Factors determining the optimal architecture of a manufacturing execution system utilised in automated production.

By

David Lee, BSc (Honours)

Waterford Institute of Technology

INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

A Thesis Submitted in Fulfilment of the Requirements for the Degree of Master of Science

Research Supervisor

Dr. Noreen Quinn-Whelton

Research Supervisor

Michael McCarthy

Submitted to Waterford Institute of Technology

March 2013

Waterford Institute of Technology

INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Declaration:

I declare that the writing of this thesis and the research contained within is my own work. Any assistance received has been acknowledged where appropriate.

Signed: ______________________________

David Lee

Date: ______________________________

Abstract

Information Technology driven manufacturing has progressively evolved since the first introduction of computer systems. As the needs of industry become more demanding, the software used to drive manufacturing must continually advance. In order to remain competitive, businesses need to keep their costs of production down, maintain high product quality and also drive maximum efficiencies. Only through automated manufacturing processes, are businesses able to achieve these goals. High levels of automation require the implementation of sophisticated Manufacturing Execution Systems (MES). This research investigates the optimal design elements for MES, with particular focus on the potential for a web based architecture. Product comparisons, technological investigations and surveys were employed to elucidate MES design and potential areas of enhancement. It was found that while a web based MES could out-perform the traditional client-server model on various aspects, there are too many security concerns to be viable as a cloud hosted application.

Acknowledgements

Thanks to Michael and Noreen for all their assistance. Without the backing of Bausch + Lomb this research would not have been possible.

Glossary

Some acronyms in the glossary have not been elaborated in first appearance of text due to them being synonymous with usage, giving diminished significance to their unabbreviated text. Also a small number of acronyms used in the text are not listed here because they have been mentioned for completeness but have no major relevancy to this research.

Acronym	**Meaning**
ActiveX	Framework for defining reusable software components
AI	Artificial Intelligence
AJAX	Asynchronous JavaScript And XML
API	Application Program Interface
ASCII	American Standard Code for Information Interchange
ASP	Application Service Provider
BAM	Business Activity Monitoring
BI	Business Intelligence
BOM	Bill Of Materials
BPM	Business Process Management
CAMS	MES utilised in Bausch + Lomb
CAPA	Corrective And Preventative Action
CDN	Content Delivery Network
CGI	Common Gateway Interface
CIM	Computer Integrated Manufacturing
COM	Common Object Model
COTS	Common Off The Shelf
CPU	Central Processing Unit
CRM	Customer Relationship Management
CSPRNG	Cryptographically Secure Pseudo Random Number Generator
CSRF	Cross-site request forgery
CSS	Cascading Style Sheets
CSV	Comma Separated Values
DAML	DARPA Agent Markup Language
DEP	Data Execution Protection
DHR	Device History Record
DLL	Dynamic Link Library

DHTML	Dynamic Hyper Text Markup Language
DNS	Domain Naming System
DOM	Document Object Model
DOS	Denial Of Service
eDHR	electronic Device History Record
EDI	Electronic Data Interchange
ERP	Enterprise Resource Planning
FDA	Food and Drug Administration
Frames	HTML technique for displaying several Web pages in the same browser window
GUI	Graphical User Interface
GUID	Globally Unique Identifier
GZIP	Open source command line data stream compressor and archiver
HCI	Human Computer Interaction
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IDS	Intrusion Detection System
IE	Microsoft Internet Explorer
iFrames	HTML technique for placing a window within a window in same browser window
IIS	Microsoft Internet Information Services
IT	Information Technology
JavaScript	Scripting language supported by all mainstream browsers
JIT	Just In Time
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LAN	Local Area Network
LIMS	Learning Information Management System
Mashup	Web page that aggregates content from different sources
MDAC	Microsoft Data Access Component
MES	Manufacturing Execution Systems
MIM	Man In the Middle
MRP	Materials Requirement Planning
MVC	Model View Controller
NPI	New Product Introduction

ODBC	Open Database Connectivity
OEE	Overall Equipment Effectiveness
OLAP	OnLine Analytical Processing
OPC	OLE for Process Control
OS	Operating System
OWL	Web Ontology Language
P2P	Peer to Peer
PC	Personal Computer
PDA	Personal Digital Assistant
PERL	Practical Extraction and Report Language
PLC	Programmable Logic Controller
PRNG	Pseudo Random Number Generator
RAC	Real Application Cluster
RAD	Rapid Application Development
RDF	Resource Description Framework
REST	REpresentational State Transfer
RFC	Request for Comment
RFID	Radio-frequency identification
ROI	Return On Investment
RS232	Standard for communication to devices through serial ports
RSS	Really Simple Syndication
RXTX	Java framework for communication to devices via RS232
SAN	Storage Area Network
SCADA	Supervisory Control And Data Acquisition
SDLC	Software Development Life Cycle
SFIS	Shop Floor Information System
SIP	Session Initiation Protocol
SOA	Service Orientated Architecture
SOAP	Simple Object Access Protocol
SOX	Sarbanes OXley
SPARQL	RDF query language
SPC	Statistical Process Control
SQL	Structured Query Language
SSI	Server Side Includes
SVG	Scalable Vector Graphics

UML	Unified Modelling Language
UNC	Universal Naming Convention
URL	Uniform Resource Locator
VBScript	VB Scripting language used for Internet Explorer browser and windows platforms.
VPN	Virtual Private Network
W3C	World Wide Web Consortium
WaSP	Web Standards Project
WIP	Work In Progress
WSDL	Web Services Description Language
WWW	World Wide Web
XHTML	eXtensible HyperText Markup Language
XLink	XML Linking Language
XML	Extensible Markup Language
XPointer	XML Pointer Language
XSL	eXtensible Stylesheet Language
XSLT	eXtensible Stylesheet Transformation Language
XSS	Cross-site scripting

Table of Contents:

1 INTRODUCTION.......16

1.1 DEFINITION OF MES.......16
1.2 RESEARCH QUESTIONS.......18
1.3 JUSTIFICATION.......18
1.4 SCOPE LIMITATIONS.......19
1.5 OUTLINE OF THESIS.......20

2 METHODOLOGY.......23

2.1 HYPOTHESES.......24
2.2 COMPARATIVE ANALYSIS OF COTS MES.......25
2.3 GOVERNANCE OF MES DESIGN THROUGH INDUSTRY STANDARDS AND REGULATION.......25
2.4 UTILIZATION OF WEB BASED SURVEY FOR IN-DEPTH MES ANALYSIS.......26
2.5 CONCLUSION.......30

3 LITERATURE REVIEW.......31

3.1 INTRODUCTION.......31
3.2 THE HISTORY AND ADVANCES OF MANUFACTURING.......32
3.3 MACRO-ECONOMIC ASPECTS OF AUTOMATION.......34
3.4 CONTEMPORARY MES DESIGN PRINCIPALS.......36
3.5 EMERGENT MES TRENDS.......37
3.6 PROBLEMS ASSOCIATED WITH DESKTOP SOFTWARE.......38
3.7 BENEFITS OF WEB BASED MANUFACTURING SYSTEMS.......41
3.8 WEB BASED DEVICE COMMUNICATION.......43
3.9 REUSABLE MODELS FOR OPTIMAL SOFTWARE DESIGN.......48
3.10 MES USABILITY.......53
3.11 UTILIZING TOUCH SCREENS IN A MANUFACTURING ENVIRONMENT.......63
3.12 MES SECURITY.......64
3.13 SYSTEM ERRORS AND FAILURE MODES.......66
3.14 REVIEW OF OTHER MES APPLICATIONS.......73
3.15 LITERATURE REVIEW CONCLUSIONS.......78

4 WEB SOFTWARE FUNCTIONALITY.......82

4.1 WORLD WIDE WEB CONSORTIUM 82
4.2 WEB BROWSER STANDARDS COMPLIANCE 83
4.3 XML 85
4.4 JAVASCRIPT OBJECT NOTATION 86
4.5 AJAX 87
4.6 CSS 88
4.7 REUSABLE JAVASCRIPT FRAMEWORKS 89
4.8 WEB 2.0 90
4.9 ADVANCED PERFORMANCE TECHNIQUES & OPTIMISATIONS 90
4.10 SEMANTIC WEB 101
4.11 NETWORK LOAD BALANCING AND WEB FARMS 101
4.12 DATABASE CONNECTION POOLING 102
4.13 NEW PROGRAMMING LANGUAGES SUCH AS ASP.NET & JAVA 103
4.14 WEB CACHING 104
4.15 VIRTUAL PRIVATE NETWORKS 105
4.16 GRID COMPUTING 106
4.17 WEB SERVICES 108
4.18 SERVICE ORIENTATED ARCHITECTURE 108
4.19 CLOUD COMPUTING 109
4.20 CLUSTERING 110
4.21 STORAGE AREA NETWORKS 111
4.22 WEB ANALYTICS 111
4.23 CONCLUSION 113

5 INDUSTRY STANDARDS AND REGULATIONS 114

5.1 REGULATORY GOVERNANCE AND COMPLIANCE 114
5.2 STANDARDS ADHERENCE 116
5.3 CONCLUSION 117

6 RESULTS AND DISCUSSION 119

6.1 INDUSTRY STANDARDS AND REGULATIONS 119
6.2 WEB BASED DEVICE COMMUNICATIONS 119
6.3 TECHNICAL SURVEY FINDINGS 120
6.4 USER SURVEY FINDINGS 133

6.5 REVIEW OF OTHER MES APPLICATIONS 141
6.6 WEB TECHNOLOGY REVIEW 144

7 CONCLUSIONS 147

7.1 CONSIDERATIONS ON DESIGN EFFORT 147
7.2 RESEARCH SURVEY COLLABORATION WITH BAUSCH + LOMB 149
7.3 MES PRODUCT INSIGHTS 150
7.4 RECOMMENDATIONS FOR THE FUTURE 152
7.5 FUTURE RECOMMENDATIONS 152

REFERENCES 154

APPENDICES 164

A. CWE/SANS TOP 25 SECURITY FLAWS 164
B. COMMERCIAL MES FUNCTIONALITY COMPILATION 184
C. GLOBAL BAUSCH + LOMB MES SURVEY QUESTIONS (END USER) 208
D. GLOBAL BAUSCH + LOMB MES SURVEY QUESTIONS (TECHNICAL) 217

Table of Figures:

FIGURE 1-1. OVERVIEW OF MES FUNCTION AND RELATED SYSTEMS (MESA 1997B) .. 17
FIGURE 1-2. MES OPERATIONAL BENEFITS MODEL (MESA 1997B) 19
FIGURE 2-1. BLOCK DIAGRAM OF RESEARCH METHODOLOGY FUNDAMENTALS 23
FIGURE 3-1. PROPOSED MULTI-TIER WEB MES MODEL 36
FIGURE 3-2. MES AND RELATED SYSTEMS 38
FIGURE 3-3. DECOMPILED .NET BINARY CODE 40
FIGURE 3-4. RS232 DEVICE COMMUNICATION UML DIAGRAM (LEE ET AL. 2007) 45
FIGURE 3-5. DEVICE COMMUNICATION UML DATA FLOW MODEL (LEE ET AL. 2007) ... 47
FIGURE 3-6. HICK'S LAW (TEACH PE 2011) 54
FIGURE 3-7. FITT'S LAW (PARTICLETREE 2011) 55
FIGURE 3-8. FITT'S LAW TARGET AREA ACQUISITION (PARTICLETREE 2011) 56
FIGURE 3-9. THE POWER LAW OF PRACTICE (ROESSINGH AND HILBURN 2000: 7) 57
FIGURE 3-10. ATKINSON AND SHIFFRIN MEMORY CONSTRUCT (SCIENCE AID 2010) 58
FIGURE 3-11. EBBINGHAUS'S SERIAL POSITION EFFECT (SIMPLY PSYCHOLOGY 2013) .. 60
FIGURE 3-18. GRACEFUL HANDLING OF AN ERROR STATE 70
FIGURE 3-19. UNHANDLED ERROR DISPLAY 71
FIGURE 4-1. W3C GOVERNING TECHNOLOGY MAP (W3C 2004) 83
FIGURE 4-2. WASP CSS TEST IN COMPLIANT BROWSER 84
FIGURE 4-3. WASP CSS TEST IN NON-COMPLIANT BROWSER 84
FIGURE 4-4. COMPARISON OF HTTP VERSUS AJAX (ADAPTIVE PATH 2009) 88
FIGURE 4-5. CONFIGURING HTTP COMPRESSION IN IIS 93
FIGURE 4-6. SAMPLE SPRITE IMAGE (IGNACIO RICCI 2009) 94
FIGURE 4-7. EXAMPLE OF DEFERRED PAGE LOADING UTILIZING AJAX 95
FIGURE 4-8. COMPLETION OF DEFERRED PAGE LOADING UTILIZING AJAX 96
FIGURE 4-9. EXAMPLE OF USING FIREBUG PLUG-IN TOOL FOR FIREFOX 99
FIGURE 4-10. EXAMPLE OF USING YSLOW PLUGIN TOOL FOR FIREBUG 100
FIGURE 4-11. DATABASE CONNECTION POOLING (ORACLE DOCUMENTATION 2011) . 103
FIGURE 4-12. VPN OVERVIEW (NETWORKED ELEMENTS 2010) 106
FIGURE 4-23. RAC OVERVIEW (ORACLE DOCUMENTATION 2010) 107
FIGURE 4-14. BASIC SOA FLOW OVERVIEW (SERVICE ARCHITECTURE 2011) 109
FIGURE 4-15. USE OF CLUSTERING FOR PROVISION OF FAULT TOLERANT SERVICE 110
FIGURE 6-1. BALANCE OF CONFIGURABLE MES VERSUS GENERIC PRODUCT 121

FIGURE 6-2. BALANCE OF RESPONSE TIMES V BUSINESS LOGIC 122
FIGURE 6-3. PROJECT TRIANGLE MODEL (MIS SERVICES 2010) 123
FIGURE 6-4. MES ATTRIBUTE RANKING 124
FIGURE 6-5. REMOTELY HOSTED MES RELIABILITY 125
FIGURE 6-6. STANDARDS COMPLIANCE 126
FIGURE 6-7. REGULATORY COMPLIANCE 127
FIGURE 6-8. INCREASING INTERFACE PROMPTING 128
FIGURE 6-9. MES INTEGRATION CAPABILITIES 129
FIGURE 6-10. MES REPORTING INTEGRATION LEVELS 130
FIGURE 6-11. REAL-TIME MES REPORTING TOOLS 130
FIGURE 6-12. MES DATA MINING 131
FIGURE 6-13. MULTI-GEOGRAPHIC DATA AMALGAMATION 132
FIGURE 6-14. PRIMARY INPUT METHOD 133
FIGURE 6-15. PREFERRED INPUT METHOD 134
FIGURE 6-16. ACCEPTABLE MES RESPONSE TIMES 137
FIGURE 6-17. LOST PRODUCTIVITY DUE TO SLOW MES RESPONSE 137
FIGURE 6-18. MES TRAINING NEEDS 138
FIGURE 6-19. PREFERRED TRAINING METHODS 139
FIGURE 6-20. OPENNESS TO MORE COMPUTER BASED TRAINING 140
FIGURE 6-21. PERCEPTION OF MES FUNCTION 141
FIGURE A-1. UNFILTERED ERROR DETAILS TRANSMITTED TO CLIENT WEB PAGE 171
FIGURE A-2. USING FIREBUG TO VIEW WITH 3RD PARTY WEB SITE CODE 172
FIGURE A-3. USING FIREBUG TO TAMPER WITH 3RD PARTY WEB SITES 173
FIGURE A-4. LISTING OF A CONSTANT VALUE HARDCODED INTO A .NET PROJECT 179
FIGURE A-5. PEEKING OF CONSTANT VALUE HARDCODED INTO A .NET ASSEMBLY 180
FIGURE A-6. FIREFOX OPTION CONFIGURATION PAGE 183

Table of Code Figures:

CODE 2-1. SURVEY CHOICE EXAMPLE 28
CODE 3-1. HICK'S LAW 54
CODE 3-2. FITT'S LAW 55
CODE 3-3. THE POWER LAW OF PRACTICE 56
CODE 3-4. PATTERSON'S COST OF DOWNTIME 67
CODE 3-5. DATABASE TRANSACTION IN VB.NET 72
CODE 4-1. XML EXAMPLE 85
CODE 4-2. JSON EXAMPLE 86
CODE 4-3. XML COMPARISON TO JSON 87
CODE 4-4. CSS CODE EXAMPLE 89
CODE 4-5. JAVASCRIPT BEFORE MINIFYING 91
CODE 4-6. JAVASCRIPT AFTER MINIFYING 92
CODE 4-7. META TAG REFRESH EXAMPLE 92
CODE 4-8. CSS SPRITE EXAMPLE 94
CODE 4-9. CALLING A DELAYED CONTENT LOADING JAVASCRIPT FUNCTION 96
CODE 4-10. JAVASCRIPT FUNCTION FOR DELAYED CONTENT LOADING 97
CODE 4-11. PRE-FETCHING IMAGES WITH JAVASCRIPT 97
CODE 4-12. HTML LINK USED FOR DELAYED CONTENT LOADING 97
CODE 4-13. META TAG FOR DELAYED CONTENT LOADING 98
CODE 4-14. HTML FORM REQUEST TAG 100
CODE 4-15. DATABASE CONNECTION POOLING PARAMETER FOR .NET 103
CODE 4-16. META TAG FOR PAGE CACHING 104
CODE 4-17. META TAG FOR NO PAGE CACHING 104
CODE 4-18. ASP.NET CACHING 104
CODE 4-19. TRACERT OUTPUT 105
CODE A-1. HACK ON IMPROPER ESCAPE ENCODING 164
CODE A-2. .NET METHOD TO PREVENT IMPROPER ENCODING 165
CODE A-3. TYPICAL SQL USED FOR WEBSITE LOGIN 165
CODE A-4. SQL INJECTION HACK 165
CODE A-5. FINAL SQL AFTER SQL INJECTION HACK 165
CODE A-6. DYNAMIC SCRIPT EXECUTION 167
CODE A-7. MALICIOUS DYNAMIC SCRIPT EXECUTION 167

CODE A-8. DYNAMIC SCRIPT EXECUTION 167
CODE A-9. SIMPLE CSRF EXPLOIT 168
CODE A-10. POST CSRF REQUEST 169
CODE A-11. HTML CODE BEFORE ALTERATION 173
CODE A-12. HTML CODE AFTER ALTERATION 173
CODE A-13. SERVER SIDE INCLUDE 174
CODE A-14. SQL DYNAMIC CALCULATION EXAMPLE 175
CODE A-15. PROPER DATABASE CONNECTION CLOSING IN JAVA 176
CODE A-16. IMPROPER C VARIABLE DECLARATION 177
CODE A-17. CORRECT C VARIABLE DECLARATION 177
CODE A-18. CORRECT RANDOM NUMBER GENERATION 181
CODE A-19. GUID GENERATION IN VB.NET 182
CODE A-20. GENERATED GUID EXAMPLE 182

1 Introduction

1.1 Definition of MES

MES is a computerised real time system used to manage and control an entire manufacturing process comprised of machines, personnel, and support services. MES tracks activities and resources on the shop-floor, and is often integrated with other applications used within organisations for the purposes of purchasing, inventory control, preventative maintenance, scheduling, real-time monitoring of production data etc. (Feng and Zhang 1998: 364 - 365). The deployment of MES can vary depending on the manufacturing process type it is implemented in (Kletti 2007: 31). Although this software is most commonly known as MES, the organic-like growth and disparate industry type implementations has resulted in a plethora of alternate names such as: Manufacturing Systems, Manufacturing Automation System, Computer Aided Manufacturing, Computer Integrated Manufacturing (CIM) Shop Floor Information System (SFIS), Shop Floor Control System and other variants. An overview of the main MES functions and some of its connecting systems is illustrated in the following Figure 1-1:

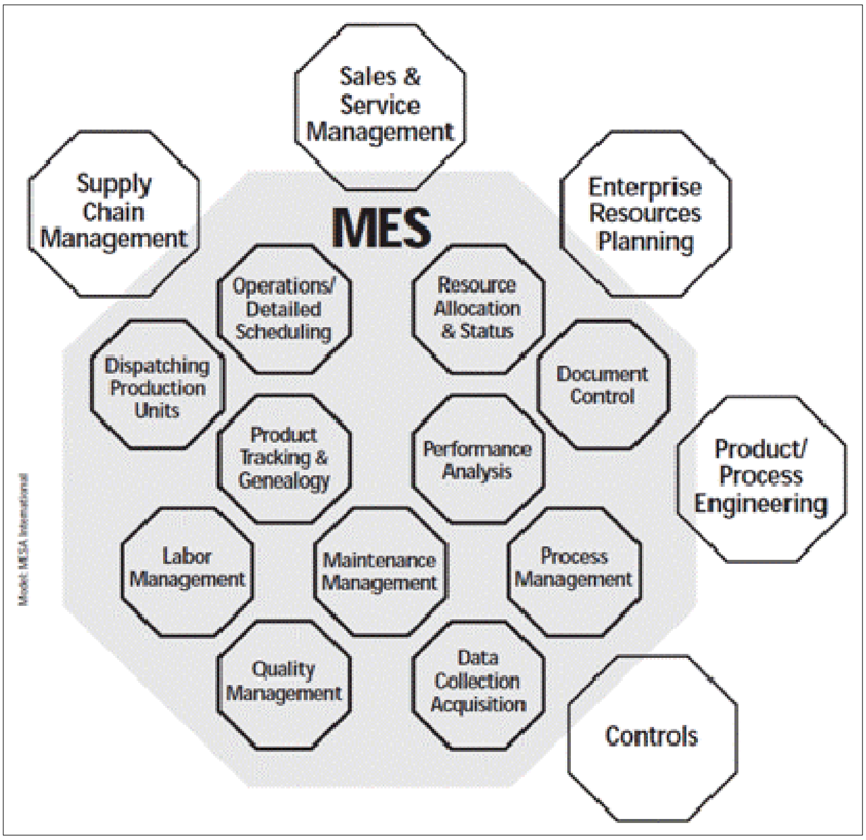

Figure 1-1. Overview of MES function and related systems (MESA 1997b)

MES rarely operates in isolation, and typically interconnects with other systems such that the delineation of MES can become ambiguous. A portion of the low-level machine control MES function is defined as being Supervisory Control And Data Acquisition (SCADA). Above that, MES integrates with Enterprise Resource Planning (ERP) systems used for visibility into plant wide operations. Currently there is a growing industry trend to label MES and its related systems as Manufacturing Intelligence Systems. Throughout this research the label of MES is predominantly employed in order to maintain consistency, however, reference sources may have used a different label such as SFIS etc.

The industry MES is employed in varies greatly with contrasting levels of automation integration. These could range from pharmaceutical drug manufacture to timber yard cutting. While these example MES implementations are quite contrasting, they have the commonality of involving a manufacturing process that produces a finished product. This research will be primarily focused on high automation MES implementations.

1.2 Research Questions

The decision to conduct research into MES was chosen due to the author's familiarity with the MES industry software, and also because of the potential to improve the custom developed MES application utilised in the Bausch + Lomb Corporation. Improvements to MES are achieved by enabling business to be more efficient and make more profit. Two questions became prominent during the investigation into MES research. These questions address the relevant aspects of the research topic and attempt to provide the foundation for the research subject.

