporate database of user IDs, access to view the BOMs is made unrestricted to allow all engineers at the Development Centre insight to the BOMs. In addition, anyone with knowledge of the password will be granted full access to the files. However, to get access to the BOMs, a user identification number and a password is required to be entered on the computers at the company.

With today's large capacity on removable storage devices, together with the fast transfer rate through the computer ports, anyone with the wrong intentions could theoretically download all the BOMs in a matter of a few minutes. The danger of a possible intrusion is mentioned in the Corporate IT Policy: "Outside disclosure could lead to serious damage of the corporation's business relationships." The potential cost associated with a security breach is intangible, but a potential leak of classified information to a competitor or a customer would account for the cost associated with R&D, marketing, missed sales, assembly worker relocations or lay-offs, lawyers, investigations, etc. which could total millions of dollars. As such, the cost could be calculated as: Σ of all projects or products involved $\{(L \$ \text{R\&D cost}) \times (M \$ \text{marketing cost}) \times (N \$ \text{missed sales cost}) \times (X \$ \text{relocation or lay-off cost}) \times (Y \$ \text{lawyer cost}) \times (Z \$ \text{investigation cost})\} = \text{dollars spent on data theft}$

Case 10 – Disconnected Integration

All the build issues and part number deviations are currently maintained as Assembly Concerns in The Engineering Change system, an SQL database accessible through the web. Attached to an issue is a reference number, which is recorded in the Build BOMs. The reference number could be connected to an Intranet hyperlink, which would link the Assembly Concern reference number to the online issue. Substitutions however, are presently communicated through Excel templates stored on a network server at the Development Centre. The substitutions are copied to a local folder and thereafter logged in an Excel spreadsheet with a reference number. Therefore the reference number merely refers to information in the logbook rather than the substitution file itself. Fortunately though, the substitutions have been planned to be integrated into The Engineering Change system by September 2006, and will thereafter be given the same type of reference number as the issues. In addition, all part and assembly drawings are referred to with a reference number in the database where they are stored. The reference number refers to the drawing number which could also be attached to an intranet hyperlink. However, to work properly in Excel, any hyperlinks need to be added with the function “HYPERLINK”. As mentioned earlier, information in Excel is easily overridden or formatted differently. Additionally, hyperlinks in an Excel file take up valuable file space, hence slowing down all other processes in the computer. The potential cost associated with disconnected integration contains an unknown parameter, but could be calculated as: $(X \text{ \#of issues per year}) \times (5 \text{ minutes per issue}) \times (2 \text{ document locations}) \times (\$26.00 \text{ per hour}) + (300 \text{ rows changed per BOM}) \times (70 \text{ BOMs per year}) \times (5 \text{ minutes per substitution}) \times (\$26.00 \text{ per hour}) = (\text{\#of issues}) \times \$4.33/\text{year} + \$45,500/\text{year}$

Case 11 – Inaccuracy and Inconsistency

Due to inaccuracy and inconsistency, the EBOM and the Build BOM need to be compared at the end of the build event. The comparison is a tedious task that involves numerous manual steps, performed in a special sequence: The EBOM needs to be extracted from the MRP database, while the Build BOM needs to be formatted in such a way that the BOMs can be combined and compared. Once the BOMs are combined, a formula is applied to the file to guide in the interpretation of the comparison. Regardless of the formula, the person who compares the BOMs has to be quite proficient in Excel. As well, an experienced person needs two hours to complete the comparison and another three hours to resolve the discrepancies in the MRP system, to close the order. The potential cost associated with inaccuracy and inconsistency can be calculated as: (70 BOMs per year) x (2 hours to compare the BOMs) x (\$26.00 per hour) + (70 BOMs per year) x (3 hours to process the BOMs) x (\$26.00 per hour) = **\$9,100/year**

Case 12 – Complexity of Information Flow

The build event is composed of many documents and processes related to each other through intricate connections, as seen in Figure 10 on page 48. These connections of information flow make the processes complex and highly coupled (inseparably joined). The couplings that exist among the processes, and therefore between the documents, make the flow of information from one document to another depend upon the successful transfer of information from yet another document. As an example, information concerning outstanding engineering changes is recorded in the finalized Build BOM. However, the information first has to be copied from the Assembly Concern to the Build

Issues document. It then needs to be transferred to the Product Issues/OK-To-Ship document, and thereafter copied over to the Build BOM. Thus, the flow of information is very fragile as there are many links that might fail to receive or transfer the information. Conversely, if the information is only stored at one location (in one document) and instead referenced to in other documents, the likelihood of failed information flow is less between documents. The coupling of the information flow among the documents is a cause of added complexity to the processes. It is also proven from observations made at the Prototype Centre that complexity is further added to the processes due to excessive use of manual operations, redundant processes, and human interference that cause human and computer errors. In addition, the lack of interconnectivity among critical documents is another source of complexity in the build event processes. The potential cost associated with complexity and coupling is intangible, but accounts for the labour cost of maintaining and using the system, as well as training new staff on the system.

Case Summary

The current system of managing the BOMs causes errors to occur during the build event, which are unintentionally transferred to the system leading to inaccuracies of the BOMs. The inaccuracies accumulate a cost, mostly associated with maintaining and correcting the erroneous data. Over the course of a year, the tangible fraction of the cost can account for as much as \$210,000 in extra work and time lost. Note however that there is also a large fraction of intangible costs, which could account for millions of dollars in a worst-case scenario. A security breach involving data theft is an example of such scenario. Table 2 on the next page shows a summary of the above mentioned cases.

Table 2 – Summary of Cases with Opportunity for Improvements

Case	Description	Concern	Cost (\$/yr)
1	Each Build BOM is a separate file BOMs stored in separate build program folders	<i>Recreation of documents and processes</i> <i>Time consuming search for specific BOM</i>	intangible
2	Limited cross-reference through multiple BOMs No backlog of issues and comments entered in BOM Manual processing of Substitutions	<i>Resolving issues proactively are quite difficult</i> <i>Duplication of efforts at PC and DC</i> <i>Tedious, redundant, and erroneous process</i>	intangible intangible \$43,300
3	No standard operating procedure between PC/DC Multiple versions of BOMs used Duplication of Substitutions	<i>BOMs not used by DC</i> <i>Not reflecting the current As-Built BOM</i> <i>300 Substitutions redundantly processed</i>	intangible intangible \$6,500
4	No integrity of information contents No reliable tracking or verification of changes Cell protection cannot be turned off while file is shared	<i>Any information can be entered or modified</i> <i>Difficult to determine why change was made</i> <i>Added processing and redundancy</i>	intangible \$9,100
5	Redundancy of processing (Build Status) Duplication of data entries (Product Issues)	<i>Information available in other documents</i> <i>Link between source and target data is lost</i>	\$19,200 \$4,500
6	Incorrect BOMs sent to DC to reference test products	<i>Investigations, Rework, Schedule changes</i>	\$10,000 - \$50,000
7	The BOMs are fragile documents Risk of using multiple copies of the BOMs	<i>Easily moved, deleted, changed, or unshared</i> <i>Errors and tedious consolidation</i>	intangible
8	Large BOM file sizes (between 5 MB to 40 MB) Multiple users maintaining shared Excel documents	<i>RAM allocated, slowing down computer processes</i> <i>Escalating file sizes</i>	intangible
9	No restrictions to view the BOM information No user access restrictions on the BOMs The BOMs are easily transferable to external source	<i>Password only to modify files</i> <i>Password gives everyone full access</i> <i>Potential data theft</i>	intangible intangible millions
10	No active links with Assembly Concerns or WorkFlow	<i>Separate process to view information</i>	intangible
11	Consolidation of EBOM and Build BOM requires expert-level of knowledge	<i>Process highly dependable on certain people that require specialized training</i>	\$9,100
12	Complex flow of information between documents Coupled design of data management system	<i>System is difficult to understand and use</i> <i>System is fragile and causes errors to occur</i>	intangible

Research Objectives

The objective of this thesis is to engineer the design of a Collaborative Information System providing all the specific requirements needed to deliver a corporate wide solution. The solution will support: planning, extracting, accessing, validating, improving, logging, referencing, linking, and archiving the Bills of Material, collaboratively and securely throughout the entire lifecycle of the prototype products for the corporation's prototype operations. Much emphasis will be placed on reducing the complexity of the assembly processes to deliver a robust and simple solution. In addition, the needs of the stakeholders, such as the engineering and the assembly groups, will be addressed to ensure a design that best suits their needs meanwhile containing the least amount of complexity as well as allowing expansion to accommodate future needs.

Literature Support

A literature review was conducted with the intention to support the findings in the cases discussed above. Supporting literature, mostly relating to information systems, was gathered from the fields of transaction errors, technology improvements, and information and data quality. The review showed that the identified weaknesses of the current information system were not unique to the Prototype Centre. Instead, the findings are commonly recognized throughout the industries that deal with human interactions and with information systems. To summarize the findings, the supporting information from the literature is further discussed in the three paragraphs that follow. Of the three fields of literature, the most important was undoubtedly the one relating to information and data quality, as the facility and its business relies on correct and valid information and data.

Literature Support – Transaction Error

The twelve cases discussed above indicate that the Bills of Material contain many errors and shows evidence for being prone to mistakes and redundant processing, causing the system as a whole to be complex and fragile. However, the scenario at the Prototype Centre is far from unique as much literature shows: According to Piasecki (2003), aside from picking the wrong quantity of parts for an order, the most common and repeated error in a manufacturing [assembly] operation are related to transactions, such as “transactions not being recorded or data entry errors” (pp. 19-23). These transaction-errors may include: missing a line item on an order, entering the wrong number in the quantity field in production reporting, forgetting to enter (save) a transaction, entering a transaction twice, and transposing numbers or letters in an item number or quantity (pp. 2-3). Piasecki also mentions that “most errors can be eliminated through process definition, employee training, and technology” (pp. 20-21). However, although all the above improvements have been made, a company may still face employees who continue to make mistakes (p. 21). The reasons for continued mistakes are explained as due to: ability to learn, pride, long-term employment objectives, and gender, which all are personal and managerial aspects which cannot be easily fixed. As such, to eliminate as much errors as possible without restructuring the organization or discriminating employees, the improvements must be made on the processes and to the technology of the system. Young (1991) acknowledges that the major challenge in designing very accurate inventory systems is “to detect and compensate for human error (and sometimes for some forms of human malfeasance)” (p. 44), as studies show that the rate of error in a typical manual data entry is “about one in every 400 characters” (p. 14). In addition, Eckerson (2002) identified that 76 per cent of data quality problems are due to data entry

by employees (pp. 24-30). It is therefore important to automate the data entry process, and to enter the data in a systematic way, says Alter (1980, p. 130). As such, the quality of the information can be upheld by eliminating or, if not feasible, by reducing the presence of errors and mistakes.

Literature Support – Technology Improvements

As mentioned by Piasecki (2003) and Alter (1980), the proper use of technology and the improvements to processes and systems becomes a vital part of bringing the accuracy of the inventory and the quality of the information and data together. Piasecki states that although “it would make sense that an accuracy improvement effort should focus on the human-machine interface” as most [data] discrepancies are “ultimately caused by human error”, most human-machine interfaces have been designed with a key objective of functionality and not usability (p. 29). This wide-ranging functionality of the ability to meet the needs of diverse businesses results in “a higher degree of complexity than a program designed with a more specific purpose”, such as a legacy system or a custom-made database. To further accentuate the importance of using the proper technology and designing and implementing a real database, Hernandez (2003) gives his expert opinion that spreadsheets should not be used as a substitute for a database when the organization “has a need to collect, store, maintain, and manipulate various types of data” (p. 494). Instead, a spreadsheet software, such as Microsoft Excel, should be “used properly and for the purpose for which it was designed”, meaning “work that involves complex mathematical calculations and statistical analysis” (pp. 493-494). In addition, although using macro (script program) could be a “very inexpensive way to make

accuracy easy”, the macro is not a software, nor a sophisticated computer program, but instead a “script of actions that will occur whenever the macro is initiated regardless of any other factors” (Piasecki, p. 42). Another negative side effect of using a macro is that the macro will disregard the undo function in most software, making it impossible to recover any mistakes caused by the macro. Therefore, if the macro is not properly designed or executed (run) it might do more damage than good.

Literature Support – Information and Data Quality

Sustaining a high level of information quality is especially important when maintaining electronic documents and their contents. Turban, Aronson, and Liang (2005, pp. 218-219) informs that “Data in organizational databases are frequently found to be inaccurate, incomplete, or ambiguous” and that “The economic and social damage from poor-quality data cost billions of dollars”. Laudon, Laudon, and Brabston (2005) indicate that “the most common source of information systems failure is poor data quality”. Strong et al. (1997) organized data quality into four categories and dimensions, shown in Table 3, after an extensive research on data quality problems (Turban, Aronson, and Liang, 2005, p. 220).

Data Quality Categories			
Contextual	Intrinsic	Representation	Accessibility
Relevancy	Accuracy	Interpretability	Accessibility
Value Added	Objectives	Ease of Understanding	Access Security
Timeliness	Believability	Concise Representation	
Completeness	Reputation	Consistent Representation	
Amount of Data			

Table 3 – Data Quality Categories

O'Brien and Montazemi (2004) view the quality of information as a measure of effectiveness in the terms of content, form, and time. The key attributes of information quality is summarized below in Table 4 (adapted from O'Brien et al. 2004, p. 250).

Measure of Effectiveness		
Content	Form	Time
Accuracy	Clarity	Timeliness
Relevance	Detail	Currency
Completeness	Order	Frequency
Conciseness	Presentation	Time Period
Scope	Media	
Performance		

Table 4 – Attributes of Information Quality

Laudon et al. (2005) explain that “data that are inaccurate, untimely, or inconsistent with other sources of information can create serious operational and financial problems for businesses”. The poor data quality is a result of “errors during data input or faulty information system” (p. 366). Simsion (2001, p. 10) states “Frequently, problems with data quality can be traced to a lack of consistency in defining and interpreting data.” and that “...data held in a database is usually a valuable business asset, built up over a long period. Turban et al. (2005, p. 218) also emphasize the significance of data quality: “Data quality (DQ) is an extremely important issue because quality determines the usefulness of data as well as the quality of the decisions based on them”. Inaccurate data (poor *data quality*) reduces the value of the asset and can be expensive or impossible to correct”. Turban et al. (2005, p. 219) show that some of the costs involved in poor data quality include “rework, lost customers, late reporting, wrong decisions, wasted project activities, slow response to new needs (missed opportunities), and delays in implementing large projects that depend on existing databases” (adapted from Olson,

2003a and 2003b). The cost associated with correcting erroneous data in the structure (code) of information systems increases over time as the development progresses. Laudon, Laudon, and Brabston (2005) identifies that the cost can multiply dramatically if errors are not corrected early on in the development stages. “A minor logic error, for example, that could take one hour to correct during the analysis and design stage could take 10, 40, and 90 times as long to correct during programming, conversion, and post-implementation respectively” (Laudon et al., 2005, p. 366). In addition, research shows that the costs of maintaining information systems are very high as a result of “the complexity of the flow of program logic” and “software complexity, as measured by the number and size of interrelated software programs and subprograms” (Laudon et al., 2005, p. 366).

With that, it becomes clear that when using spreadsheet software, such as Microsoft Excel, the data quality can not be upheld as the system is inherently weak in sustaining the integrity of the data. Hernandez (2003) shows that in a relational database, “data integrity is imposed at the field, table, and relationship levels” (pp. 17, 33), as well as with business rules which help guarantee the data consistency and accuracy. The four levels of data integrity in a relational database can be summarized as (pp. 71-72):

- Table-level (*entity*): No duplicated records. Only unique records. No null values
- Field-level (*domain*): Solid field structure. Valid, consistent, and accurate field values. Fields of same type are consistently defined
- Relationship-level (*referential*): Sound table relationship. Synchronized data entry, update, and removal
- Business rules: Restrict or limit certain aspects of the database

As such, the [relational] database can therefore inherently be accurate if designed using sound data modeling techniques. The proper design of the database will ensure that the data and information are consistently of high quality, offering the corporation a more accurate information system. However, to maximize data quality in the business, the following best practices should be used (adapted from Stackpole, 2001, pp. 101-114):

- Data scrubbing is not enough: Approach data standardization
- Start at the top: Top management must be aware of data quality issues
- Know your data: Understand what data you have, and what they are used for
- Make it a continuous process: Develop a culture of data quality
- Measure results: Regularly audit the results to ensure standards are enforced

In addition, as Laudon et al. (2005, p. 367) point out: “To minimize errors, disasters, interruptions of service, computer crimes, and breaches of security, special policies and procedures must be incorporated into the design and implementation of information systems”. These policies (controls) must be an integral part of the company.

CHAPTER III

REVIEW OF LITERATURE

Rationale for Literature Review

The main focus of the research was to engineer an in-house information system that would be proven to have a robust design and have the least amount of complexity. As the information system would replace multiple documents used in a prototype facility within a larger corporation, it would have to provide and encourage collaboration. In addition, the information system would also have to be tailored to accommodate the specific needs of the stakeholders. As such, the literature survey was broken down into four topics: Quality Function Deployment, Robust Design, [System] Complexity, and Information Systems Design. The graph shown in Figure 6 represents the findings from the literature survey in relationship to the different topics, as well as indicates the opportunity gaps where this thesis would make a unique contribution.

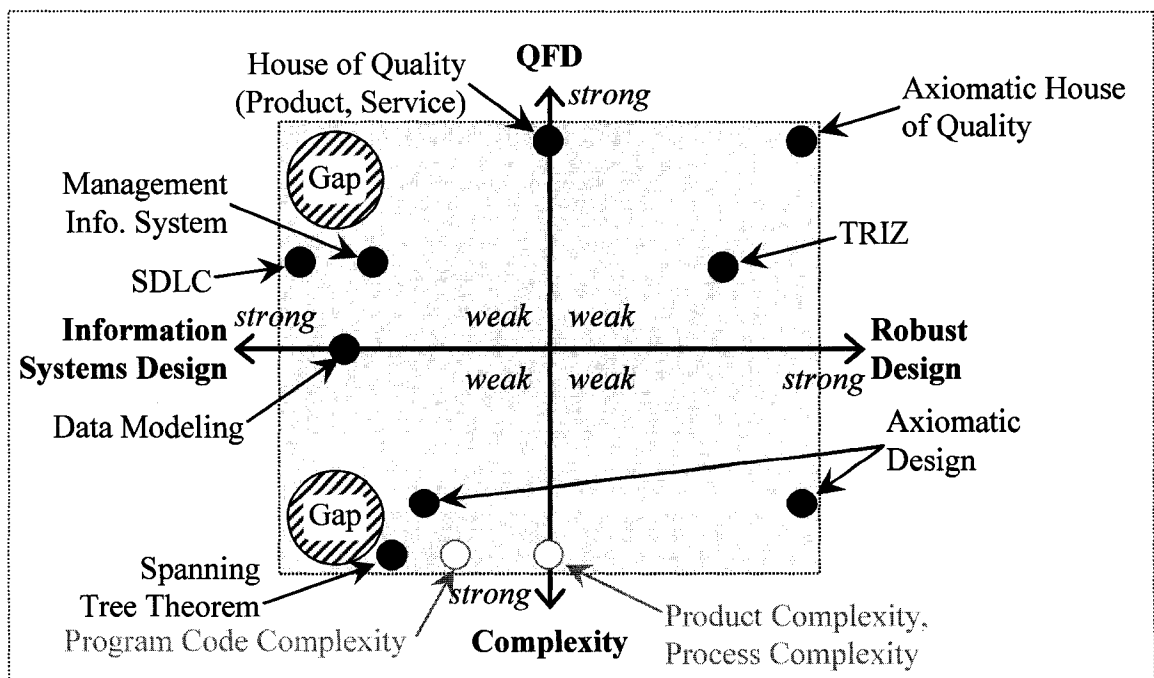


Figure 6 – Literature Survey with Opportunity Gaps

The graph can be viewed as having four quadrants, each with two topics. In the graph, the filled dots represent current literature that covers the topics as indicated, with either a weak or a strong relationship. The indicated literature concentrates on the topics of House of Quality, TRIZ, Axiomatic Design, Axiomatic House of Quality, Spanning Tree Theorem, Data Modelling, and Management Information System. The shaded dots, on the other hand, represent current research that fit into the sections of Information Systems and Complexity in the literature survey, but have a focus on applications or systems not applicable to this thesis. The discussion that follows elaborates on the findings.

Quality Function Deployment

The history of implementing a Quality Function Deployment (QFD) dates back to 1972 in Kobe, Japan, when two engineers, Nishimura and Takayanagi, and two consultants, Mizuno and Furukawa, developed a matrix based quality chart for the shipyards of Mitsubishi Heavy Industries Ltd. (Franceschini, 2002, p. 21). However, QFD had been written about in articles in Japan since 1967 (Kogure and Akao, 1983, pp. 16, 25-29). QFD appeared in the United States in 1986 as a result of the “commitment of Don Clausing, professor at Massachusetts Institute of Technology” (Franceschini, 2002, p. 22). Franceschini explains: through the research Clausing conducted at Fuji Xerox Ltd., QFD was introduced at Ford Motor Co. Later a “series of study missions” were organized to Japan through the American Supplier Institute (Franceschini, 2002, p. 22). However, the credit for the well known development matrix used in QFD, the House of Quality (HOQ), is given to Toyota who introduced HOQ in their product design process (Suh, 2001, p. 14).

QFD was developed to capture the quality standards used in the Japanese product development industry. As such, most implementations of QFD to date have involved the development of physical products, although the technique allows QFD to be successfully used for other [intangible] products, such as software, as well as other industries, such as service (Franceschini, 2002, p. 24). The development process of QFD is thus very flexible as “QFD is a customer-driven process for planning products and services” (Pysdek, 2003, p. 133). Today, QFD is commonly used as a well established design methodology in larger industries: “Many companies, especially in Japan and in the United States, have benefited from QFD in that it has been instrumental in achieving notable improvements in planning cycles while at the same time attaining reduced product development and costs.” (Franceschini, 2002, p. 32). This is due to the common awareness that money spent during the design phase accounts for about 75% of the overall manufacturing cost, for about 70% of the life cycle cost, and for more than 80% of the quality features, while only contributing to an average of 5% of the total cost of the product (Franceschini, 2002, p. 5).

In recent research, QFD has been used to aid in the development of software, through the modified “Software QFD”. However, the Software QFD is used to transform business requirements into coding, thus focuses only on the planning and development of the software code as its guidelines are closely connected with both the ISO 9126 software quality standards and the IEEE 830:1998 software requirement standards (Zrymiak, 2003). As such, no current research can be found that uses QFD in the development of an information system or a database on a systems level (the logical flow of information).

Robust Design

According to Dhillon (2006, p. 2), robust design, in the context of reliability engineering, appeared for the first time in U.S. literature in 1957 with a report from the Advisory Group on the Reliability of Electronic Equipment (AGREE). However, the history of reliability dates back to the 1940s in Germany when the reliability concept was used to improve crucial military equipment. With that, most applications involving the theory of reliability were first to be found in the military and electronics industry.

Both reliability and robustness deal with uncertainty, although in slightly different ways. In engineering, reliability deals with uncertainty of the design solution (failure of the design), whereas robustness deals with uncertainty of the design parameters (the data about the product) (Snyder, 2003, pp. 3-4). The robustness can be seen as a product or process that performs as intended even during less than ideal conditions. The variations that negatively affect the product or process to a non-ideal condition are usually referred to as noise (Ulrich and Eppinger, 2004, p. 266).

Axiomatic Design (AD) encompasses both robustness and reliability through the Independence Axiom: the reliability is ensured as the failure of one Design Parameter (DP) does not affect any other Functional Requirements (FRs) than that of the failed DP (Suh, 2001, p. 16-17, 95). When compared with other design methodologies, such as Statistical Process Control (SPC), the Taguchi method, and Theory of Inventive Problem Solving (TRIZ), “Axiomatic Design deals with principles and methodologies rather than with algorithms or tools” as used by the compared methodologies, explains Suh (2001, p. 57-58). However, if the design does not satisfy the Independence Axiom, a robust design cannot be achieved using the Taguchi method (Suh, 2001, p. 58).

The Taguchi method uses the Design of Experiments (DOE) methodology to ensure the quality is upheld in the design of the product. To allow the focus on quality to be combined with the design for robustness, Manchulenko (2001) identified that Axiomatic Design could be applied to the House of Quality, as both AD and HOQ are built upon Functional Requirements and Design Parameters placed in a matrix structure. In addition, Chakrabarti Ed (2002, p. 142-143) also showed how TRIZ can be integrated with other design methodologies, such as QFD, during the Collect, Create, Construct, and produce (4C's) design phase to bring robustness to the design process.

As with Quality Function Deployment, Robust Design has mainly been used for designing physical products. Nevertheless, "Axiomatic Design is applicable to all designs: products, processes, systems, software, organizations, materials, and business plans." (Suh, 2002, p. 58).

Complexity

According to Nicolis and Prigogine (1989, p. 8), complexity as we know it today was born from the 1960s revolution in both mathematical and physical sciences, affecting the view of topics such as thermodynamics and classical mechanics. Both studies brought new insight to the respective fields of science, causing the gap between "simple" and "complex", and between "disorder" and "order" to be much narrower than previously believed. Nevertheless, complexity had been talked about before the sixties, seeing that "initial work in complexity theory in the late 1920s and early 1930s was concerned with subclasses of the effectively computable functions" (Jones, 1997, p. 24). However, although numerous efforts have been made to define complexity, there is still no common

definition of complexity accepted throughout all sciences (Suh, 2001, p. 470). In engineering, complexity is today generally described as a system where “a great number of interacting elements are involved” (Nicolis and Prigogine, 1989, p. 6). Sometimes, complexity is also referred to as something that acts with randomness or something that is chaotic. In addition, complexity can be directly related to the size or scale order of systems: very small or very large systems tend to be more complex than those that are closely related to the size of our immediate environment (a virus is viewed as a complex biological system, while a galaxy is seen as a complex astronomical system). This complexity however, is referred to as imaginary or cognitive complexity.

In Axiomatic Design, complexity is related to information, which is defined as “a logarithmic function of the probability of achieving the specified functional requirements” (Suh, 2001, p. 471). As such, Suh classifies complexity into two categories and four sub-classes: time-dependent complexity (combinatorial and periodic) and time-independent complexity (real and imaginary). Complexity can thus be defined as a measure of uncertainty.

In information systems, Simsion (2001) states that “The most common communication problems arise from high level of complexity, new concepts, and unfamiliar terminology.”, showing that even a data model of twenty or thirty tables will be “overwhelmingly complex for most non-specialists” (p. 15).

Studies show that there are more than thirty-five different ways the word complexity is used by scientists (Suh, 2001, p. 470). With that, there are just as many or more ways to measure complexity. Therefore, there was a need of finding a quantitative way to measure the complexity of the system intended to be studied in this thesis.

Guenov (2001) discusses the underlying structure of complexity when placed in a relation with the “architectural design process of composite systems”. Guenov’s study proposes an estimate on measuring the complexity of designs when presented in an Axiomatic Design matrix. Thus, the complexity between Functional Requirements and Design Parameters is measured within the matrix to facilitate a comparison between different design proposals. Nevertheless, the result is restricted to measure the level of coupling in an AD matrix in the sense of complexity induced to the system.

Latva-Koivisto (2001) conducted a study on business process models to find a measure for its structural complexity. In the report, Latva-Koivisto evaluates different complexity measures, such as: Coefficient of Network Complexity (CNC), Cyclomatic Numbers (S), Complexity Index (CI), Restrictiveness Estimator (RT), and Number of Trees in a Graph (T). The study shows that T, when used with a logarithm, can be applied to process graphs to produce a quantifiable measure of its complexity. Another important finding was that the value of T increases as a graph becomes intuitively more complex, which shows an imperative relationship between analytical and cognitive complexity. That however, was not the case with CNC, S, and CI.

Throughout the literature review, the most difficult task was undoubtedly to measure the level of complexity in any given system. The most promising measure of complexity at the time of the literature review seemed to be “T”, the Number of Trees in a Graph. Although, proof needed to be found that T would accurately measure the complexity in the system as devised.

Information Systems Design

Data modeling is an essential part of today's systems design as most of the information systems recently developed contain some form of database. In the early 1970s, there were three kinds of database structures: hierarchical, network, and relational. Today, the most common structure of a database is the relational database, developed by Dr. Edgar F. Codd in 1969 while he was working at IBM (Hernandez, 2003, p. 12). The theory of data modeling can thus be traced back to the late 1960s when the commercial use of database management systems emerged. Nonetheless, the basic concepts of data modeling have changed very little since then (Simsion, 2001, p. 28). The difference is that; today, organizations buy packaged software (such as Microsoft Access or Enterprise SQL) when they have a need to develop a database. Before, organizations developed the database system in-house, as a legacy system using a coding language such as FORTRAN. However, as Simsion (2001, p. 29) points out: "Owning a sophisticated tool is not the same thing as being able to use it effectively, and much time and effort is wasted... attempting to build applications without an understanding of basic design principles".

The main focus in today's literature on information systems development fall under two categories: systems design and database design. Although database design is an integral part of most systems design, its methodology has a different focus than that of systems design. In database design, the literature can currently be separated into three areas: how to design a database (data modeling), how to build a database (code programming), and how to maintain a database (management information system). Conversely, systems design is a management tool with a focus on the design of a system as a whole. In conventional development and design of information systems, specialists

are custom to use the methodology of the Systems Development Life Cycle (SDLC), also known as the Information Systems Development Cycle. SDLC generally consists of four phases which are closely related to the stages of a systems approach. Although, sometimes the last phase is divided into two separate steps, making the SDLC consist of five steps. The traditional SDLC methodology is shown in Figure 7 (adopted from O'Brien and Montazemi 2004, p. 332).

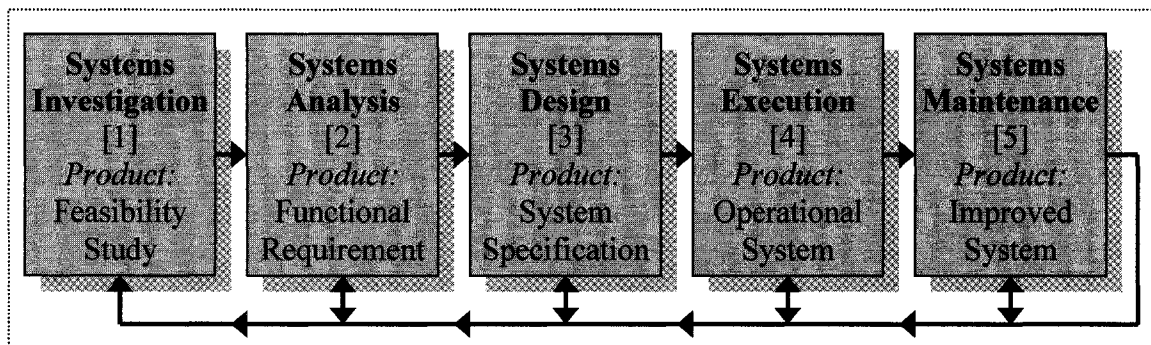


Figure 7 – Systems Development Life Cycle Steps

As seen in the SDLC process, each step generates a product which, if found inadequate, can be redefined by recycling back to any previous step, if more work should be deemed necessary. During a systems development, the SDLC process frequently takes the form of one of four approaches (Valacich, George, and Hoffer, 2004, p. 26). These approaches are Prototyping, Rapid Application Development (RAD), Joint Application Design (JAD), and Participatory Design (PD). However, the systems design approaches can not be proven successful with a failure rate of up to seventy percent (Laudon et al., 2005, p. 415) on all business reengineering projects, contrary to most engineering design methodologies that have well established procedures and a history of successful implementations throughout the industry. Laudon et al. (2005, p. 365) also points out: “Studies show that about 60 percent of errors discovered during testing are a result of specifications in the design documentation that were missing, ambiguous, in error, or in

conflict". Therefore, to increase the rate of success and to establish an analytical approach, tailored to fit the design of information systems, it would be beneficial to incorporate the use of engineering design methodologies, such as QFD, as a substitute to the traditional systems design tools.

As an example of using engineering design methodologies in the design of information system, Suh (2001) shows how Axiomatic Design can be used in software design (p. 239). Nevertheless, the method places more focus on the software architecture and the computational sequence than the flow of information and interactions between people, hardware, and software.

As such, no literature can be found describing the use of a combination of engineering design methodologies for the development of information systems. In addition, no common approach can be found that makes the development of information systems solid while following a quantifiable approach with a focus on low complexity.

Summary of Literature Review

Throughout the literature review it became apparent that although design methodologies, such as Quality Function Deployment, Axiomatic Design, and Theory of Inventive Problem Solving, have been used successfully in many industries, its implementation is almost entirely found in the development of physical products, mechanical applications, or industrial processes. Documentation of similar methodologies for the development of information systems is more or less insignificant in comparison. Likewise, literature on robust designs has been published in the fields of mechanics and product development since the late 1950s. However, no indications can

be found that the concept has been implemented to the development phase of information systems.

Quality Function Deployment is a matrix-based design methodology with a focus on meeting customer demands, leading to increased quality of the products. Robust Design on the other hand, has a focus on ensuring that the functionality of the product is satisfied, even during unpredictable circumstances. The Axiomatic House of Quality (AHOQ) is a novel approach, bringing the quality and the robustness together as a design methodology. However, the AHOQ approach is described very general in literature without any real-life implementations exemplified. Nevertheless, the methodology seems very promising and should be examined in more detail to allow it to gain recognition as a valuable design methodology if proven useful.

When designing systems that interact with humans it is important that the design is as simple as possible. As such, there is a focus on complexity in various fields of design and development. However, the complexity that is of interest in literature deals with the interaction between physical parts in products or processes, or with the computability of software codes. To date, the information to be found on how to effectively and easily measure the complexity of the design intent of information systems is almost insignificant.

Thus, there is an opportunity to use the vast knowledge from the methodologies used in Quality Function Deployment and Robust Design, in combination with pioneering ways to measure the Complexity of a system, to develop a quantifiable step-by-step approach to the design of [robust and simple] collaborative information systems.

CHAPTER IV
DESIGN AND METHODOLOGY

Design Framework

The implementation plan for the project objectives can be visualized by using the Integration Definition method (IDEF). There are different levels of the IDEF methods, thus the project plan for this thesis is illustrated in Figure 8 as an IDEF₀ diagram, showing the model of the functions and activities. The intended outcome of the project can be derived from the IDEF₀ diagram: an improved process of handling the prototype assembly process documents. The diagram also shows the control functions that would impact the improvements of the system, namely the corporate standards, future needs, and the boundary, or scope, of the research project. As control functions, the corporate standards are items such as selection of available software, corporate security policies, ISO-9000, and design guidelines, whereas the boundary contains restrictions on which processes and documents would be included in the project. These restrictions are discussed in the chapter Feasibility Study on page 51.

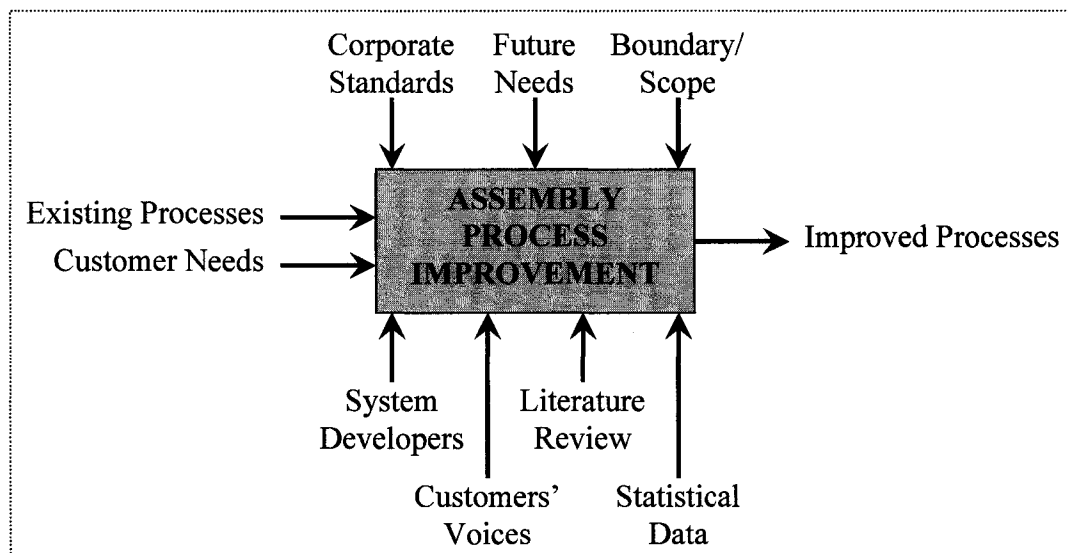


Figure 8 – IDEF₀ Diagram of Project Plan

The IDEF₀ diagram also indicates the resources that were available for the project. These resources came from the System Developers, in form of knowledge about the current system and their expertise in future developments, the voice of the customers, review of literature, journal papers, and theses, and statistical data captured at the Prototype Centre. As illustrated in the IDEF₀ diagram, the design process, and its result, was dependent upon various inputs, controls, and resources, which would impact the engineering system differently at all stages throughout the design and evaluation of the project. It was therefore essential to not only effectively use the appropriate engineering design methods, but also to combine different engineering design methods to achieve the desired result.

Engineering Design Methodologies

To improve data quality, a business improvement process designed to identify and eliminate the root causes of poor-quality data must be implemented. The IDEF₀ diagram in Figure 8 shows that the project involved reengineering the existing processes, indicated as input to the project plan. Therefore, the current processes and documents used in the prototype build event had to be detailed and categorized to bring an understanding of their purpose, contents, pattern of usage, and connections to other documents. An information flow diagram was used to map those connections, as flow of information between documents and processes. In addition, as the project was based upon an existing system, a Quality Function Deployment (QFD) tool, namely the House of Quality (HOQ), was thus an appropriate engineering design method to implement into the methodology of the initial stages of the project. The main intent of the HOQ is to indicate which Customer Attributes (CAs) are sought after the most. The CAs are

converted into Functional Requirements (FRs) and thereafter related to existing Design Parameters (DPs) that are known to be able to accommodate the CAs. As the HOQ only relates the CAs with known DPs, another design method was needed to redesign, or transform, the existing system into the desired end result. Here it seemed most logical to implement design principles from Axiomatic Design (AD) and Theory of Inventive Problem Solving (TRIZ) in combination with the HOQ. TRIZ as a methodology has the ability to radically change the design of a system through contradicting statements. Therefore it is useful when attempting to combine features such as “Security” and “Access”. Similarly, Axiomatic Design would also aid in the development of the new system, as it gives insight on the level of coupling of the FRs and the DPs. Coupling in AD is associated with the amount of DPs that are dependent on multiple FRs to function, making the DPs share FRs with other DPs and thus disrupting the robustness. Figure 9 illustrates the engineering design methodologies that were used in this project, and are discussed in the paragraphs that follow.

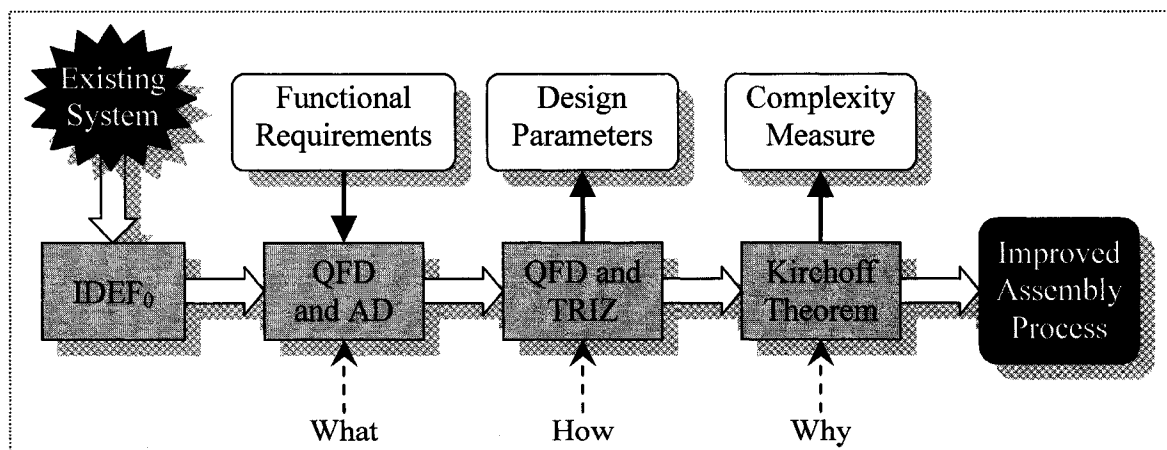


Figure 9 – Engineering Design Methodologies

Once an improved system had been designed, its complexity level had to be measured to determine whether the new system really was of less complexity when

compared with the original. Kirchoff's theorem allows the complexity of graphs to be measured, and thus compared and evaluated. The theorem and the results of measuring the complexity are discussed in the chapter "Analysis of Results" on page 83.

Information Flow Mapping

In order to effectively and accurately assess the processes and documents involved in the build event, information had to be gathered to gain knowledge about the document contents and patterns of usage. The first step was to create an information flow diagram, to visualize the flows of information between documents and entities. The information was gathered through interviews with the users of the documents, and through using, or having used, the documents throughout the various stages of the prototype build event as discussed in the introduction. Much knowledge was also gained from developing and reengineering many of the documents to fulfil the needs of the managers and employees at the Prototype Centre. Once a clear picture of the processes and documents was established, an information flow diagram was developed. The detailed information flow diagram shows, at the time of the research, each process, procedure, and document used during the product assembly at the Prototype Centre. The diagram, shown in Figure 10, was created after having meticulously examined all entities as discussed above. The ordering system is shown in a simplified view in the diagram as it was discussed in more depth in the paragraph "Product Order Process" on page 3. Nevertheless, the simplified view is sufficient for the purpose of identifying and illustrating the documents and information flow, as the ordering system would remain in its original design. Neither would it be improved upon within the scope of this thesis.

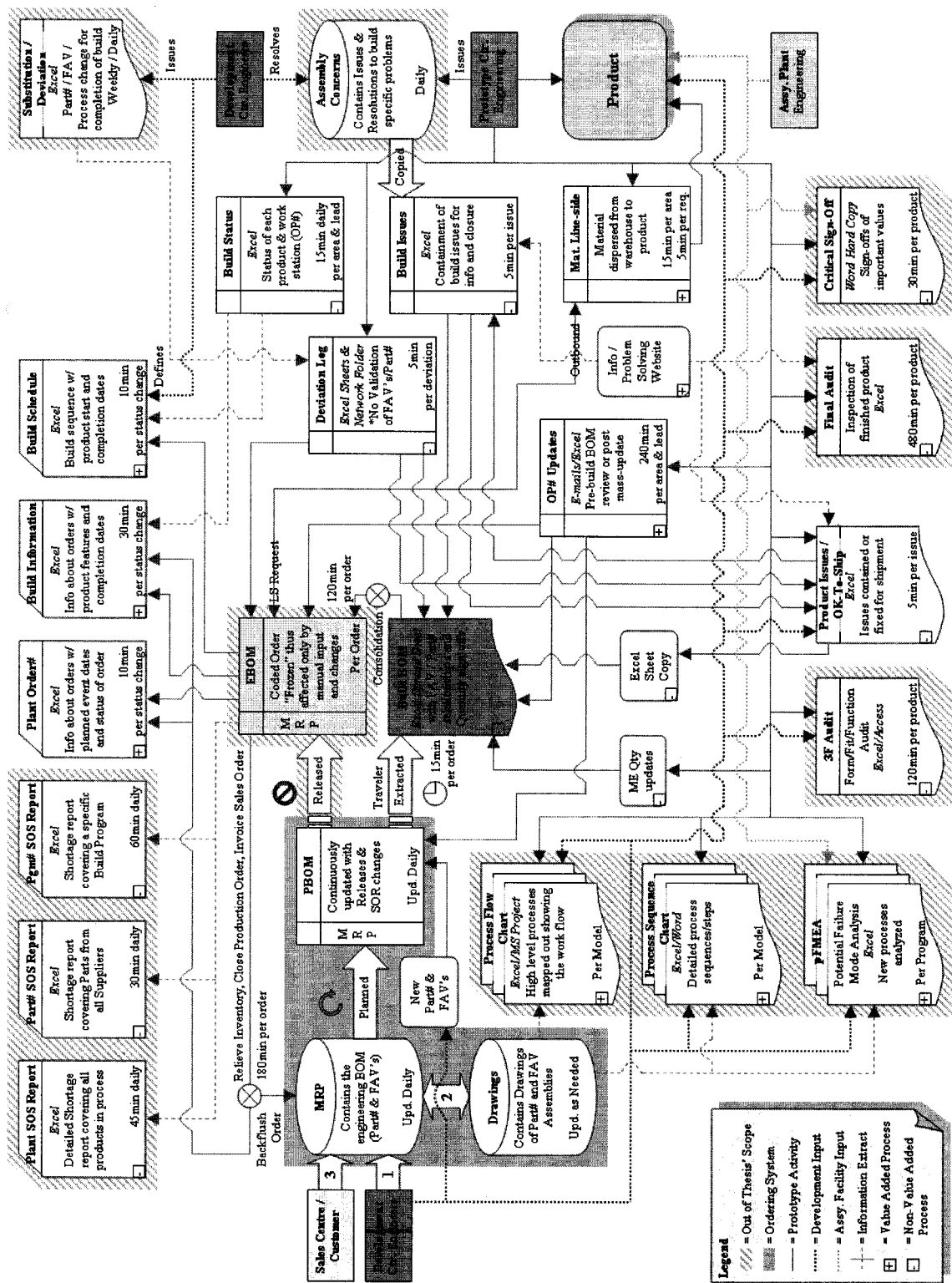


Figure 10 - Current Information Flow Diagram

As shown in Figure 10 there are basically six types of entities involved in the prototype build event: processes, databases, electronic files, reports or schedules, hard copies, and multi-page hard copies. Each entity is represented by a unique icon in the information flow diagram. The icons are illustrated by the flowchart shapes in Figure 11.

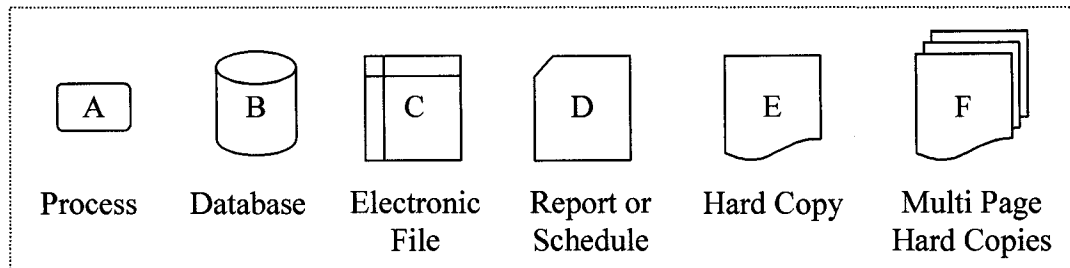


Figure 11 – Information Flow Entity Icons

In addition to the icons, the documents themselves are categorized and labelled to give an overview of its contents, storage, and usage. The legend to the categorization of the documents is shown in Figure 12.

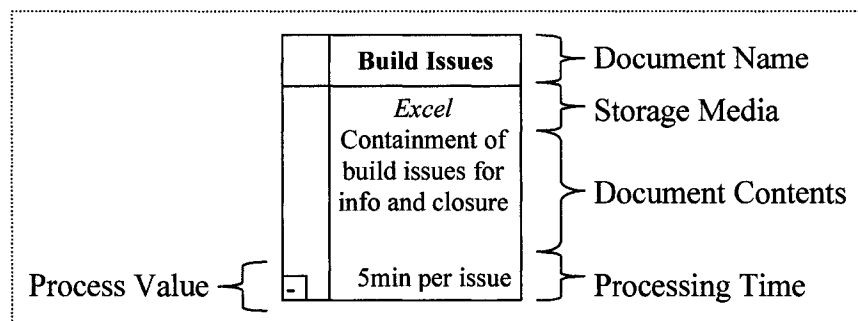


Figure 12 – Document Categorization Legend

The labelling scheme of the document icons is conducted as follows: The bold text on the top shows the name of the document. The italic text underneath shows the storage media used for the document. The text following thereafter describes the contents and purpose of the document. At the bottom is a time indication as well as a per unit value which shows the frequency and time allocation of maintaining or updating the document. Finally, in the bottom left corner of each document is a square with either a plus sign [+] or a minus sign [-]. The sign indicates whether or not the document adds

value to the assembly process. The documents are also described in more detail in the next chapter.

The arrows connecting the documents in Figure 10 show flows of information, such as inventory data, engineering changes, quantity allocations, schedules, etc. Some information however, is extracted electronically (shown with a dashed line) whereas other information is updated by a person (shown with a solid line). In addition, there is also information that flows as a means of electronic or verbal communication from the Development Centre and/or the assembly plants (shown with a dotted line). The shaded and hatched areas are not included in the scope of the thesis. However, recommendations on how to improve those processes are found in the concluding remarks of this paper.

Document Categorization

The documents and electronic files included in the information flow diagram previously discussed were thoroughly identified, categorized, and documented. By doing so, each document could be scrutinized by evaluating the documents based on a set of predetermined categories. The categorization of the documents would not only become the framework for the improvements and design solutions, but also act as blueprints for the detailed design that resulted from the research and analysis. Some categories used for evaluating the individual documents are: type of media, source of original information, flow of information, storage point, access point, update and review occurrences, document lifespan, archiving point, function of document, etc. See Appendix C for the template used for the individual categorization.

Feasibility Study

An overview of the feasible improvements to the prototype build event processes and documents could be established from the information flow diagram and the document categorization. This feasibility study showed that some processes were currently aligned with ongoing or future initiatives within the corporation. For example, the Substitution process at the time of the research was undergoing integration with the Engineering Change database at the Development Centre and was completed and operational by the end of 2006. The Build Issues process was also planned to be integrated into the database, but its implementation had tentatively been scheduled for early 2008. In addition, the Process Flow and Sequence Charts were currently under development at one of the assembly plants. Therefore, to prevent redundancy in efforts, but instead take advantage of the corporate approved projects, the previously mentioned processes were excluded from the objectives of this thesis.

Some processes were also either necessary to remain in its given form, or had little or no direct relationship with the other processes. For example: the Critical Sign-Off is a legislated document that needs to be signed by the assembly workers, hence it is required to be a hard-copy document. The 3F Audit has only been conducted during one build program and might never be requested again according to the prototype build manager at the Prototype Centre. In addition, the Final Audit is performed when the product is fully assembled, thus making it difficult to reference a specific part number or installation. Therefore, those processes were also disregarded from the scope of the thesis. As a result of the above discussions, the prototype build processes and associated documents this thesis had its focus on are shown in Figure 13.

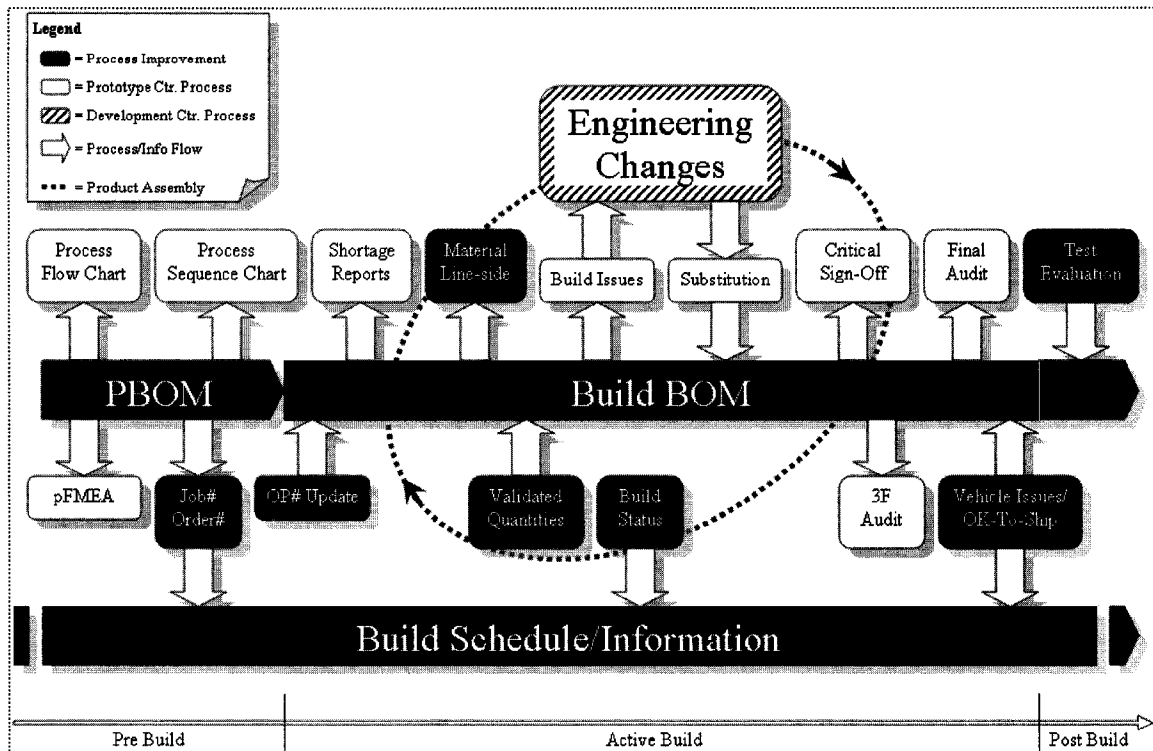


Figure 13 – Processes to Improve

As mentioned, processes currently aligned with ongoing or future initiatives within the corporation, as well as processes with very low reoccurrence or with limited relation to other processes were omitted from the scope of this thesis. However, recommendations for improvements to the excluded processes and documents will be included in this thesis. In addition, the omitted processes certainly allow for improvements in future projects and should be revisited at a suitable opportunity.