The first question is: “What factors determine the optimal design of MES software utilised in automated manufacturing?”. MES has a relatively mature design considering it has been in existence for decades. From an academic perspective, it is hoped that the best possible architecture can be formulated without the constraining influence from industry norms and legacy design.

The second question is: “Can web based software perform the functionality of client-server MES software?”. Due to the complexity and large amount of functionality, it is not possible for one person to fully develop a web based MES as proof of comparison. Instead this research will attempt to map out relevant design aspects.

1.3 Justification

Manufacturing has gone through numerous advances from its early stages of cottage industries to current day high volume manufacturing. Every aspect of it has evolved towards the goal of manufacturing the highest volume of product possible at the greatest efficiency. In the beginning, the advances were primarily through socio-technical improvements, often as giant leaps in progress. Current day manufacturing advances are not so revolutionary but still have the potential to benefit through greater efficiencies. Businesses need to differentiate from the competition by developing a strategy which offers customers additional added value to products such as high flexibility, short delivery times and high quality; This leads on to the need for “adaptive manufacturing” (Kletti 2007: 1). Research into MES usage is very clear on the many benefits gained through the use of MES (MESA 1997a: 5 -7). MES is employed for the majority of manufacturing and has the potential to further drive advances. Figure 1-2 illustrates the different types of benefits that MES can develop:

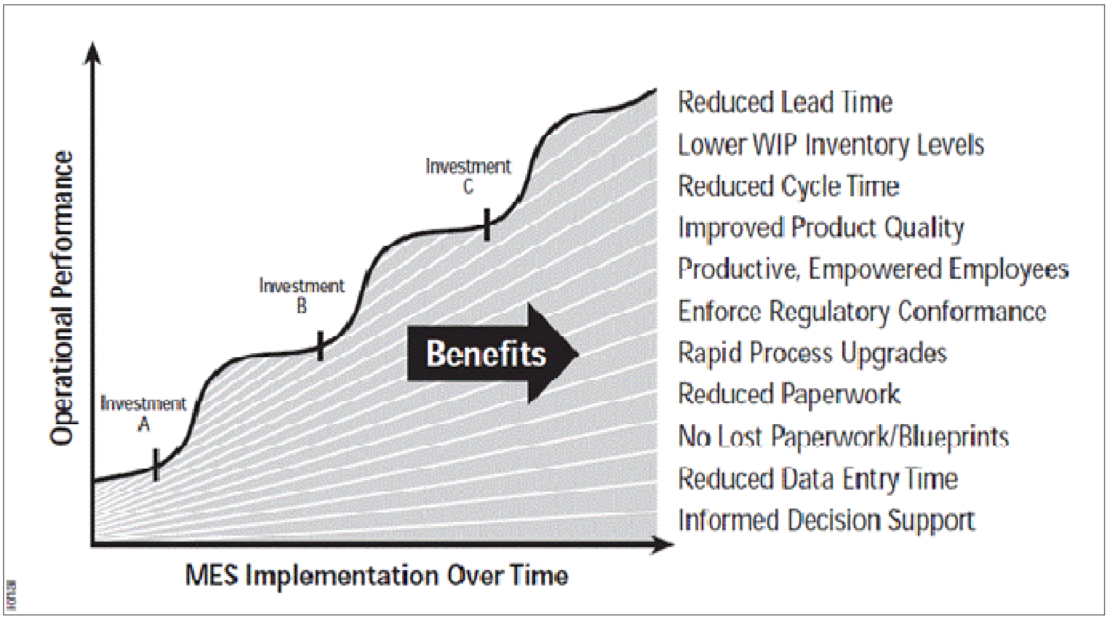

Figure 1-2. MES operational benefits model (MESA 1997b)

This research aims to elucidate the optimal design factors concerning MES, and prescribe how it can assist in the continuance of manufacturing advancement and attainment of greater levels of operating efficiencies.

1.4 Scope limitations

MES is utilised in varying heterogeneous manufacturing environments. Industry sectors such as pharmaceutical and heavy industrial processing have contrasting needs with differing MES design requirements. Even within a single industry type, manufacturing processes and techniques constantly evolve, which in turn drive changing MES requirements. Where possible, this research has been conducted so as to benefit as many MES variations as possible. Where this generic focus reduces the efficacy of the research, more focus is placed upon highly automated healthcare environments. Scope is further reduced to the specific operating environment utilised within Bausch + Lomb's manufacturing environment, for the purposes of potentially yielding greater benefit to Bausch + Lomb. This narrowed focus is also applied to reduce the scope of research work where it would be infeasible to cover all environments.

There are many related and interconnecting systems to MES. It is the purpose of this research to concentrate on the core MES architecture. While related systems will be discussed where appropriate to elucidate aspects of MES, they will not be covered in any detail.

1.5 Outline of thesis

The research input to this thesis has come from online surveys, comparative evaluation of MES Common Off The Shelf (COTS) software and also guidance from a broad array of industry experts in various fields such as: Global Supply Chain Director, Global MES Director, Database Manager, IT Manager, Software Development Manager, Software Engineer, Software Quality Engineer, Network Operation Manager and other related IT disciplines. A technical director for a leading software development company based in Waterford, has also provided direction in this thesis on emergent industry trends and future growth. Guidance and direction on MES design has also been taken from the MESA International association. The researcher also has extensive experience in a number of IT disciplines, working closely with the bespoke MES and related systems utilised in Bausch + Lomb.

Because technology is an ever-changing landscape, attempting to prescribe a specific MES design would have little value. No sooner than an optimal design is created, newer technology swiftly makes it outdated and effectively sub-optimal. It is also not possible to design a MES system that fully meets the needs of all manufacturing types. To deliver the most significant impact, this thesis is constructed to outline the factors relevant to MES design. While virtually all mentioned technology will be superseded in a matter of years, the aim will remain the same: to enable manufacturing to produce product as cheaply, as quickly and as efficiently as possible.

An underlying goal of this research is elucidate the design of a web based MES system. Separate areas of technology exist to support this idea, however, the concept of browser based device communication for MES was found to be lacking feasibility. For a fully effective design, proof of cross browser and cross Operating System (OS) device communication was necessary. No available MES COTS had this architecture previously and without the capability, a legacy client-server architecture was the only possible MES design. In order to add validity to the proposed research approach, Lee et al. (2007) developed a Java based prototype framework that could perform cross browser and cross OS device communication for MES. This framework utilised the low level communication capabilities of the RXTX project for serial and parallel communications (RXTX 2011).

Due to the rapid change of new technological advancements, references to supporting IT literature typically becomes outdated in a few years. On occasion in this research pursuit, older references to supporting material will be used because they are still relevant to the core MES architecture; an architecture that has remained unchanged since first development. Older references have also been included where they are less technology specific and more prescriptive in effective methodologies. As another departure to the typical referencing composition, there is a higher occurrence of web page references. This is due to the commercial nature of MES relevant web technology and how it is mostly publicised through commercial channels as opposed to academic sources. In order to provide factually correct support to this research, only prominent web sites have been used as they are subject to the greatest amount of public scrutiny.

The research methodology is addressed in chapter 2 where the input from COTS MES is discussed. The reasoning behind the particular research direction is explained. In particular, this chapter covers the analysis of two online surveys formulated with the aim of highlighting gaps in contemporary design and areas for potential improvement.

In chapter 3, the literature review has been formulated to address the areas of influence around MES design, with relevancy to web based architecture. It begins by looking at how manufacturing has advanced and also examines the influencing factors on a global scale that drive change. Chapter 3 then moves on to discuss contemporary MES design models in comparison with an alternate web based version. Other contributing areas of importance that are covered are reusable design models, security, usability, the use of touch screens and also the importance of how errors are handled with relevancy to MES. Section 3.13 of chapter 3 displays an analysis of commercially available MES packages. The structure of section 3.13 was derived through the logical grouping of common functionality exhibited by the evaluated software packages. The listed functionality combines to give an overview of what MES should be capable of performing. Some of the more advanced packages give an indication of areas for further advancement in application design.

In order to provide solid support to the migration of MES architecture from desktop to web, chapter 4 highlights technology and protocols that are available for use in web based architecture. When combined together, they support the concept that web based software can perform, if not surpass, the functionality of the desktop architecture.

Chapter 5 explores the influences of design and functionality from official sources such as standards authorities and governing regulatory bodies. There is a vast array of potential authorities, with some being MES specific. This chapter briefly discusses some of the notable and more influential sources.

Chapter 6 presents the application of findings based on the conducted research. Direction of future design is taken from emerging trends in MES functionality highlighted in chapter 3 and available technology covered in chapter 4. This chapter then covers the findings of the surveys and highlights recommendations.

Post research conclusions are covered in chapter 7. In addition to general conclusions, it lists future recommendations and also covers work that would have been performed differently based on hindsight.

2 Methodology

This chapter outlines how the research questions and testing hypotheses formed the basis for this research. In order to attempt worthwhile research into MES design, it was necessary to investigate what capabilities currently exist and also elucidate shifting industry trends. This investigation was performed primarily through the evaluation of existing MES and relevant technologies, the completion of MES surveys, and also the development of a proof of concept for web based device communication. Figure 2-1 represents the significant elements surrounding optimal MES design:

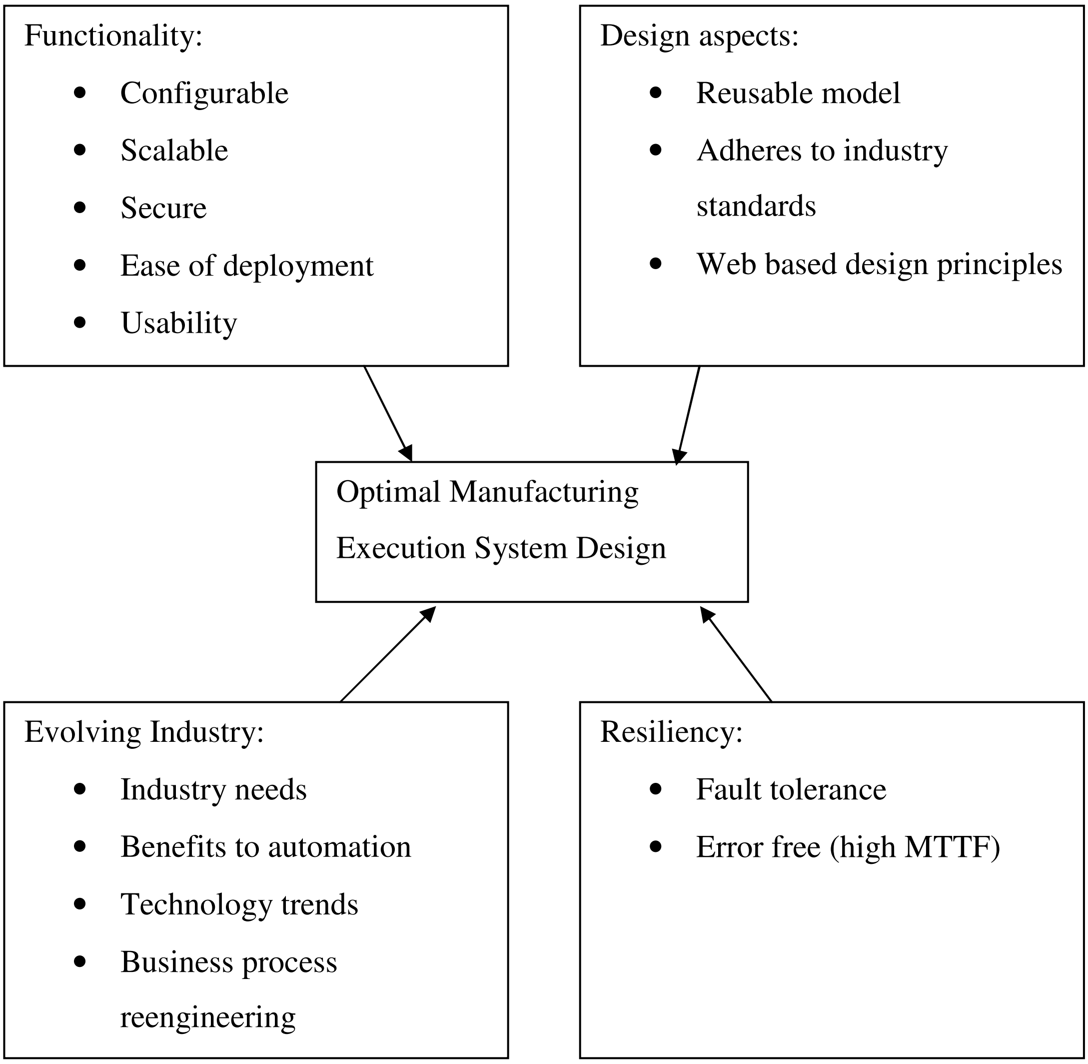

Figure 2-1. Block diagram of research methodology fundamentals

2.1 Hypotheses

The following hypotheses provide the foundation for achieving the research objectives.

2.1.1 Serial devices can be communicated reliably to via a web browser.

Currently, there are no COTS web based MES that can communicate to devices. In order to be utilised in automated manufacturing environments, it is necessary to possess this capability. This capability has already been demonstrated by Lee et al. (2007) in a communication protocol prototype.

2.1.2 Web based MES can perform the same functionality of client-server MES.

The traditional client server MES architecture has remained the same since it was first devised. This persistence was due to the inability of web technology to perform at the level of sophistication that desktop applications possessed. The capabilities of web technology have increased greatly in recent years but MES design has been slow to adopt alternate designs. Due to the critical nature of MES, it can be perceived as unwarranted to risk attempting the utilization of web based architecture. This will be elucidated by the literature review of available technology in conjunction with a technical IT professional survey.

2.1.3 It is possible to define an optimal design of MES software that is reusable.

There are many challenges around this hypothesis due to the varying demands of MES. Industry types differ greatly so attempting to prescribe a design that is universally beneficial would not be feasible. To minimise this challenge, the scope of this research has been limited to American multi-national healthcare companies operating in Ireland. The concept for reduced MES scope can also be seen for COTS MES companies such as Camstar, whereby they have different MES products specialised for their intended industry. This will be elucidated by the literature review of available technology, evaluation of COTS MES, and also with a technical IT professional survey.

2.2 Comparative analysis of COTS MES

A list of the top COTS MES applications was compiled for the purposes of ascertaining what functionality currently exists. More than 22 products were appraised, but continued research became unproductive after 22 products due to no new functionality being uncovered. While not all of the MES packages were applicable to highly automated environments, all relevant factors were noted. Each of these applications were appraised for functionality through a variety of electronically available media such as webcasts, whitepapers, product data sheets, industry partner reviews, phone calls, online demonstrations, corporate email correspondence and available information from company websites. All capabilities of these applications were then compiled into a list of industry design factors for MES. Information compiled was carefully filtered to remove any marketing embellishments or product exaggerations. The collected points are either directly taken from source or paraphrased for conciseness. The compilation of current MES industry capabilities is shown in Appendix B.

There is an inherent complication with COTS software generally in relation to maintaining unused functionality. It is very likely that in a complex system, no one particular implementation utilises the full product feature-set. This can give rise to an unnecessary maintenance overhead. There are also security and safety ramifications of users not fully understanding COTS functionality, giving rise to unanticipated errors. Effective application design will work towards mitigating this. One such example is through the use of fine grained access control so that users are only exposed to the parts of the application that they need.

2.3 Governance of MES design through industry standards and regulation

In addition to industry capabilities, there are other influences that mandate the functionality of MES. All sectors of manufacturing are presided upon by various regulatory bodies governing how products must be manufactured. Any systems used in the manufacturing of products are therefore within the scope of these regulations. Enforcement of business logic and regulatory rules is a core requirement of MES. In any one sector type, there may be multiple regulatory bodies of relevance. This research has discussed at a high level the types of regulations applicable to general manufacturing environments. The scope of this research was kept to American multi-national healthcare companies operating in Ireland, although there can also be regulations from other country specific governing bodies.

2.4 Utilization of web based survey for in-depth MES analysis

As another means of deriving optimal MES design factors, web-based surveying was conducted. Surveying made it possible to gain a better understanding of current MES capabilities, strengths and weaknesses. For practicality, the survey was only extended to personnel in the Bausch + Lomb Corporation; this enabled the approach to be tightly controlled and tailored to the specifics of one MES product. Also because of the researchers employment within Bausch + Lomb, a high response rate could be guaranteed. The concept of surveying was very appealing to the Bausch + Lomb global MES team so a collaborative effort was undertaken. The team aimed to use the survey data to improve the effectiveness of the main Bausch + Lomb MES product called CAMS. Through surveying, the team also hoped to get unambiguous justification to deploy the CAMS product to various Bausch + Lomb sites around the globe. Not all Bausch + Lomb manufacturing sites use CAMS, so the survey was being used to compare the different MES tools already in operation. The collaboration was greatly beneficial because it was supported and enhanced by various members of the Bausch + Lomb technical community, ranging from technical director, software engineers, software quality engineers, technical support and a global MES director. The data collected would not only be beneficial for this research, but also yield real-world business benefits to Bausch + Lomb. Conducting a survey with the support of the business also gave significantly more credibility and increased the size and quality of participant involvement, with cooperation from multiple European and American Bausch + Lomb manufacturing plants.

2.4.1 Reasons for choosing web based survey method of data gathering

A web based survey was chosen for its ease of access to the different geographic sites. Strict data entry validation and control could also be implemented so that data could be collected in a clean and standardised manner. The online method also provides a single point of entry so no collating of results is required, since the data is already populated into a database. A number of different options were assessed for conducting an electronic survey. StatPac was the initial choice because of its powerful statistical capabilities, in-built GUI designer and it also had a free use version. After testing of the product, it became apparent that there were too many complexities to be navigated purely for a one-off survey endeavour. The downside of the high degree of customisation was that there was a steep learning curve even for basic product competency. There are also a lot of complex statistical capabilities inherent that were not needed; this added to the difficulties of trying to create a survey because of the steep learning curve associated with becoming proficient in StatPac's use. Another notable free online survey tool is www.kwiksurveys.com. While it adequately met the functional requirements, it was deemed to be visually basic and not promote a good image for the delivery of the survey. Various alternate hosted web survey providers were appraised and www.surveymonkey.com was found to be the most suitable, containing a selection of reasonably comprehensive reports available for use. It also had an easy to use interface for building and modifying surveys.

2.4.2 General design principles for survey construction

The general survey construction aimed at requiring as little work as necessary for completion by participants through use of pre-populated fields. They were composed into logical groupings with questions of a similar theme on each page. This grouping enabled a clear and logical structure that was easier to follow by the user and incurred less context switching of thought, with the intention of being less cognitively demanding for users, allowing them to be more engaged with the questions.

In cases where possible answers were distinctly separate and not on a sliding grade of similarity, semantic differential choices were employed as opposed to Likert scaling. This enabled participants to choose defined concepts as opposed to gauging nearest matches through numerical representation. This was a redesign of certain questions and combated phenomenon that was discovered in debriefing of pre-release survey testing. In the test phase, participants were found to have selected answers that they did not deem to be fully accurate. As an example, a question such as "Should data historian functionality be integrated into MES applications or should there be a separate data historian solution/application?" was found to have less accurate responses on a one to four Likert scale of importance, as opposed to the greater accuracy when presented with choices listed in Code Figure 2-1:

```
Fully integrated
Integrate some elements
Totally separate
Don't need data historian functionality
```

Code 2-1. Survey choice example

When participants reviewed their answers to the redesigned questions, they found the answers given to the semantic differential choices more accurately represented their opinion. Also it was found that participants were less likely to give indecisive neutral answers. As part of the redesign, some questions from the trial were re-worded in standard text with the removal regional vernacular.

Nearly all questions were qualitative in nature, forcing a closed ended response. This was possible because the potential answers could be predetermined, thus facilitating clear trending and some basic statistical comparisons. For each question, provision was also made for participants to additionally add any relevant comments in an open ended qualitative format. This allowed the collection of some potentially invaluable insights that the participant may have on the subject.

2.4.3 User based survey

In order to elicit the correct responses from the relevant employees, two separate surveys were created. The first survey was user based, where the users of MES systems within Bausch + Lomb were questioned. This survey focused primarily on perceived performance and usability of MES. The sample of users to participate were specifically chosen to give an even spread of the different manufacturing stages and geographic locations, as too many responses representing a particular manufacturing operation may yield skewed and inaccurate results. This group would provide a look at the MES landscape in its entirety and all the various configurations and functionality available within CAMS. Based on business requirements, some sites use specifically customised functionality that would give an inaccurate representation such as the product is very slow purely because a particular site connects to a remote database or else the product cannot perform a certain action because an old version is being used. Care had to be taken with the wording because a significant portion of the survey participants were not completely proficient in English and had an alternate primary language. In fact for some participants, the survey was completed with the aid of a human interpreter. The survey was also designed to ensure the participants that they were not the focus of the survey, as some regions and cultures tend to take a more cynical view on fact finding initiatives and deem it to have negative intent.

The ability to enter additional freeform text for each question made it possible for users to provide feedback that had been otherwise unanticipated. They could then provide direct feedback and communicate any learning's they have made through the close interaction they have on a daily basis with the CAMS product. Departing from standard survey composition, the option of choosing "don't know" was removed from each question. This design is normally required for surveys to prevent misleading and skewed results, when participants click a random answer due to either not knowing the answer or not wishing to provide an answer for some other reason. These issues were not going to be present for the controlled participant sample group and so the "don't know" option was removed where feasible, to force all questions to be answered.

2.4.4 Technical based survey

The second survey was technical in nature, and aimed at the personnel that have influence in the design of CAMS. This survey was primarily focused on MES capabilities and potential functionality enhancements. The mix of questions consisted of current areas of contention within Bausch + Lomb regarding certain design principles, speculated topics of potential enhancement and also emergent industry trends in manufacturing system design. All of the significant elements of an MES system were covered. Most of the questions in the technical survey were constructed so that participants were forced to choose one answer over the other and assign different weighting. This ordinal ranking aimed to provide clarity and direction as business pressures often result in conflicting mandatory requirements. Forcing participants to compromise on conflicting requirements was performed by mutually exclusive single choice options, in addition to mutually exclusive list ranking.

Unlike the user survey, the technical survey used “don’t know” answers for some questions that were possibly outside the knowledge of the participant sample. The level of technical expertise varied between geographic sites, so use of the “don’t know” option was intended to highlight possible training needs for areas that should be understood more. Also because of the differences in expert job functions, some questions were outside the scope of their occupation. For questions of this nature, an option of selecting “Question not relevant to my job function” was provided. This option intended to prevent participants from providing random answers purely to proceed to the next question.

2.5 Conclusion

Chapter five has outlined the methods through which this research endeavours to meet the research objectives and deliver academic benefit. The research input was conducted through a number of means such as MES analysis, industry standards and regulations review, web based device communication prototyping, technology review and MES related surveys. It is envisaged that the input will highlight the current MES landscape and assist in guiding how MES design can be improved. The varying sources of input are believed to give sufficient elucidation of all pertinent aspects of MES design. While the surveys were limited in scope for feasibility purposes, the findings of this research will be of benefit to all MES types, including those outside the survey scope.

3 Literature Review

The literature review direction is guided by the review variables listed in the methodology in chapter 2. These review variables were formulated to address the research questions highlighted in research questions section 1.2. They address the relevant aspects of the research topic and attempt to provide the foundation for the research subject matter. They were formulated by looking at the anticipated research outcomes for MES design around the following concepts: Optimal software design, MES design, Automated manufacturing, Web software functionality, Client-server MES, Reusable MES design model.