Quality Function Deployment

The literature reviewed in the chapter of Information and Data Quality shows that sustaining a high level of quality is essential to the success of maintaining electronic documents, especially in a collaboration system. However, the quality must also be assured when designing for these systems. Kenneth Crow is president of DRM

Associates in Palos Verdes, California and a publisher of a product development Internet forum. On his website, much information can be found on various topics of product development, such as Design for Manufacturing and Assembly, Design to Cost and Cost Management, and Quality Function Deployment (QFD). In the introduction to QFD Crow states that: "Quality must be designed into the product, not inspected into it". As well "Quality can be defined as meeting customer needs and providing superior value. This focus on satisfying the customer's needs places an emphasis on techniques such as Quality Function Deployment to help understand those needs and plan a product to provide superior value". Therefore, QFD was used in this research to address the needs of the customers, to ensure a design with attention to quality and customer value. When constructing and analyzing a HOQ, the methodology follows a step-by-step approach. However, the steps are mainly intended for the design of physical parts, thus the steps were changed to fit the intent of this research. For example, it is recommended to have the customers rate the competition, to evaluate how well the company stands against its competitors in accommodating the needs of the customers. This step was removed as the intent of this research was to reengineer a current system, in-house. There is also a step in which the direction of improvements are indicated. This step was changed to specify the desirability of each technical descriptor. By showing the desirability, it was possible to instead give an indication of whether a certain design, although able to satisfy a requirement, was desired or not. Moreover, as the project would be a redesign of an in-house system, the step in which competitors' products are analyzed was changed to a technical evaluation of available products, to allow different solutions to be evaluated side-by-side. The step indicating target values for the technical descriptors was also

removed, as the descriptors did not need to be weighted: the descriptors were chosen knowingly that they would meet the requirements. Lastly, the roof of the HOQ, the correlation matrix, was also removed because the system was intended to be designed using methodologies from Axiomatic Design, in which the optimal solution does not have correlations between the technical descriptors. Therefore, the steps that were used to develop the HOQ for this research are:

Step 1. Customer Attributes (CAs) – “Voice of the Customer”

Step 2. Regulatory Attributes (RAs) – “Voice of the Experts”

Step 3. Functional Requirements (FRs)

Step 4. Customer Importance Ratings (CIR)

Step 5. Design Parameters (DPs) – “Voice of the Engineer”

Step 6. Desirability

Step 7. Relationship Matrix

Step 8. Importance Weighting (IW)

Step 9. Technical Evaluation of Available Products

Note that some of the names have been changed to standardize the notation with the other design methodologies.

Voice of Customer and Experts

An essential part of QFD is, as mentioned, the needs of the customers. These needs, the voice of the customer, can be captured in many different ways. According to Kenneth Crow, the industry commonly uses direct discussion or interviews, surveys, focus groups, customer specifications, observation, warranty data, and field reports, to get

an understanding of the customers' needs. The list of CAs below resulted from interviewing the assembly line workers, the plant managers, the engineering group, the test group, the BOM augmenters, the material schedulers, and the BOM validators, whereas the RAs resulted from my personal experience of developing many of the documents used throughout the prototype build event. The list shows the CAs and RAs:

1. Information must be kept securely and from unauthorized usage.
2. Access to key data must only be granted to certain users.
3. An historical view of information changes should be made available
4. Data entries should be defined as to what data type and size is allowed.
5. Users must be able to simultaneous access and modify the information.
6. The system must be accessible throughout the entire corporation.
7. Information should be presented differently to users depending on user and type of information requested.
8. Users must be able to cross-reference information from all BOMs.
9. The information kept in the system must be reliable and accurate.
10. Information should be accessible through a one-point entry to the system.
11. Information relating to other information should be accessible within the system.
12. BOM modifications should be initiated directly from the BOMs.
13. Identical BOM modifications affecting different BOMs should be combined and processed simultaneously in all BOMs.
14. Assembly milestones should be automatically indicated on the assembly schedule.
15. Redundant processing should be avoided and minimized as much as possible.

Functional Requirements

To follow the normal notation of the HOQ, the CAs and RAs had to be converted into Functional Requirements (FRs) as the CAs and RAs were written in sentences stated as a want, need, or must. The FRs should be stated as briefly as possible, and should also be divided into subgroups that meet similar functionalities, such as in this case “security”, or into subgroups in which the requirements are used by the same functional group of people, for example “managers”. The following list is the result from converting the previously listed CAs and RAs into corresponding FRs:

1. Provide Secure Access to Information
2. Define User Access to Certain Key Data
3. Record User Activity
4. Define/Restrict Data Type Entries
5. Allow Multiple Simultaneous Users
6. Allow Corporate-Wide Access
7. Provide Customized Views of Information
8. Provide Simultaneous Visibility to all BOMs
9. Provide Reliable/Accurate As-Built BOMs
10. Provide One-Point Entry of all Processes
11. Provide Direct Link to Relating Information
12. Initiate Modifications Directly from BOMs
13. Simultaneously Update BOM Modifications
14. Automatically Log Assembly Milestones
15. Minimize Redundant Processes

As mentioned before, the FRs should be separated into subgroups as logical clusters, making the evaluation process more lenient. From the list of FRs above, three subgroups were identified into which the FRs could be divided. The first six FRs are in some way related to accessing the system, either securely or restricted. FR seven to eleven are associated with using the system, in the sense of reliability and usability. The last four FRs are connected to the processing of the system, by automated functions. The following list shows the three subgroups that were used to categorize the FRs:

- FR1 to FR6 belong to subgroup “Access”
- FR7 to FR11 belong to subgroup “Use”
- FR12 to FR15 belong to subgroup “Process”

The FRs discussed above and the correlating subgroups initiate the design of the HOQ.

House of Quality

The HOQ is a well established design methodology which has been used in many engineering applications. One great advantage of the HOQ, aside from the ability to design with the customer in mind, is the ability to reconfigure the rooms of the matrix to meet the specific needs of the design project. The basic structure of the HOQ is illustrated with the schematic view as shown in Figure 14 (derived from Logan and Radcliffe, 1997, p. 107). To reiterate the changes to the traditional HOQ matrix in Figure 14, the shaded areas (the correlation matrix and the competitive evaluation) have been removed. Added however, are the desirability row and the technical evaluation matrix. The technical evaluation matrix allows the different systems to be compared and ranked, and thus indicates where focus should be placed within each system.

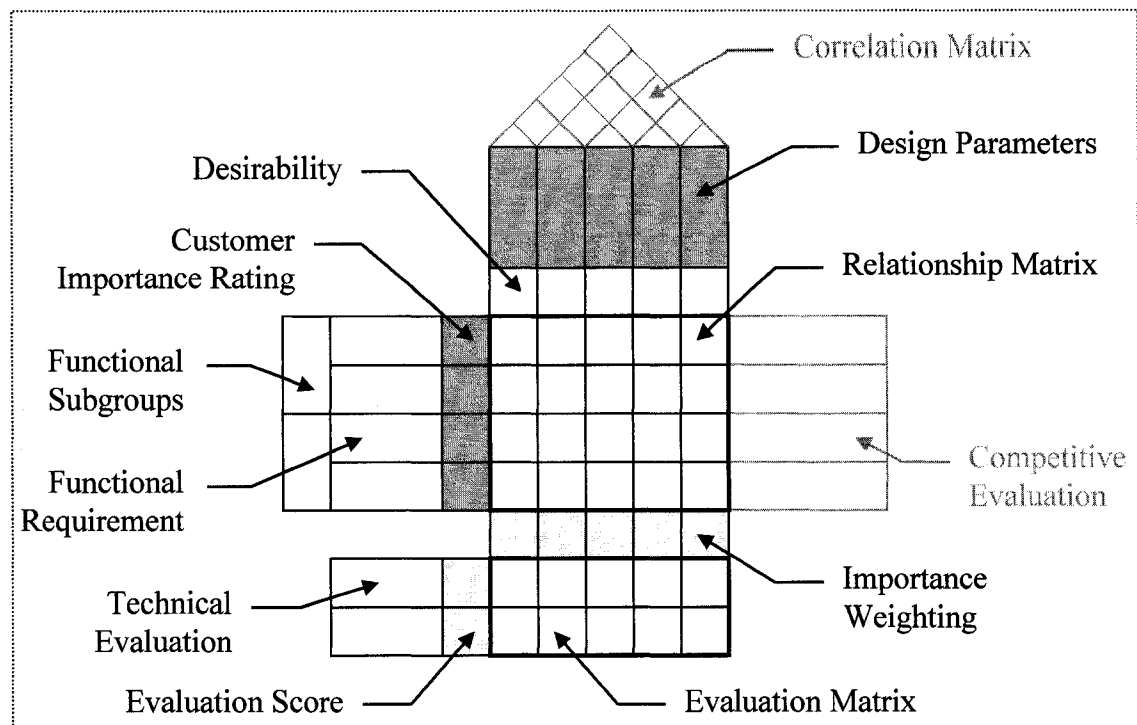


Figure 14 – House of Quality Scheme

As previously mentioned, the FRs were divided into three subgroups and were given an individual priority as a Customer Importance Rating (CIR), denoted 1, 2, or 3. The CIR was based upon the level of importance as seen by the customers as well as in conjunction with the knowledge about some regulatory requirements. Figure 15 shows the complete HOQ matrix for the prototype assembly process improvements.

Once the FRs were derived from the CAs, the corresponding DPs were identified and listed in the table. Note that there are DPs from both the existing system as well as from future possible design solutions. It should also be noted that this procedure is one of the weaknesses with the HOQ: identifying DPs. The process of identifying DPs is not clearly defined and does not promote innovative or creative solutions, but in contrast merely lists the possible design solutions known to fulfil the desired FRs. Therefore, the HOQ shown in Figure 15 has sometimes more than one DP to each relating FR. However, as this is a somewhat traditional HOQ, this was beneficial and sought after as

the purpose of the HOQ was to indicate the importance of the different DPs, as well as to evaluate and grade the different possible design solutions, or DPs.

Design Parameters (DPs)		#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
		Customer Importance Rating (1,2,3)	Data Encryption	Firewall	Login User-ID	Login Password	History Log-File	Data Field Formatting	Shared Documents	Granted Server Access	Graphical User Interface	Linked Documents	Tabulated Data Storage	Time/Event Driven Processing	Intranet Hyperlink	Script Programs	Visual Basic Macro	Copy and Paste Information
Functional Requirements (FRs)		Desirability ↑=high, ↓=low	↑	↑	↑	↑	↑	↑	↓	↑	↑	↑	↑	↑	↑	↑	↓	↓
Access	Provide Secure Access to Information	3	●	Δ	○	●												
	Define User Access to Certain Key Data	2			●	○		Δ										
	Record User Activity	1			○		●	Δ										
	Define/Restrict Data Type Entries	3		Δ	○		●											
	Allow Multiple Simultaneous Users	3			Δ			●	○				●					
	Allow Corporate-Wide Access	3	●	Δ				Δ	●									
Use	Provide Customized Views of Information	1			Δ						●							
	Provide Simultaneous Visibility to all BOMs	3										● ^[3]						
	Provide Reliable/Accurate As-Built BOMs	3					Δ					○	●					
	Provide One-Point Entry of all Processes	1										● ^[3]						
	Provide Direct Link to Relating Information	2										○		●				
Process	Initiate Modifications Directly from BOMs	2										●	○		○	○	Δ	
	Simultaneously Update BOM Modifications	3										●	○		Δ	○	Δ	
	Automatically Log Assembly Milestones	2										Δ	●		Δ	Δ		
	Minimize Redundant Processes	3										○	○		●	Δ	Δ	
Importance Weighting $\Sigma(RL \times CIR)$			27	33	46	33	9	30	33	36	9	18	122	36	24	32	20	8
Rank			10	5	2	5	14	9	5	3	14	13	1	3	11	8	12	16
Evaluation	Excel 2003 ^[1] Current System	49	x	x	Δ	●	○	○	●	Δ	Δ	Δ	x	Δ	Δ	Δ	●	●
	Access 2003 ^[1]	87	○	○	○	●	●	●	x	●	○	○	●	●	○	○	●	○
	Enterprise SQL + Visual Studio .Net ^[1]	121	●	●	●	●	●	●	x	●	●	●	●	●	●	●	Δ	○
^[1] = Product of Microsoft®																		
^[2] = Ability to meet required feature																		
^[3] = Only satisfied through Parameter #11																		
Score ^[2]																		

Figure 15 – House of Quality Matrix

The identified DPs were also given a desirability rating which was either “high”, indicated with an arrow pointing up [↑], or “low”, indicated with an arrow pointing down [↓]. The desirability of the DPs show, even though the DP satisfies the FR, whether that

DP is wanted or not. Sometimes a compromise has to be met, using undesired DPs, to allow a design to function. Therefore, this rating was useful when attempting to decouple the FRxDP matrix to engineer a robust design. However, more of this is discussed later.

The next step of constructing the HOQ was to relate the FRs with appropriate DPs. This was done by marking the relationship between an FR and the DPs in the FRxDP matrix. To rate how well the DPs met a given FR, the DPs were graded on a scale of 1, 3, or 9, where 9 is a perfect match between the FR and the DP. As previously mentioned, there is sometimes more than one possible DP per FR, which creates a highly coupled FRxDP matrix. Nevertheless, the coupling serves the purpose of indicating which DPs the focus should be placed on, as well as identifying redundant, obsolete, or less effective design solution attempts. As an example; the FR “Initiate Modifications Directly from BOMs” has five DPs listed as possible solutions: “Linked Documents”, “Tabulated Data Storage”, Intranet Hyperlink”, “Visual Basic Macro”, and “Copy and Paste Information”. However, only one will effectively generate the desired result, namely “Linked Documents”, as indicated with the High (9) relationship.

Once all FRs and DPs were ranked in the matrix, the DPs were given an Importance Weighting (IW) by multiplying the CIR with the Relationship Legend (RL) grade given for that FR and DP, which thereafter were summarized with all FRs for the ranked DP. As an example, the IW for the DP “Data Field Formatting” was calculated as $(3 \times 9) + (3 \times 1) = 30$. Once all DPs were counted for, they were ranked from first to last, where the DP with the highest IW was given first place, and so on. Thus, the DP “Tabulated Data Storage” was ranked number one as it accumulated the highest IW, and should therefore be noted as the most important DP to focus on.

The last step performed on the HOQ was the evaluation of different technical systems. Normally this would entail a comparison of the company against different competitors. However, in this HOQ the comparison was based upon different software packages from Microsoft, namely Excel 2003 (the system used at the time of the research), Access 2003, and Enterprise SQL with Visual Studio .Net. It should be noted that Microsoft was the only vendor in the evaluation due to the corporate software guidelines, which specifies the approved vendors and products. Nevertheless, in the evaluation the products were rated on how well the software accommodated the listed DPs. As shown in Figure 15, Enterprise SQL with Visual Studio .Net received the highest score and is therefore the best choice for the company.

To summarize the evaluation of the HOQ matrix; more focus should be placed on the Design Parameters that received a high Importance Rating value, such as “Tabulated Data Storage”, “Login User-Id”, and “Granted Server Access” and “Time/Event Driven Processing”, which received the top three IWs in the matrix. It is also important to notice that two Functional Requirements, “Provide Simultaneous Visibility to all BOMs” and “Provide One-Point Entry of all Processes” could only be accommodated by one single DP, namely “Tabulated Data Storage”, which was the highest weighted DP. In addition, “Provide Simultaneous Visibility to all BOMs” was also ranked with a “High” importance by the customers. Lastly, the system most fit to accommodate the FRs (the voice of the customer) was Microsoft Enterprise SQL with Microsoft Visual Studio .Net as that system received the highest score in the technical evaluation. Microsoft Access 2003 would also accommodate the needs, although with compromises.

Axiomatic House of Quality

To reiterate, the HOQ does not promote innovative solutions, nor does it support a means for designing a robust system with the least amount of complexity. Although the solution will work, the design parameters (the elements that make up the solution) are established from either the knowledge of previous and current systems or the level of technical expertise acquired by the designer. There is also a great deal of decision making left to be done after the HOQ matrix is completed. As shown in Figure 15, there are more DPs than needed for the solution, and there are also some “undesired” DPs that affect the outcome of the decision. Manchulenko (2001) identified that “many organizations have experienced problems with the implementation of the current HOQ model” and that “most problems with the HOQ resulted from customer requirement dependencies”. Manchulenko researched on the topic of Axiomatic Design in combination with the House of Quality, and presented his results in his thesis titled “Applying Axiomatic Design Principles to the House of Quality”. In his research he identified a refined engineering methodology, where the rules of the first axiom of AD were used to resolve dependencies among FRs within the HOQ. An axiom is explained by Dr. ElMaraghy as a “truth that cannot be derived, but for which there are no counterexamples or exceptions”. AD was developed by Prof. Nam Suh and contains two axioms. The first axiom declares that the design must “maintain independence of Functional Requirements”, meaning that the design matrix should be uncoupled or at worst decoupled, but never coupled. The second axiom states that the design should “minimize the Information Content”. Information is in this case related to uncertainties of the success of the design.

Coupling of a design matrix means that there exist interdependencies between FRs and DPs, such that an FR requires more than just one DP to function. In a design solution, coupling becomes a trade-off between functionality and simplicity, causing the design to be either less optimal or complex and unreliable. Three levels of coupling exist in AD: uncoupled, decoupled, and coupled. The coupling of a design matrix is explained in Figure 16. As indicated, a design that fulfils the first axiom is inherently robust, and will require the least amount of maintenance.

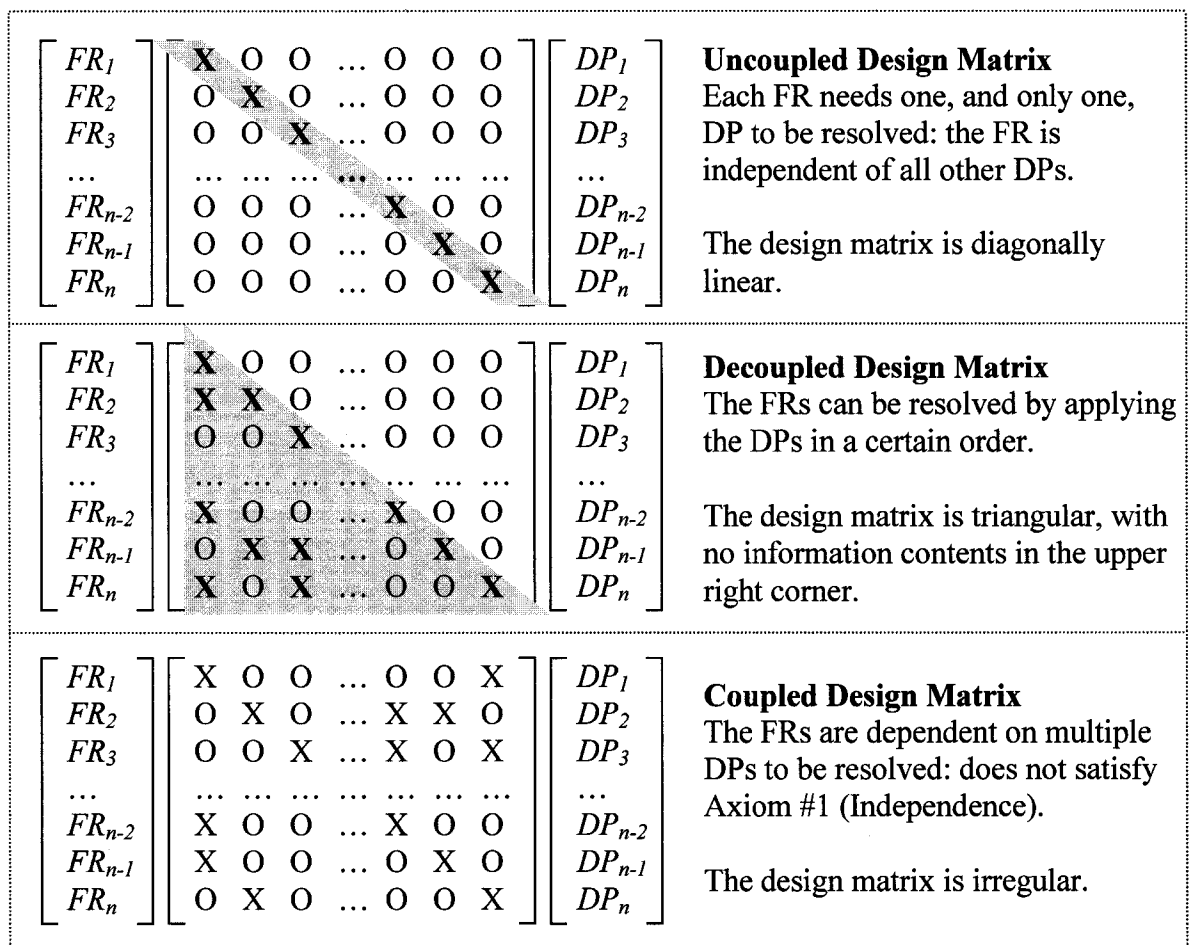


Figure 16 – Coupling of Design Matrices

The information content is, in contrast to coupling, an indistinct measure of the performance of a system. Nevertheless, it can be viewed as the complexity of a system in the terms of the predictability of the success of the design. Most designs using AD as the

design tool overlook the second axiom as little research has been conducted on the subject, and for the difficulty of measuring the information contents. However, the complexity can be measured of both the design matrix and the design itself, and will be discussed in more detail in the chapter Analysis of Results on page 83.

The design methodology Manchulenko identified in his thesis combines the structure of the House of Quality with the logic of Axiomatic Design, and is called the Axiomatic House of Quality (AHOQ). By using his proposed approach, the dependencies induced to the HOQ matrix can be resolved, thus a robust design can be realized. The following list of steps is derived from the approach recommended by Manchulenko (Manchulenko, 2001, p. 46, Step 7 removed and the rest renumbered):

- Step 1. List Customer Attributes (CAs)
- Step 2. Convert CAs into Functional Requirements (FRs)
- Step 3. Identify Constraints
- Step 4. Formulate Design Parameters (DPs)
- Step 5. Formulate the Design Matrix and Initial Design
- Step 6. Resolve FR Dependencies (Decouple FRs)
- Step 7. Correlation of DPs
- Step 8. Comparison of Competing Products
- Step 9. Listing of Constraints
- Step 10. Evaluation of Final Model Results

As the list indicates, many steps are shared between the traditional HOQ and the refined AHOQ. The steps that have been added or changed relate to the methodology of AD, such as Step 6 and Step 7. However, Step 3 and Step 9 differ from the methodology of

the traditional HOQ as they allow certain FRs to be converted to a measurable constraint in the AHOQ and thus removed from the coupling of the design matrix.

The first two steps of Manchulenko's methodology are the same as the initial steps for HOQ. Therefore, the CAs, RAs, and FRs previously identified were reused when designing the AHOQ matrix. However Step 3 focuses on the intent to "identify requirements that are not functionally related, and determine if they are a design constraint". The previous list of FRs contain four FRs that would qualify as constraints, namely FR# 5, 8, 9, and 10. If these FRs are proven to be constraints, it should be possible to convert them to a quantifiable measure, or mathematical formula. The following list shows the FRs that were converted to constraints:

- FR5 "Allow Multiple Simultaneous Users". This FR attempts to enable more than one user to access the system at any given point in time. Therefore, the FR could be stated as "Simultaneous Users > 1 ".
- FR8 "Provide Simultaneous Visibility to all BOMs". This FR means that all BOMs must be connected to each other to allow for cross-referencing. Therefore, the FR could be stated as "Simultaneous Visibility $\cup \forall$ BOM",
- FR9 "Provide Reliable/Accurate As-Built BOMs" can be represented as "BOM Content $\approx 100\%$ Assembled Parts", which allows this FR to qualify as a Volume Constraint, and thus not an FR. However, it is important to recognize the source of any discrepancies between the "As-Built BOM" and the true "Parts List" for the finished product. After having researched this, the source of the discrepancies was found to be related to human errors, complex processes, the number of changes made to the BOMs, and the disconnection

between the different BOMs as well as the disconnection between the BOMs and the related information.

- FR10 “Provide One-Point Entry of all Processes”. This FR could instead be described as “Documents Access Point = 1”.

Therefore, the FRs mentioned above were hereinafter considered being constraints and thus moved from the list of FRs to the section, or subgroup, of constraints in the AHOQ matrix. There is no longer a need for weighting the constraints in the matrix. Instead, the constraints need only to indicate whether the requirements will be met or not. Thus, the low (Δ) and the moderate (\circ) weights were simply removed, whereas the high (\bullet) was changed to (OK) to confirm that the requirement had been met. The technical evaluation matrix and the customer importance rating were also removed as they had no significant meaning in the AHOQ matrix where the weight is binary and not decimal as in the traditional HOQ.

Step 4 in the approach entailed formulating the DPs. However, as the DPs were identified in the design of the HOQ matrix they could be reused for the AHOQ in the same way as with the FRs. To make a comparison between the HOQ and the AHOQ possible, the initial numbering sequence of the DPs in the HOQ was kept throughout the design stages of the AHOQ matrices.

With the next step of Manchulenko’s approach, the design matrix and its initial design was formulated. As much ground work had already been done with the HOQ matrix, the initial AHOQ matrix was derived from the HOQ matrix with the modifications mentioned above. Figure 17 shows the initial stage of the AHOQ matrix. Note the changes and differences from the HOQ matrix in Figure 15 as discussed earlier.