3.1 Introduction

The literature review starts by looking at the path that MES has evolved from its first implementation to contemporary designs. The review then progresses to the factors of influence on MES capability. The fundamental goal of advancing MES capability is to enable businesses to remain competitive. Implementing a highly automated system may not be financially viable for some sectors and could cost the company more in the long term (Brown and Hellerstein 2005). One such example would be that of heavy industrial manufacturing where their relatively simple business processes do not require MES automation integration. Those industries are the exception however, as ever increasing business pressures are forcing companies to operate with less cost and produce at higher capacities and lower defects.

Through investigating the traditional client-server MES architecture, the review then looks at the relevant problems, and also the potential benefits in moving to a web based architecture. As noted in the thesis outline section 1.5, device based communication through a browser is a new and unverified area for MES applications. The literature review highlights potential in this area and the conducted proof of concept work.

MES needs to be adaptable in order to meet changing business requirements (Kletti 2007: 27). This flexibility should have the ability to be rapidly implemented through reconfiguration as opposed to development which would need to progress through the comparatively lengthy Software Development Life Cycle (SDLC). Hence, this flexibility needs to be built in from the outset. Reusable design models are covered in the review in order to prescribe the foundation for a flexible MES application. End user interaction with automated systems can develop a number of issues if HCI aspects are not managed correctly (Sarter 1997: 3-14). Because MES is utilised on shop floors and has a direct impact on the manufacture of product, it is essential that it is as easy to use as possible. The literature review covers this specialised area of Graphical User Interface (GUI) design for touch screens and their most appropriate use. Usability design for MES is of paramount importance. Even slight alterations to a design can have powerful consequences for productivity and even safety. The review covers these aspects of usability and design relevant to MES.

Another crucial design factor for MES is that of security. A faulty application could cause serious financial loss to a company. Either through system error or malicious intent, systems could potentially become inoperable with lost or corrupted data. This chapter also covers techniques for mitigation so that MES is designed with safeguards that prevent these unforeseen error states that have the potential to produce very undesirable results.

3.2 The history and advances of manufacturing

Automation is a very topical area of interest as it is the key to manufacturing industry's competitiveness. Business pressures are forcing organisations to continually reduce costs, improve quality, increase efficiencies and react quicker to customer demands. MES plays a central role in all manufacturing automation (MESA 1994: 1-2), Computer Science and Telecommunications Board 2005: 1-2). The software typically spans the entire manufacturing process and is integrated into a variety of manufacturing stages (Spera 2005: 1). Precursor MES type software was first introduced in the 1960's and has undergone many technological advances including the move from mainframe applications to client-server based applications (Teunis 1998: 1). The traditional client-server based architecture has remained unchanged since the 1980's (Teunis 1988: 1, Chiu 2001: 1, Shaw 2006: 110).

MES software is generally tailor made by organizations to suit their specific needs and requires significant integration and maintenance. Demand is growing however, for COTS MES packages in order to avail of assured software quality, functionality and associated benefits (Spera 2005: 1). Therefore, research into a reusable MES design would be of particular interest to industry. A web based MES also has the potential to enter new business markets such as Application Service Provider (ASP) manufacturing software. There are a number of terms for this business market, with Cloud Computing being the prominent contemporary label. With the cloud model there is also the potential for virtually unlimited scaling to suit changing organisational needs.

There are some parallels between the world of business and that of anthropology, with the adage of “survival of the fittest” being quite pertinent. Continual technological innovation is being applied in manufacturing to deliver products and services with higher quality, lower cost and in a shorter time to market. At the very least, businesses must deliver products and services at equal price and quality as their competitors. However, even that may not be enough for some companies that seem to have the mantra of “grow or die”, which appears to have been the case for the software giant Oracle as it went on an acquisition spree in 2010, having bought out numerous other companies (Oracle, 2010).

Higher degrees of automation result in much lower production costs. In Ireland, as with all developed countries, labour costs are high (Forfás 2010: 34). By comparison, less developed countries such as those in Eastern Europe, can maintain significantly lower unit costs because of cheaper labour. By reducing the amount of people required in manufacturing, labour costs can be minimised and it becomes possible to be competitive without relocation to low cost locations. Increased automation is a key enabler for the competitiveness of manufacturing in developed economies.

When manufacturing uptime and throughput are maximised it is possible to meet production targets with less equipment and hardware. By keeping expensive machinery running at higher uptime levels, otherwise known as Overall Equipment Effectiveness (OEE), it may be possible to delay, or defer indefinitely, the purchase additional machinery. Automation and greater efficiencies are two of the derived benefits of MES usage.

The alleged Henry Ford business mantra of "you can have any colour as long as it's black" is no longer appropriate. Businesses must become much closer aligned to customer needs to fend off substitution from competitor offerings (Kletti 2010: 1). Supporting functions such as IT must also become more closely aligned to the business to enable better customer responsiveness. This means that IT needs to be highly adaptive to changing demands. MES has a very large part to play in reducing labour costs, maximising machine utilization and also enabling businesses to react quickly to global market pressures (Gifford et al. 2007: 22 - 23*)*.

3.3 Macro-economic aspects of automation

A brief discourse of the global MES influencers is required in order to develop a solid understanding of MES design requirements. While the technology landscape continually evolves, global market drivers provide a consistent guide of MES capabilities and enhancements.

3.3.1 Changing economy

Even before the economic downturn that was experienced worldwide in the recession of 2009, the manufacturing landscape has been facing numerous challenges. Many small labour intensive businesses folded when faced with the insurmountable competition of low cost developing countries. Where possible, companies relocated their manufacturing plants to these lower cost sites such as those in Eastern Europe and further afield to China. A typical relocation of manufacturing operations to these low cost sites can yield cost savings of between 20 and 60 per cent (Vestring et al. 2005: 1). As the cost of living increases, so does the cost of conducting business. Even Eastern Europe is not immune to this and is already attempting to mitigate the growing threat from even cheaper countries such as China and India. In order for manufacturing to survive in developed countries, aggressive automation is required to minimise costs and maximise output.

3.3.2 Rapid adaption to changing environment

The time to market for New Product Introduction (NPI) is continually being reduced. An understanding of Speed To Market (STM) is being applied to deliver products quicker than competitors, at a cheaper cost and higher quality (Smith 1999). Leverage from product innovation is eventually lost as competitors catch up so there is a need for continual advancement. A MES design that facilitates rapid configuration and deployment gives a distinct advantage to businesses and has potential to build significant market share. In addition to speed of deployment, MES must also enforce high levels of quality and enable seamless high volume production. This rapid adaption is made possible by having the ability to meet business logic requirements by configuration of existing product functionality. Systems that need to enter the SDLC to meet business requirements greatly hamper rapid adaptation endeavours.

3.3.3 Economies of scale

Cottage industry type businesses are no longer sustainable when providing easily substitutable goods and services. Early advances arose through use of machinery and evolved to mass production processes (Shaw 2001: 4). Only through large-scale production and increased operational efficiencies can economies of scale be taken advantage of. Larger companies can reduce their running costs by utilizing shared services and efficiently supporting different divisions of business. This in turn enables a lower unit cost and the ability to deliver lower priced products to customers (The Economist 2008). To realise these benefits, MES needs to be highly scalable and facilitate production to grow with the minimum of cost.

3.3.4 High availability of service

With the cost of performing business being relatively high in developed economies, it is imperative that maximum productivity is achieved. There is the potential for a prolonged service outage to force a company out of business. Even a momentary outage of MES service could cause substantial loss of manufactured product (Managing Automation 2011). In the event of problems caused by faulty MES, the supply chain could be disrupted, resulting in a shortage of manufactured product available to customers. This unavailability of product could influence customers to use alternatives. Very few products have the luxury of being irreplaceable or have the backing of an extremely powerful brand, as the majority of businesses cannot guarantee their customers will return once supply of their product has resumed.

3.4 Contemporary MES design principals

While the traditional MES client server architecture performs adequately, significant benefits can be obtained from redesigning to a multi-tier web based model (Chiu 2001: 1, Shaw 2001: 20), as illustrated in Figure 3-1. New technologies available for web based systems have been evolving at a rapid pace (Hayes and Cai 2001: 1). These technological advances may facilitate the redesign of traditional MES architecture. There are also other traditional architecture aspects that have not grown with the advances in automation software, such as security (Shaw 2006: XIII) and usability (Kasvi 2000: 45).

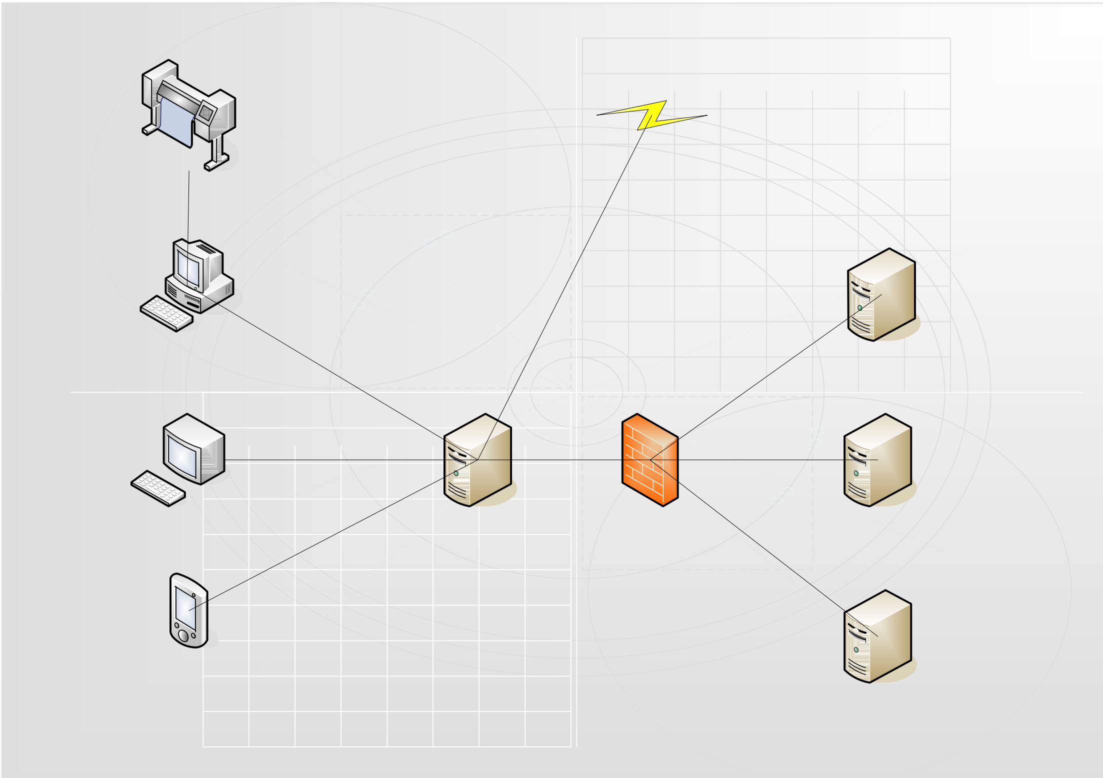

Figure 3-1. Proposed Multi-tier Web MES Model

MES software is generally tailor made by organisations to suit their specific needs and requires significant integration and maintenance. There is a growing demand however for COTS MES packages in order to avail of assured software quality, functionality and associated benefits (Spera 2005: 1).

The utilization of automated systems has helped improve quality and reduce costs in a variety of ways (Sarter 1997: 1). MES software was first introduced in the 1960's to help drive automation and has undergone many technological advances (Griffiths

2002: 5). One of the major evolutionary changes includes the move from a mainframe application to a client-server based architecture (Teunis 1998: 1). However, the traditional client-server based architecture has remained unchanged since the 1980's (Chiu 2001: 1, Shaw 2006: 110). Resultantly, current MES design does not make full use of the latest available technologies.

While the traditional MES architecture meets industry needs, significant benefits can be obtained from redesigning to a multi-tier web based model (Chiu 2001: 1, Shaw 2001: 20). The client-server architecture is no longer the most cost effective or flexible solution for manufacturing systems (Lee et al. 2007: 3). In addition to hardware costs for high powered workstations to drive manufacturing, there are also many associated administrative overheads in maintaining desktop applications. Currently implemented technology is inadequate to support the manufacturing requirements of the 21st century (Computer Science and Telecommunications Board 2005: 5). There are inherent limitations on scalability, flexibility, performance and management for client-server MES models (Chiu 2001: 1-2). New technologies available for web based systems have been evolving at a rapid pace. (Hayes and Cai 2001: 2). These technological advances may enable the redesign of traditional SFIS architecture and cater for the real-time demands of MES (Chiu 2001: 1, Sullivan 1997: 269). Bowden (1994: 121) states that flexibility, inter-operability and portability are key aspects of information technology utilised in manufacturing. Web based design inherently meets these key design aspects (Chiu 2001: 1-2). A number of additional benefits of a web-based MES are envisaged, including rapid deployment of systems to manufacturing, high versatility in implementation with operating system independence, and serving remote manufacturing sites remotely. There is also evidence to suggest that web based designs are easier to maintain after system creation than traditional rich-client counterpart designs (Hayes 2001)

3.5 Emergent MES trends

Early MES design was a stand-alone closed system (Teunis 1998), implementing process control on various manufacturing activities. That design has been superseded with the need for increasingly more integration to related systems (Scholten 2009: 2). Already the boundary of MES is becoming somewhat blurred as it becomes intertwined with a multitude of manufacturing, business and automation systems (Scholten 2009: 17). Even the concept is altering with shifts in industry referring to MES as manufacturing intelligence, evolving from basic data collection and reporting to a real-time tool driving manufacturing optimisation. As the product matures, elements of other systems are being incorporated into MES such as data historian functionality, scheduling, Corrective And Preventative Action (CAPA) etc. An overview of MES and its related systems can be seen from Figure 3-2:

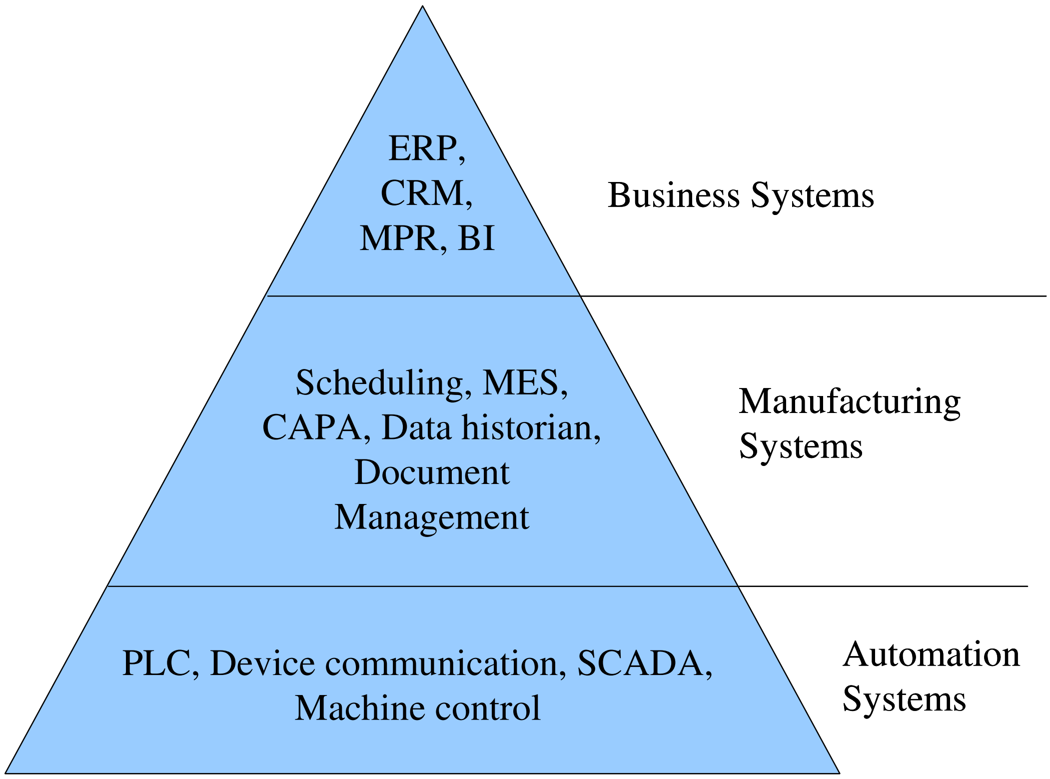

Figure 3-2. MES and related systems

A number of web based MES solutions have been released to market in the last couple of years, taking advantage of web technological advancements. However, these solutions do not have device communication capabilities. While it may not be suitable for all MES implementations, web based architecture can provide numerous advantages for its client-server counterpart.

3.6 Problems associated with desktop software

Installation and maintenance issues are inevitable with software deployment. Even a failure rate of one percent can cause considerable financial loss in a large-scale manufacturing environment. While the cost of high-performance computing continues to drop, the cost of failure continues to rise (Swift 2005: 1). For the duration of OS problems, machines and operators are non-productive, the time of technical personnel is wasted and work orders are not being filled. It is not uncommon for PCs to need a total software reinstallation due to irresolvable software issues. With OS software there are various dependant subsystems that can impact the installation and operation of a desktop application. An example of one such subsystem is the Microsoft Data Access Component (MDAC) kit associated with Windows OS. This subsystem caters for database connectivity to nearly all database types. Microsoft created a component checker application to help resolve MDAC issues because they occur so frequently due to software installation problems. Short (2003) as cited by Swift (2005: 1), claims that on Windows XP, 85% of crashes are caused by an OS subsystem called 'device drivers'. The phrase 'DLL hell' is used to reflect the operational issues that can arise with the installation of multi-component applications. Often a subsequent program installation puts an incompatible version of a DLL, device driver or ActiveX file on their system. Problems are also caused by situations like the windows path environment being changed or a file not properly documented in the system registry. Although Microsoft .NET technology alleviates DLL type installation issues but is still reliant on OS subsystems and the Windows OS platform.

The frequency of these issues is accelerated by the constant need for patching. Microsoft has a monthly 'Patch Tuesday' with releases fixes for security flaws and software bugs. This 'patch and vulnerability cycle' is also on-going for other OSs such as Linux. Patch rollout is so problematic for Linux and Solaris that a company called Aduva developed a product called OnStage that attempts to prevent these issues (Koch 2005). With each patch installation there is a possibility that manufacturing systems are disrupted (Erbschloe, 2005). An even more undesirable outcome is when the running of desktop software does not fail outright. Often issues only arise when attempting to use a particular aspect of the software. Indeterminate and undesired application results can occur from subsystem component issues.

Company employees can bring undesirable software inside company Local Area Networks (LANs) through a variety of methods. Securing organisational infrastructure is becoming harder with the prevalence of the internet, email, instant messaging, file sharing applications and portable storage devices such as USB keys.

Viruses, worms, trojans, key loggers, stealware, spy ware and other malicious software can all negatively impact manufacturing systems and often operate by manipulating the OS software through software alteration (Erbschloe, 2005). Any uncontrolled alteration to the OS can have negative ramifications for the operation of manufacturing systems. Innocuous software such as games, screen savers and even functional utilities can all harbour malicious code with the potential to cause desktop software to function incorrectly. Even the installation of a genuine application can overwrite a necessary operating system component that manufacturing software requires for error free operation. Other forms of tampering can also occur as the surface area of attack for manufacturing software is wide open with open access to the client PC operating system. Seemingly harmless user actions such as changing the system clock or regional settings could have disastrous consequences such as altering the correct expiry of product processed through the particular PC station.

An additional vulnerability for desktop software is the capability for client software to be decompiled, revealing potential weaknesses to potential malicious parties. Figure 3-3 shows the decompiled code of a binary obtained by a freely available .NET decompiler tool called Reflector:

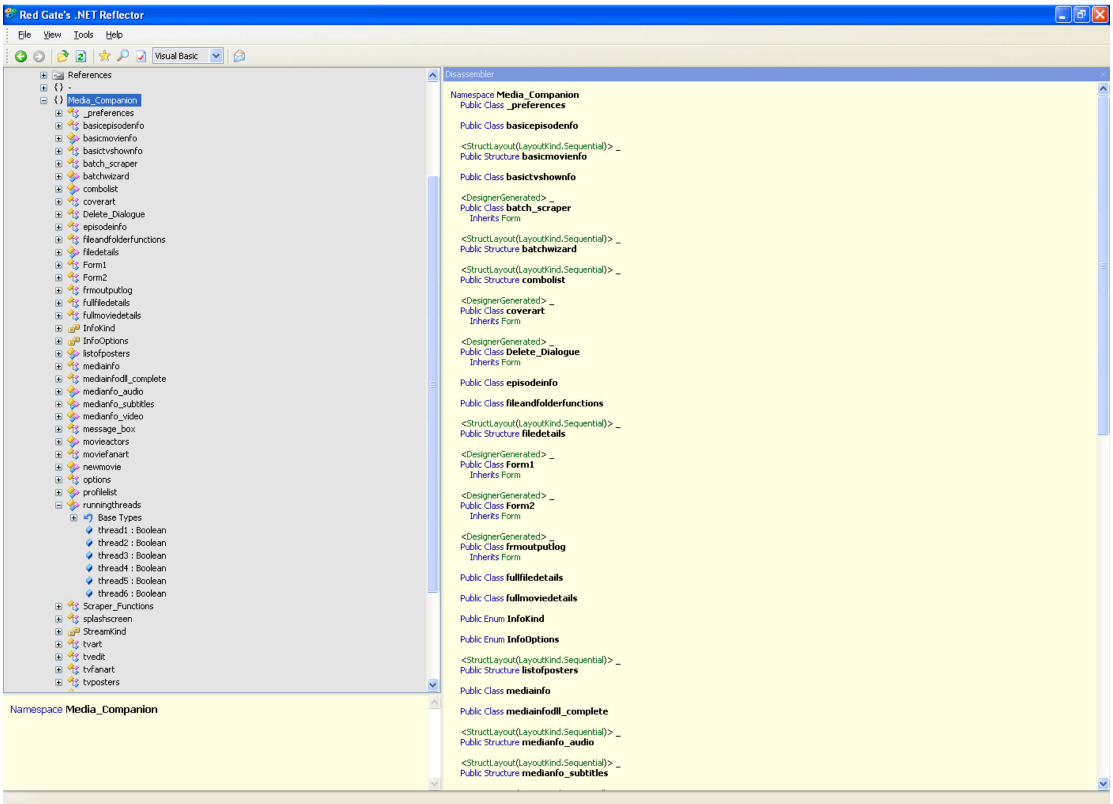

Figure 3-3. Decompiled .NET binary code

As manufacturing systems become more sophisticated, there are growing demands for processing power on client PCs. Hardware costs rise correspondingly in support of manufacturing system software needs. These costs are amplified through business continuity measures, such as having backup hardware prepared for when client PCs eventually fail. These backup PCs typically have the required MES installation and configuration to enable it to take the place of the faulty hardware.

With the client server architecture in automated manufacturing systems, clients PCs remain idle for a large spans of time, often only fully utilised at certain parts of the manufacturing cycle. The cost of processing demands is increased as more expensive hardware is required to mitigate outages for highly automated manufacturing processes. Avoiding downtime and the cost of actual downtime make up more than 40% of the total cost of ownership for modern IT systems (Schroeder and Gibson 2007: 1). This extra expenditure is not unwarranted however. For example, eight percent of hard disk drives fail in the first two years alone, according to a study conducted by Google (Pinheiro 2007: 4).

Another drawback of desktop software is that it can only be deployed to the OS that it was developed for. It is not even possible for the highly portable Java software to be deployed on different OS's without changes and customization. Multi-platform support is just not an option for other types of software created with programming languages such as Visual Basic (VB), Delphi, C++, etc. VB applications for example are limited to the Windows OS and so cannot take advantage of the free Linux OS. In large MES implementations, OS licensing can be a significant cost.

3.7 Benefits of Web Based Manufacturing Systems

An internet browser is the only software pre-requisite for running web applications. Standards compliant web applications are cross-browser compatible and so can be run by any platform or browser type. This low footprint removes the variability of deployment issues experienced with system rollouts. The manufacturing software installation is as simple as loading the correct Uniform Resource Locator (URL). This enables a seamless and rapid deployment without technical personnel support. In this manner, Honeywell reduced its software deployment costs by 98% when it migrated to a web based architecture for a manufacturing system (IBM 2000).

Manufacturing is becoming more automated in an attempt to reduce costs, increase quality and yield higher production output (Walker 1996: 2). Machines are replacing people, with control being handed over to computer systems. Because the bulk of the processing is performed on the server for web-based applications, client PC demands are low. As a result, client PC hardware specifications can be reduced to a minimum, giving financial savings on the cost of hardware. Cheap and low specification hardware enables cost effective growth in automated manufacturing. The portability of web based applications enables even more cost savings with the potential to migrate to a free OS such as Linux. The number of backup PCs required for maintaining production is also significantly reduced. There is no need to have cloned backup PCs for each manufacturing node because machine dependant configuration is taken directly from the web server. This lowers the amount of backup hardware required to maintain a manufacturing system. It also becomes harder to tamper with manufacturing systems as all processing occurs on the server side. The database and any additional tiers can be hidden behind a firewall, protecting otherwise vulnerable elements. Web applications have the effect of segregating manufacturing systems logic from client operating systems. While the web brings a new set of vulnerabilities such as cross-site scripting, Structured Query Language (SQL) injection etc, development platforms such as ASP.NET have inherent safeguards against such attacks and mitigate these vulnerabilities (Esposito 2005). Some technology vendors employ additional safeguards to mitigate this. Oracle for example, have a database firewall tool to protect against SQL injection.

Scalability has always been a critical factor for manufacturing systems. In addition to expanding client bases, increased processing demands arise from changes arising from customer requirements, markets, governmental and other regulatory bodies. These factors all place increased demands on existing systems; through a web based architecture, plant-wide PC infrastructure is no longer required to accommodate extra processing requirements. Use of web farms in web based applications allows relatively easy scalability by adding an additional node in the web server pool. This load balancing is a lot more cost effective in comparison to the desktop model whereby all PCs on the shop floor need to be upgraded or replaced.

There is a wide range of COTS desktop MES software currently available as listed in Appendix B, and possess varying strengths and functionality. The packages with device communication capabilities are limited to a desktop architecture, and must be setup for each client implementation. Databases, application servers, client-side desktop setup, security, service redundancy, configuration, backup & recovery and all other application dependencies also need to be configured and maintained by the client. In contrast, web architecture has the potential for COTS MES to take advantage of new business models. Apart from client PCs and an internet connection, there are no requirements for client side infrastructure. A MES software manufacturer can become service orientated and lease out hosted systems to its clients through a Cloud model. The knowledge burden is kept to a minimum so the client only needs to know how to operate the application, not how to maintain and troubleshoot the system and all of its dependencies. Clients can then fully outsource their MES application and make significant financial savings. The MES service provider benefits from economies of scale and can develop a 'best in class' application at a lower cost. It would then potentially be economically viable for small to medium enterprises to take advantage of MES that would otherwise be too costly. Manufacturing software has developed into over a three billion dollar market each year (Walker 1996), so there is a large market for COTS MES. An example of such a business model in action is www.Salesforce.com where customer relationship management services are leased on-demand through a cloud model.

3.8 Web based device communication

A major function of MES automation is to connect to and control devices used in manufacturing. These devices can range from simple label printers to highly advanced robotics and control systems. Interfacing to these devices can be a complex task that requires resilient design to ensure consistent, scalable, reliable and error free device communications. The ramifications for indeterminate device communication could result in injury or even fatality in a heavy industry type-manufacturing environment. With the stakes being so high, a client-server model has traditionally been a safer means to execute the required business logic. This over reliance on client-server architecture is no longer warranted however with the onset of Web 2.0 and the semantic web. Recent growth brings a whole armoury of technologies such as AJAX, Extensible Markup Language (XML), SOAP, JavaScript, Cascading Style Sheets (CSS), Web Services, SVG, XHTML, SPARQL, OWL, SIP, RSS, RDF, DAML etc. (Wikipedia 2009). The statelessness of Hyper Text Transfer Protocol (HTTP) can now be easily overcome through a combination of additional protocols and techniques to deliver desktop like features (Meschkat 2007: 1-2). It has become possible for the automation, integration and reuse of data across various web applications, further overcoming HTTP limitations (Passin 2004: 2). Traditional drawbacks no longer exist as web based software can now not only perform the functionality of desktop software, but also add rich GUI features (Van Der List, 2007: 167). Technological advances can enable the redesign of traditional MES architecture and cater for the real-time demands of MES (Sullivan 1997: 5).

As proof of the ability of web based system to communicate on a cross browser and cross platform basis, a prototype was successfully developed and tested by communicating to printing devices. These devices were programmed with the RS232 communication protocol via a web browser (Lee et al. 2007). Figure 3-4 illustrates a Unified Modelling Language (UML) Activity diagram overview of the RS232 communication used:

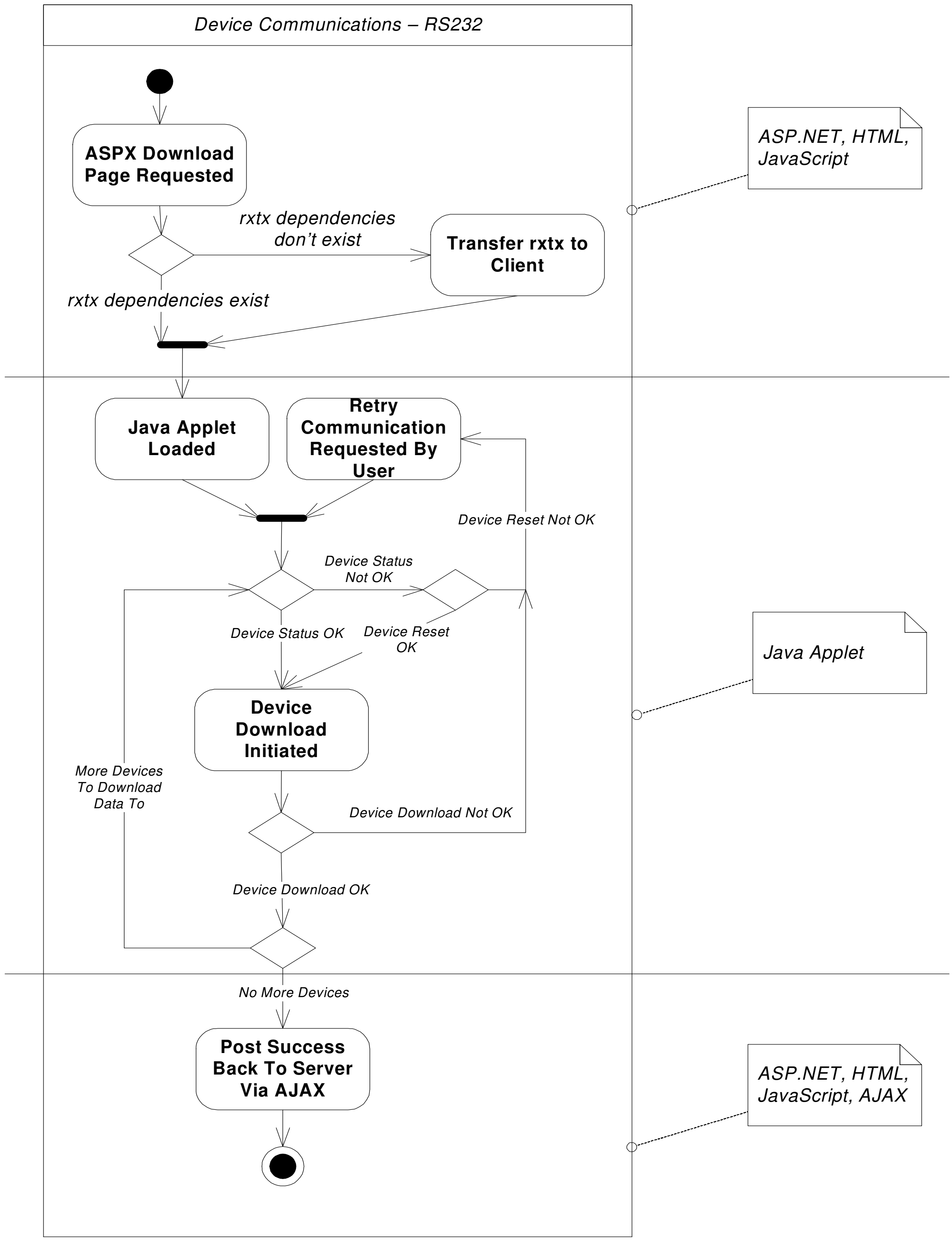

Figure 3-4. RS232 device communication UML diagram (Lee et al. 2007)

As illustrated in the preceding figure, a comprehensive failure detection model has been incorporated so that multiple devices can be communicated to in a serial transaction-like manner. If communication with any one of the devices fails, all the devices will be reset, ensuring no dangerous error states occur. One such example would be where a machine saw has been successfully turned on with the first device download but a subsequent device communication to lock protective guarding fails. Automation failures if not mitigated could give rise to a person being seriously hurt or killed (Sarter 1997: 12-14). In this scenario however, the device communication prototype is configured to disarm the saw device to prevent risk and flag the error appropriately.

The prototype consisted of a Java based framework that enables any device type to be communicated to a bi-directional manner via customisable device drivers. As new device types and communication protocols arise, they can be plugged into the framework so it is fully future-proofed without limitation. The following Figure 3-5 illustrates a UML Activity Overview diagram of the prototype data flow:

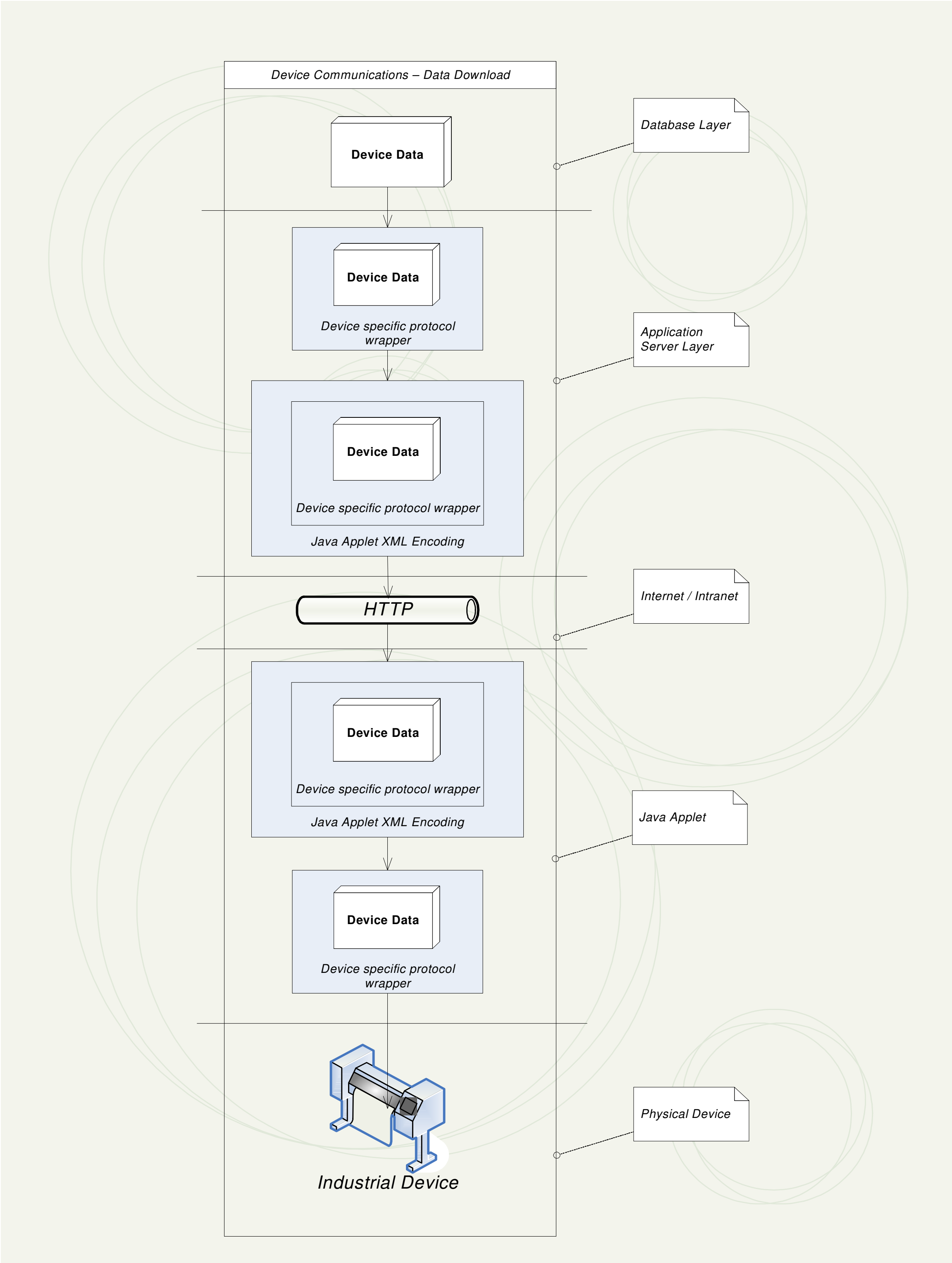

Figure 3-5. Device communication UML data flow model (Lee et al. 2007)

3.9 Reusable models for optimal software design

The use of templates is a well-known idiom, employed for a vast array of physical and computerised disciplines. Template use is highly prevalent because it enables the reuse of work with pre-proven efficacy, and fast tracks the completion of tasks. In the realm of software engineering, templates are in the form of design patterns. These patterns prescribe the underlying framework to various software development goals and enable reuse of seasoned software developer experiences (Riehle, Züllighoven 1996: 1). In practice, the templates will not achieve everything that is required, but will provide an excellent foundation that can be modified as needed. Where some models are mutually exclusive, others can be merged into hybrids so that they more closely meet design objectives. The use of design patterns is essential for all large software development projects. All of these models are abstract from programming language and are documented via notation languages such as the widely used Unified Modelling Language (UML), Z Notation and others. The models can then be implemented into the required language specific code. It is not practical to fully describe each of the available design models, so the following sections have a brief comment on some of the most relevant patterns to MES design.

3.9.1 Architectural patterns

Architectural patterns govern the underlying framework of a program with which all code is developed upon. It is important to decide what architecture is most suitable before development as attempting to alter this model type retrospectively is an arduous task (Wikipedia 2008). This pattern type provides the foundation to employ other hybrid patters to create fully functioning applications.

3.9.1.1 Model View Controller

The Model View Controller (MVC) architectural pattern specifies the separation of business logic and GUI design. By keeping business logic encapsulated, it can be used in multiple points in GUIs but only needs modification from a central location. This practice allows developers with specific skill sets to simultaneously work on a project and also means that developers with narrower skillets can still contribute. There are numerous language specific frameworks that enable software projects to inherit the design concept. For example, .NET has options to plug-in ready-made MVC code. This model is particularly relevant to the design of web based MES.

3.9.1.2 Peer to peer

Peer to Peer (P2P) is essentially direct communication between clients, cutting out the round trip required in transmitting via a server. These peers are all of equal importance and do not rely on a server to relay messages. In the event of one peer becoming unavailable, the P2P network can still remain active. Synonymous with file sharing, P2P technology has potential benefits to particular manufacturing process designs. While it is not suitable for the core MES architecture where information needs to be shared between all clients, P2P has expedient messaging and communications; something which may grow more popular as manufacturing stages become more autonomous.

3.9.1.3 Multi-tiered architecture

This is the most applicable architectural pattern for MES design. Even with client-server based MES, multi-tier structure is often employed to separate business logic and maintain a scalable and efficient design. This would typically entail separate tiers for business logic and database. With the onset of web based MES, the web server element creates another tier, allowing further options for scalability and function specific tuning. It should be noted however that separating tiers onto separate servers/hardware may aid scalability but it also has a downside of creating an additional dependencies and points of failure.

3.9.2 Creational patterns

Creational type patterns specify the creation mechanism for objects. These patterns detail how specific parts of code are to be initialised, their visibility to other objects and also how they interact (Wikipedia 2008).

3.9.2.1 Abstract factory

This creational pattern provides a way to encapsulate code units that have a similar function or theme. Utilizing polymorphism, a generic code unit (otherwise known as class), is created to be generic enough so that is can be as widely used as possible, but also detailed enough so that it captures as much business logic as required (Shalloway and Trott 2001: 163). Use of this pattern ensures code does not have to be duplicated each time to perform the same action, and so makes development times quicker and makes modifications/maintenance much cleaner. Sub-classes can then be used to inherit from the abstract factory class, so that they have the same interfaces and base methods. As an example, an abstract class of a car could be derived from by sub-classes of car models such as Seat Ibiza, Nissan Almera etc. With relevancy to MES development, this model is essential in order to handle the production of varying product and machine types in manufacturing.

3.9.2.2 Singleton

When this creational pattern is implemented, only a single copy can be created at any one time. This is necessary for multithreaded concurrent operations that utilise shared resources. In this scenario, a singleton object can be created to perform coordination tasks. There are also other nuances to this type such as providing a global mechanism to detect object existence and also mechanisms to enable create on demand that enable more efficient computer resource usage (Shalloway and Trott 2001: 256).

3.9.2.3 Multiton

Being another creational pattern, the multiton is similar in design to the singleton. The multiton is used for managing multiple instances of objects and provides a global mechanism for obtaining handles to the instantiated objects. In reality, modern development languages negate the necessity for explicitly defining this pattern because pre-defined objects that contain that functionality such as dictionary and hash are readily available for use.

3.9.2.4 Object pool

An object pool is a set of initialised objects that are maintained in a ready state for use. When a client requests an object from the pool, it can be quickly processed because the object has already been instantiated. When the client has finished with an object, it is returned to the pool as opposed to being terminated. This pattern specifies a method for achieving high performance and scalability through partial pre-processing of commands so that user driven requests can be serviced much quicker than would otherwise be. A typical example of this pattern type is in database connection pooling for web applications where state is not maintained by HTTP so each page load is a new request.

3.9.3 Structural patterns

Structural patterns help simplify the collection of attributes required to represent business logic (Wikipedia 2008). They often combine multiple objects into larger structures for reuse in some form of Application Programming Interface (API). These design patterns are very important due to the fact MES must interface with different systems in order to meet the needs of an organisation in its entirety. There are a number of variants of this pattern, all of which essentially modify how objects can be interfaced.

3.9.3.1 Facade

This pattern provides a simplified interface when interacting with objects. By encapsulating object interfaces, it becomes possible to fully control what functionality can be provided with an API and other such integration methods. This is useful for controlling what options are exposed to other systems and reduces the scope for intentional misuse or due to errors (Shalloway and Trott 2001: 87). For security considerations, as little as possible should be exposed for external system control. There is a balance however, due to the numerous adjoining systems of MES, extensive interoperability is necessary.

3.9.3.2 Adapter

Adapter patterns are used to convert interfaces so that they match the expectations of a connecting system. This is required when disparate systems need some form of communication but they have dissimilar interface methods. By creating an adapter, the underlying objects remain unchanged while provision is given for the other system to interconnect through a custom wrapper (Shalloway and Trott 2001: 95). This type of pattern is suitable for enhancing a pre-existent MES because to connect with an adjoining system through a less intrusive alteration to existing system design. Otherwise the alternative would be to modify existing interfaces with the potential to break existing compatibility.

3.9.4 Behavioural patterns

Behavioural patterns identify common communication patterns between objects and define optimal communication structures. By doing so, these patterns increase flexibility and scalability in carrying out communications (Wikipedia 2008).

3.9.4.1 Iterator

The iterator pattern provides a way to access the elements of an aggregate object sequentially without exposing its underlying representation. In essence, it abstracts the complexities of dissimilar objects so that they can be handled in consistent manner. This pattern is more of a programming good practice as opposed to critical architecture design but does ensure a greater level of scalability. All the modern development languages have this type of functionality inherent in its design and available for use.

3.9.5 Concurrency patterns

These pattern types govern how software should be designed when dealing with concurrent operations such as multi-threading. Multi-threading is crucial for rapid response to user requests and makes use of multiple CPU's and cores that exist in contemporary server architecture. Reuse of these patterns is crucial due to the intricacies of marshalling multiple threads. These pre-tested solutions will significantly reduce the potential for synchronizing and deadlocking errors that are typical for multi-threading and use of shared resources.

3.9.5.1 Thread pool

The thread pool pattern is similar to the creational object pool pattern type. A number of threads are pre-created and available for processing. These threads sleep until they are requested for use. This prevents the overhead of continually creating and destroying handles. Implementation of this pattern ensures better user response times and increased application performance. For web-based applications, the underlying web hosting software performs this thread pooling inherently. Thread pooling can also be built into the application architecture for additional benefits.

3.9.5.2 Scheduler

When interacting with single threaded shared resources, the scheduler pattern can be used to ensure only one thread/object has access at any one time. It marshals access sequentially, granting exclusive access to a list of requestors. For example, a lot of MES interfacing with legacy systems takes place via text file transfer. These files need to be created with an exclusive lock while all of the data is being dumped to a network location. Through utilization of the scheduler patter, operations such as these can be performed with minimal errors and consistent performance.

3.10 MES Usability

With advances in industrial automation, machines and systems are progressively performing additional work as humans take a more supervisory role (Sarter 1997: 2). These advances mean humans need to increase their skill base in accordance with increasing system complexity. Ironically part of the motivation of using automated systems is to reduce operator workload and training requirements (Sarter 1997: 1). Therefore systems need to be as usable and intuitive as possible to prevent the over-burdening of users (Kasvi 2000: 40). MES software often lacks inherent usability design principles (Kasvi 2000: 45). The technology-centred approach to automated system design often causes breakdown in communication between humans and machines. A typical failure of MES is the lack of adequate feedback to the user (Sarter 1997: 19). This communication breakdown results in financial loss, injury and general system failure (Sarter 1997: 12). To yield the full benefits of automated manufacturing, its supporting software must become user-centred with more intuitive HCI design. The optimal balance of information must be obtained by providing adequate feedback without inducing cognitive overload.

3.10.1 Cognitive Modelling

Through the evolutionary process and natural learning practices, people perform tasks, solve problems and interact with their environment generally through distinguishable patterns. Many of these patterns have been scientifically modelled through psychological principles. In relation to computer systems, these models enable us to predict the time it takes for people to perform tasks, the kinds of errors they make, the decisions they make and even what buttons and menu items they choose. These models can be used to determine ways of improving the user interface so that a person's task has fewer errors or takes less time. They can also be implemented in systems so that they are more effective tools that adapt to how people naturally interact with systems. Highly advanced systems can even extend their usefulness by anticipating user abilities and adapting accordingly.