		#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Design Parameters (DPs) Relationship Legend: ● = High (9) ○ = Moderate (3) Δ = Low (1)																		
	Functional Requirements (FRs)	Data Encryption	Firewall	Login User-ID	Login Password	History Log-File	Data Field Formatting	Shared Documents	Granted Server Access	Graphical User Interface	Linked Documents	Tabulated Data Storage	Time/Event Driven Processing	Intranet Hyperlink	Script Programs	Visual Basic Macro	Copy and Paste Information	
Desirability ↑=high, ↓=low			↑	↑	↑	↑	↑	↑	↓	↑	↑	↑	↑	↑	↑	↑	↓	↓
Access	Provide Secure Access to Information	●	Δ	○	●													
	Define User Access to Certain Key Data			●	○			Δ										
	Record User Activity			○		●		Δ										
	Define/Restrict Data Type Entries		Δ	○		●												
	Allow Corporate-Wide Access		●	Δ					Δ	●								
Use	Provide Customized Views of Information			Δ							●							
	Provide Direct Link to Relating Information											○		●				
Process	Initiate Modifications Directly from BOMs											●	○		○		○	Δ
	Simultaneously Update BOM Modifications												●	○		Δ	○	Δ
	Automatically Log Assembly Milestones											Δ	●		Δ	Δ		
	Minimize Redundant Processes											○	○		●	Δ	Δ	Δ
Constraint	Allow Multiple Simultaneous Users							OK				OK						
	Provide Simultaneous Visibility to all BOMs											OK						
	Provide One-Point Entry to all Processes											OK						
	Provide Reliable/Accurate As-Built BOMs												OK					

Figure 17 – Initial Axiomatic House of Quality Matrix

The initial AHOQ, shown in Figure 17 has many couplings among the FRs and DPs. The coupling exists in part because there are sixteen DPs but only eleven FRs. As well, there are couplings due to the many relationships between FRs and DPs. Therefore, emphasis had to be placed on decoupling the matrix as well as removing the additional five DPs. Although not indicated in the approach defined by Manchulenko, it is strongly recommended to keep the original relationship legends between the FRs and DPs from the traditional HOQ. The legends will be valuable in the decoupling of the matrix, indicating how well a DP accommodates an FR.

As mentioned, the AHOQ matrix contained too many DPs in relation to the FRs at the initial stage. Therefore, Step 6 of the approach focused on resolving the FR dependencies, meaning decoupling the FRxDP matrix. Figure 18 shows the AHOQ matrix after unneeded DPs had been removed through the process discussed below.

DP7 “Shared Documents” was removed by using the relationship legends as well as the information given in the desirability row. The desirability row has two possible entries: high (↑) or low (↓). These ratings are based on the knowledge and expertise gained from the system developers and stated in the perspective as: If the listed DP would have to be used, how desirable would that be? Thus, in the traditional HOQ, the DP7 “Shared Documents” is shown to be able to resolve the FR “Allow Multiple Simultaneous Users” as it has a high (●) relationship. However, the same FR was redirected to be a constraint in the AHOQ matrix, which now had two DPs that would meet the requirement of the said FR. As well, DP7 was rated with a low (↓) desirability index, and had only low (Δ) relationships with the other three FRs it was connected to. In addition, all other FRs with relationship to the said DP7 had other DPs with a high (●) relationship. Therefore, DP7 was removed from the AHOQ matrix without any negative impact to the listed FRs or constraints.

The same reasoning as above was also used to remove DP15 “Visual Basic Macro” and DP16 “Copy and Paste Information” from the AHOQ matrix. Both FRs had a low (↓) desirability index and either low (Δ) or moderate (○) relationship to the FRs, but were also resolved by other DPs with a high (●) relationship to the same FRs. Therefore, DP15 and DP16 served no purpose in the design of the AHOQ matrix and were removed. The resulting DP-reduced matrix is shown in Figure 18, without DP7, DP15, or DP16.

As a guideline; whenever there is an FR connected to a DP with a low (↓) desirability index and either low (Δ) or moderate (○) relationship, in presence with other DPs with a high (●) relationship to the same FR, the low rated DP can be removed without any negative impact to the design.

		#	1	2	3	4	5	6	8	9	10	11	12	13	14
Design Parameters (DPs)			Data Encryption	Firewall	Login User-ID	Login Password	History Log-File	Data Field Formatting	Granted Server Access	Graphical User Interface	Linked Documents	Tabulated Data Storage	Time/Event Driven Processing	Intranet Hyperlink	Script Programs
Functional Requirements (FRs)															
<i>Desirability</i> ↑=high, ↓=low			↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
Access	Provide Secure Access to Information		●	Δ	○	●									
	Define User Access to Certain Key Data				●	○									
	Record User Activity				○		●								
	Define/Restrict Data Type Entries			Δ	○			●							
	Allow Corporate-Wide Access			●	Δ				●						
Use	Provide Customized Views of Information				Δ					●					
	Provide Direct Link to Relating Information											○		●	
Process	Initiate Modifications Directly from BOMs										●	○		○	
	Simultaneously Update BOM Modifications											●	○		Δ
	Automatically Log Assembly Milestones											Δ	●		Δ
	Minimize Redundant Processes											○	○		●
Constraint	Allow Multiple Simultaneous Users											OK			
	Provide Simultaneous Visibility to all BOMs											OK			
	Provide One-Point Entry to all Processes											OK			
	Provide Reliable/Accurate As-Built BOMs												OK		

Figure 18 – DP-Reduced Axiomatic House of Quality Matrix

However, there were still too many DPs to allow the matrix to be uncoupled. As mentioned before, the coupling exists because of multiple FR-DP connections; therefore, the AHOQ matrix should, if possible, be left with only high (●) relationships. In other

words, wherever there was an FR that could be solved through many DPs, only the relationships rated high (●) for that FR were left in the table. Note however, in those cases where there was more than one FR-DP relationship rated high (●), all of those with high (●) relationships had to be left in the matrix, as those caused the couplings. In this case, as all FRs contained at least one high (●) relationship to the DPs, only the high (●) were left in the AHOQ matrix, whereas all the moderate (○) and low (Δ) were removed. In addition, the DPs were also rearranged, resulting in a decoupled matrix with eleven FRs and thirteen DPs as shown in Figure 19.

		#	1	4	3	5	6	8	2	9	13	10	11	12	14
Design Parameters (DPs)			Data Encryption	Login Password	Login User-ID	History Log-File	Data Field Formatting	Granted Server Access	Firewall	Graphical User Interface	Intranet Hyperlink	Linked Documents	Tabulated Linked Data Storage	Time/Event Driven Processing	Script Programs
Functional Requirements (FRs)															
Access	Provide Secure Access to Information		●	●											
	Define User Access to Certain Key Data				●										
	Record User Activity					●									
	Define/Restrict Data Type Entries						●								
	Allow Corporate-Wide Access							●	●						
Use	Provide Customized Views of Information									●					
	Provide Direct Link to Relating Information										●				
Process	Initiate Modifications Directly from BOMs											●			
	Simultaneously Update BOM Modifications												●		
	Automatically Log Assembly Milestones													●	
	Minimize Redundant Processes														●
Constraint	Simultaneous Users ≥ 1													OK	
	Simultaneous Visibility $\cup \forall$ BOM													OK	
	Documents Access Point = 1													OK	
	BOM Content \approx 100% Assembled Parts													OK	

Figure 19 – Decoupled Axiomatic House of Quality Matrix

As a guideline to the previous reduction of FR-DP couplings; wherever there is an FR connected to multiple DPs, and at least one of those DP has a high (●) relationship to the said FR, all DP connections with a low (Δ) or a moderate (○) relationship to the said FR can be removed without any negative impact to the design. However, if the high (●) relationship does not exist between the said FR and the DPs, an additional DP should be introduced which allows for a high (●) relationship, or else the next highest relationship must be kept. Note, however, that the latter will create a less optimal design, with only a moderate (○) or low (Δ) relationship between the said FR and the DP.

As seen in Figure 19 there were still couplings between some DPs in the AHOQ matrix, namely between DP1 “Data Encryption” and DP4 “Login Password” as well as between DP8 “Granted Server Access” and DP2 “Firewall”. Recall from the section about coupling of design matrices: a robust design only contains connections in a linear diagonal manner across the matrix, as shown in Figure 16. The effort was therefore continued to decrease the level of coupling to an uncoupled AHOQ matrix. However, as all the connections left in the matrix had a high (●) relationship between the FRs and the DPs, an alternative approach had to be used that could further decouple the matrix. Therefore, inventive design principles from the Theory of Inventive Problem Solving (TRIZ) were used to resolve the couplings of the design.

Theory of Inventive Problem Solving

TRIZ has been widely covered in literature during the last decade. There are also websites, such as <http://www.triz-journal.com>, that offer free journal papers on the subject, which provides a great overview of TRIZ. The design methodology of TRIZ

consists of 40 “Inventive Principles”, which are derived from the study of thousands of patents and their solutions, see Appendix D. Many of them relate more to the field of mechanical engineering than anything else, but some have a general characteristic and can be very useful when engineering the design of systems, such as this project. In addition, Rea (2001) and Fulbright (2004) converted the 40 principles to suit the design process of software development, giving examples on how the principles can be used in the field of computer science. Therefore, the purpose was to find, in the list of the 40 Inventive Principles for Software Development, design ideas that could resolve the coupling effect in the AHOQ matrix. The first step was to identify those principles that potentially could be used for the design parameters in the AHOQ matrix. As the original principles were developed for physical items, it became necessary to “ignore the wordings” of the principles, but instead focus on the true meaning and intention of the principles, which also was concluded by Rea and Fulbright. Therefore, the following seven principles were identified as potentially useful:

- Principle 1) **Segmentation**: Divide an object into multiple parts
- Principle 2) **Taking Out**: Single out the only necessary part of an object
- Principle 3) **Local Quality**: Make each part of an object fulfil a useful function
- Principle 5) **Merging**: Combine identical or similar object; make contiguous
- Principle 6) **Universality**: Make a part perform multiple functions; elimination
- Principle 7) **Nested Doll**: Place an object inside another
- Principle 24) **Intermediary**: Merge one object temporary with another

Due to the coupling of the AHOQ matrix, the purpose of introducing the TRIZ principles was to reduce the number of design parameters (DPs) so there would be an equal amount

of DPs as there are functional requirements (FRs), which is an absolute requirement of the first Axiom. With that, two DPs had to be removed from the matrix. Therefore, it was important to view the TRIZ principles from the point of view of how they could be used in attempting to remove a DP. Through analysis of the AHOQ matrix as a complete system and by using the above listed TRIZ principles, the DPs causing the couplings could be resolved, thus decoupled.

Of the above listed TRIZ principles, Principle 5 “Merging” stood out as a candidate to solve the coupling between DP1 “Data Encryption” and DP4 “Login Password”. However, it was by including the adjacent DP3 “Login User-ID” in the analysis that allowed the coupling between DP1 and DP4 to be solved. As the prototype assembly documents were operating through Microsoft Excel 2003 spreadsheets, there was a need to differentiate between Login User-ID and Login Password in the traditional HOQ matrix. In addition, it was the HOQ matrix that laid the groundwork for the initial AHOQ matrix, hence the DPs remained intact. The need to differentiate between DP3 and DP4 was due to the fact that Excel 2003 does not have the functionality of a User-ID. However, the HOQ matrix showed that both Access 2003 and Enterprise SQL with Visual Studio .Net have the ability to accommodate both a Login User-ID and a Login Password. Besides, both Access 2003 and Enterprise SQL were proven to be better than Excel 2003 for the intended processes. Therefore, by using Principle 5, DP3 and DP4 were merged to perform a parallel function, namely “Defining User Access to Certain Key Data”. As such, by merging DP3 “Login User-ID” with DP4 “Login Password” to form the new DP_i “Login User-ID and Password”, the DPs were decoupled.

For the coupling between DP8 “Granted Server Access” and DP2 “Firewall”, Principle 6 “Universality” was used to uncouple the DPs. This principle guides the functions to “make an object perform multiple functions; eliminating the need for other objects”, which would lend the Server to use the functionality of the Firewall to “grant access” to itself. Therefore, by using Principle 6, DP8 and DP2 were combined to perform a joined function, namely “Allow Corporate-Wide Access”. Hence, by making DP8 “Granted Server Access” universal with DP2 “Firewall”, to form the new DP_{ii} “Server Access through Firewall”, even these DPs were now decoupled.

Uncoupled Design Matrix

The end result from using TRIZ (the totally uncoupled FRxDP matrix) can be seen in Figure 20, given the standard binary Axiomatic Design notation. Thanks to the uncoupled matrix, each Functional Requirement is now dependent on only one Design Parameter, making the design of the system inherently robust. Suh (2001) describes robustness as “directly related to the level of coupling in an FR_j-DP_j matrix”.

As shown in Figure 20, each DP lends an FR to be utilized as desired. Recall that the FRs are the “voice of the customers”. Thus, the design has successfully met all the requirements set out by the customers. Although the design presented in the traditional HOQ matrix in Figure 15 also managed to successfully meet all the customer requirements, the Axiomatic design below is proven to be robust and guaranteed to function as necessary, lending it to become the framework for the detailed model. Nevertheless, it is important to be aware that the design can only be considered robust if, and only if, all functional requirements of the system are present in the matrix.

By following the analytical approach of systems design and quality function deployment (QFD), there is a greater chance of having most, or all, requirements counted for. Although, there is no way of being completely certain that all requirements are captured. As such, without having ways of identifying “missed” requirements at the design stage, the systems design and QFD will reduce the risk of overlooking vital requirements. As well, using QFD allows the design to be easily revisited and updated in the event that further requirements would be revealed later on in the development stage or while testing the system.

		Design Parameters (DPs)													
		#	1	i	5	6	ii	9	13	10	11	12	14		
		Functional Requirements (FRs)													
		Relationship Legend: 1 = DP _j Resolves FR _i 0 = DP _j Unrelated to FR _i													
Access	Provide Secure Access to Information	1	0	0	0	0	0	0	0	0	0	0	0	0	
	Define User Access to Certain Key Data	0	1	0	0	0	0	0	0	0	0	0	0	0	
	Record User Activity	0	0	1	0	0	0	0	0	0	0	0	0	0	
	Define/Restrict Data Type Entries	0	0	0	1	0	0	0	0	0	0	0	0	0	
	Allow Corporate-Wide Access	0	0	0	0	1	0	0	0	0	0	0	0	0	
Use	Provide Customized Views of Information	0	0	0	0	0	1	0	0	0	0	0	0	0	
	Provide Direct Link to Relating Information	0	0	0	0	0	0	1	0	0	0	0	0		
Process	Initiate Modifications Directly from BOMs	0	0	0	0	0	0	0	1	0	0	0	0		
	Simultaneously Update BOM Modifications	0	0	0	0	0	0	0	0	1	0	0	0		
	Automatically Log Assembly Milestones	0	0	0	0	0	0	0	0	0	1	0	0		
	Minimize Redundant Processes	0	0	0	0	0	0	0	0	0	0	0	1		
Constraint	Simultaneous Users ≥ 1										OK				
	Simultaneous Visibility ∪ ∇BOM										OK				
	Documents Access Point = 1										OK				
	BOM Content ≈ 100% Assembled Parts											OK			

Figure 20 – Uncoupled Axiomatic House of Quality Matrix

Detailed Engineering Model

As indicated in the AHOQ matrix in Figure 20, each Functional Requirement has a unique Design Parameter that allows the system to be uncoupled, which in turn lends the engineering solution to be robust and simple in its design. By using the information in the AHOQ matrix above, the detailed engineering model for the system can be illustrated as shown in Figure 21.

Aside from the different processes and functions, the individual DP-numbers from the uncoupled and finalized AHOQ design matrix from Figure 20 are also indicated in the detailed engineering model in Figure 21.

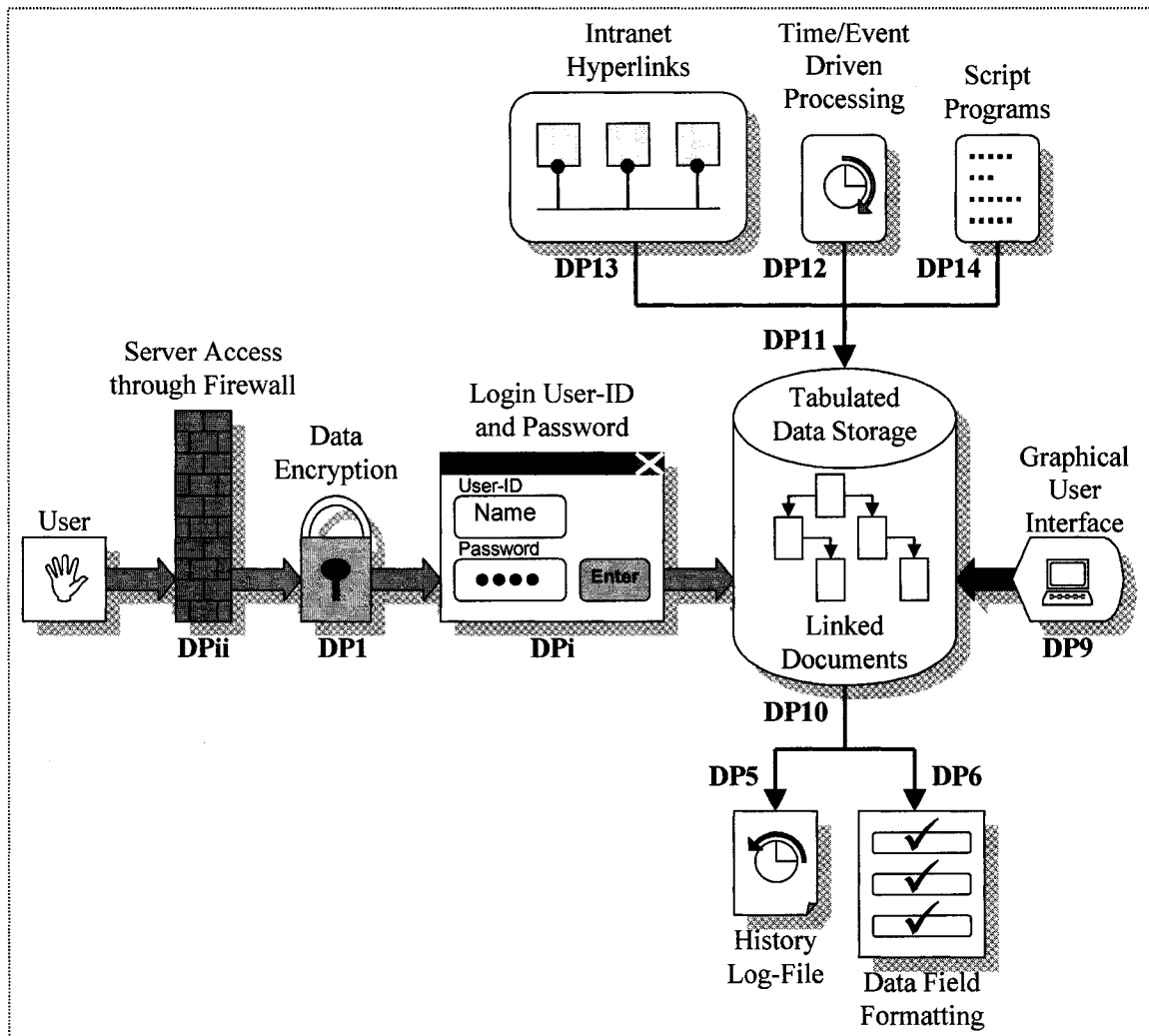


Figure 21 – Detailed Engineering Model

In the detailed engineering model, processes are effectively executed by a unique function or process, as part of the engineered system. This is in contrast to the original system which previously required multiple documents or processes to function as desired. As well, the new system is made robust, meaning that if one process (Design Parameter) fails to function, the other processes will still continue to function. As such, no design parameters are directly affected by the failure of another.

As the detailed engineering model shows, users can enjoy a corporate wide access to the server through a firewall. The data stored on the server is secured by encryption, such as a 128 bits Secure Socket Layer. To ensure the highest level of data quality, each user is identified on the server via a login user-id and password that restricts the access to certain key data. The users will navigate and maintain the data through a familiar graphical user interface that allows both personalized and customized views of the data, which is stored on the server in a tabulated manner. Having the data tabulated allows the users to access all data simultaneously, such as the BOMs. To aid the engineers in the substitution of parts, the data is also linked to the corporate server where engineering changes are made. This would allow the engineers to initiate a substitution directly from the BOMs, rather than from a separate document or system. In addition, certain information in the BOMs is also directly linked through Intranet hyperlinks to the storage point of valuable information, such as drawings, issues, and changes. To ensure that only the intended data is entered or changed in the system, all fields are controlled as to how and by whom they are formatted. To reduce much of the previously manual and tedious maintenance, data are processed through time and event driven indicators to automatically show important milestones, as an example. As well, script programs will

be executed at certain instances to reduce redundancy of processing, such as substitutions and changes to WIP locations. Lastly, a history log is continuously updated with all changes and entries made to the data. Thus, statistics can be extracted from the system to show all modifications and other activities in the system.

Improved Engineering Design

The detailed engineering model was used in conjunction with both the HOQ matrix and the AHOQ matrix to improve the documents and processes used in the prototype assembly build event. Figure 22 illustrates the improved design of the system as derived from the results of the matrices and the detailed engineering model.

As shown in Figure 22, the resulting system consists of a data management system: a database in which many of the previously manual operations are instead automatically processed. Most of the documents from the original system are embedded in the database, which will act as a central access point for all operations performed within the system. The manual transfer of information is replaced with direct hyperlinks to the source of the information, which will minimize redundancy and erroneous data. The collaboration effort is also greatly improved as the database is accessible from any office in the world where a connection to the Internet exists. In addition, engineering changes can be conducted more efficiently and with greater accuracy using the improved system, as the database is directly connected to all related information. The improved system also allows all stakeholders a simultaneous and immediate view of the BOMs and the information therein. As indicated in the diagram, the previous three shortage reports have been combined to one report that accommodates all reporting needs of part

shortages. Likewise, the entities and connections to associated documents are of a lesser amount in the improved system, making it cognitively much simpler when compared to the original.

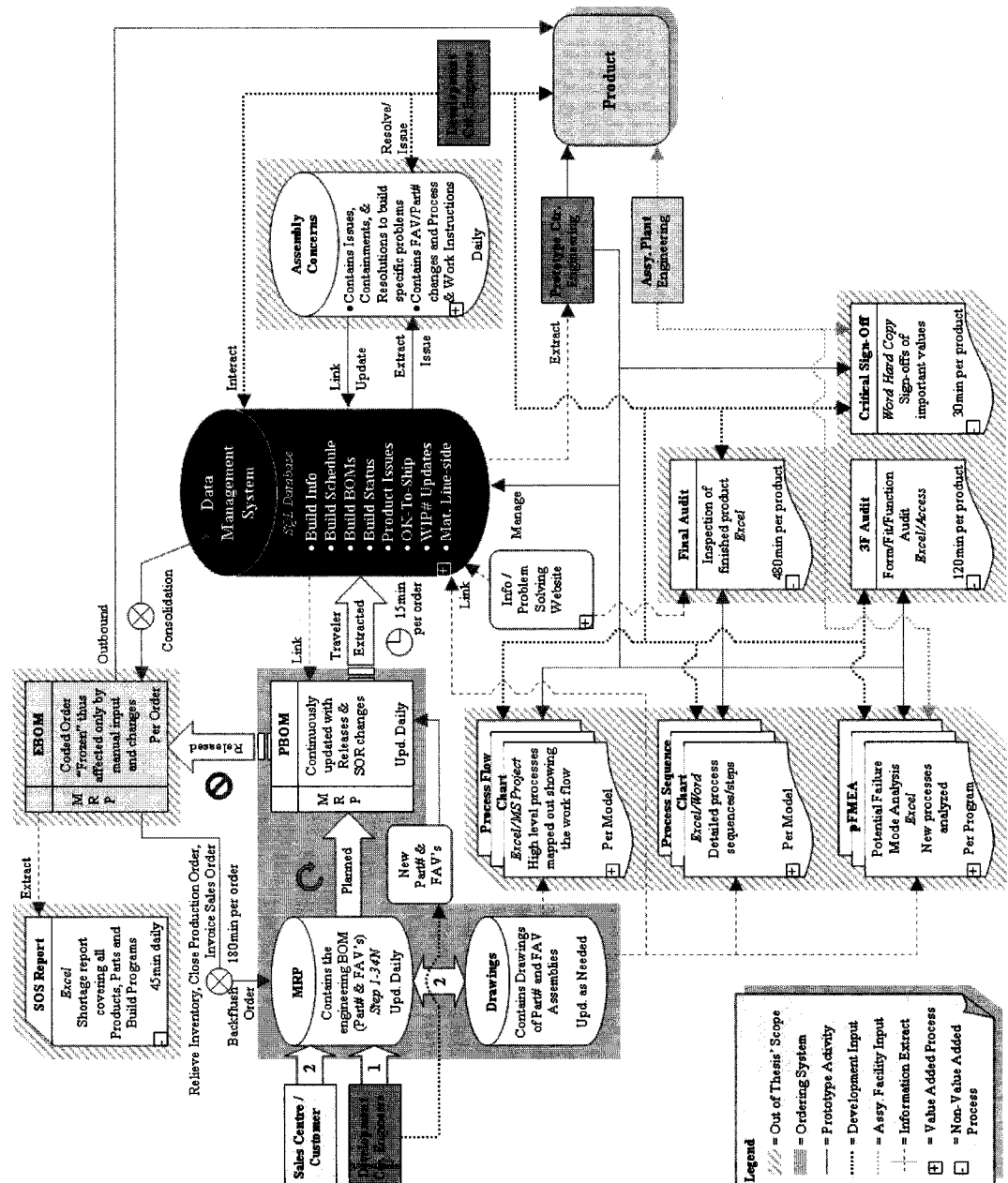


Figure 22 – Improved Information Flow Diagram

Data Flow Diagram

The improved engineering design shown in Figure 22 consists of a database in which most of the previous separate Excel documents are stored. To illustrate the logical structure of the data management system (the database), a Data Flow Diagram (DFD) can be used to show the individual entities and how the entities are connected. A DFD exemplifying the logical structure of the data management system in the improved engineering design is shown in Figure 23.

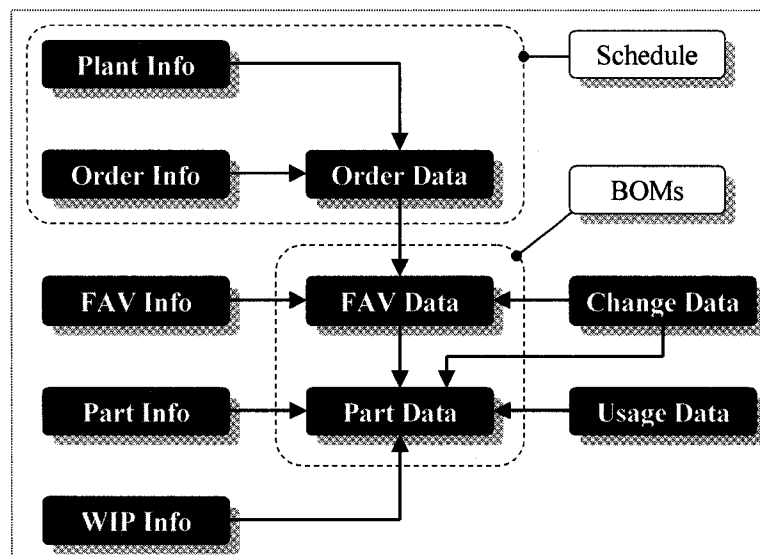


Figure 23 – Data Flow Diagram of Database

The DFD shows two logical units: the Schedule and the BOMs. The Schedule is made up of three entities (tables): “Plant Info”, “Order Info”, and “Order Data”. The BOMs however, consists of two entities: “FAV Data” and “Part Data”. Other entities, such as “FAV Info”, “Part Info”, and “WIP Info” support the BOMs with information surrounding the BOMs. Two additional entities, “Change Data” and Usage Data” make up the augmentation and validation of the BOMs. The “Change Data” contains information about all changes made to the BOMs, such as engineering changes,

deviations, substitutions, assembly concerns, etc., whereas the “Change Data” includes all validated part quantities used in the assemblies.