3.10.1.1 Hick's Law

Hick's law, also known as the Hick-Hyman law, is a cognitive model that defines the time it takes for a user to make a decision, based on the number of the possible choices available, such as screen menus, buttons and other input controls (Lidwel et al. 2010: 120). Figure 3-6 illustrates the law:

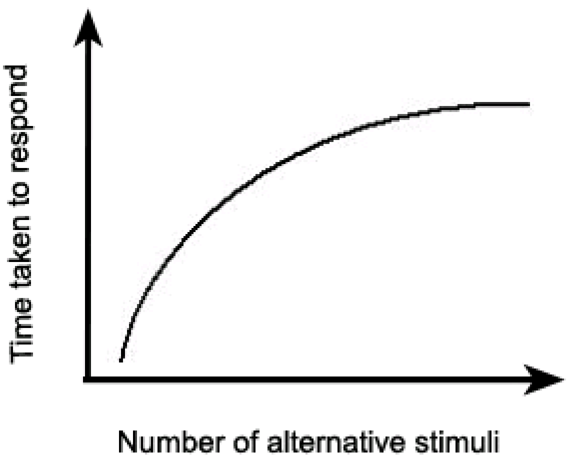

Figure 3-6. Hick's Law (Teach PE 2011)

Given '*n*' equally probable choices, the average reaction time '*T*' required to choose among them is approximately calculated in Code Figure 3-1:

$$T = b.\log_2(n + 1)$$

Code 3-1. Hick's Law

Hick's law has been demonstrated in experiments where users were presented with buttons and corresponding lights. As lights were randomly illuminated, users had to press the corresponding button as quickly as possible. Because the experiment decision process was unrealistically simple and required little conscious thought, it helped to elucidate the ramifications of control placement within an interface.

In relation to MES GUI development, it may not be feasible to calculate reaction times for all screens however it is important to limit the amount of choices displayed on screen at any one time.

3.10.1.2 Fitt's Law

Published in 1954, Fitt's law is a model of human movement, predicting the time required to move from a starting position to an end point. It takes into account the distance between the points and also the size of the target (Wikipedia 2010). Fitt's law therefore can be used to model the act of pointing, for example with a finger or stylus on a touchscreen or also through using a mouse, as illustrated in Figure 3-7:

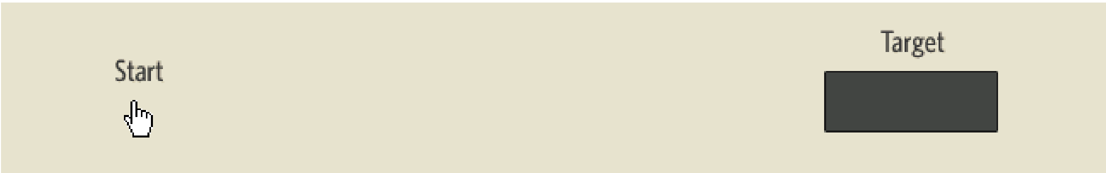

Figure 3-7. Fitt's Law (Particletree 2011)

Fitt's law has been formulated in several different mathematical models. One of the commonly used equations is shown in Code Figure 3-2:

$$MT = a + b \log_2(2A/W)$$

Code 3-2. Fitt's Law

Where '*MT*' is the movement time, '*a*' and '*b*' are regression coefficients, '*A*' is the distance of movement from start to target centre and *W* is the width of the target. From the equation, a speed-accuracy trade-off can be seen with pointing. Targets that are smaller and/or further require more time to acquire. There are however significant gains by slightly increasing the size of smaller targets. These gains taper off as a target becomes bigger, so that there is no real benefit to making a big target even bigger. It is worth noting that Meyer's Law was developed to mitigate inherent limitations with the Fitt's model (Usability First 2011). One such limitation is that it is only effective when measuring single dimensions that are straight-line movements, as illustrated in Figure 3-8:

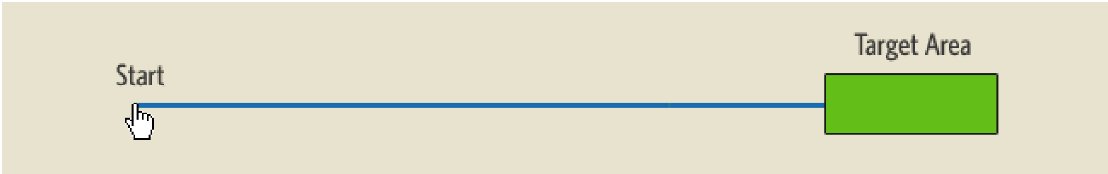

Figure 3-8. Fitt's Law target area acquisition (Particletree 2011)

Meyer's Law is used to predict how much time it takes for a user to accomplish a task involving selection of targets such as icons, menus, or Hypertext links. It also predicts that human movements consist of a primary sub movement and a possible corrective secondary sub movement toward a target.

In combination with Hick's Law, Fitt's Law can prescribe highly ergonomic GUI designs that reduce cognitive and reaction burdens on end users. While there are some noted limitations of the model, Fit's Law can be loosely applied by designers by moving target buttons closer and also making them larger if frequently used. Generally, MES usage does not warrant time critical responses by users but there may be particular facilities that require prompt attention – in this scenario it may be of use to model the data to ensure the GUI is as suitable as possible.

3.10.1.3 The Power Law of Practice

The Power Law of Practice states that the logarithm of the reaction time for a particular task decreases linearly with the logarithm of the number of practice trials taken. Put simply, it prescribes the learning curve effect on performance and how it rises sharply and tapers off (Wikipedia 2010b). Code Figure 3-3 mathematically represents the concept:

```
Tn = T1 n-a, a ~ 0.4
```

Code 3-3. The Power Law of Practice

Where '*Tn*' is the time to perform a task after '*n*' trials, '*T1*' is the time to perform a task in the first trial and '*n*' is the number of trials. The following Figure 3-9 illustrates this law:

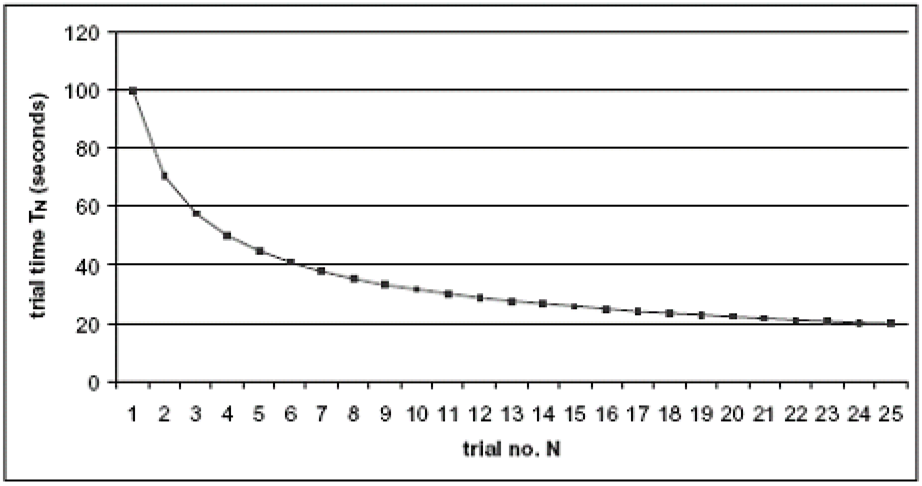

Figure 3-9. The power law of practice (Roessingh and Hilburn 2000: 7)

It is beneficial to take this model into consideration when designing GUI interfaces that minimise the amount of training needed to successfully use the application. It also helps design applications that enable users to cross train quickly on a number of tasks in manufacturing environments. Employees benefit from job enrichment and the business benefits from a more flexible workforce.

3.10.2 Understanding User Memory Functions

Having the most advanced and feature rich of systems is of no use if they do not abide by the nuances of human memory capabilities. Overburdening users' memory abilities lead to high frequencies of errors and low adoption rates. Through understanding human capabilities and boundaries, systems can be designed for optimal navigation by its users.

3.10.2.1 General Human Memory Construct

While there are very little gains to delving into the depths cognitive psychology for the purpose of MES design, significant benefits can be derived from a high-level understanding of human memory and perception. The multi-store model as defined by Atkinson and Shiffrin gives a general overview of memory and how it is comprised of short term and long term functions. The following Figure 3-10 illustrates this concept:

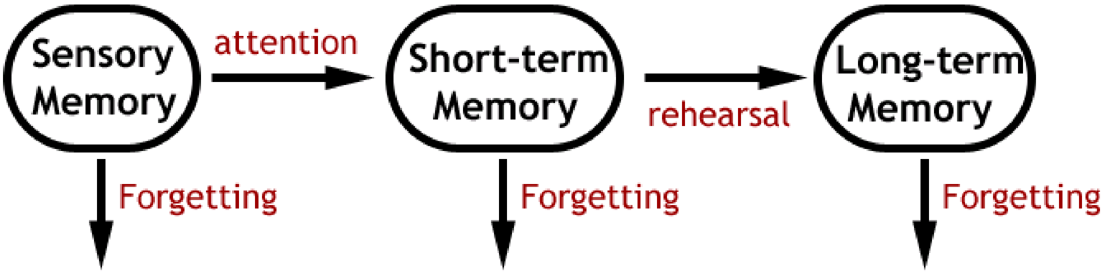

Figure 3-10. Atkinson and Shiffrin memory construct (Science aid 2010)

This model highlights that humans will not always remember information indefinitely. Unless something is recalled on a periodic basis, even information in long-term memory may no longer be available.

In the context of a manufacturing environment, users many not always be operating effectively due to stress factors such as excess noise, uncomfortable temperatures, excessive pressures to meet expected output targets, tiredness induced from shift work etc. Therefore it is important to limit user actions when navigating MES so that they cannot make grossly incorrect choices. It is also essential to have online contextual help so that the user can refresh their memories for tasks they are not completely sure about. Elizabeth Loftus, a psychologist and prominent expert on human memory, has been widely accepted for her work on the fallibility of memory. Through various means, human memory can either shift over time or even become falsely remembered in the first place (Loftus 2003). So for even highly experienced users, it is beneficial to have access to online help, tutorials and other sources of self-service job training and information.

3.10.2.2 Miller's Magic Number

In 1956, the cognitive psychologist George A. Miller published a paper called "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information". This paper highlighted the limitations of human short-term memory capacity and perceptual task abilities (Lidwel et al. 2010: 40). It became apparent that humans could only remember between five and nine concurrent bits of information. Miller did not draw any firm conclusions and simply hypothesised that the recurring limit of seven items might "represent something deep and profound or be just a pernicious, Pythagorean coincidence". His paper has since been validated numerous times, confirming the working memory capacity limitations. He noticed that the memory span of young adults was around seven elements, called 'chunks'. These chunks consist of digits, letters, words, or other unit types.

Later research revealed that memory ability also depends on the type of chunks used so that we can generally manage around seven digits, six letters or five words. Also, the number of chunks we can retain is lower for longer words. In general, short term memory retention strongly depends on the time it takes to speak the contents aloud, and also on how familiar they are to the individual. There are other contributing factors that can influence an individual’s capacity, such as fatigue or background noise; both of which may exist for a manufacturing operative working on night shift. Therefore to minimise cognitive burden and ensure the application can be used effectively, tasks should be broken down to manageable sizes and all required information should be displayed on the same screen. Where possible, tasks should be automated and field’s automatically pre-populated. Where pre-population is not possible, assistive population such as autosuggest functionality could be of benefit. The lower limit of five concurrent items of information should be adhered to when designing MES GUI and underlying workflow so that the majority of users can successfully navigate.

3.10.2.3 The Serial Position Effect

To augment Millers Magic Number theory, Herman Ebbinghaus’s theory of the Serial Position Effect can be used to highlight the likelihood of an item being remembered when multiple items are presented to individuals. The effect of recency and primacy is manifested whereby people are more likely to remember serially presented items that are at the start and the end (Lidwel et al. 2010: 220). Putting this into practice, essential information should always be presented first or last so that the user of a system is not burdened with having to go between different screens and menus to complete an operation. The following Figure 3-11 illustrates this concept:

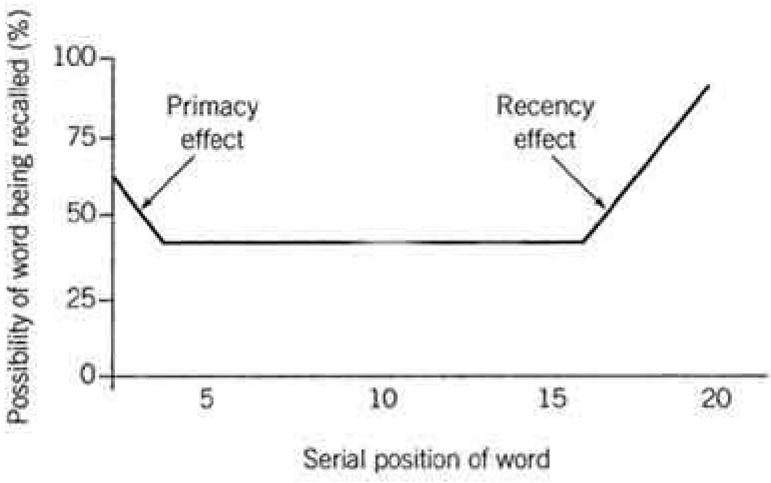

Figure 3-11. Ebbinghaus's serial position effect (Simply Psychology 2013)

3.10.3 HCI Guidelines

The importance of effective HCI cannot be overstated. Problems in this area for MES design can lead to a whole array of issues ranging from reduced productivity to manufacturing faulty product. The following is a list of the HCI models that are most pertinent to the design of MES visual interfaces.

3.10.3.1 Raskin's Design Rules

Jeff Raskin devised a set of design rules that all GUI's should ideally adhere to. These rules are abstract enough that they can be applied to all types of GUI design. They are human focused so that interfaces are designed to fit with users as opposed to the other way around. While they are common sense in nature, they are often lost in the design process and remain unimplemented. The rules are as follows (Bilconference 2010):

1. When using a product to help you do a task, the product should only help and never distract you from the task.
2. An interface should be reliable.
3. An interface should be efficient and as simple as possible.
4. The suitability of an interface can only be determined by testing.
5. An interface should be pleasant in tone and visually attractive.

These guidelines cannot comprehensively prescribe a highly usable and intuitive interface but are very useful due to their simplicity. In the design and development of MES GUI, adhering to guiding principles has more potential for success as opposed to trying to formally specify each detailed element of design.

3.10.3.2 Shneiderman's Design Principles

Progressing from Raskin's design rules, Ben Shneiderman developed a set of principles that further prescribe ergonomic interface design (Shneiderman 2003):

1. Strive for consistency – "Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent commands should be employed throughout."
2. Enable frequent users to use shortcuts – "As the frequency of use increases, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user."
3. Offer informative feedback – "For every operator action, there should be some system feedback. For frequent and minor actions, the response can be modest, while for infrequent and major actions, the response should be more substantial."
4. Design dialog to yield closure – "Sequences of actions should be organised into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and an indication that the way is clear to prepare for the next group of actions."
5. Offer simple error handling – "As much as possible, design the system so the user cannot make a serious error. If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error."
6. Permit easy reversal of actions – "This feature relieves anxiety, since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry, or a complete group of actions."
7. Support internal locus of control – "Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders."

8. Reduce short-term memory load – "The limitation of human information processing in short-term memory requires that displays be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions."

Of all the HCI and memory principles, this is possibly the most valuable due to its simple but comprehensive guideline on how to build highly usable interfaces. This encompasses not just visual design but the architecture and behaviour of an application. This human centred nature serves as a blueprint for a really usable application. All the eight principles have high relevance to MES design. The descriptions have been included here with the points due to the fact they are so informative and helpful.

3.10.3.3 Nielsen's Usability Heuristics

Any discussion of usability and the web would be incomplete without mentioning the work of Jakob Nielsen. Nielsen is a prominent web usability figure and has a relevant guide list called usability heuristics. He called them heuristics because they are loosely defined as opposed to being specific guidelines (Nielsen 2010):

1. Visibility of system status.
2. Match between system and the real world.
3. User control and freedom.
4. Consistency and standards.
5. Error prevention.
6. Recognition rather than recall.
7. Flexibility and efficiency of use.
8. Aesthetic and minimalist design.
9. Help users recognise, diagnose, and recover from errors.
10. Help and documentation.

The work of Nielsen is not without criticism as he is very vocal on the point of having as simplistic a design as possible without the use of unnecessary rich content. This paradigm may not be desirable for websites that need to attract and retain user attention by creating visually attractive aesthetics. The nature of MES is different however, being predominantly designed for its functionality. A lot of Nielsen's guidance may be outdated due to greater internet bandwidth, more visually capable internet browsers and also increasing levels of end user sophistication. Taking this into account, Nielsen's heuristics on web design into account can still yield benefits to MES.

3.11 Utilizing touch screens in a manufacturing environment

Touch screen utilization in manufacturing is an ergonomic method of interfacing with computer systems. Keyboard & mouse usage can be cumbersome in an environment where users may be required to stand or adopt some form of non-standard posture in order to carry out their tasks. Touch screens will ideally be elevated shoulder high on a particular point at a machine so that the operator can have visibility of both computer terminal and machine simultaneously while they conduct their work based activities.

Touch screens allow quick navigation but must be specifically designed for in an application, as there are a number of differences in GUI layout requirements between touch based and standard keyboard & mouse interaction. The dexterity of point and touch actions cannot measure up to navigation with a mouse so all GUI controls need to be enlarged for ease of touching on screen (Frontend 2011). This leads to significantly lower available screen space for GUI design so applications need to make full use of the available screen space. Because of this limitation, the GUI should be designed to minimise the amount of components on the screen. Complementary to Human Computer Interaction (HCI) principles discussed in the previous section 2.9, these requirements tie in conveniently when designing usable interfaces.

There are some nuances to be taken into account with this departure from traditional interface design which often only become apparent through real-world testing. While any application can become touch enabled through presentation via a touch screen, applications must also be touch optimised to cater for the alternate interface method. For example, immediate feedback should be provided to indicate that options are being selected. This is necessary due to the fact that the hand being used to navigate is blocking the screen, therefore the user is not completely sure they have acquired their on-screen target (SAP Design Guild 2011: 6). There may be the possibility of introducing haptic feedback as used with smartphone devices. Also, backgrounds should generally be kept bright (ELO Touchsystems 2011). This not only reduces glare but also reduces the visibility of fingerprints and other debris that can accumulate through use in a manufacturing environment. Conventional idioms for interfacing need to be re-evaluated for touch screen usability. Scroll bars can become extremely tedious to operate and should be replaced by touch friendly controls such as tab-strips and other screen switching idioms. The same is true for complications in navigating multiple open windows via touch screen.

A touch friendly MES is ineffective if high portions of activities require keyboard and mouse input. As a guideline, no less than 80% of navigation should be performed through a single input method (Galitz 2001: 439). On-screen keyboards are becoming more popular and provide the functionality of a physical keyboard without the user having to leave the screen. New mobile “smartphones” have arisen, whereby the phone has no buttons with all interfacing performed through a touch sensitive panel. Touch enabled technology is not limited to mobile phones however as highlighted by Microsoft with its work on its Surface product, where a table with a touch surface can be used for all manner of interactive activities. Windows 8 with a planned release of 2012 is to be designed with touch capability being a fundamental design factor (Softpedia 2011).

In the future there may be scope for even more advancement in computer interfacing and navigation. Currently, Microsoft’s Xbox Kinect is a revolution in the computer gaming industry, whereby users can interact with a computer through body movements. Its use is also being investigated for business and other purposes. However as with touch screen navigation, intensive interaction may be exhausting and be less suitable than the traditional keyboard and mouse (Galitz 2007: 438).

3.12 MES Security

In general, MES security design is inadequate (United States Congress 2004: 35). MES security developed pre September 11th 2001 in particular is often inadequate as it was not designed in cognisance of potential malicious exploitation (Shaw 2006: xvii). MES used in industrial and infrastructure control are particularly vulnerable to potential malicious exploitation (Lewis 2004: vii). The ramifications of inadequate MES security can result in serious financial loss and jeopardise human safety. Therefore, security is an aspect of MES design that warrants research when proposing a revised architecture. Web based COTS MES security is of additional relevance due to the fact that critical manufacturing systems could be accessed remotely by 3rd parties. While a reusable Rapid Application Development (RAD) architecture is excellent for quickly designing fully featured web applications, there are also security risks involved in using code that has inherent flaws. In particular there is a growing trend of web based systems exploitation as it becomes the main conduit for attacks (Symantec Internet Security Threat Report, 2008). A public example of this was when the slammer worm crashed the Ohio nuclear plant network in 2003 (Security Focus 2003). That was not just an isolated incident however as there have been multiple accounts of malicious software causing real-world problems, including a plane crash that was suspected of being caused from malware (MSNBC 2010). 2010 was the year that saw MES security threats progress to a whole new level. The Stuxnet worm was found to have infected 60% of Iran's nuclear power plants. It was subsequently discovered that the worm was written with specific intent to target a certain type of SCADA system, found primarily in Iranian power stations (Silicon Republic 2010). This cyber terrorism event highlights the possibility of malicious threat and the need to mitigate potential methods of MES subversion.

3.12.1 Security policies

The formulation and implementation of security policies provides reasonable assurance of maintaining a secure system. Fundamentally, security policies mandate minimising accidental or intentional inappropriate actions within a system. Password management for example is a preventative form of security measure so that malicious parties cannot easily gain access to systems. Such password control measures would be to ensure reasonable password complexity, account locking and enforced expiration. Depending on MES design, authentication could even be handled by an intermediate system such as Microsoft's Active Directory. Other security measures are in the form of activity logging. Logging user and system activity is just as important as preventative measures as they enable determination of issues when preventative measures fail. Because failure can never be ruled out, logging is crucial to finding out what exactly went wrong.

3.12.2 CWE/SANS top 25 security flaws

The CWE and SANS Institutes in collaboration with a number of other organisations, released a list of the 25 most common programming errors that can lead to security vulnerabilities (SANS Institute 2011). By publicizing these error types, the organizations endeavoured to make public software more secure. According to the group, just two of these error types led to more than 1.5 million security breaches in 2008 (CWE Institute 2009). Simply knowing about these error types is not enough as defences and checks must be incorporated into code to minimise security vulnerabilities. The security firm Kapersky is a leading world expert on computer security and has often been first to diagnose major software vulnerabilities. Ironically in February 2009, news was released that the Kapersky US website was hacked, resulting in customer information being leaked from its internal database (PC Advisor 2009).

Specific exploits are technology specific and resolved by software patching on a regular basis. The top 25 list spans all technologies that are used to create a software based service such as a website or other publicly available utility. At the very least, the process of mitigating the 25 error types will give rise to more error resilient software. The exploits all operate on the principle of doing something that the developer had not considered or catered for. By preventing these errors, software will not only be more secure but also more reliable and operate with fewer errors, less downtime and reduced technical intervention; ultimately this will all result in a more successful and profitable business. Appendix A covers these error types in more detail with relevancy to MES design.

3.13 System Errors and Failure Modes

System errors can present themselves in numerous forms, ranging from minor user annoyances to complete system outage. While minor annoyances can be frustrating to users, it is still possible to continue working. In the case of total outages however, it can be very costly to businesses. A rough formula for calculating the cost, as devised by Patterson (2002) is illustrated in Code Figure 3-4:

```
Estimated average cost of 1 hour of downtime =
```

```
Employee costs per hour * Fraction employees affected by outage
+ Average Income per hour * Fraction income affected by outage
```

Code 3-4. Patterson's cost of downtime

This is quite a relevant guide for manufacturing environments where workers are employed for the purposes of manufacturing products. With higher levels of automation, workers activities are tied in with computer systems and are unable to proceed with productive value adding work. In addition to the employee costs, there are other associated financial losses that are incurred when production is halted due to system errors. These costs can include heating, lighting, equipment rental, plant services and other similar expenses.

To minimise these financial losses, errors should be reduced in both scope and frequency. Where possible, systems should be designed to fail gracefully and rectify themselves. At the very least, they should contain the issue and request human intervention in a timely manner. Features should also be built in that enable easy failure recovery. The following sections list the main areas for potential errors.

3.13.1 Software Bugs

With the requirement for increasing levels of automation, systems of higher complexity are being developed. While this complexity enables much greater innovation and cost savings, it also incurs a much higher cost in the event of system outage. This growing complexity is caused by significantly more business logic being encapsulated in code. However, the more lines of code written into programs, the more potential for issues and failure. The industry standard count of bugs per 1000 lines of code is approximately between 15 and 50 although larger projects have a greater propensity for error frequency (McConnell 2004). While there are techniques such as Z notation that can be used to significantly reduce if not completely remove defects, the time and cost involved is not feasible for MES design. If a company attempted to employ these techniques, costs would outweigh Return On Investment (ROI) and make time to market so slow as to destroy any competitive advantage. Therefore as more code is developed for growing automation, there will be a proportionately greater amount of bugs. Unfortunately not all of these bugs will be discovered through testing and validation, no matter how rigorously conducted. Some bugs can be deeply embedded in code, remaining dormant for years; only activated when multiple unusual conditions arise, invoking a previously unused piece of functionality. In the case of the North American power blackout of 2003, a system bug contributed to the loss of electricity to about 55 million people (Kahl and Vasovski 2004). This software bug resided in a General Electric energy management system. Mike Unum, a General Electric energy spokesperson said that the system was in operation for over three million hours in which the bug had not surfaced. Unum also believed that due to the nature of the bug, it would not have been revealed even with more extensive testing. This example highlights the fact that while every precaution may be taken to develop safe and error free software, there will always be the potential for system failure caused by previously undiagnosed bugs. In cognisance of this, systems should be developed with appropriate error handling so that in the eventuality of a bug surfacing, it does not cause escalating uncontrolled problems.

3.13.2 Hardware

Unlike software, hardware has a limited lifespan. Depending on the hardware component type, it could last for months in the case of consumable items, to decades for items that do not have any moving parts. In large-scale enterprise applications, hardware failure is becoming an increasing issue due to the number of components involved (Schroeder and Gibson 2007: 1). As will be highlighted in section 3.20 and 3.21 of the web software functionality chapter, clustering and Storage Area Networks (SANs) provide a greater level of system uptime through hardware redundancy and load balancing of user requests. These technologies have multiple layers of duplicate hardware, removing any single point of failure. As always, there is a cost associated with avoiding downtime. Some estimates believe this can be as much as 40% of the initial system cost (Schroeder and Gibson 2007: 1). The most comprehensive hardware redundancy may not be cost effective for every MES system so a balance is needed between risk avoidance and cost.

3.13.3 Usability

In addition to GUI usability, there can also be deficits in the usability of system functionality. Automation technology was originally developed for the purposes of efficiency and cost saving, with a significant reduction in operator workloads, putting humans in a more supervisory role. The assumption was that automation could transparently replace human actions without any negative impact on the overall system operation. This view is predicated on the notion that a complex system is decomposable into a set of independent tasks. Many automated systems have been designed without consideration for the modified role that humans have when interacting with these systems (Sarter et al. 1997). With the changing role that humans take when interacting with automated systems, care must be taken to ensure systems behave in a manner to how the users could reasonably expect. Negative consequences can arise from unanticipated outcomes when the user thinks a particular event will happen, but in reality, the system pursues another course of action. The importance of usability not only covers interface design but also how it behaves in response to user requests.

Quite often, adequate feedback is not provided to users. This can result in information overload with resultant catastrophic failure. An example of this type of usability failure happened with a radiotherapist operating an X ray machine in a hospital. Due to a glitch in the machines control software, a patient received a lethal dose of radiation. The radiotherapist was unaware of the problem because the machine falsely indicated that no X rays were administered. The patient died a few months later from complications related to the overdose received (Leveson and Turner 1993). Although some error types are virtually impossible to eradicate, usability related problems are reasonably avoidable through proper testing and consultation with end users.

3.13.4 Inadequate error handling

It's not feasible to completely irradiate every software bug that exists in code. Therefore all systems must cater for the eventuality that something will go wrong. Even a perfectly written application can run into problems when a dependant resource or 3rd party interface errors or responds incorrectly. When these issues arise, sometimes applications can fail gracefully by handling the issue and reporting status to the user, as demonstrated in the following Figure 3-18:

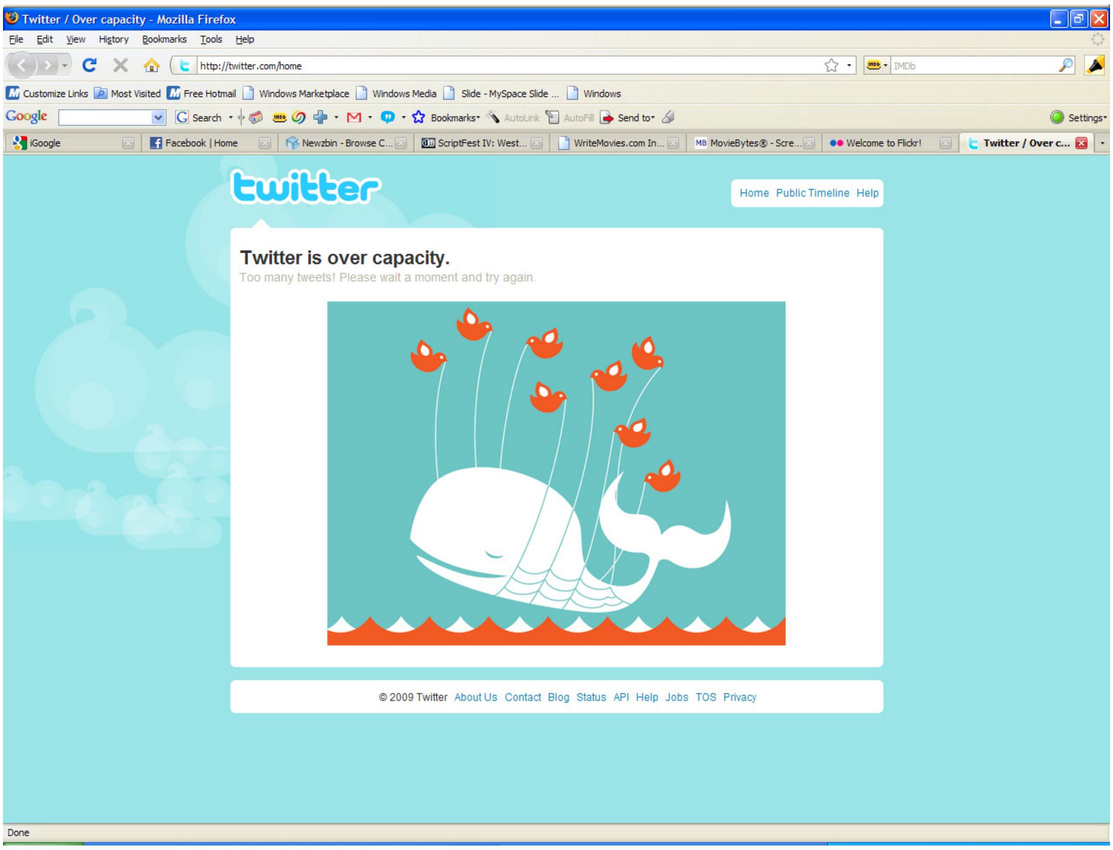

Figure 3-12. Graceful handling of an error state

Sometimes however uncontrolled errors happen, giving rise a complete failure to process requests as can be seen in the following Figure 3-19:

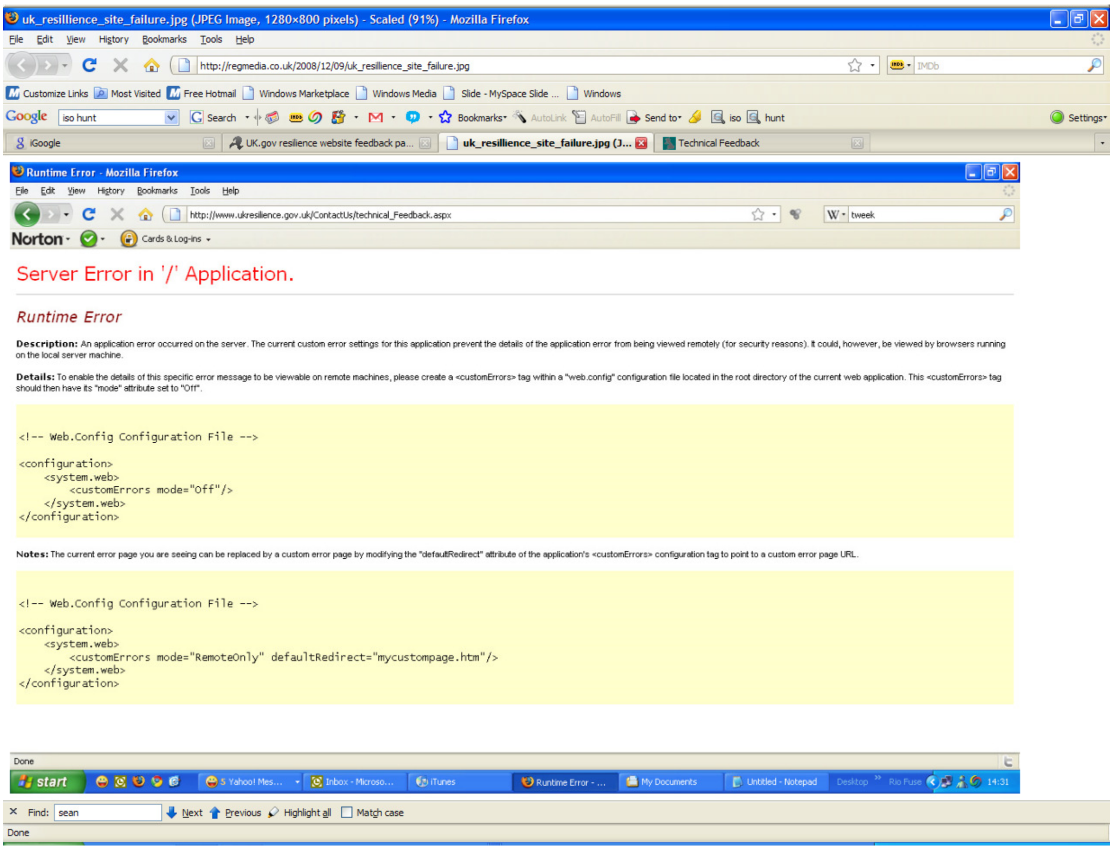

Figure 3-13. Unhandled error display

These types of errors are very dangerous as the system can be left in an indeterminate state with partially updated data. In two-phase transactions involving multiple system interfaces, one system may have been updated successfully with the other one in a state of failure. In the scenario of a banking system, this error could manifest itself in a customer being debited for a payment without the payment being passed on.

In addition to capturing the error, system resources should be released with all actions returning system state and data to how it was before the action began processing. This state restoration is known as a rollback. All relational databases have inherent support for transaction rollback. A sample of VB.NET code correctly handling a SQL Server transaction is illustrated in Code Figure 3-5:

```
Try
    ' Open Connection
    thisConnection.Open()
```

```
        ' Begin transaction and attach it to command
        thisTransaction = thisConnection.BeginTransaction()
        thisCommand.Transaction = thisTransaction

        ' Run first  command
        thisCommand.CommandText = “Update bank_balance set funds = funds + 100”
        thisCommand.ExecuteNonQuery()

        ' Run second command
        thisCommand.CommandText = “update money_paid_out set funds = funds - 100”
        thisCommand.ExecuteNonQuery()

        ' Commit transaction
        thisTransaction.Commit()

    Catch ex As Exception
        ' Roll back transaction
        thisTransaction.Rollback()
    Finally
        thisConnection.Close()
    End Try
```

Code 3-5. Database transaction in VB.NET

As illustrated, the action is all or nothing. If the first table was updated and there was an error updating the second table, the first update would be undone in the exception handler. Retrying is a typical user response to an error; by making actions idempotent, requests can be retried without compounding the issue with further problems.

3.14 Review of Other MES Applications

The following sub-sections present a compilation of the functionality of 22 commercially available MES packages, as listed in Appendix B. All of the available packages were sourced through a variety of channels such as industry partner recommendations, product resellers and general internet search. Of the available packages, 22 were chosen because of their adequate product quality, available product information and relevant product functionality.

Package appraisal was based on criteria such as vendor web site information, online news reports, online web casts, white papers, articles, customer appraisals, industry reviews and partner reports. All compiled MES information was correct as of 10th April 2008. Due to progression of technology, some functionality may be outdated or no longer offered. It is also possible that marketing and public relation forces may exaggerate product abilities.

Most of these products have full feature sets of related functionality such as Customer Relationship Management (CRM), ERP, Materials Requirement Planning (MRP), Learning Information System (LIMS), etc. This research is focused on the MES layer only however.

3.14.1 General core functionality:

MES architecture is best suited to a modular design. Service Orientated Architecture (SOA) is becoming a popular method of modularising functionality and enabling features to be turned on/off so that customers can get flexible price models based on the particular needs. Smaller MES packages provide fully comprehensive feature-sets to run small organisations, including functionality such as: job costing, accounting, financials, ordering, payroll, inventory, quotation, labour management, time & attendance, warehouse management and kanban replenishment. The larger MES providers tend to specialise their products and enable integration with “best in class” enterprise applications such as CRM, ERP, MRP, etc.

A basic business requirement is to have full visibility into manufacturing data for product as it moves through processing. This includes lot and Bill Of Materials

(BOM) tracability so that all manufactured product can be tracked and controlled. In efforts to become more productive and efficient, scrap, rework and downtime tracking is essential as part of the core MES functionality.

While skills tracking capability is not an essential component of MES, it is employed successfully by a number of MES vendors to monitor training needs and prevent expired/untrained operators from performing certain actions. Some MES systems take this further and have integrated training capabilities that provide context sensitive help.

A few of the more comprehensive MES implementations provide innovative features such as message alerting upon user login, statistical process control alerting, non-conformance management, work flow management, change control integration and other forms of document management functionality. There is also a growing trend from these more sophisticated packages to provide paperless manufacturing through extended manufacturing data capture that meets regulatory compliance for electronic Device History Record (eDHR).

3.14.2 HCI/Usability

Operator interfaces are generally tailored for touch screen use. COTS MES state that the GUI needs to be intuitive, easy to use, and well thought out. Badly designed GUIs can lead to manufacturing disasters. Work is generally minimised for the operator, with features such as data entry through an onscreen keypad, removing the need for a physical keyboard.

User prompting is generally used to reduce cognitive loads, where only relevant information is displayed. The user should be guided safely and easily through manufacturing steps, and items that the user does not have access to, are hidden as opposed to disabled, in order to prevent confusion. Colour coding is also used to direct attention such as traffic light warnings onscreen.

3.14.3 Web based functionality

Of the MES offerings that had web capability, they maintain that all data entered into the system should be available in real-time. Where possible, support should also be given to mobile devices so as to enable mobile working. The main reported benefit of a web based application was that of rapid software rollouts. This rapid rollout is highly desired for COTS MES products (Scholten 2009: 110). Companies are also aware of the Application Service Provider (ASP) business model potential whereby customers can have remotely hosted for leased system or client side installation. No COTS MES packages were found possessing cross browser device communication functionality.

3.14.4 Data capture / Device communications

Data capture is an essential component of MES. Manually entered data should be validated upon entry. It should also be possible to receive data from devices such as: weigh scales, scanners, printers, etc. More advanced MES packages can have additional connectivity to devices such as Programmable Logic Controllers (PLCs) through various technologies, such as OLE for Process Control (OPC), Ethernet etc.

3.14.5 Industry Standards

Depending on the target environment, there are numerous standards and regulations that are required to be adhered to. These are generally industry norms that are not mandatory, but do ensure MES design is robust and capable of easy interconnect with related systems. For a COTS MES, adherence to industry standards is essential so that it is as versatile as possible. All MES caters for work ordering and BOM handling in some form or another as it is a basic requirement. While not industry standards as such, techniques such as Six Sigma and Kazan are often built in to help MES drive more successful operations.

Other standards such as OAG, S95/B2MML, OPC-UA, ISA-95 and ISA-88 prescribe design factors such as device communications and data exchange. Even governance such as GUI layout is provided through ergonomic standards set out by ISO 9241.

3.14.6 Compliance

MES utilised in some industries and countries must have certain capabilities in order to be permissible for use within an organisation. In general, MES must always perform an audit-trail of system changes and user action. There must also be a complete lot tracability and genealogy so that any information related to the manufacture of specific product can be recalled. This mandates that all historical data must be readily accessible for the life of product. Various elements of security and control are also required as a basic function so that users have to provide an electronic signature when performing an action.

Healthcare companies sending product to the US, are required to be compliant with Food and Drug Association (FDA) governance on access control and security, such as defined in its regulation called 21 CFR Part 11. There are also standards for specific industries, such as ISO9000 and HACCP for the food processing industry or Payment Card Industry for financial companies. The Sarbanes-Oxley (SOX) Act is required for public companies trading on the American stock exchange (Zhang 2007: 74-75).

3.14.7 Customisation

Systems should be designed so that they can be tailored as much as possible to meet client needs. Rapid configuration and deployment is also desirable to enable quick reaction to changing business needs. Where required, there should be capability for the coding of business rules although this should not be necessary for most deployments. In addition, the GUI should also be customisable without the need for code alterations. As part of this configurability, many MES packages can have different modules turned on/off as required to meet required functionality.

Rather than trying to tailor one product to fit all industry types, MES is made available under different builds. Some products have extensible libraries (that can be programmed with) for additional functionality, as per customers' requirements. Multi-lingual and localisation capabilities are also prevalent in a lot of MES packages.

3.14.8 Business Intelligence

Business Intelligence (BI) functionality is becoming more of an integral part of MES, as opposed to a sophisticated add-on. Real time monitoring of user data entry and automated machine activities is tracked by some MES variants, so that non-conformances trigger alarms. It can monitor Key Performance Indicators (KPIs) so that action can be taken to prevent issues before they happen. Management toolsets like Business Activity Monitoring (BAM) and Business Process Management (BPM) dashboards, enable high-level real-time views of manufacturing performance. There are many add-on packages that can be used to model data and perform meaningful representation that can be used to make informed business decisions. Some notable examples of business intelligence tools are Pentaho and QlikView. There are also some other tools that enable data mining and OnLine Analytical Processing (OLAP) that can give further advantage, enabling manufacturing to become more efficient and effective. Hyperion Cubes and Oracle Discoverer are two such products for this deep analysis of data. It is essential that a COTS MES package has BI functionality either through custom development or integration with 3rd party BI products. It is no longer adequate to drive manufacturing without having full visibility into every aspect of its operations.

3.14.9 Reporting

Comprehensive reporting and manufacturing process visibility is extremely important for MES. Dashboards are used to display user-relevant information in a summarised format that allows drill-down of information. Like a web based MES, reports should also be accessible via a browser and have an array of pre-configured reports ready for user viewing and on demand generation. Effective visualisation through the use of colour coded information, graphing, charting etc. reduces cognitive burden and clearly displays areas within manufacturing for attention. As users are becoming more skilled with technology, there should also be the facility to export the data into various formats such as CSV and XML so that the user can perform further data modelling. To further minimise the work required by users, reports can be automatically generated and delivered according to pre-defined schedules, or on an exception basis. There can often be an overlap between standard reporting functions and BI. The distinction being that, reporting is focused on performing the day to day actions, whereas BI has a longer term strategic focus.

Reporting can often be added as an after-thought to systems with their benefits not fully realised. This is short sighted as continued improvement can only be sustained through in-depth understanding of manufacturing operations. Inadequate reporting prevents visibility into how the business is running and what improvements are possible.

3.14.10Security

Security access control should prevent unauthorised access and possibly use a single sign-on method, so that a single account can be used for various systems. The access should have fine grained control so that individual action items can be enabled, and so users are given only the access they need to perform their duties.

3.14.11Interfacing and interoperability

While it may not be possible to natively interconnect into all related systems, MES should adhere to common standards and facilitate easy integration to other systems that are used in an enterprise such as ERP, Supply Chain Management (SCM), LIMS, MRP, SCADA etc. Typically this interoperability is performed through a common protocol such as TCP/IP, XML, ODBC etc. Some MES applications even have inbuilt Electronic Data Interchange (EDI) capabilities so that data can be synchronised between manufacturing plants. A flexible and interoperable design should be given more focus, because historically, integrating MES with adjoining systems has been a difficult task (Vijayan 2000).

3.14.12Mobile platform

Dependent upon the customer, mobile working will have varying levels of importance. Support for mobile devices is necessary due to the prevalence of smart phones, Personal Digital Assistants (PDAs), iPads, etc. and the need to react much quicker to customer demands and changes in manufacturing.

3.15 Literature Review Conclusions

Various aspects of MES design have not been covered in the literature review. In particular, specific features such as Process Control, Hold Management etc. have not been discussed as they are implementation specific and more aligned with functionality as opposed to system architecture. Depending on industry type and organisational needs, MES functionality can vary greatly, often with conflicting design requirements. Alternately, this review concentrates on the core elements used to construct a MES application.

While the underlying technologies form a fundamental part of web based MES architecture, new technologies and innovative implementations of existing technologies are continually emerging. The literature review technology listing is subject to aging from the progression of future technologies and business models. Where possible, concepts are discussed as opposed to technical specifics so as to provide more enduring benefits to MES design.

There are a number of pertinent theories and guides for the GUI design of MES. Additional researchers not covered in the review such as Douglas Englebart, Alan Kay, Donald Norman, Deborah J. Mayhew, Rick Oppedisano and James Hobart have all produced similar usability guidelines for website GUI construction. These theories are orientated towards general website construction as opposed to the more industrial nature of MES and so are not as relevant. The most important guides are those that reduce cognitive burden, reduce learning times, and reduce the potential for user-induced errors in addition to making it as easy as possible to recover from errors. As established as the mentioned researchers findings are, GUI implementations should always be tested for usability post construction. What may seem appropriate in theory may in fact not be optimal for the particular needs of the end user navigation. While not so critical, other guides such as providing consistent interfaces, providing shortcuts and allowing users to feel they are in control, can increase user productivity and lead to better user acceptance and adoption. The general theme from the various experts in HCI is similar in nature and essentially strives for an interface that is “humanistic”.