Discourse on Evaluation of Designs

The system described with the detailed engineering model in Figure 21 and the resulting information flow diagram in Figure 22 had analytical proof to be inherently robust and to have a high probability for success. The robustness was a direct result from the uncoupled Axiomatic House of Quality matrix. As such, if the first axiom of the Axiomatic Design methodology is fulfilled (meaning the design matrix is uncoupled), the design of the system is proven to be robust. However, to say that the new design would be better than the original, analytical proof had to be provided showing the improved system to be of a simpler nature than that of the original. However, the intent of measuring the complexity would be to assess the level of complexity of the system as a whole, not the complexity, or the structure, of the individual entities (documents). In other words, the complexity would be determined as the level of difficulty of comprehending the design of the system. With that, there will be no attempt to assess the complexity within any of the entities, such as the structural complexity of a relational database. However, it is necessary to appreciate the difference between the complexity of the system as a whole and the complexity within the entities the system is composed of. The complexity within an entity is dependent upon the layout and structure of the entity, such as forms, fields, tables, graphs, and reports, whereas the complexity within the system as a whole depends on the connections, relationships, and flow of information

between the entities. Appropriately, as this research has a focus on the design of a system, the complexity will only be evaluated on a systems level.

To support the decision in the evaluation of the simplest design, a formula for calculating the complexity had to be modified to suit the needs of this research. With the ability to calculate the complexity a few questions needed to be answered: How complex was the original system? How complex is the new system? And, could the complexity be reduced even further, through other design implementations? The chapter that follows will answer the above questions as well as discuss the matter in more detail.

CHAPTER V
ANALYSIS OF RESULTS

Matrix Tree Theorem

To reiterate what was stated in the literature review, complexity of a graph (such as a business process diagram or an information flowchart) can generally be viewed as the relationship between the number of nodes and the number of arcs within a given graph. Although there are many formulae available to calculate complexity in graphs, the complexity theorem introduced by Temperley (1981) is, in the research of Latva-Koivisto (2001), believed to be the most reliable measure of complexity in graphs. The Temperley complexity theorem is in reality a modified version of the “Matrix Tree Theorem” which was developed by Gustav Kirchoff in 1847. Kirchoff originally devised the complexity calculation to gain insight on the flow of current in electrical networks.

Nonetheless, when using the Matrix Tree Theorem, the complexity of a graph is calculated as the number of spanning trees in the graph that is evaluated. A spanning tree is explained by Wilson (1996) as a subgraph of the original graph G which contains all nodes of G and where all nodes are connected in a tree structure without cycles. A cycle in a graph is explained as a redundant path between nodes as such that all cyclic nodes can be reached through other paths. Therefore, the Matrix Tree Theorem calculates the complexity as directly related to the amount of connecting arcs to a node.

Robin Whitty explains the theory of the Matrix Tree Theorem: the complexity denoted T is measured by calculating the diagonal minor D_{ij} of a matrix G consisting of the relationship between the nodes N_i through N_j and the connecting arcs A_{ij} as exemplified with the illustration shown in Figure 24.

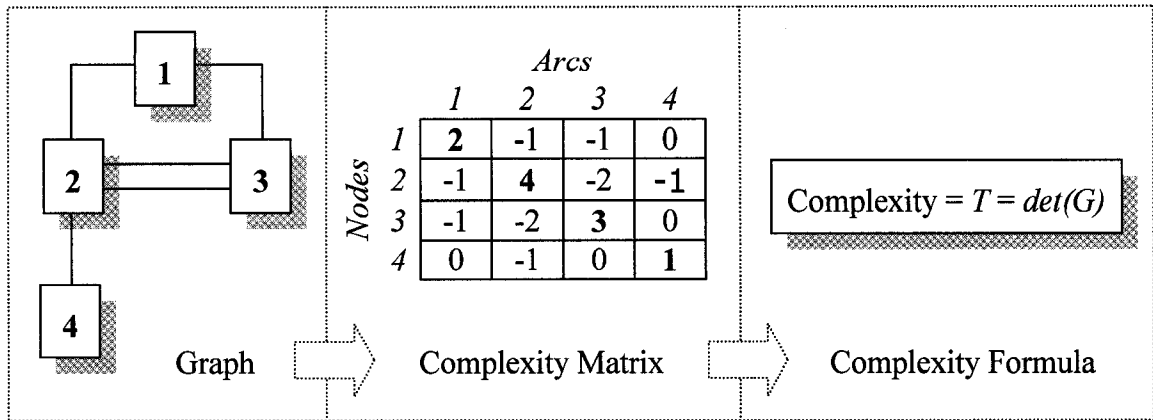


Figure 24 – Illustration of the Matrix Tree Theorem

When calculating the complexity T of the graph shown in Figure 24 using the Matrix Tree Theorem, the value of T is five, as the graph can be redrawn as five distinct spanning trees.

Improved Complexity Formula

The research conducted by Latva-Koivisto was based on what is explained as a Kaimann process graph. Kaimann studied the Coefficient of Network Complexity (CNC) for which he designed a generic process graph with twenty-two nodes. To evaluate the formulae, the complexity of the graphs was increased by adding arcs between the nodes, increasing the coupling of the system. However, as Kaimann studied network, and in particular the connections between the nodes in the network, the sample system always had the same number of nodes. Therefore, the scale of the system was not taken into consideration: the number of nodes was kept at twenty-two. Conversely, Latva-Koivisto used the Kaimann process graphs to evaluate the different complexity measures in the research. The research concluded that the Matrix Tree Theorem was the most reliable measurement, and that it correctly responded to the increased and decreased level of complexity of the Kaimann graphs. Nevertheless, the research also showed that the

relationship between size and complexity was weak, not only in the Matrix Tree Theorem formula, but in all of the evaluated formulae.

The weak relationship between the scale (number of nodes) of a system and the [measured] level of complexity would pose a problem in evaluating the original and the improved information flow diagrams. Recall that the intent of measuring the complexity of the system was to compare the two systems, which inherently would be of different size and composition. Hence, the scale of the systems (the number of documents) would have a significant impact on the results from the complexity formula.

Therefore, there was an opportunity to improve the complexity formula with a goal of [more] accurately measuring the cognitive complexity of graphs which would be of different size and have different amount, and paths, of connections between the entities. Not to reinvent the wheel, the Matrix Tree Theorem was chosen as a foundation for the improved formula, as it was proven that the formula [at least] correctly calculated the complexity of the connectivity of a graph. In an attempt to evaluate and improve the Matrix Tree Formula, a set of simulated graphs were designed. As such, Figure 25 and Figure 26 show comparable examples of the simulated graphs labelled [a] to [k] in addition to different calculations of the complexity. The intent of the comparison was to illustrate how the original complexity formula (the Matrix Tree Theorem) correctly accommodated the level of coupling in graphs of fixed size, but was less accurate when the size of graphs and connections between nodes was compared. In the figures, the Matrix Tree Theorem value T was calculated as shown in Figure 24. In addition, a binary logarithm was used to reduce the scale order of the results, as the value of T becomes very large with an increase of the number of nodes N and number of arcs A .

The comparison of the graphs and the rationale of the formulae are described in more detail following Figure 25 and Figure 26.

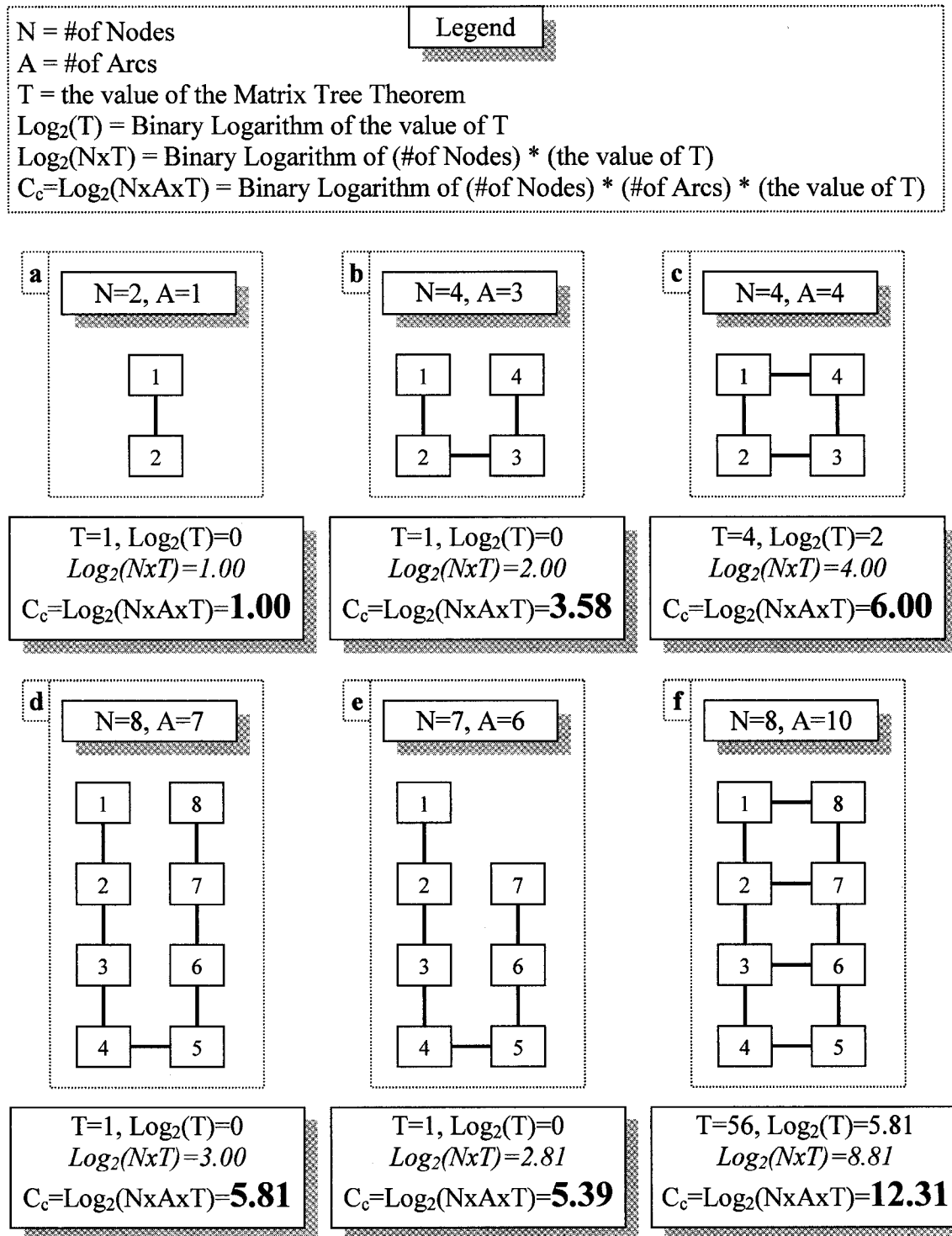


Figure 25 – Example (1) of Process Graphs

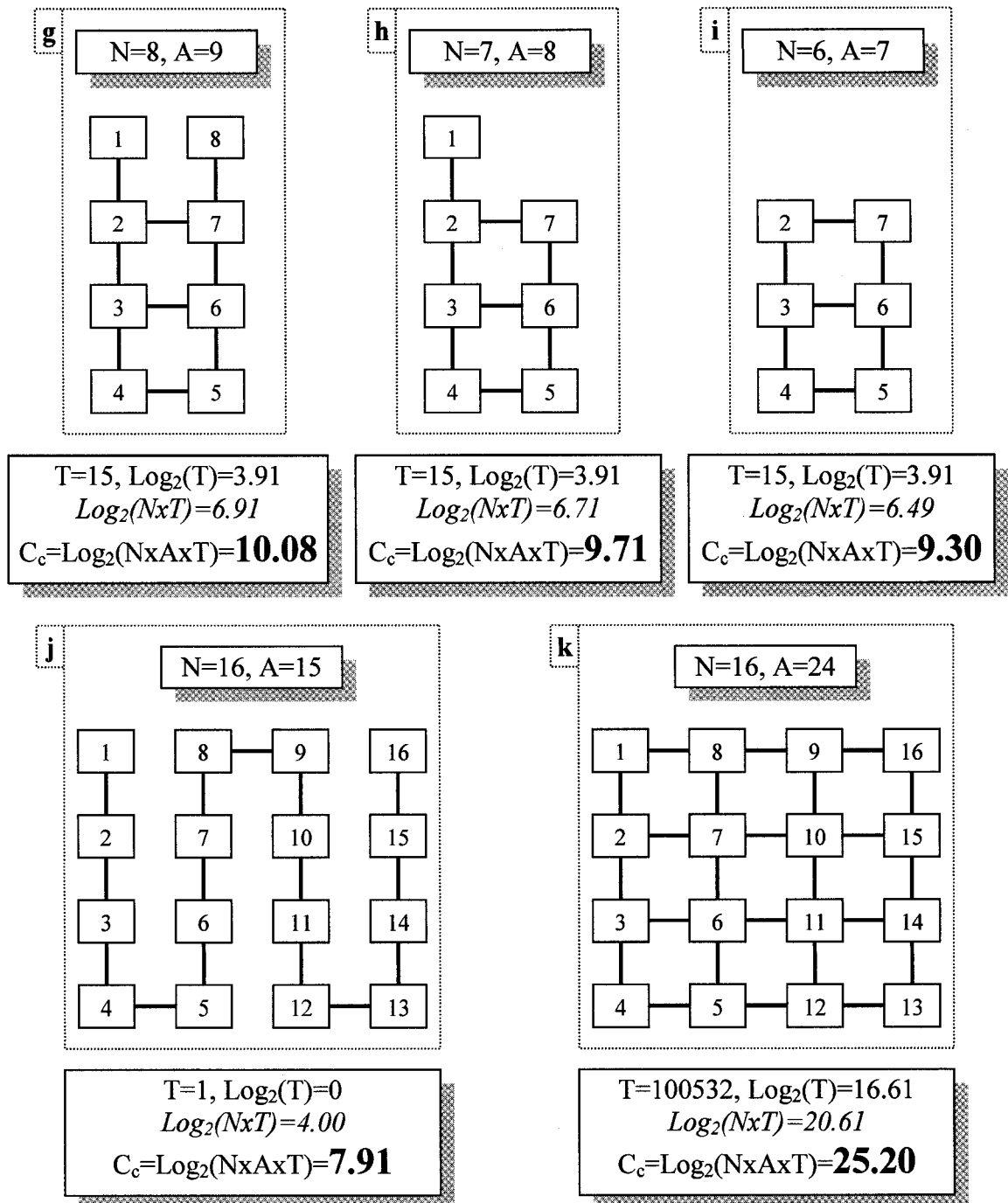


Figure 26 – Example (2) of Process Graphs

When comparing the simulated graphs in the figures above, the Matrix Tree Theorem shows a higher value of T with an increase of complexity, as the example of comparing graph [f] and [g] where [f] has a complexity of $\text{Log}_2(T) = 5.81$ whereas [g] has a value of $\text{Log}_2(T) = 3.91$. As a result of calculating the complexity as the number of

spanning trees, nodes with only one connecting arc do not affect the level of complexity, as the example of comparing graphs [g], [h], and [i] show. All three graphs have the same level of complexity: $\text{Log}_2(T) = 3.91$, but the graphs evidently contain different amount of nodes and arcs. In addition, graphs that only consist of nodes serially connected between arcs show no complexity at all: $\text{Log}_2(T) = 0$, as indicated with the serially connected graphs [a], [b], [d], [e], and [j].

Therefore, the size of the graph had to be added to the calculation. As an experiment, the number of nodes N was multiplied with the value of T from the Matrix Tree Theorem as $\text{Log}_2(NxT)$, where $T = \det(G)$. Again, the binary logarithm was used to reduce the scale order of the calculated complexity. This resulted in a distinct difference of the serially connected graphs, where graph [j] was the more complex graph of the previously mentioned serially connected graphs. However, the formula still showed inconsistency, when graph [c] was compared to graph [j], as both showed the same complexity value: $\text{Log}_2(NxT) = 4.00$. By looking at the graphs, the smaller graph [c] should have a lower value of complexity than the larger graph [j], even though graph [j] is serially connected.

Using the $\text{Log}_2(NxT)$ formula, the serially connected graphs and the nodes with only one connecting arc were correctly indicated, but the scale was still inaccurate. Therefore, the number of arcs A was multiplied with the previously improved formula, to lend the complexity of the process graphs to be calculated as shown in Figure 27.

$$C_c = \text{Log}_2(NxAxT) \text{ where } T = \det L(G)(1|1)$$

Figure 27 – Modified Complexity Formula

Note that the complexity formula was given the notation C_c for *Cognitive Complexity*. In the formula, N = number of nodes, A = number of arcs, and $T = \det L(G)(1|1)$, where $L(G)(1|1)$ equals the matrix obtained by deleting the first row and column of $L(G)$. As such, the simulated graphs in Figure 25 and Figure 26 could therefore be evaluated for its complexity, regardless of serial connectivity, single connected nodes, and size of graph, which was a requirement for measuring the complexity of the information flow diagrams. When comparing the simulated graphs, the complexity value C_c increases with seemingly more intricate graphs, such as when comparing graphs [c], [i], [f], and [k]. The formula also allows single connected nodes to impact the evaluation, such as when comparing graphs [g], [h], and [i], as well as graphs [d] and [e]. In addition, graphs [c] and [j] are also correctly calculated, showing separate values for the individual graphs.

Interestingly, calculating the complexity using $C_c = \text{Log}_2(NxAxT)$ lends graph [a] (the simplest graph) to act as a reference, with a complexity of $C_c = 1.00$. In addition, a graph with no connections (only a node) shows no complexity, thus $C_c = 0$. This should be regarded as an important finding: *no complexity should be zero (0) and the lowest level of quantifiable complexity in the decimal system should be one (1)*.

Thus, by using the modified Matrix Tree Theorem, the complexity C_c could be calculated for graphs with different number of nodes and arcs, and with different connections of the arcs to the nodes. As such, by modifying the formula for calculating the complexity, the graphs representing the original and the improved information flow diagrams could therefore be evaluated analytically, using quantifiable and comparable measures.

Complexity Matrices

To measure the complexity level of the original information flow diagram shown in Figure 10 and the improved information flow diagram shown in Figure 22, the documents were first numbered to aid in the formulation of the complexity matrices for the Matrix Tree Theorem. As mentioned however, the diagrams do not only show flow of information, but also work being done on the documents as well as verbal communication between the different stakeholders. Due to the scope of the research, rather than showing the individual workers the diagrams instead show a generic flow of information in the form of facilities, to which the workers belong. It was therefore not feasible to include those connections of flow of work in the calculations of the complexity, as it was not expected of one worker to perform all duties illustrated in the diagrams. Nor was the scope of the research to improve the work performed on the system, but rather to improve the system itself. Consequently, as the intention was to evaluate the complexity of the information flow of the prototype assembly processes, direct connections between people and documents were removed, resulting in two modified diagrams shown in Figure 28 and Figure 29 respectively. As a result of removing the redundant connections, some documents became disconnected (isolated) from the rest, shown as crossed out in the diagrams. These documents had to be excluded from the calculation due to the non-existing flow of quantifiable information. However, the disconnection of the documents could be viewed as an indication for the need to improve, integrate, or remove those documents from the prototype assembly processes. Recall, the documents are disconnected as there are no natural flows or exchanges of information between the overall system and the recently isolated documents.

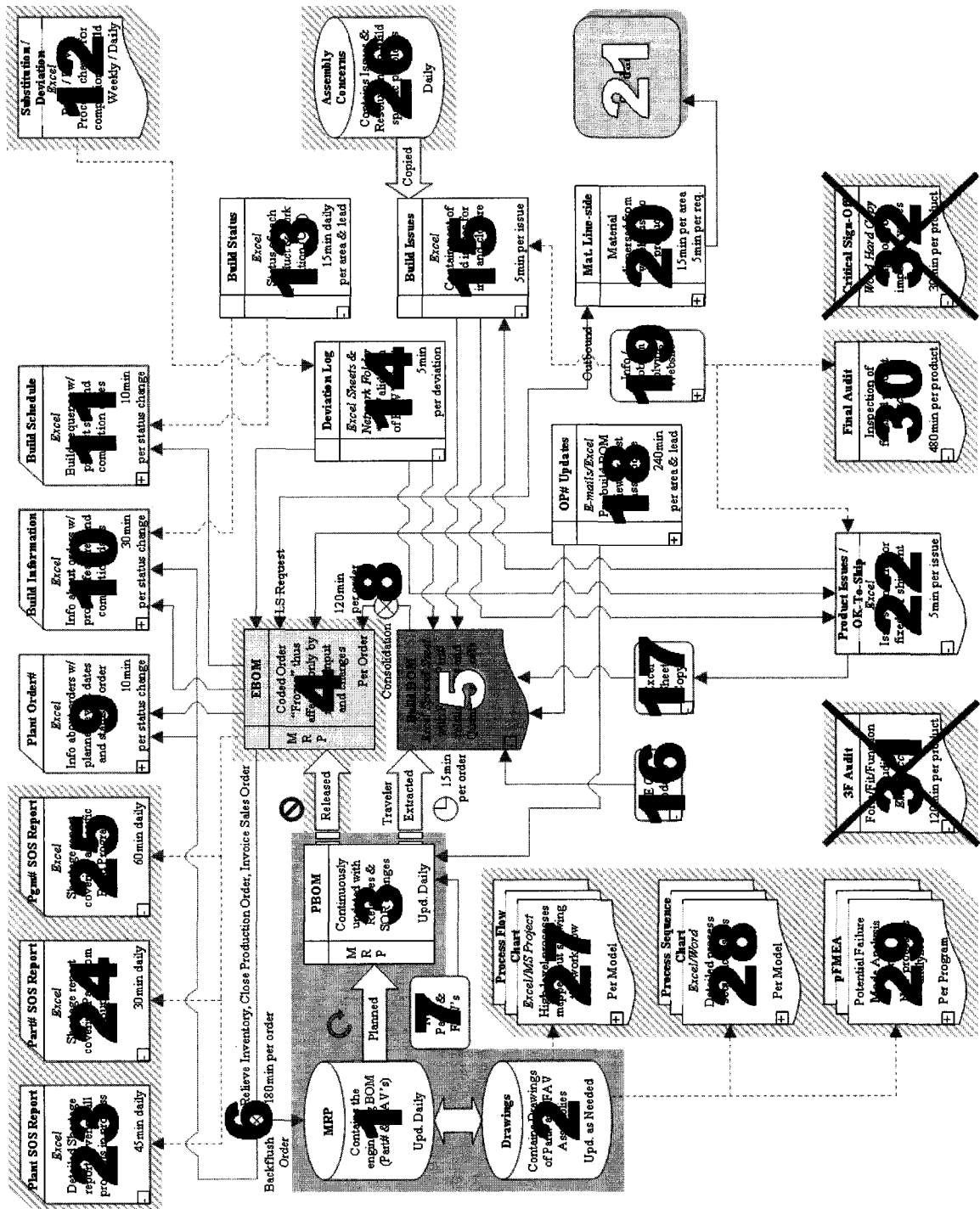


Figure 28 – Numbered Original Information Flow Diagram

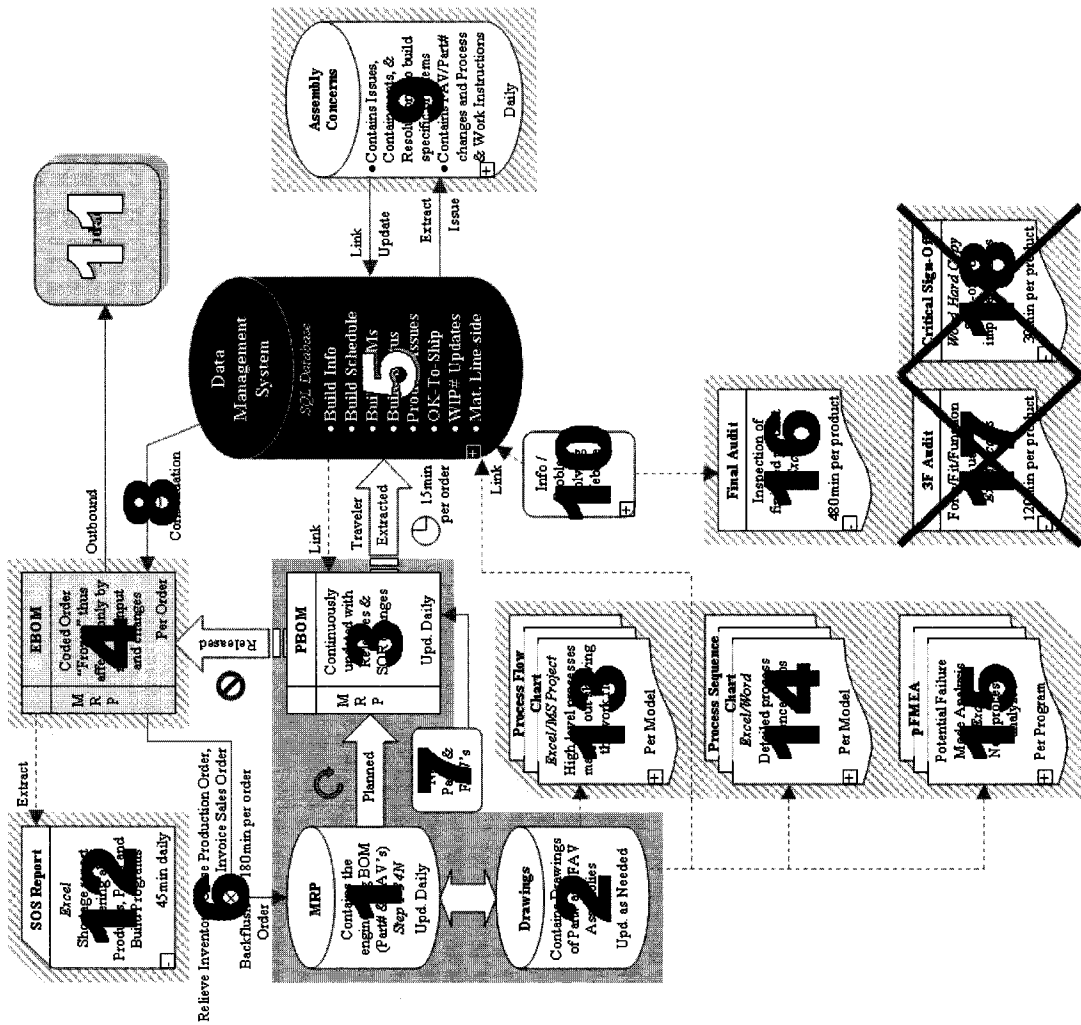


Figure 29 – Numbered Improved Information Flow Diagram

As previously mentioned, the level of complexity C_c of a graph can be calculated as $C_c = \text{Log}_2(NxAxT)$ using the determinant of the corresponding complexity matrix of the graph. The logarithm is used, as recommended by Latva-Koivisto among others, to lower the scale-order of the complexity as it tends to become very high when calculating the determinant of large-scale matrices. This is of course due to the fact that the Matrix Tree Theorem calculates the number of spanning trees in the graph, which will increase exponentially with the number of nodes and arcs. As such, the resulting complexity matrices used for calculating the complexity of the diagrams shown in Figure 28 and Figure 29 are illustrated in Figure 30 and Figure 31 respectively.