An understanding of the SANS/CWE list of 25 security flaws provides a very helpful framework towards implementing more secure software. For practicality purposes, the list can be condensed down as there are some items of a similar nature such as "External Control of File Name or Path" being nearly the same as "Untrusted Search Path". Mitigating against the discussed exploit types will provide a reasonable level of application protection but outside of system design, there are frequent publications on inherent flaws in 3rd party software such as web based client software (internet browsers), media players and other components. Organisations should install patches for known flaws in all software that comes into contact with MES. Security needs to be not only considered for the MES hosted webserver but also for the clients that utilise it, as a vulnerability on either side has the potential to negatively impact production. Using inherently secure development frameworks such as ASP.NET facilitate more robust applications. There may be a downside to this as reusable frameworks may also come with inbuilt flaws. These are otherwise known as zero day exploits, where the defect has always existed but only uncovered at a later date by hackers. The main attack method on a system is to gain some basic foothold to leverage for further manipulation. Many exploits can only be taken advantage of when gateway exploits are used to gain leverage for further exploit. SQL injection, improper validating and failure to escape special characters are the main entry point for initial attacks. Attackers may also utilise weaknesses in underlying networking technologies such as Domain Naming System (DNS) and lower level hardware software such as exists in routers and other devices.

Policies are another effective tool in the protection of systems. Unfortunately these are often not correctly adhered to, with a lack of monitoring controls and personnel training. A hacking technique often overlooked is the power of social engineering to gain useful information. Overall system security is only as strong as the weakest element. Just because a MES system has been operating successfully for some time does not mean there are no security holes with potential for malicious exploitation. Periodic reviews are essential to continually enforce and uncover potential weaknesses.

Larger companies prefer to leverage the power of “best in class” applications. This entails using standalone COTS applications that are developed for specific purposes. One such example of this is for employee training using a tool called Compliance Wire. While there are many benefits to be gained from using specialised tools, they cannot exert the same benefits of tightly coupled integration. As was found with some MES applications with integrated training elements, they were much more effective than their standalone counterparts.

There are varying levels of sophistication in the reviewed MES applications. A core feature set is supported by all applications, with the more advanced packages possessing more advanced capabilities. There are also some industry leaders that exhibit capability to maximise business effectiveness and directly impact company profitability and survival. While the fundamental MES feature set was satisfied by all, there were contrasting levels of emphasis, with some packages aiming for simplicity and speed whereas others opted to focus on comprehensive integration with the supply chain. Of all the packages reviewed, Camstar was the most prominent because of how aligned it was to manufacturing and how it attempts to drive a more successful business, not just fulfilling a software requirement. This highlights that technological advancements alone will not drive MES improvements. Ultimately, MES improvements can only be measured through improvements delivered to the business. Through review of the various MES packages, it is apparent that there is a growing shift towards web enabled applications. This trend continues further with more and more integration to the supply chain and increased visibility to the business through advanced reporting and business intelligence.

4 Web software functionality

The perception of web based applications has changed with the numerous technologies that have evolved in recent years. The stateless nature of Hyper Text Markup Language (HTML) does not lend itself readily to serving fully functional applications. Fortunately there are a number of methods to enhance the basic HTML/HTTP framework and overcome this limitation. Depending on the scale and functionality requirements, elements from the following technologies and concepts can be chosen when developing a MES solution.

4.1 World Wide Web Consortium

The most prominent body regulating web standards and compliance is the World Wide Web Consortium (W3C). The W3C has been creating interoperable technologies through specifications, guidelines, software and tools since 1994. Its main goal is "To lead the World Wide Web to its full potential by developing protocols and guidelines that ensure long-term growth for the Web" (W3C 2011). In order for the Web to reach its full potential, the most fundamental Web technologies must be compatible with one another and must allow any hardware and software used to access the Web to work together. W3C does not implement technology however. It simply creates a common blueprint for software to follow. There is an online tool provided by W3C (W3C 2010) that can be used to validate the W3C standards compliance of a websites client HTML.

It has often been argued that standards kill innovation and slow technological advancement. For large-scale growth, however there must be interoperability so standardization does in fact foster innovation in practice. The governance of W3C is growing rapidly as the web becomes ubiquitous through mobile computing and other emerging innovative applications. Modern browsers also put a greater emphasis on standards compliance in order to reduce the quirks that must be developed around so that a web page can be viewed the same in different browsers (WaSP 2010).

In order to future proof developed web software, it is important to comply with W3C regulations. Over time, standards are being more closely adhered to. A web site that does not properly comply with regulations may display/work correctly on current browser builds. It is also possible that compatibility will be broken in the future as browsers become less tolerant of non-compliance.

In addition to W3C there are also other governing bodies that should also be considered such as Internet Standard and Request for Comments (RFC) documents published by the Internet Engineering Task Force. There are also more specific regulatory bodies such as Ecma International who govern a standard of JavaScript. Figure 4-1 gives an overview of the W3C governance.

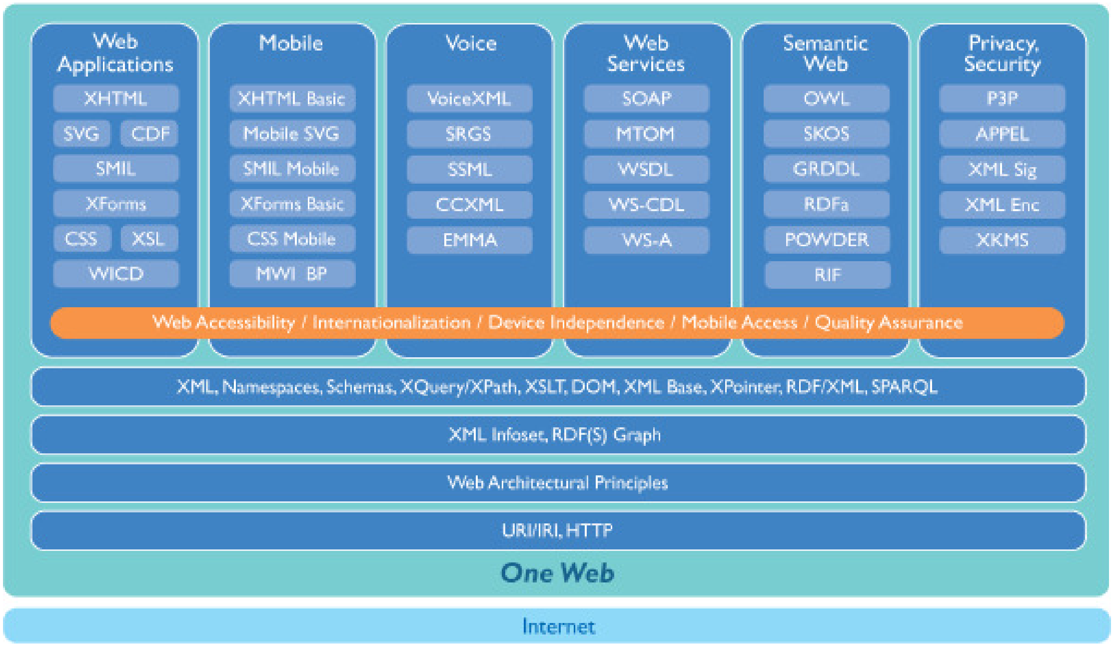

Figure 4-1. W3C governing technology map (W3c 2004)

4.2 Web browser standards compliance

Developing cross browser web sites has always been a challenge. A site may appear and function perfectly on one browser and completely malfunction on another. An example of this is highlighted with the acid test, created by The Web Standards Project (WaSP). The test uses a combination of web technologies to highlight how differently pages can be rendered (WaSP 2009). The following Figure 4-2 shows how the page should be displayed, as is performed via Firefox 3:

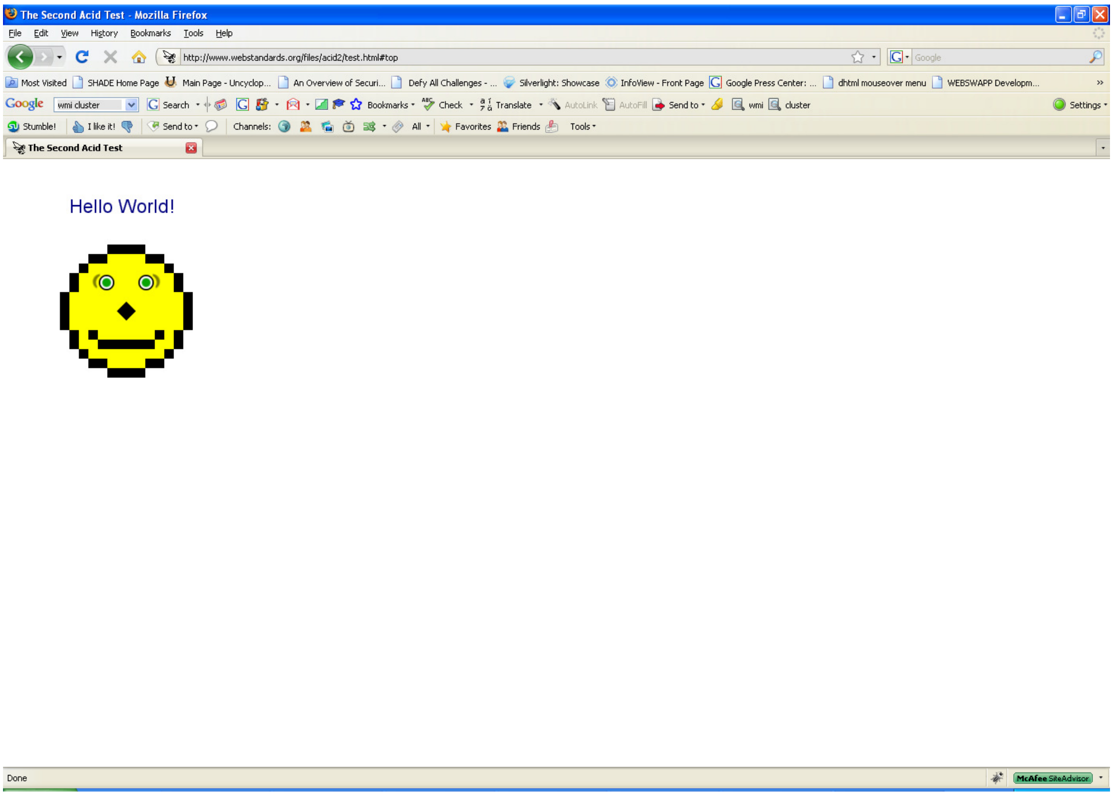

Figure 4-2. WaSP CSS test in compliant browser

However Figure 4-3 highlights the ramifications of non-conformance to standards, performed via Microsoft Internet Explorer (IE) 6:

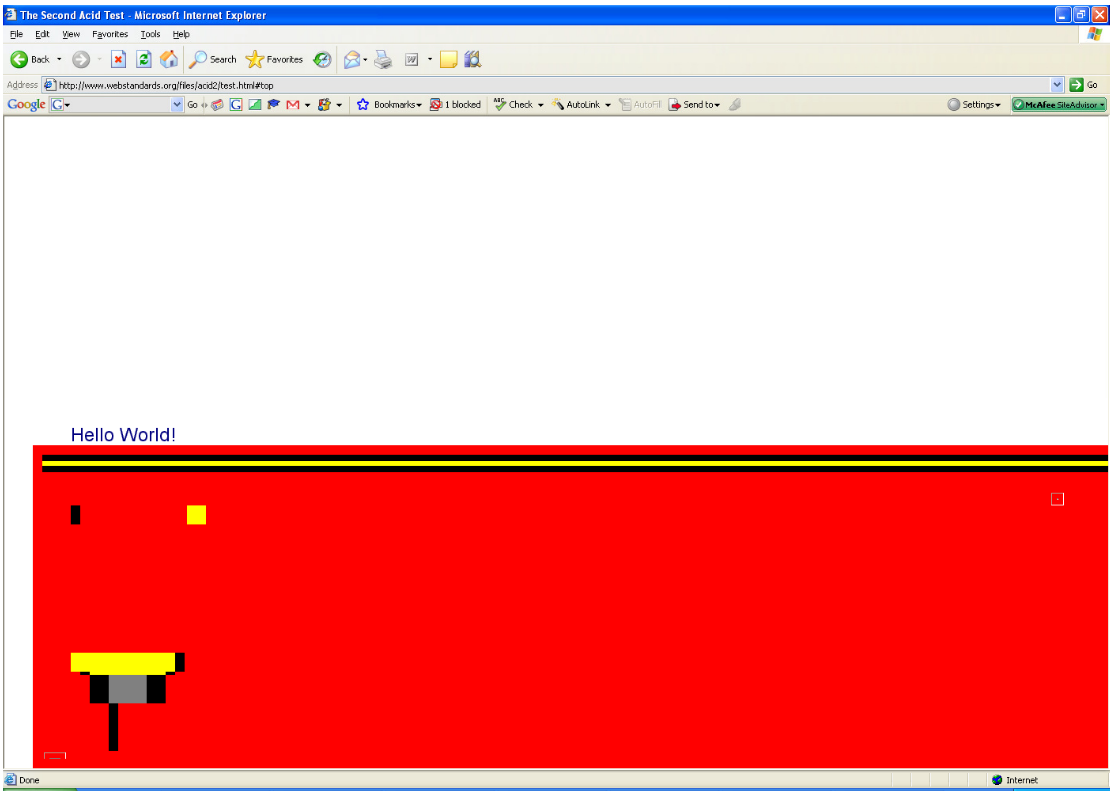

Figure 4-3. WaSP CSS test in non-compliant browser

A website can be tailored to the nuances and quirks of how a particular browser behaves. This may work for initially but as browsers become more standards compliant, quirk coding will be de-supported; leaving web pages open to complete malfunction, as shown in the IE 6 Acid2 test, figure 4-3. Web site development should first and foremost strive for standards compliance, only lending support to individual browser quirk compatibility if it can co-exist. In web browser terminology, ‘quirk mode’ is when a browser lends compatibility to web pages that are not standards compliant. As an indication of things to come, Microsoft decided that from IE8 onwards, it will by default interpret web content in the most standards compliant way it can (Microsoft 2008). There will still be an optional mode for compatibility with sites with non-compliant code. Over time however support for non-compliance will be phased out.

4.3 XML

XML is an ASCII based method for representing data structures in a readable format. There is an opening and closing tag for each piece of information. These tags can be grouped and repeated to enable representation of any required data structure. Code Figure 4-1 illustrates an XML list with one record from an address book:

```
<?xml version="1.0"?>
<addressbook>
  <person>
    <name>Some Body</name>
    <gender>F</gender>
    <phone>123456789</phone>
    <address>
      <street>111 New St.</street>
      <county>Waterford</county>
      <country>Ireland</country>
    </address>
  </person>
</addressbook>
```

Code 4-1. XML example

While XML defines basic data structures as shown, it also refers to the whole suite of protocols in the XML family, with the most notable being: eXtensible Stylesheet Language (XSL), eXtensible Stylesheet Transformation Language (XSLT), eXtensible HTML (XHTML), XML Pointer Language (XPointer) and XML Linking Language (XLink).

Because XML can so powerfully represent data in dynamic structures, it has been used as the underlying framework in a wide variety of other technologies ranging from AJAX and the Semantic Web to Web Services. It has become the industry standard for data exchange because of its open standards as created by W3C and relative ease of implementation, enabling dissimilar systems to communicate (W3C 2009b).

4.4 JavaScript Object Notation

JavaScript Object Notation (JSON) is a lightweight data-interchange protocol developed to minimise the amount of characters required to represent data. Speed of data transfer is highly desirable when transferring data over the web. As can be seen, less characters being transmitted means quicker loading time. JSON was developed as a more efficient protocol to replace XML in certain uses. XML in comparison is more verbose. The following is a sample of a JSON script:

```
{"menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuitem": [
      {"value": "New", "onclick": "CreateNewDoc()"},
      {"value": "Open", "onclick": "OpenDoc()"},
      {"value": "Close", "onclick": "CloseDoc()"}
    ]
  }
}}
```

Code 4-2. JSON example

The following Code Figure shows the same data encoded with XML:

```
<menu id="file" value="File">
  <popup>
    <menuitem value="New" onclick="CreateNewDoc()" />
    <menuitem value="Open" onclick="OpenDoc()" />
    <menuitem value="Close" onclick="CloseDoc()" />
  </popup>
</menu>
```

Code 4-3. XML comparison to JSON

4.5 AJAX

AJAX was publicly released in early 2005 (Zakas 2006: 1). AJAX in itself is not a new technology but it uses asynchronous JavaScript calls in an innovative approach that reduces the gap between web and desktop applications. This method involves sending only a small amount of information to and from the server in order to give the user the most responsive experience possible, without the necessity for costly Frames, iFrames or other cumbersome methods to compartmentalise functionality on a web page. Instead of the traditional web application model where the browser itself is responsible for initiating requests to, and processing requests from the web server, the AJAX model provides an intermediate layer that can perform autonomous actions (Zakas 2006: 1). Instead of loading a webpage, at the start of the session, the browser loads an AJAX engine written in JavaScript. The engine is responsible for both rendering the interface the user sees and also for communicating back to the server. This enables a web page to load faster with only minimal content displayed as required. Any update to the user interface can be performed without having to fully reload the page. The following Figure 4-4 highlights the differences between AJAX and the standard HTTP model:

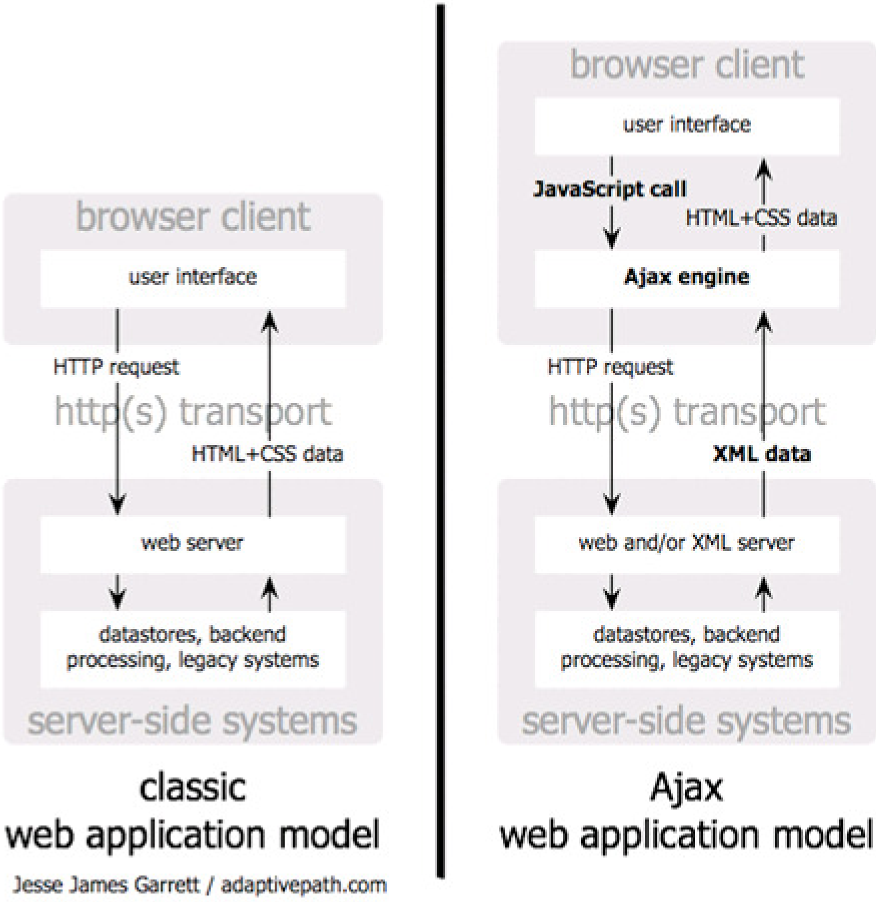

Figure 4-4. Comparison of HTTP versus AJAX (Adaptive Path 2009)

4.6 CSS

CSS is a technology used for visually representing elements in the browser. It is a very efficient way to format colours and layout of web pages and can elegantly implement designs that would not be possible otherwise. Style formatting code can be either embedded in the relevant web pages or maintained in separate pages. When CSS is implemented, it enables the underlying HTML to be much cleaner and easier to read. CSS code can additionally be applied to a multiple web pages, enabling a simple and controlled method to change the GUI layout across a whole web site. W3C created a useful online CSS validator that can assist in developing standards compliant CSS (W3C 2009a). Web pages using CSS tend to have much smaller file sizes than those using tabular layouts. Smaller file sizes mean reduced bandwidth costs, quicker page loading and ability to use lower specification server hardware to do the same job. In addition, CSS is readily compatible and does not need modifications to be displayed correctly on devices such as PDAs, mobile phones, WebTV etc. CSS also enables a higher ranking in Search Engines due to the fact it is much easier to understand and enables a better classification. An example of a basic CSS style and its use is illustrated in Code Figure 4-4:

```
#divContents
{
	width:95%;
	height:100%;
	overflow:auto;
	border:1 solid;
}

<div id="divContents">hello</div>
```

Code 4-4. CSS code example

4.7 Reusable JavaScript frameworks

In comparison to desktop applications, the web is regarded as having a rich and dynamic GUI. While this is true, it requires substantial development for robustness across different browsers and versions. To enable more RAD, there are a number of framework libraries and toolkits that can be used to create rich web interfaces. In general, toolkits are more GUI based and are often extended from the framework libraries. While framework libraries can be extended into other uses, they can also be used directly within code. Both methods have reusable functions and GUI controls that can be plugged into any web project, enabling a much quicker product creation cycle. Due to the explosion in web technologies, it is improbable that a developer will be expert in all areas of development. Toolkits enable general developers to make use of well written code from experts in various technology fields.

A notable framework is called Prototype. It is a class-based package of commonly used JavaScript routines that can be reused without having to redevelop the same functionality in different projects. The Prototype framework has been so effective that it has been further extended by other JavaScript toolkits. The following is a list of some of the other notable toolkits, most of which extend from Prototype are: jQuery, qooxdoo, Ext JS, YUI, Dojo, MooTools, script.aculo.us, Yahoo! UI Library, MochiKit, Google Web Toolkit, Rialto and Spry Frameworks (Web Designish 2011).

In order to deliver the relevant functionality, these toolkits typically use a combination of technologies such as: XML, CSS, Document Object Model (DOM), Dynamic Hyper Text Markup Language (DHTML), JSON and AJAX.

4.8 WEB 2.0

Web 2.0 is not a technology as such. Instead it is a definition for the collection of standards and technologies that are used to create fully functional, interactive, visually rich and highly dynamic web content. These web sites are usually business based performing services and/or functions useful to consumers. While some sites do not generate revenue from customers, the can generate revenue from related avenues such as advertising. The arrival of the various toolkits and technologies was a large enabler of Web 2.0. Social networking sites such as Facebook and Google+ are typical examples of Web 2.0.

4.9 Advanced performance techniques & optimisations

As web applications encompass growing amounts of functionality, their performance often degrades due to higher amounts of data transfer the internet. By reducing the amount of data that needs to be transferred, performance degradation can be negated for growing web applications. One method to reduce data transfer is to reduce the amount of individual HTTP requests made, as there is extra overhead for each individual trip. This can be conducted by amalgamating content from multiple files into one central file, such as, placing styles from multiple CSS files into one.