The complexity matrices above were formed by placing the number of arcs (flow of information) connected to each node (document) on the diagonal in the respective matrix. Each row in the matrix represents one node with its connecting arcs noted in the columns. The connecting arcs (number of connections to one particular node) are shown with a negative sign, originating from the tree-generating determinant formula introduced by Temperley (1981), illustrated in Figure 32 (Latva-Koivisto, 2001, p. 16).

$$D = \begin{vmatrix} \sum_{j \neq 1} a_{1j} & -a_{12} & -a_{13} & -a_{14} \\ -a_{21} & \sum_{j \neq 2} a_{2j} & -a_{23} & -a_{24} \\ -a_{31} & -a_{32} & \sum_{j \neq 3} a_{3j} & -a_{34} \\ -a_{41} & -a_{42} & -a_{43} & \sum_{j \neq 4} a_{4j} \end{vmatrix}$$

Figure 32 – Tree-Generating Determinant

To exemplify the values in the complexity matrix in Figure 31: row two indicates the connections to document number two “Drawings”. The number in the intersecting cell of “row two column two” shows that “Drawings” is linked via four connections to other documents, each connection indicated by the values in the cells on row two: document number one “MRP”, number thirteen “Process Flow Chart”, number fourteen “Process Sequence Chart”, and number fifteen “pFMEA”.

The size of the complexity matrices was directly proportional to the amount of documents in the described processes. Hence, the matrix for the original design was 30x30, whereas the improved design was 16x16. Due to the large size matrices, it was beneficial to ensure that all nodes and arcs were counted for: the sum of each row and column in the respective matrix should be zero. Thus, $\Sigma(N_i + A_{ij}) = 0$ and $\Sigma(A_i + N_{ij}) = 0$.

Complexity Calculation

Once the complexity matrices were developed, the calculations could be made. Microsoft Excel was recommended to be used for the purpose of setting up the matrices and for calculating the determinant by using the built-in formula **=MDTERM()**. Calculating the determinant of a 30x30 matrix would otherwise have been a tedious task. Similarly, the logarithmic value of the determinant was also recommended to be calculated using Excel as the base of the logarithm in the complexity calculation was two, whereas an ordinary scientific calculator computes a logarithm with the base of ten. The formula for calculating the logarithmic value in Excel with the base of two is **=LOG(a,2)**, where **a** is the value to be calculated (the determinant). Note as well, when calculating the determinant, the first row and column of the matrices must be excluded from the calculation. Although, as stated by Robin Whitty, “in fact, any row and column of $L(G)$ may be deleted without changing the absolute value of the result”. Robin Whitty produces and maintains a website “<http://myweb.lsbu.ac.uk/%7Ewhittyr/MathSci/>” which contains useful formulae, tips, and theorems, such as the Excel formulae above.

As such, the complexity value for the original system, represented by the diagram in Figure 28 and the corresponding matrix in Figure 30 was calculated as:

$$C_{cOriginal} = \text{Log}_2[30 \times 42 \times \det L(G_{Original,2,2;30,30})] \rightarrow C_{cO} = 29.11$$

Likewise, the complexity value for the improved system, represented by the diagram in Figure 29 and the corresponding matrix in Figure 31 was thus calculated as:

$$C_{cImproved} = \text{Log}_2[16 \times 20 \times \det L(G_{Improved,2,2;16,16})] \rightarrow C_{cI} = 15.64$$

As a comparison, the improved system has a complexity level at the magnitude of $C_{cOriginal}/C_{cImproved}$ less than the original system, and was calculated as:

$$\Delta C_{cOriginal/Improved} = C_{cOriginal}/C_{cImproved} = 29.11/15.64 \rightarrow \Delta C_{cO/I} = 1.86$$

However, as complexity is not a quantifiable measurement, the correct assessment of the above calculations should be; when compared to the original system, the improved system is considered to be **less complex**. As a result of having measured and compared the complexity of the two information flow diagrams, the questions of whether the improved design is better than the original can thus be answered: the robustness of the design resulting from the uncoupled Axiomatic Design matrix in combination with the irrefutably lower level of complexity of the improved information flow diagram makes it a better system than the original.

Enhanced Engineering Design

The improved design shown in Figure 22 was regarded as having fulfilled all customer and regulatory requirements with the least amount of design parameters. To reiterate however, the complexity of a system is directly related to the amount of nodes as well as the amount of connecting arcs to any one node. Thus, by reducing the amount of nodes and arcs to the lowest practical level, a further enhanced design might be realized. To attempt to reduce the complexity beyond what was achieved by the design of the Axiomatic House of Quality matrix; a few system constraints set out in the chapter Feasibility Study on page 51 had to be disregarded. To reiterate, as the Axiomatic Design matrix was completely uncoupled, the design of the system was as simple as it possibly could be, without compromising on any Functional Requirements. Therefore, for the Functional Requirements to still be nonnegotiable, further improvements to the system could only be realized if constraints would be broken. As the complexity increases with the number of connections, the following constraints were tweaked:

- Integrate the Data Management System with the Assembly Concerns database
- Integrate the Process Documents (Charts) with the Data Management System

As such, by integrating three documents into an existing database and by merging two databases to form one united database, a significant number of connecting arcs could be removed from the system. The system described above is illustrated in Figure 33.

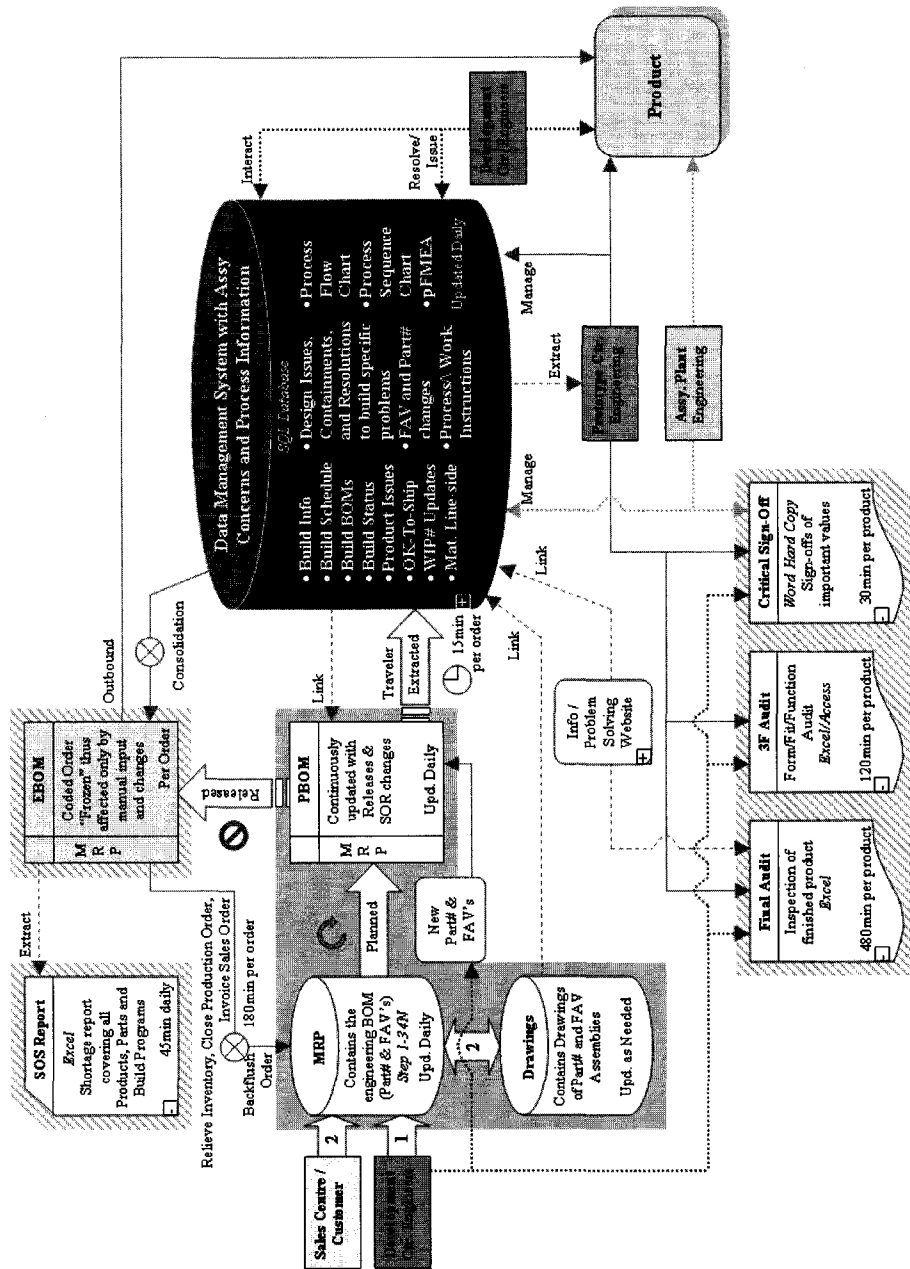


Figure 33 – Enhanced Information Flow Diagram

The enhanced design shown in Figure 33 underwent the same reduction of connections as the previous information flow diagram, meaning all non-generic flow of information was removed. The diagram was thereafter numbered as previously, resulting in the diagram shown in Figure 34, with the corresponding matrix illustrated in Figure 35.

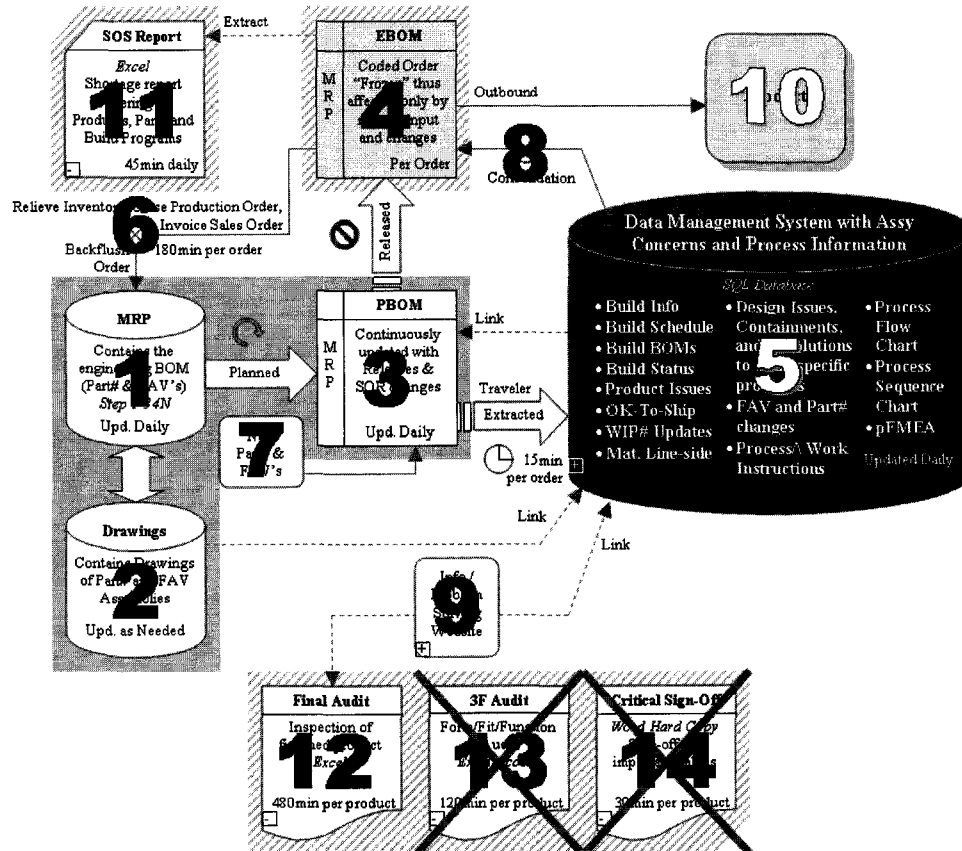


Figure 34 – Numbered Enhanced Information Flow Diagram

Arcs

	1	2	3	4	5	6	7	8	9	10	11	12
1	3	-1	-1	0	0	-1	0	0	0	0	0	0
2	-1	2	0	0	-1	0	0	0	0	0	0	0
3	-1	0	5	-1	-2	0	-1	0	0	0	0	0
4	0	0	-1	5	0	-1	0	-1	0	-1	-1	0
5	0	-1	-2	0	5	0	0	-1	-1	0	0	0
6	-1	0	0	-1	0	2	0	0	0	0	0	0
7	0	0	-1	0	0	0	1	0	0	0	0	0
8	0	0	0	-1	-1	0	0	2	0	0	0	0
9	0	0	0	0	-1	0	0	0	2	0	0	-1
10	0	0	0	-1	0	0	0	0	0	1	0	0
11	0	0	0	-1	0	0	0	0	0	0	1	0
12	0	0	0	0	0	0	0	0	-1	0	0	1

Nodes

Figure 35 – Complexity Matrix for Enhanced System

The complexity value for the enhanced system, represented by the diagram in Figure 34 and the corresponding matrix in Figure 35 was calculated as:

$$C_{c\text{Enhanced}} = \text{Log}_2[12 \times 15 \times \det L(G_{\text{Enhanced}2,2;12,12})] \rightarrow C_{cE} = 13.81$$

Thus, the enhanced system has a complexity level at the magnitude of $C_{c\text{Original}}/C_{c\text{Enhanced}}$ less than the original system, and was calculated as:

$$\Delta C_{cOE} = C_{c\text{Original}}/C_{c\text{Enhanced}} = 29.11/13.81 \rightarrow \Delta C_{cO/E} = 2.11$$

Therefore, by having identified possible causes for increased complexity of the information flow, the system derived from the HOQ and the AHOQ matrices was further enhanced. The ability to add improvements to the system would not have been as apparent without the possibility to measure and compare the level of complexity between design solutions. Nevertheless, it is important to point out that although the cognitive complexity of the overall system in the enhanced design has been reduced with the introduction of the merged database, the internal complexity of the database has increased between the Data Management System, the Assembly Concerns, and the Process Charts. Nonetheless, measuring the complexity within the database goes beyond the scope of this research and will therefore not be explored any further. In addition, to maintain a robust system, the databases should remain separated as in the improved design. Therefore, regardless of the lowered level of complexity in the enhanced design, the improved design illustrated in Figure 22 should still be regarded as optimal, as none of the boundaries of the design were broken while complying with all functional requirements. As well, as the modified formula was demonstrated to accommodate scalability and composition of graphs, the improved design was analytically proven to be better than the original.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

Implementation and Change Management

To increase the success of a project, O'Brien et al. (2004) and Valacich et al. (2004) recommend using a systematic approach when designing or reengineering information systems. The systems approach (to problem solving) is generalized by O'Brien et al. (p. 331) using the following interrelated activities:

1. Recognize and define a problem or opportunity using systems thinking
2. Develop and evaluate alternative system solutions
3. Select the system solution that best meets your requirements
4. Design the selected system solution
5. Implement and evaluate the success of the designed system

The methodology used in this thesis is directly related to the systematic approach as the following activities were implemented:

1. Problem Statement
2. Design Methodology
3. Analysis
4. Deliverables

The fifth stage in the systems approach (Implementation) was not part of the scope of the thesis as the intent was to evaluate the possibilities and benefits of a reengineered and improved system design. Nevertheless, O'Brien et al. indicate that the design and implementation of a corporate-wide system should be the responsibilities of Information System (IS) professionals and specialists.

During the years, a conventional knowledge has emerged that shows two conditions that are essential for a successful implementation of a project. Valacich et al. state the conditions to be “management support of the system under development” and “the involvement of users in the development process” (p. 383).

This project was initiated as a response to the increased accountability of the prototype assembly plant, which unfortunately amplified the complexity of the assembly documents and documentation processes. Fortunately, as the opportunity arose, the project was well supported by management who gave it a high priority and provided support from the central Information Technology (IT) department. In addition, the end-users participated in the project by bringing insight to the current processes as well as bringing forward the needs of the employees, customers, and other stakeholders. Valacich et al. also states, “Despite the support and active participation of management and end users, the implementation of information systems still sometimes fails” (p. 383), which is explained by:

- *Risk* (financial and time constraints in the development process)
- *Commitment to the project* (the problem being solved should be well understood and the system being developed should solve the problem)
- *Commitment to change* (users and management should be keen to change)
- *Extent of project definition and planning* (extensive planning efforts)
- *Realistic user expectations* (the users should early on have realistic expectations about the system and its capabilities)

The concerns listed above should not pose any threat to the implementation of the improved system: the project has been developed without any real time or budgetary

constraints; the problem was well identified, solved, and accommodated using engineering methodologies and product development tools; both management and users show great interest in the development of the system and appreciate a new and improved system; the project has been thoroughly defined, although the planning of the implementation will fall in the hands of the managers; and lastly the users have from the time of initiation of the project been well informed about the expected outcome and potential of the system. Therefore, the implementation of the system should have an optimistic outlook with a significant chance of success. As well, the built-in robustness of the system and low level of complexity mandate a high probability of success.

When deciding to take action in developing and later implementing the designed collaborative data management system, it is important to have awareness of the factors involved: people, processes, and technology are all vital dimensions of change management. O'Brien et al. state "people are a major focus on organizational change management". Thus, activities such as "developing innovative ways to measure, motivate, and reward performance, and designing programs to recruit and train employees in the core competencies" are all required in a changing workplace according to O'Brien et al. (p. 317).

Once the collaborative information system has been tested through alpha and beta versions, the old processes can be moved to the new system. Valacich et al. call this process "installation", but other terms are also used, such as "conversion" as adopted by O'Brien et al. Regardless of the terms, the literature shares the same fundamental approaches to how systems can be converted. The four common approaches to converting an information system are shown in Figure 36 (Valacich et al., 2004, p. 374).

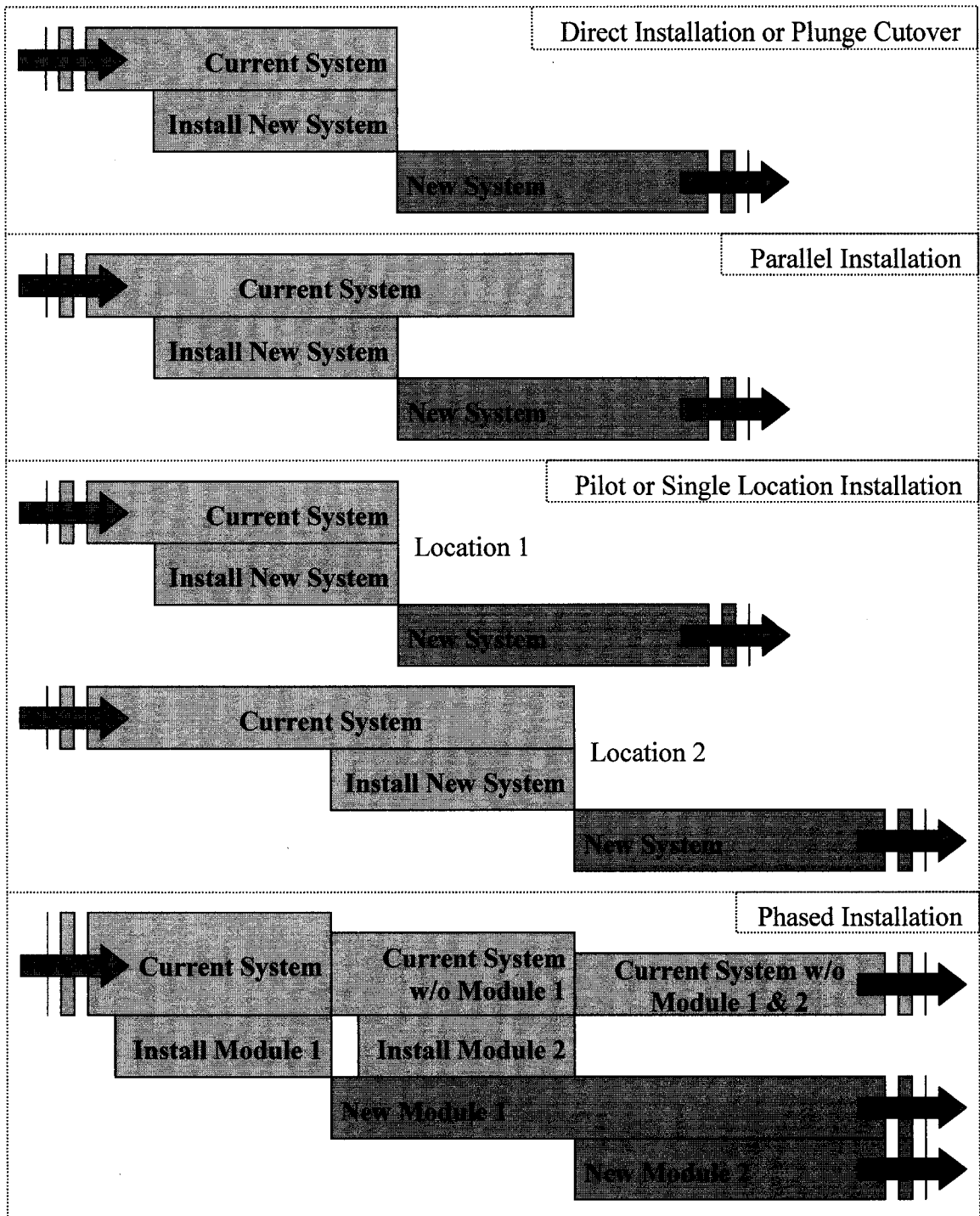


Figure 36 – System Installation Strategies

The most abrupt strategy is the direct installation, where the old system is disconnected as soon as the new system is in place. This approach may be the only approach if the old and the new system cannot coexist, side-by-side. Pulling the plug on the old system can be a good thing though, as the success of the installation is forced to be highly prioritized. In addition, there will be no cost spent on maintaining multiple systems once the new system is operational.

The parallel installation is used when there is a need for playing it safe: the old system is used as a backup in the event of failure or malfunctions to the new system. However, the cost of running the dual systems is very high, as all data must be processed in both systems simultaneously. In addition, having two, or more systems, are both confusing to the users and will cause an increase in redundancy and duplication of processing data.

The pilot installation strategy is a learn-as-you-go approach. This approach can be very successful when users need to be convinced of the potential of the system. However, as with the parallel approach, there is an increase of cost by maintaining different systems. As well, data sharing between the pilot and old systems require the systems to be bridged with the capability of communicating information between them.

The phased installation allows the risk and cost to be spread over time. Each phase should be made small and thus more manageable. However, as with the pilot installation, bridges between the old system and the installed modules need to be in place for the information to be shared among the system as a whole.

As the strategies discussed above involve converting not only the system itself, but also data, hardware, documents, and how work is performed on the system, a single

strategy is commonly not enough. Therefore, in order to accurately assess the probability of a successful implementation, multiple strategies need to be considered. Perhaps different strategies are even needed during the course of a major systems installation to ensure the highest success. However, at the bottom line, it will become the responsibility of the IT department and the managers to decide, regardless of the strategy used.

System Specifications

To make the most of the strengths and potential of the system, Hernandez (2003) recommends using a real database to build the system upon: Microsoft Enterprise SQL with Visual Studio .Net. The results from the House of Quality also showed that Microsoft Enterprise SQL with Visual Studio .Net would suit the company's short and long term goals the best and should undoubtedly be the favourable choice.

The complexity calculations showed that removing obsolete nodes and redundant arcs as well as integrating objects into a database had a positive effect on lowering the overall complexity of the system. It is therefore highly recommended to further research the possibilities to design a database system specifically tailored to include the Process Flow Charts, the Process Sequence Chart, and the pFMEA documents. In addition, the data management system devised to encompass the bills of material, build schedules, product issues, etc., should be integrated with the Assembly Concerns database. Both the Prototype Centre and the Development Centre use information from both systems and the Assembly Concerns database is already coded in SQL. Therefore, it seems most logical to close the gap of information transfer between the two centres by bringing the information together into one database that would service the entire corporation.

Integration with UGS Teamcenter

One request repeatedly addressed from the company was the ability to integrate the engineered information system with UGS Teamcenter. Teamcenter is a Product Lifecycle Management (PLM) tool that is designed to address the key issues of a corporation by “driving product makers – accelerating time-to-market, delivering the right products to market at the right time, extending product life to maximize revenues, increasing productivity, and reducing product costs”. Teamcenter is built on UGS modular PLM platform, which includes “a robust scalable foundation and rich application modules” as shown in Figure 37 originating from UGS.

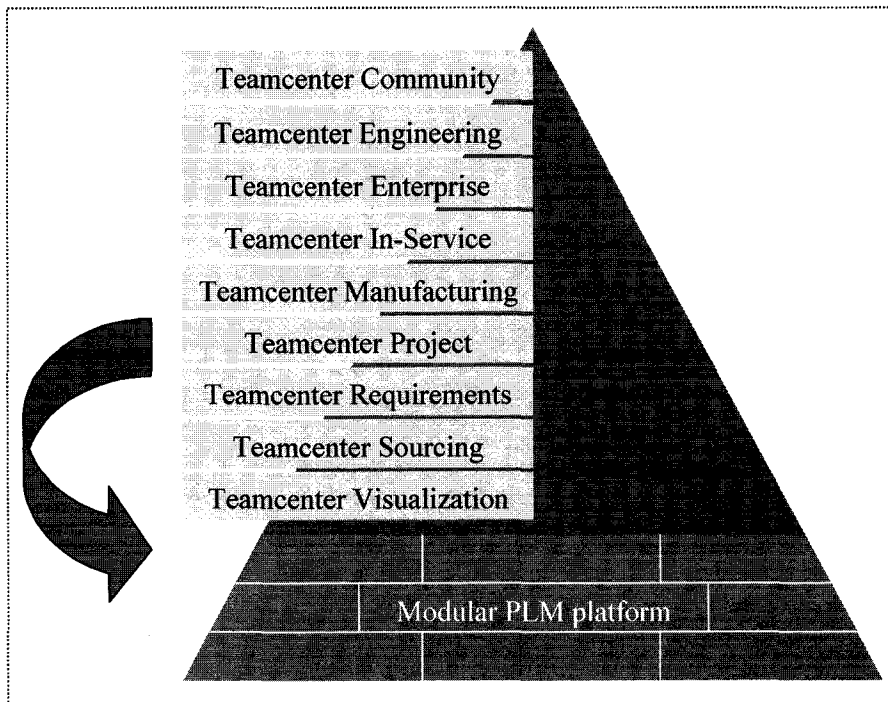


Figure 37 – UGS Teamcenter Modules

Having discussed the implementation strategy with technicians at UGS, it was assured that regardless of design solution of the improved system, or even without improvements at all, Teamcenter can be integrated (in steps) with both the current and with future systems. Teamcenter would at the end be the backbone of all processes.

Further Possibilities and Research

In the research of Latva-Koivisto (2001), it was pointed out that there was a need for converting real-life process charts into graphs, and for accommodating the possibility of measuring complexity independently of size. This research has shed more light on those concerns as the research was based on real-life data and that the Matrix Tree Theorem was modified to more accurately determine the complexity of graphs independent of size or composition. However, there is definitely a need for conducting more research using real data. In addition, there is also an opportunity for evaluating the complexity formula using a survey based approach as devised by Cardoso et al. (2006). In addition, the formula should also be compared and scrutinized against the other formulae used for measuring complexity of systems, as discussed by Latva-Koivisto (2001) and Cardoso et al. (2006). With a positive outlook, a study as such would confirm the formula to be useful in evaluating the complexity of a system.

The formula could then be used in determining the complexity of information flow diagrams, process graphs, hierarchical organization charts, assembly and disassembly sequence charts, product trees, and other similarly visualized system. To allow the complexity to be measured more accurately, research should be conducted on finding ways to weight the nodes and arcs. Using weighted nodes and arcs, the complexity could be converted to a cost or a time unit, which would allow for a system to be evaluated using a currency as a quantitative measure. An example of a weighted graph as such and its legend that could be used for its representation is shown below in Figure 38. The weighted graph is converted from the graph previously shown in the illustration of the Matrix Tree Theorem in Figure 24 on page 84.

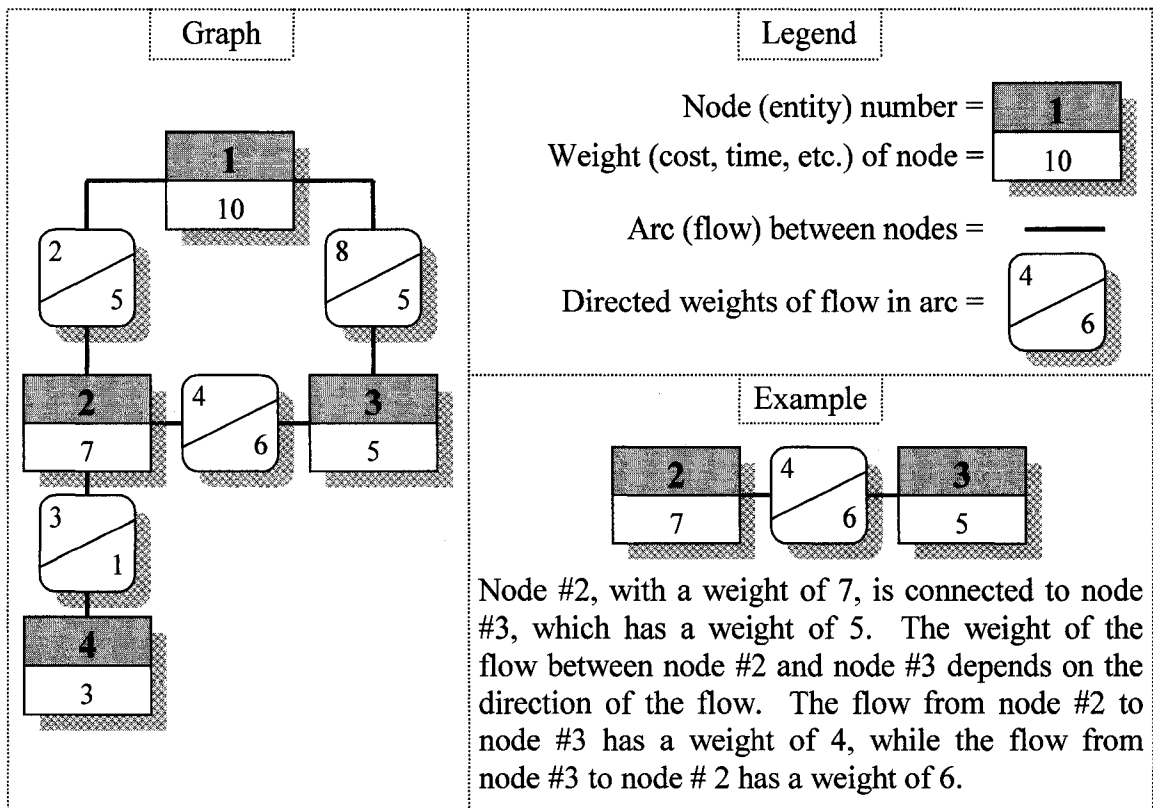


Figure 38 – Example of Weighted Graph

As the illustration shows in Figure 38, the graph is represented with both weighted nodes and weighted arcs, which here are directed as well. The weight of the nodes and the directional flow between the nodes can be associated with quantitative measures, such as currency, time, or distance. With that, the graph can now be evaluated for both its weight as well as its composition, and thus be analytically compared with other similar graphs. However, as with the complexity formula devised in this research, the above mentioned weighted representation of graphs must be tested and confirmed using real data.

Nonetheless, regardless of the outcome from evaluating the modified formula or the weighted graphs, there is an increasing need to find a simple straight-forward approach to measure and evaluate the complexity of business processes and information systems.

Concluding Remarks

The study of this research shows that by using sound engineering methodologies and by not limiting the research to only one field of science, it is possible to design a data management system that has analytical proof for being robust and simple in its design. Having used a systems approach, it was therefore important to acquire a diverse knowledge in the fields of engineering, mathematics, computer science, and management. For example, the system was designed using methodologies from product development, such as IDEF₀ diagram, HOQ, AHOQ, and TRIZ, while it was evaluated by comparing the level of complexity of the systems using Graph Theory and a modified formulation of the Matrix Tree Theorem. In addition, the design of the systems was realized by using tools from product development with a Detailed Engineering Model, and from computer science with deliverables such as Data Flow Diagram and Entity-Relationship Diagram. Recommendations on a successful implementation could also be given using Change Management and System Installation Strategies.

The engineered system consists of a design that successfully met all the requirements set out by the customers and experts. The design entails a database in which many of the previously manual operations are automatically processed. The majority of the documents from the original system were embedded in the database, which acts as a central access point for all operations within the system. Direct links to the source of information replaced the manual transfer of information, minimizing the redundancy and erroneous data. The collaboration effort was greatly improved with Intranet world-wide access to the database. The system also allows all stakeholders simultaneous and immediate views of the information contents, through the robust and cognitively simple design of the collaborative data management system.

CHAPTER VII
DELIVERABLES
Detailed Model

The engineered information system resulting from the methodologies and analyses used in this research is depicted in the Detailed Engineering Model in Figure 39. Through the research, the information system is proven to accommodate all the needs as set out by the various stakeholders and experts.

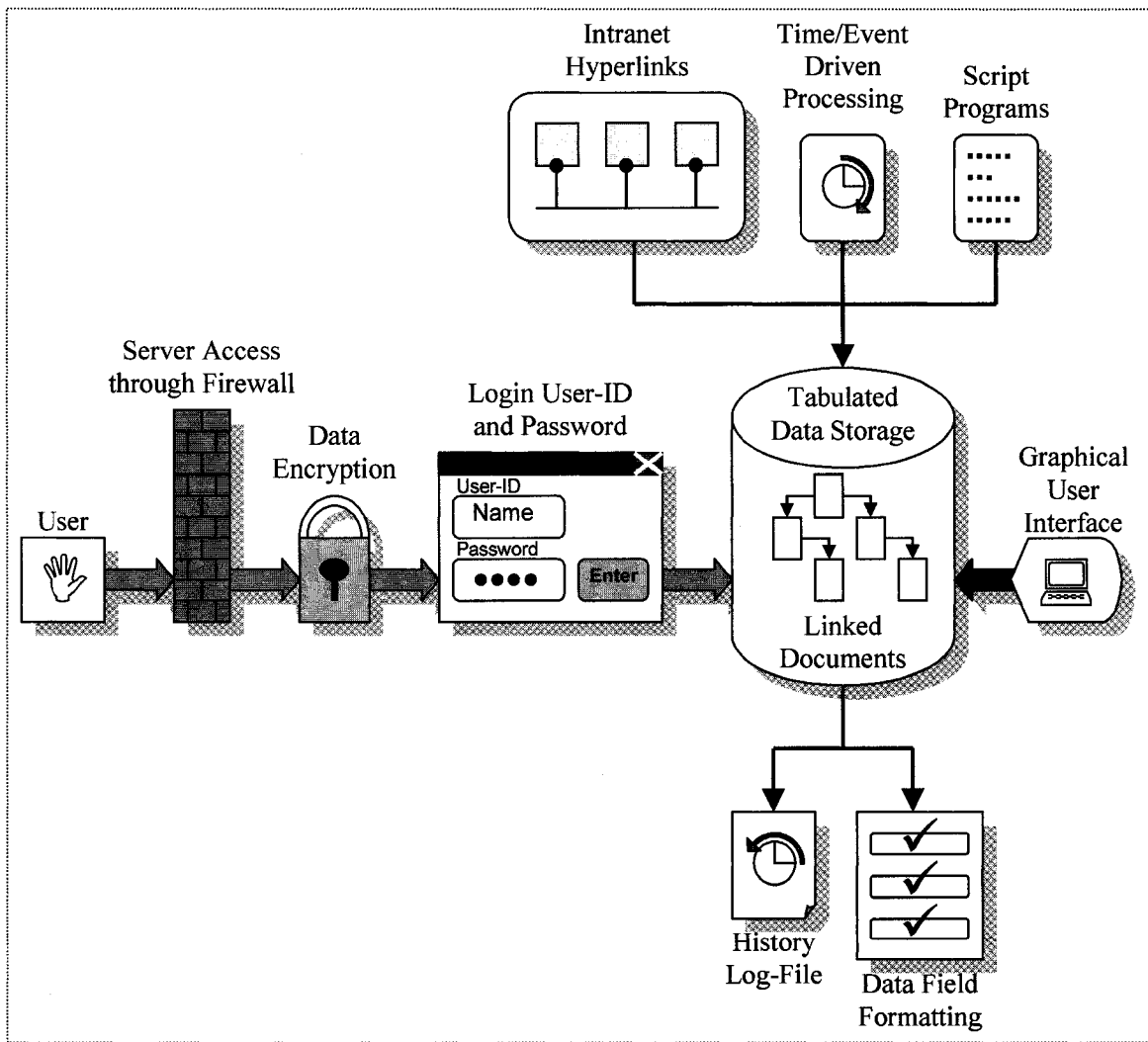


Figure 39 – Detailed Model of the Information System

Data Flow Diagram

The database of the reengineered system can be exemplified in a Data Flow Diagram as shown below in Figure 40.

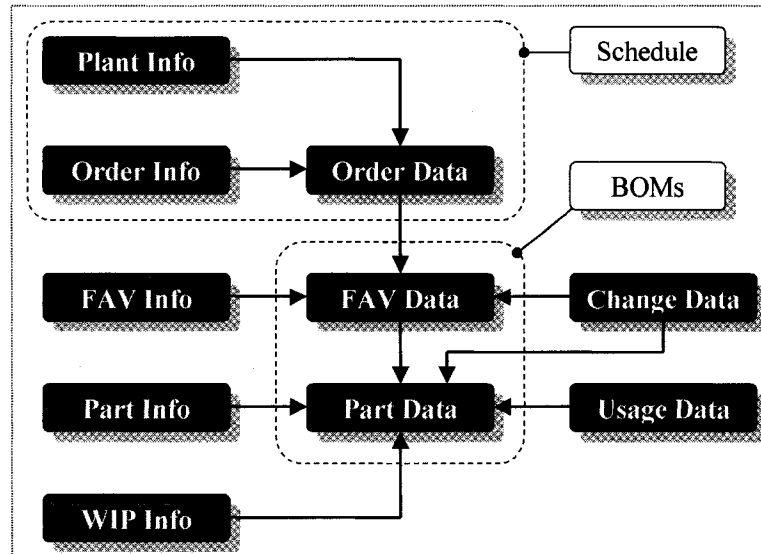


Figure 40 – Data Flow Diagram of Database

The DFD shows two logical units: the Schedule and the BOMs. The Schedule is made up of three entities (tables): “Plant Info”, “Order Info”, and “Order Data”. The BOMs however, consists of two entities: “FAV Data” and “Part Data”. Other entities, such as “FAV Info”, “Part Info”, and “WIP Info” support the BOMs with information surrounding the BOMs. Two additional entities, “Change Data” and Usage Data” make up the augmentation and validation of the BOMs. The “Change Data” contains information about all changes made to the BOMs, such as engineering changes, deviations, substitutions, assembly concerns, etc., whereas the “Change Data” includes all validated part quantities used in the assemblies.

Entity-Relationship Diagram

The relationship between the different entities of the database in the reengineered system can be illustrated using an Entity-Relationship Diagram as shown in Figure 41.

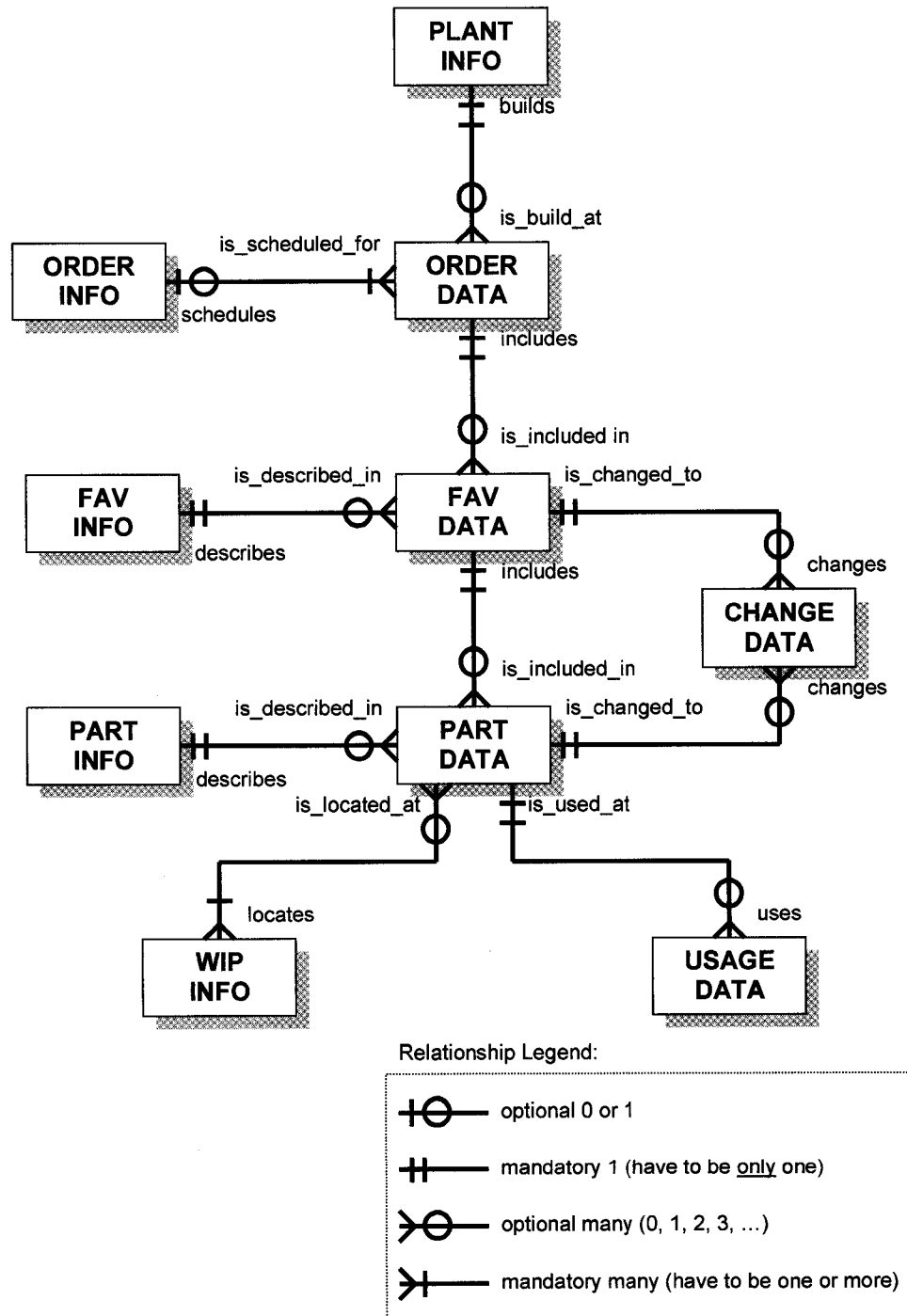


Figure 41 – Entity-Relationship Diagram of Database

Database Table Structure

The logical structure within the different entities of the database, as well as the connections and relationship between the entities, can be seen in Figure 42.

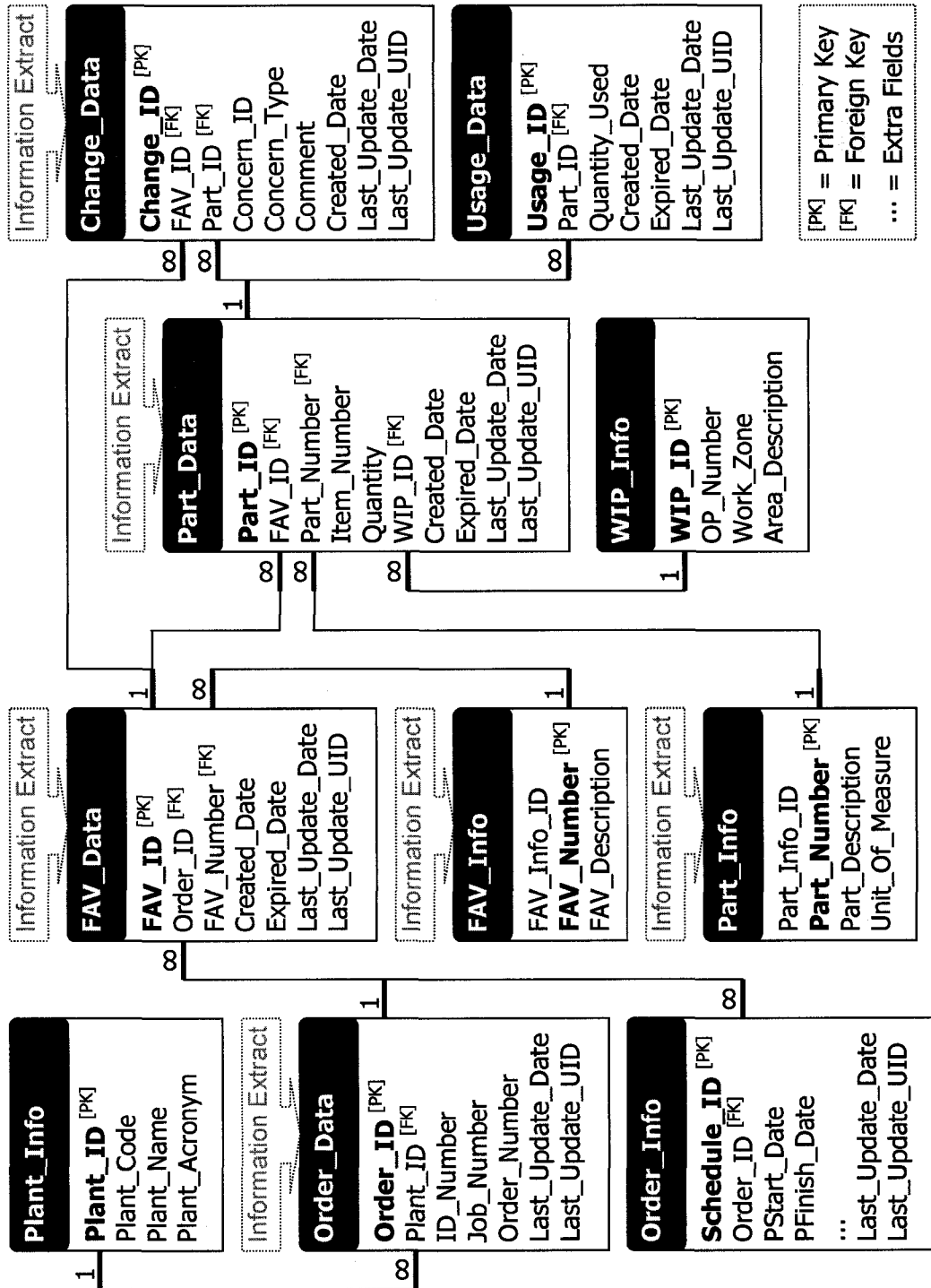


Figure 42 – Database Table Structure

Example of Table Data

To illustrate the relationship between the entities and the logical structure of the database, the tables that follow can be used as an example of how the data would be created, stored, modified, and viewed. The data used in the tables originate from the partial Bill of Material, as shown below in Figure 43 as an Excel spreadsheet.

Installation	Asm Var	Item	Part#	WIP#	New WIP#	Description	Qty	UoM
N010053477	0300					ASSEMBLY A		
N010053477	0300	1	30021R1	120		PART 1	2	PC
N010053477	0300	1	30754R1	110		PART 2	2	PC
N010053477	0300	2	507419C1	120	110	PART 3	1	PC
N010053477	0300	3	3661320C1	120		PART 4	1	PC
N010053477	0300	4	503505C1	120		PART 5	2	PC
N010053477	0300	5	194046H1	840		PART 6	2	PC
N080056049	0104					ASSEMBLY B		
N080056049	0104	3	306132C1	710		PART 7	4	PC
N080056049	0102					ASSEMBLY C		
N080056049	0102	18	306132C1	830		PART 7	1	PC
N080056049	0103					ASSEMBLY D		
N080056049	0103	2	3536111C1	710		PART 8	4	PC
N080056049	0103	2	3544557C1	710		PART 9	4	PC
N080056049	0103	2	306132C1	710		PART 7	4	PC
N090053326	0100					ASSEMBLY E		
N090053326	0100	4	3557745C3	110		PART 10	1	PC
N090053326	0100	4	3571199C1	710		PART 11	2	PC
N090053326	0100	5	3557746C3	110		PART 12	1	PC
N090053326	0100	5	3571199C1	710		PART 13	2	PC
N090053326	0100	6	3531907C2	520		PART 14	2	PC

Figure 43 – Bill of Material (Excel Spreadsheet)

The stricken through lines are removed from the BOM through substitutions.
 “~~double~~ strikethrough” = the whole FAV (Assembly B) is removed.
 “~~single~~ strikethrough” = only the Part# is changed (from 3544557C1 to 306132C1).

Please note that the tables below are shown in its simplest form, but can easily be modified and expanded to accommodate the individual needs of the facilities.

The first database table from the Entity-Relationship Diagram is the **Plant_Info** table. This table contains the data about the different assembly plants, such as the Prototype Centre and the Development Centre.

Plant_Info			
Plant ID [PK]	Plant Code	Plant Name	Plant Acronym
1	021	Prototype Centre	PC
⋮	⋮		⋮
68	034	Development Centre	DC

The second table, the **Order_Data** table, connects the incoming product orders to a specific assembly plants through the “Plant_ID” field. As well, the table shows the order numbers and job schedule numbers for the different orders.

Order_Data						
Order ID [PK]	Plant ID [FK]	ID Number [FK]	Job Number	Order Number	Last Update Date	Last Update UID
1	1	124578XCSD986532	123456	234567	7/08/06	U00L043
2	1	235689WESD875421	123457	456987	7/14/06	U00L043

The third table is the **Order_Info** table. This table shows the current build schedule for the ordered products, which are connected to the Order_Data table through the Order_ID field. As such, the three tables discussed make up the complete schedule.

Order_Info						
Schedule ID [PK]	Order ID [FK]	PStart Date	PFinish Date	...	Last Update Date	Last Update UID
1	1	19/10/06	12/20/06	...	7/08/06	U00L043
2	2	03/11/06	05/02/07	...	7/14/06	U00L043

The Bills of Material are essentially made up of two tables with three supporting tables for the redundant information. The first BOM table, and thus the fourth table, is the **FAV_Data** table. This table contains all FAV numbers used in all the ordered products. The FAVs are connected to the orders through the Order_ID field. In the table, the second row of data have been removed (expired) and changed to the fifth row instead.

FAV_Data						
FAV ID [PK]	Order ID [FK]	FAV Number [FK]	Created Date	Expired Date	Last Update Date	Last Update UID
1	1	N0100534770300	7/08/06		7/08/06	U00L043
2	1	N0800560490101	7/08/06	7/12/06	7/12/06	U00L043
3	1	N0800560490103	7/08/06		7/08/06	U00L043
4	1	N0900533260100	7/08/06		7/08/06	U00L043
5	1	N0800560490102	7/12/06		7/12/06	U00L043
6	2	N0100534770300	7/14/06		7/14/06	U00L043
7	2	N0800560490102	7/14/06		7/14/06	U00L043
8	2	N0800560490103	7/14/06		7/14/06	U00L043
9	2	N0900533260100	7/14/06		7/14/06	U00L043

Job# 123456 and 123457 share the same FAVs, but Job# 123457 was coded without the need of substituting FAV N0800560490101 as it already had N0800560490102.

The second BOM table, the fifth table, is the table where the data about all the parts connected to the FAVs in the orders are stored. The **Part_Data** table contains the information about the parts that is not redundant to other information. The connection to the FAV_Data table is through the FAV_ID field.

Part Data									
Part ID [PK]	FAV ID [FK]	Part Number [FK]	Item Number	Qty	WIP ID [FK]	Created Date	Expired Date	Last Update Date	Last Update UID
1	1	30021R1	1	2	2	7/08/06		7/08/06	U00L043
2	1	30754R1	1	2	1	7/08/06		7/08/06	U00L043
3	1	507419C1	2	1	2	7/08/06	7/11/06	7/11/06	U00L043
4	1	3661320C1	3	1	2	7/08/06		7/08/06	U00L043
5	1	503505C1	4	2	2	7/08/06		7/08/06	U00L043
6	1	194046H1	5	2	32	7/08/06		7/08/06	U00L043
7	3	3536111C1	2	4	25	7/08/06		7/08/06	U00L043
8	3	3544557C1	2	4	25	7/08/06	7/12/06	7/12/06	U00L043
9	4	3557745C3	4	1	1	7/08/06		7/08/06	U00L043
10	4	3571199C1	4	2	25	7/08/06		7/08/06	U00L043
11	4	3557746C3	5	1	1	7/08/06		7/08/06	U00L043
12	4	3571199C1	5	2	25	7/08/06		7/08/06	U00L043
13	4	3531907C2	6	2	23	7/08/06		7/08/06	U00L043
14	4	379990R1	6	2	23	7/08/06		7/08/06	U00L043
15	4	3501427C1	7	1	23	7/08/06		7/08/06	U00L043
16	1	507419C1	2	1	1	7/11/06		7/11/06	U00L043
17	3	306132C1	2	4	25	7/12/06		7/12/06	U00L043
:	:	:	:	:	:	:	:	:	:
56	6	30021R1	1	2	2	7/14/06		7/14/06	U00L043
57	6	30754R1	1	2	1	7/14/06		7/14/06	U00L043
58	6	507419C1	2	1	1	7/14/06		7/14/06	U00L043
59	6	3661320C1	3	1	2	7/14/06		7/14/06	U00L043
60	6	503505C1	4	2	2	7/14/06		7/14/06	U00L043
61	6	194046H1	5	2	32	7/14/06		7/14/06	U00L043

The first six Part# belong to FAV# N0100534770300 on Job# 123456. However, the last six Part# are identical as the same FAV is also found on Job# 123457.

Part# 3544557C1 in FAV# N0800560490103 on row 8 is substituted on Job# 123456 to be replaced with Part# 306132C1 on row 17.

Part# 507419C1 in FAV# N0800560490103 on row 3 is rerouted to OP# 110 on Job# 123456 on row 16. However, the same Part# on row 58 on Job# 123457 was coded correctly.

The first supporting table to the BOM tables, and thus the sixth table, is the **FAV_Info** table. This table basically contains the description of the FAVs. This table is connected to the FAV_Data table through the FAV_Info field.

FAV_Info		
FAV Info ID	FAV Number [PK]	FAV Description
1	N0100534770300	ASSEMBLY A
⋮	⋮	⋮
348	N0800560490101	ASSEMBLY B
349	N0800560490102	ASSEMBLY C

The seventh table, the second BOM support table, is the **Part_Info** table. As with the FAV_Info table, the Part_Info table contains the descriptions of the parts used in the FAVs, in addition to the unit of measure for the different parts. This table is connected to the Part_Data table through the Part_Info field.

Part_Info			
Part Info ID	Part Number [PK]	Part Description	Unit of Measure
1	194046H1	PART 6	PC
2	30021R1	PART 1	PC
⋮	⋮	⋮	⋮
1065	507419C1	PART 3	PC

The last supporting table for the BOMs, the eighth table, is the **WIP_Info** table. The table, connected through the WIP_ID field, contains all work area descriptions.

WIP_Info			
WIP ID [PK]	OP Number	Work Zone	Area Description
1	110	F01	Area One
2	120	F01	Area Two
3	210	A01	Area Three
⋮	⋮	⋮	⋮
32	840	C11	Area Thirty-Two

The ninth table, the **Usage_Data** table, contains the information about all the quantities used in the assembly operations, connected through the Part_ID field.

Usage_Data						
Usage ID [PK]	Part ID [FK]	Quantity Used	Created Date	Expired Date	Last Update Date	Last Update UID
1	1	1	7/10/06		7/10/06	U00L043
2	1	1	7/11/06		7/11/06	U00L043
3	4	1	7/11/06	7/13/06	7/11/06	U00L043

Part# 30021R1 was used 1 piece of on 7/10/06 and 1 piece on 7/11/06 which fulfilled the demand of required quantity of 2.

Part# 3661320C1 was indicated to have been used 1 piece on 7/11/06 but was thereafter recalled on 7/13/06 and thus indicated as not used.

The last table, the **Change_Data** table contains the information about all the changes. The information is connected through the FAV_ID and the Part_ID fields.

Change_Data								
Change ID [PK]	FAV ID [FK]	Part ID [FK]	Concern ID	Concern Type	Comment	Created Date	Last Update Date	Last Update UID
1	2		245613	Assy	FAV not needed	7/10/06	7/12/06	U00L043
2		3		OP	Should be OP#110	7/11/06	7/11/06	U00L043
3		16		OP	New OP# assigned	7/11/06	7/11/06	U00L043
4		8	345123	Assy	Wrong part	7/11/06	7/11/06	U00L043
5	2		424123	Sub	Remove	7/12/06	7/12/06	U00L043
6		8	456723	Sub	Remove	7/12/06	7/12/06	U00L043
7		17	456723	Sub	Add	7/12/06	7/12/06	U00L043

On row 1, FAV# N0800560490101 was issued an Assembly issue against (#245613) and was later, on row 5, removed from the BOM through a substitution (#424123).

On row 2 and 3, Part# 507419C1 was rerouted from OP# 120 to OP# 110.

On row 4, Part# 3544557C1 was issued an Assembly issue against (#345123), removed from the BOM at row 6 and substituted for Part# 306132C1 on row 7 through a substitution (#456723).

APPENDICES

APPENDIX A

Substitution Statistics

Product#	Total Original Parts	# of Original Parts Sub'd	% of Original Parts Sub'd	Total Final Parts	Total Added Parts	# of Total Sub's	% of Total Parts Sub'd	# of TTL # Qty (After Check)	# of OP# Changes	% of OP# Changes
TOTAL SUM	108875	9572	8%	116958	8357	18392	15%	1907	15983	15%
AVERAGE	1728	152		1856	135	292		32	296	
WORST	2848	614	25%	3065	585	1153	38%	120	1164	60%
1	1395	102	7%	1482	87	174	12%	40	34	2%
2	1409	90	6%	1516	107	180	12%	43	31	2%
3	1443	142	10%	1543	100	240	16%	25	29	2%
4	1300	157	12%	1397	97	253	18%	19	53	4%
5	1332	161	12%	1443	111	257	18%	26	N/A	N/A
6	1479	158	11%	1564	85	237	15%	21	84	6%
7	1645	144	9%	1729	84	224	13%	17	76	5%
8	1662	161	10%	1754	92	257	15%	16	145	9%
9	1642	165	10%	1731	89	257	15%	30	102	6%
10	1559	147	9%	1621	62	220	14%	24	260	17%
11	1625	147	9%	1701	76	225	13%	45	217	13%
12	1607	157	10%	1686	79	238	14%	18	89	6%
13	1647	99	6%	1729	82	189	11%	22	136	8%
14	1713	175	10%	1836	123	300	16%	34	115	7%
15	1396	63	5%	1468	72	132	9%			
16	1799	74	4%	1955	156	213	11%			
17	1494	31	2%	1621	127	169	10%			
18	1436	95	7%	1528	92	192	13%	39	243	17%
19	1389	103	7%	1494	105	214	14%	41	136	10%
20	1380	115	8%	1485	105	223	15%	50	269	19%
21	1428	114	8%	1541	113	231	15%	39	203	14%
22	1994	294	15%	2339	345	670	29%	35	N/A	N/A
23	2489	614	25%	3000	511	1152	38%	43	N/A	N/A
24	1969	278	14%	2269	300	607	27%	30	N/A	N/A
25	2236	335	15%	2675	439	796	30%	25	N/A	N/A
26	2004	264	13%	2309	305	566	25%	18	251	13%
27	2153	390	18%	2576	423	818	32%	15	103	5%
28	2480	439	18%	3065	585	1153	38%	21	128	5%
24	2116	297	14%	2514	398	724	29%	30	N/A	N/A
25	1502	154	10%	1574	72	230	15%	20	72	5%
26	1560	78	5%	1630	70	148	9%	42	104	7%
27	1540	101	7%	1606	66	169	11%	40	116	8%
28	1627	104	6%	1709	82	186	11%	21	110	7%
29	1572	104	7%	1635	63	169	10%	29	148	9%
30	1465	91	6%	1528	63	155	10%	24	76	5%
31	1543	74	5%	1611	68	153	9%	23	149	10%
32	1415	102	7%	1500	85	192	13%	26	140	10%
33	1480	107	7%	1566	86	198	13%	29	218	15%
34	1412	101	7%	1501	89	191	13%	30	168	12%
35	1415	67	5%	1512	97	169	11%	36	152	11%
36	1511	111	7%	1588	77	190	12%	26	167	11%
37	1569	100	6%	1644	75	177	11%	40	137	9%
38	1365	78	6%	1467	102	183	12%	34	113	8%
39	1359	78	6%	1448	89	170	12%	34	143	11%
40	1361	77	6%	1445	84	163	11%	32	142	10%
41	1422	90	6%	1507	85	177	12%	40	160	11%
42	1350	80	6%	1438	88	176	12%	35	134	10%
43	1565	95	6%	1618	53	156	10%	30	141	9%
44	1382	92	7%	1462	80	175	12%	44	190	14%
45	1728	133	8%	1828	100	235	13%	6	188	11%
46	1713	76	4%	1803	90	167	9%	23	220	13%
47	2179	35	2%	2246	67	107	5%	75	776	36%
48	2848	180	6%	2574		269	10%	12	1054	37%
49	1637	84	5%	1817	180	263	14%	13	369	23%
50	1710	92	5%	1860	150	243	13%	53	413	24%
51	1929	32	2%	2041	112	150	7%	70	1148	60%
52	2619	262	10%	2669	50	315	12%	120	1105	42%
53	2611	293	11%	2776	165	460	17%	2	1164	45%
54	2577	283	11%	2662	85	372	14%	28	1089	42%
55	1695	104	6%	1827	132	241	13%	21	453	27%
56	2648	252	10%	2737	89	352	13%	66	1084	41%
57	1695	100	6%	1822	127	232	13%	15	460	27%
58	2650	251	9%	2736	86	348	13%	16	976	37%

APPENDIX B

Microsoft Help and Support Article IDs

File Cannot Be Saved:

Article ID : 130494
Last Review : December 2, 2005
Revision : 3.3

Article ID : 214073
Last Review : April 12, 2005
Revision : 4.0

Article ID : 271513
Last Review : February 28, 2006
Revision : 1.1

Article ID : 814068
Last Review : October 5, 2004
Revision : 4.1

Article ID : 913770
Last Review : February 27, 2006
Revision : 1.0

Empty Rows Increases File Size:

Article ID : 244435
Last Review : August 15, 2005
Revision : 4.5

Article ID : 816952
Last Review : January 11, 2006
Revision : 2.1

Article ID : 313275
Last Review : January 23, 2006
Revision : 4.2

Document Categorization Template

Document Name

Document/Media Type:

Originates From:

Information Flow (From):

Information Flow (To):

Storage Point:

Access Point:

Update Occurrence:

Updated By:

(Re)View Occurrence:

(Re)Viewed By:

Document (Active) Lifetime:

Archive Point:

Archive Occurrence:

Archiving Purpose:

Document (Primary) Purpose:

Document (Secondary) Purpose:

Information Contents:

Name	Data Type	Field Size	Example	User*	Owner*

Additional Information: (Printed page header and footer)

Type	Position	Information	Example
Header			
Footer			

File Name Structure:

Sheet Name Structure:

Future State:

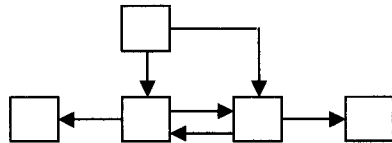
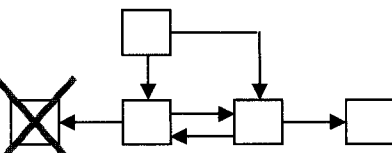
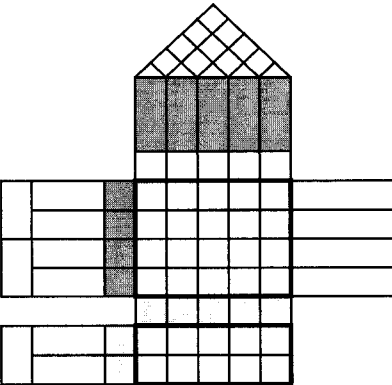
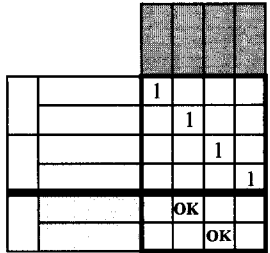
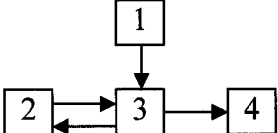
Comments:

TRIZ 40 Inventive Principles**40 Inventive Principles**

Principle 1	Segmentation
Principle 2	Taking out
Principle 3	Local quality
Principle 4	Asymmetry
Principle 5	Merging
Principle 6	Universality
Principle 7	“Nested doll”
Principle 8	Anti-weight
Principle 9	Preliminary anti-action
Principle 10	Preliminary action
Principle 11	Beforehand cushioning
Principle 12	Equipotentiality
Principle 13	“The other way round”
Principle 14	Spheroidality - Curvature
Principle 15	Dynamics
Principle 16	Partial or excessive actions
Principle 17	Another dimension
Principle 18	Mechanical vibration
Principle 19	Periodic action
Principle 20	Continuity of useful action
Principle 21	Skipping
Principle 22	“Blessing in disguise” or “Turn Lemons into Lemonade”
Principle 23	Feedback
Principle 24	“Intermediary”
Principle 25	Self-service
Principle 26	Copying
Principle 27	Cheap short-living objects
Principle 28	Mechanics substitution
Principle 29	Pneumatics and hydraulics
Principle 30	Flexible shells and thin films
Principle 31	Porous materials
Principle 32	Color changes
Principle 33	Homogeneity
Principle 34	Discarding and recovering
Principle 35	Parameter changes
Principle 36	Phase transitions
Principle 37	Thermal expansion
Principle 38	Strong oxidants
Principle 39	Inert atmosphere
Principle 40	Composite materials

APPENDIX E

Step-By-Step Approach to Robust Design of IS

<p>Step 1. Analyze the current system by mapping all transfer of information in an information flow diagram. Identify all entities (nodes) such as files, documents, processes, and databases as well as the communication flow (arcs) between the entities.</p>																										
<p>Step 2. Examine all entities to form a feasibility study which will aid in determining which entities to include in the scope of the improvement. Note: the excluded entities may still need to be included in the new system unless they are made obsolete.</p>																										
<p>Step 3. Examine the system using House of Quality. The HOQ will show which DPs to focus on and which products to best suit the project. HOQ Step-by-Step:</p> <ul style="list-style-type: none"> • Customer Attributes (CAs) “Voice of Customer” • Regulatory Attributes (RAs) “Voice of Experts” • Functional Requirements (FRs) from CAs and RAs • Customer Importance Rating (CIR) assigned to FRs • Design Parameters (DPs) current and potential • Desirability Index (DI) assigned to FRs • Relationship Matrix (RM) • Importance Weighting (IW) • Technical Evaluation of Available Products 																										
<p>Step 4. Design a robust system using the Axiomatic HOQ with a goal of an uncoupled design matrix:</p> <ul style="list-style-type: none"> • Use the FRs and DPs from the HOQ • Remove all “rooms” but (FRs, DPs, DIs, RM) • Convert appropriate FRs to Constraints • Remove obsolete DPs • Convert Relationship Matrix to binary notation • Decouple FR-DP dependencies 																										
<p>Step 5. Design an information flow diagram for the improved system using the result from the HOQ and the AHOQ. Number all entities in the diagram. Remember to add the entities which were excluded in the feasibility study but are still needed.</p>																										
<p>Step 6. Set up the complexity matrix in a spreadsheet using the Matrix Tree Theorem to calculate T. The value of T is calculated as $T = \det L(G)(1 1)$, where $L(G)(1 1)$ equals the matrix obtained by deleting the first row and column of $L(G)$.</p>	<table border="1" data-bbox="1128 1627 1315 1779"> <tr> <td></td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>-1</td> <td>0</td> </tr> <tr> <td>2</td> <td>0</td> <td>2</td> <td>-2</td> <td>0</td> </tr> <tr> <td>3</td> <td>-1</td> <td>-2</td> <td>4</td> <td>-1</td> </tr> <tr> <td>4</td> <td>0</td> <td>0</td> <td>-1</td> <td>1</td> </tr> </table>		1	2	3	4	1	1	0	-1	0	2	0	2	-2	0	3	-1	-2	4	-1	4	0	0	-1	1
	1	2	3	4																						
1	1	0	-1	0																						
2	0	2	-2	0																						
3	-1	-2	4	-1																						
4	0	0	-1	1																						
<p>Step 7. Calculate the complexity C_c of the diagram using the formula $C_c = \text{Log}_2(NxAxT)$ where $N = \# \text{of Nodes}$, $A = \# \text{of Arcs}$, and Log_2 is the binary logarithm.</p>	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> $C_c = \text{Log}_2(NxAxT)$ </div>																									

REFERENCES

Literature

- [1] Alter, S. L. (1980). *Decision Support Systems: Current Practices and Continuing Challenges*. Reading, MA: Addison-Wesley, p. 130.
- [2] Cardoso, J. (2005). *About the Complexity of Teamwork and Collaboration Processes* [Electronic Version]. Proceedings of The 2005 Symposium on Applications and the Internet Workshops.
- [3] Cardoso, J., Mendling, G., Neumann, G., Reijers, H. A. (2006). *A Discourse on Complexity of Process Models* [Electronic Version]. BPM 2006 Workshops, pp. 115-126.
- [4] Chakrabarti, A., (Ed). (2002). *Engineering Design Synthesis: Understanding, Approaches and Tools*. Great Britain: Springer-Verlag London Ltd., pp. 131-148.
- [5] Dhillon, B., S. (2006). *Maintainability, Maintenance, and Reliability for Engineers*. Boca Raton, FL: Taylor & Francis Group, LLC., pp. 2, 65-67, 74-76.
- [6] Eckerson, W. (2002). *Data Quality and the Bottom Line*. Application Development Trends. May 2002, pp. 24-30.
- [7] Flood, R., L. and Carson, E., R. (1988). *Dealing with Complexity: An Introduction to the Theory and Application of Systems Science*. New York, NY: Plenum Press, pp. 1-61.
- [8] Franceschini, F. (2002). *Advanced Quality Function Deployment*. Boca Raton, FL: St. Lucie Press, pp. 1-9, 11-17, 21-32, 35-58.
- [9] Guenov, M. (2001). *Complexity and Cost Effectiveness Measures for Systems Design*. Bedford, United Kingdom: Cranfield University.

- [10] Hernandez, M. (2003). *Database Design for Mere Mortals*. (2nd ed.). Boston, MA: Pearson Education Inc.
- [11] Jones, N., D. (1997). *Computability and Complexity From a Programming Perspective*. Cambridge, MA: The MIT Press, pp. 22-24.
- [12] Kamrani, A., K. and Salhieh, S., M.. (2002). *Product Design for Modularity*. (2nd ed.). Norwell, MA: Kluwer Academic Publishers, pp. 85-104.
- [13] Latva-Koivisto, A. (2001). *Finding a complexity measure for business process models*. Helsinki, Finland: Helsinki University of Technology.
- [14] Laudon, K., C., Laudon, J., P., and Brabston, M., E. (2005). *Management Information Systems: Managing the Digital Firm*. (2nd ed.) Toronto, ON, Canada: Pearson Education Canada Inc., pp. 365-367, 415.
- [15] Logan, G. D., Radcliffe, D. F. (1997). *Potential for Use of a House of Quality Matrix Technique in Rehabilitation Engineering* [Electronic Version]. IEEE Transactions on Rehabilitation Engineering. Vol. 5, No. 1, p.107 & 110.
- [16] Manchulenko, N. (2001). *Applying Axiomatic Design Principles to the House of Quality*. Windsor, ON, Canada: University of Windsor.
- [17] Nicolis, G. and Prigogine, I. (1989). *Exploring Complexity: An Introduction*. Munich, Germany: R. Piper GmbH & Co. KG Verlag, pp. 6, 8, 217-242.
- [18] O'Brien, J. A., Montazemi, A. R. (2004). *Management Information Systems – Managing Information Technology in the Business Enterprise*. McGraw-Hill Ryerson Ltd.
- [19] Olson, J. E. (2003a). *Data Quality: The Accurate Dimension*. San Francisco CA: Morgan Kaufman Publishers.

- [20] Olson, J. E. (2003b). “*The Business Case for Accurate Data.*” Application Development Trends. *June*.
- [21] Ontiveros, B., Soto, I., Wolomberg, M. L., Martinez, L. M., Bouyssi, G., Rios, L., (2002). *Quality Function Deployment applied to the design of Educational Intranet* [Electronic Version]. Argentine: Universidad Nacional de San Juan, and Chile: Universidad de Santiago.
- [22] Piasecki, D., J. (2003). *Inventory Accuracy – People, Processes, & Technology*. Pleasant Prairie, WI: Ops Publishing., pp. 2-3, 19-23, 25-31, 40-42.
- [23] Pyzdek, T. (2003). *Quality Engineering Handbook*. (2nd ed.). New York, NY: Marcel Dekker, Inc., p. 133.
- [24] Simsion, G. C. (2001). *Data Modeling Essentials – Analysis, Design, and Innovation*. Scottsdale, AZ: The Coriolis Group LLC.
- [25] Snyder, L. (2003). *Supply Chain Robustness and Reliability: Models and Algorithms*. Evanston/Chicago, IL: Northwestern University.
- [26] Stackpole, B. (2001). “*Dirty Data Is the Dirty Little Secret That Can Jeopardize Your CRM Effort.*” CIO. *February 15, 2001*, pp. 101-114.
- [27] Strong, D. M. et al. (1997). “*Data Quality in Context.*” Communications of the ACM. *May*.
- [28] Suh, N. P. (2001). *Axiomatic Design. Advances and Applications*. Oxford, NY: Oxford University Press, Inc., pp. 14, 16-17, 57-58, 95, 162, 177-179, 239-296, 470-499.
- [29] Suh, N. P. (2001). *Axiomatic Design Theory for Systems*. Research in Engineering Design. *10*, pp. 189–209.

- [30] Suh, N. P. (2005). *Complexity: Theory and Application*. Oxford University Press, Inc.
- [31] Suh, N. P. (1990). *The Principles of Design*. Oxford, NY: Oxford University Press, Inc.
- [32] Temperley, H. M. V. (1981). *Graph Theory and Applications*. England: Ellis Horwood Ltd.
- [33] Turban, E., Aronson, J. E., Liang, T. (2005). *Decision Support Systems and Intelligent Systems*. (7th ed.). Upper Saddle River, NJ: Pearson Education Inc., pp. 211-295.
- [34] Ulrich, K., T. and Eppinger, S., D. (2004). *Product Design and Development*. (3rd ed.). New York, NY: McGraw-Hill/Irwin, pp. 266-269.
- [35] Valacich, J. S., George, J. F., Hoffer, J. A. (2004). *Essentials of Systems Analysis and Design*. (2nd ed.). Upper Saddle River, NJ: Pearson Education Inc., pp. 154-223, 234-261, 290-291, 314-352, 362-396.
- [36] Young, J. B. (1991). *Modern Inventory Operations*. New York, NY: Van Nostrand Reinhold, pp. 14, 44.

Periodical Documents on the Internet

- [37] Celko, J. (2005, June). It's the Code, Stupid!. Retrieved from
<http://www.dmreview.com>
- [38] Fulbright, R. (2004, June). TRIZ and Software Fini. Retrieved from
<http://www.triz-journal.com/archives/2004/08/02.pdf>

- [39] Hilker, S. and Stover, S. (2006, July 14). Validate Your Way to High Quality PL/SQL Code. Retrieved from <http://www.dmreview.com>
- [40] Inmon, B. (2005, August). Information Management: Encryption at Rest. Retrieved from <http://www.dmreview.com>
- [41] King, R. and Sato, T. (2004, March 12). Relational Database Basics. Retrieved from <http://www.dmreview.com>
- [42] McKnight, W. (2006, July). Building Business Intelligence: Get to the Point. Retrieved from <http://www.dmreview.com>
- [43] Rea, K. C. (2001, September). TRIZ and Software 40 Principles Analogies, Part 1. Retrieved from <http://www.triz-journal.com/archives/2001/09/e/index.htm>
- [44] Rea, K. C. (2001, November). TRIZ and Software 40 Principles Analogies, Part 2. Retrieved from <http://www.triz-journal.com/archives/2001/11/e/index.htm>
- [45] Vaidya, A. (2006, April 11). Master Data Management: A Data Integration Path to Success. Retrieved from <http://www.dmreview.com>
- [46] Zrymiak, D. (2003, July 9). Software Quality Function Deployment: Modifying the “House Of Quality” for Software. Retrieved from <http://www.isixsigma.com/library/content/c030709a.asp>

Nonperiodical Documents on the Internet

- [47] #2214 What are the main differences between Access and SQL Server (February 7, 2006). Retrieved July 4, 2006, from <http://www.aspfaq.com/show.asp?id=2214>
- [48] 40 Inventive Principles With Examples (July 1997), Retrieved November 22, 2006 from <http://www.triz-journal.com/archives/1997/07/b/index.html>

- [49] A discussion of what's new in Access 2007 (formerly "Access 12") (2006, June 05).
Retrieved July 4, 2006, from
<http://blogs.msdn.com/access/archive/2006/06/05/618366.aspx>
- [50] Annual Product Review (n.d.). Retrieved July 6, 2006 from
<http://www.dmreview.com>
- [51] Comparison of Oracle and Microsoft Access SQL (n.d.). Retrieved July 4, 2006,
from *http://eis.bris.ac.uk/~ccmjs/ora_sql.htm*
- [52] Customer-Focused Development with QFD. By Kenneth Crow at DRM Associates.
Retrieved November 23, 2006, from *<http://www.npd-solutions.com/qfd.html>*
- [53] Microsoft Access Specifications and Limits (n.d.). Retrieved July 4, 2006, from
http://www.databasedev.co.uk/access_specifications.html
- [54] Microsoft Help and Support pages (2004-2006). Retrieved July 27, 2006, from
*http://support.microsoft.com/kb/*Article ID*/en-us*
where "**Article ID**" equals the article number referenced to in the text
- [55] SQL Server 2000 and Access 2000 limits (n.d.). Retrieved July 4, 2006, from
http://www.mssqlcity.com/Articles/Compare/sql_server_vs_access.htm
- [56] Theorem of the Day. By Robin Whitty at Brown University. Retrieved November
22, 2006, from *<http://myweb.lsbu.ac.uk/%7Ewhitty/MathSci/>*
- [57] Graphs – Computer Science 016. Brown University. Retrieved November 22, 2006,
from *<http://www.cs.brown.edu/courses/cs016/book/slides/Graphs2x2.pdf>*

VITA AUCTORIS

Bjorn Franzon was born in 1972 in Borås, Sweden. He graduated from Sven Eriksson College in 1994. After having worked in the Injection Mould industry for six years he went on to the University of Jönköping where he obtained a Bachelors degree of Science with a Major in Mechanical Engineering in 2004. He is currently a candidate for the Master's degree in Industrial Engineering at the University of Windsor and plans to graduate in June 2007.