All interactive and non-static websites use some amount of JavaScript. Because JavaScript is not compiled to machine code, it can often be quite verbose with the repetition of large identifiers. E.g. a variable called varTagToHoldOldUserSelection and also large comment sections included to aid future modifications. This can result in JavaScript taking up a significant portion of page download times. To get around this problem, there are tools to minify the JavaScript code so that it still does what it was written to do, but takes up as little space as possible. One such tool to minify JavaScript code has been developed by Douglas Crockford (Crockford 2009). The following Code Figure 4-5 is example provided by the Crockford website of pre minified code:

```
// is.js
// (c) 2001 Douglas Crockford
// 2001 June 3
// is
// The -is- object is used to identify the browser.  Every browser
edition
// identifies itself, but there is no standard way of doing it, and
some of
// the identification is deceptive. This is because the authors of web
// browsers are liars. For example, Microsoft's IE browsers claim to
be
// Mozilla 4. Netscape 6 claims to be version 5.

var is = {
    ie:      navigator.appName == 'Microsoft Internet Explorer',
    java:    navigator.javaEnabled(),
    ns:      navigator.appName == 'Netscape',
    ua:      navigator.userAgent.toLowerCase(),
    version: parseFloat(navigator.appVersion.substr(21)) ||
             parseFloat(navigator.appVersion),
    win:     navigator.platform == 'Win32'
}
is.mac = is.ua.indexOf('mac') >= 0;
if (is.ua.indexOf('opera') >= 0) {
    is.ie = is.ns = false;
    is.opera = true;
}
if (is.ua.indexOf('gecko') >= 0) {
    is.ie = is.ns = false;
    is.gecko = true;
}
```

Code 4-5. JavaScript before minifying

Code Figure 4-6 is an example provided by the crockford website of post minified code. Visibly the code is much more condensed and bandwidth friendly but is also much harder to modify. As a result, it is necessary to maintain a copy of the original JavaScript code for future modifications:

```
var is={ie:navigator.appName=='Microsoft Internet
Explorer',java:navigator.javaEnabled(),ns:navigator.appName=='N
etscape',ua:navigator.userAgent.toLowerCase(),version:parseFloat(
navigator.appVersion.substr(21))||parseFloat(navigator.appVersion)
,win:navigator.platform=='Win32'}
is.mac=is.ua.indexOf('mac')>=0;if(is.ua.indexOf('opera')>=0){is.ie
=is.ns=false;is.opera=true;}
if(is.ua.indexOf('gecko')>=0){is.ie=is.ns=false;is.gecko=true;}
```

Code 4-6. JavaScript after minifying

Browser caching can be an easily implemented method to increase application response. For example, the following Code Figure 4-7 illustrates the Meta tag, that tells a browser that the web page will not expire until a certain date. This saves the client from having to reload the full page on every request:

```
<meta http-equiv="Expires" content="Sat, 16 Jul 2016 18:45:00
GMT">
```

Code 4-7. META tag refresh example

The first time the web page loads, all data is transferred as normal. But for each subsequent fetch, the browser can use the client cache instead of retrieving from remote web servers. Unfortunately this option will only work with static web pages. Dynamic web content needs to be transferred each time without cache.

Page compression techniques also enable quicker application responses. This compression can significantly reduce the volumes of data being downloaded by performing content encoding of text based web content such as: CSS, JavaScript and HTML. Images and similar media are already in a compressed state and so cannot benefit from the content encoding. There is a slight CPU overhead on the server for compressing and on the client for decompressing but in general there is a greater supply of CPU than bandwidth. There should also be backwards compatibility for client browsers that are incapable of using compressed web pages. GZIP and Microsoft Internet Information Services (IIS) are two methods to perform HTTP compression. An example of configuring HTTP compression in IIS can be seen in Figure 4-5:

Figure 4-5. Configuring HTTP compression in IIS

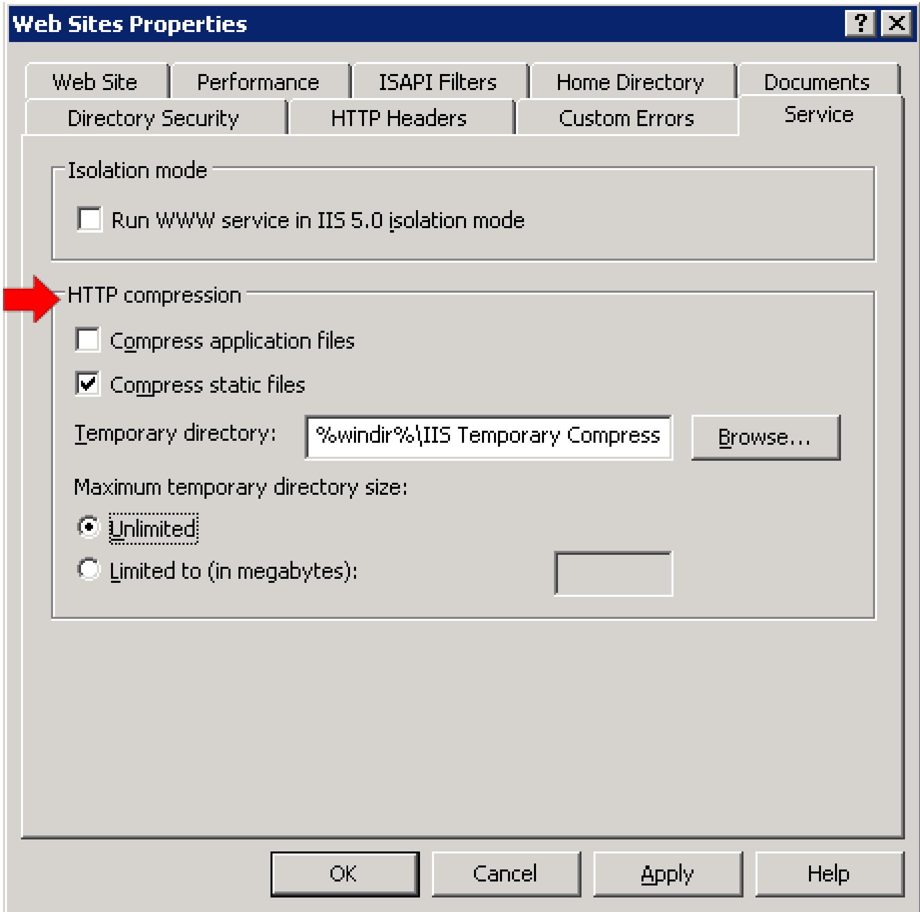

Image spriting is a technique for reducing the amount of images transferred in separate requests. Even images of one Kilobyte can consume significant bandwidth resources when being downloaded in mass quantity. To get around this, multiple images are combined into one larger image file. By downloading a single image, the overhead of making a separate request for each image is removed. When the 'sprite' image is downloaded, depending on the method used, CSS or JavaScript in the client's browser can then isolate relevant sections of the larger image before displaying on page. There is an obvious technical overhead of having to configure sprites in websites. Because of this, spriting is usually only applied to highly viewed web pages with high quantities of images displayed. An online resource that enables the automated creation of sprites can be used to simplify the process (Project Fondue 2009). An example of an image sprite can be seen in Figure 4-6:

Figure 4-6. Sample sprite image (Ignacio Ricci 2009)

On the client, the following CSS code as illustrated in Code Figure 4-8 could be used to isolate a particular image and show it separately, where “XX” and “YY” are used to isolate the image:

```
.i_anyimage
{
width: 17px;
height: 17px;
background-repeat: no-repeat;
background-image: url('/img/sprite.png');
background-position: -XXpx -YYpx;
}
```

Code 4-8. CSS sprite example

In addition to directly increasing web application performance, it is also possible to give the end user a perceived faster loading application. This can be performed by downloading additional content to the page after it has performed an initial load. Instead of the user waiting five seconds to see all of the page contents, individual items and loaded separately. This is especially important for the evolving software models of SOA where functions of an application can be performed by varying service providers. Rather than one poorly performing service effecting the whole application, the user is able to utilise the functionality that is available. This type of behaviour is typically performed by separate AJAX requests that are able to autonomously retrieve data and content to the client. As an example, Figure 4-7 demonstrates how the Pageflakes mashup website loads up initially without displaying main page content:

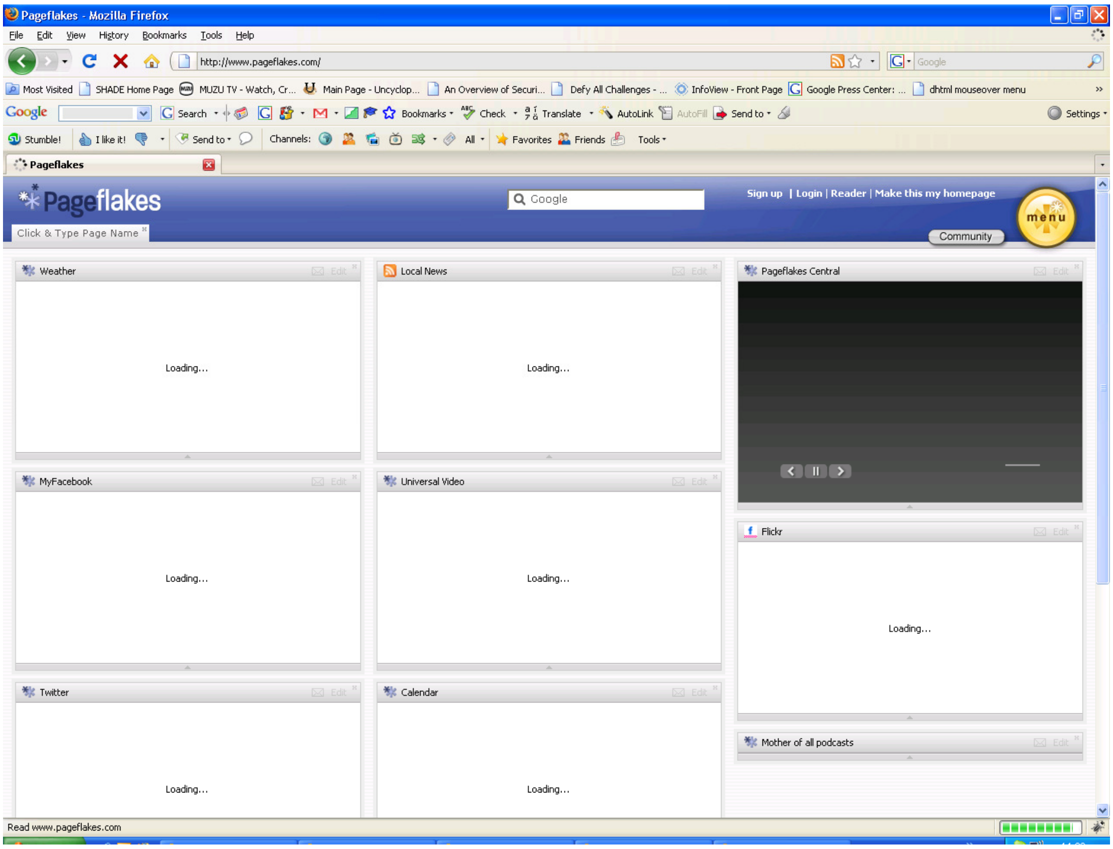

Figure 4-7. Example of deferred page loading utilizing AJAX

However as each of the individual functions become available, they are displayed independently on the page without it having to reload, as illustrated in the following Figure 4-8:

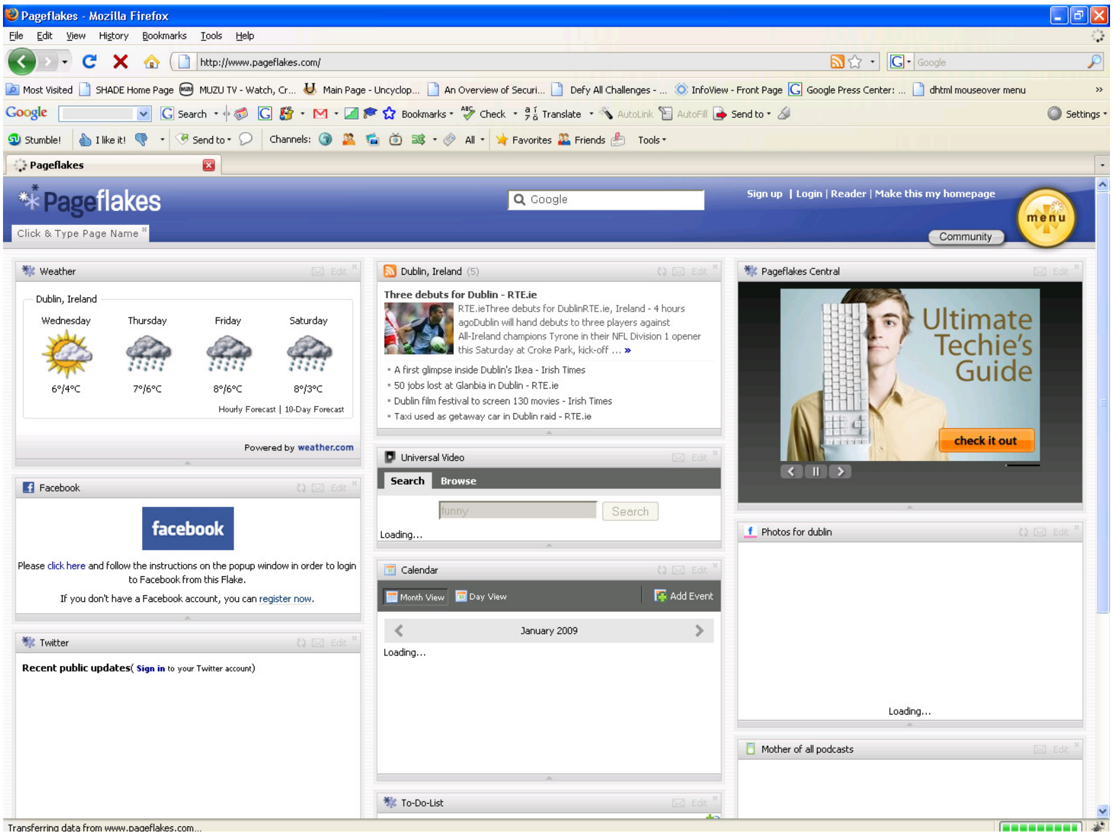

Figure 4-8. Completion of deferred page loading utilizing AJAX

Another method for page loading optimisation is to only load JavaScript and CSS files after the page has loaded. In most cases the user will not request the relevant functionality immediately, so additional files can be loaded in the background as the user is navigating the web application. This delayed loading can be performed by putting JavaScript include files at the end of the page or alternately setting a timed delay such as in Code Figure 4-9:

```
setTimeout("LoadJavaScriptFile('myroutines.js');", 5000);
```

Code 4-9. Calling a delayed content loading JavaScript function

The function to load the JavaScript would be as illustrated in the following Code Figure 4-10:

```
function LoadJavaScriptFile (src){
  var scriptElem = document.createElement('script');
  scriptElem.setAttribute('src',src);
  scriptElem.setAttribute('type','text/JavaScript');
document.getElementsByTagName('head')[0].appendChild(scriptEle
m);
}
```

Code 4-10. JavaScript Function for delayed content loading

Image downloads can also take a significant portion of page loading time. Instead of waiting for images to download when the user navigates through a website, images can be pre-fetched (via a number of methods) and cached on the client side to enable quicker loading when called. An example of pre-fetching an image via JavaScript is illustrated in Code Figure 4-11:

```
function preload(){
    var imageUrls = ["img1.jpg","img2.jpg","img3.jpg"];
    var images = [];
    for(var i=0;i<imageUrls.length;i++){
        var image = new Image();
        image.src = imageUrls[i];
        images.push(image);
   }
}
```

Code 4-11. Pre-fetching images with JavaScript

An example of using the HTML link element for pre-fetching is as illustrated in Code Figure 4-12:

```
<head>
  <link rel="prefetch" href="websitebanner.jpg">
</head>
```

Code 4-12. HTML link used for delayed content loading

Alternately a meta header can be used for image pre-fetching as illustrated in Code Figure 4-13:

```
<meta HTTP-EQUIV="Link" CONTENT="<websitebanner.jpg>;
rel=prefetch">
```

Code 4-13. META tag for delayed content loading

Various technology specific quirks and hacks exist for performance optimisation but are not relevant here due to the constantly changing software versions and capabilities. There is a free online tool (Website Optimization 2009) that offers web site optimisation analysis and gives a breakdown of where the waits are when downloading page contents. It also makes recommendations based on best practices for usability, HCI, and website optimization and is easy to use without requiring an in-depth knowledge of web development.

For the more advanced needs of large website developers, there are tools that enable more in-depth analysis of page loading composition. Firebug is a tool that plugs into the Firefox browser. It is capable of debugging all client web output and provides clear visibility to all downloaded code, provided the code has not been obfuscated. It also shows updated HTML content from the response header, such as that created with AJAX requests. This is very useful as viewing this code is not possible with the generic browser "show source" function. The following Figure 4-9 shows the Firebug plug-in in the bottom frame of Firefox, with the Google home page loaded:

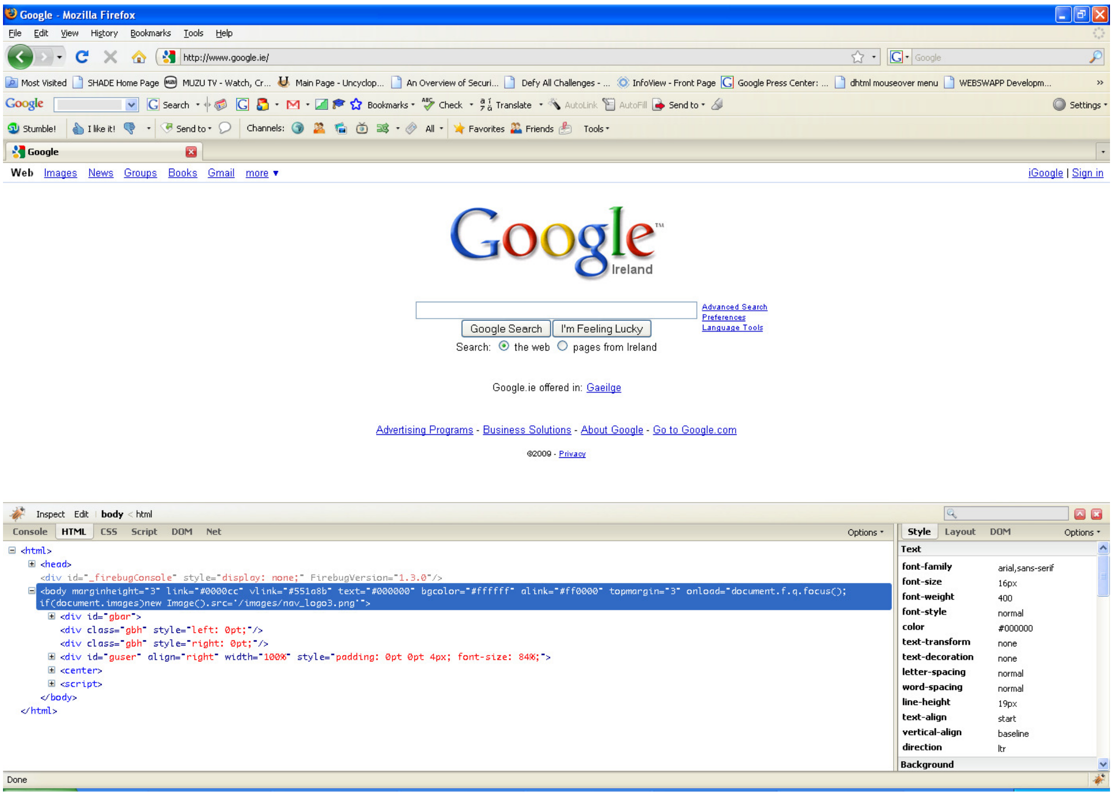

Figure 4-9. Example of using Firebug plug-in tool for Firefox

Tamper Data is another Firefox add on that enables inspection of HTTP headers and has the power to completely modify the response stream. As an add-on to Firebug, there is also a Yahoo tool called Yslow. This powerful tool extends the capabilities of Firebug to give a comprehensive analysis of each page component that downloads and provides recommendations based on design good practices that are used for high performance websites. Figure 4-10 lists the Yslow recommendations for the Google home page.

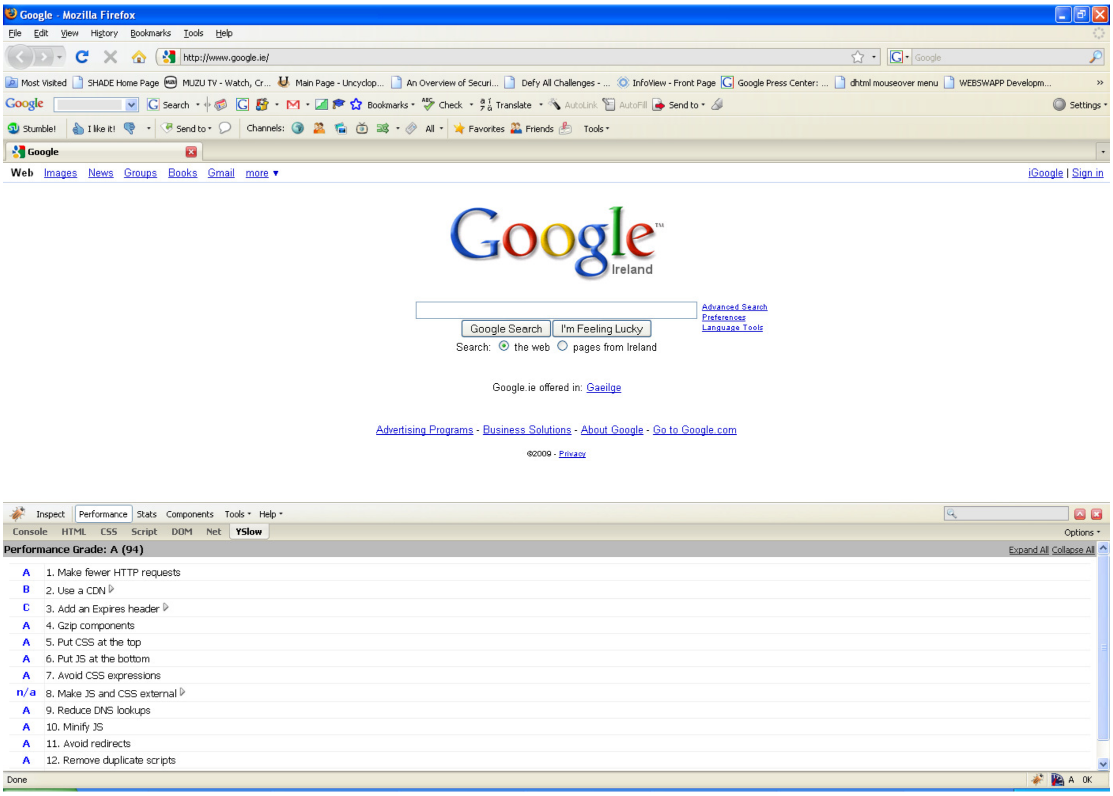

Figure 4-10. Example of using Yslow plugin tool for Firebug

When faced with the decision to use GET or POST for website requests, GET should be used to take advantage of potential cache benefits. However, POST can send data without it being visible in the URL and so is not as easy to tamper with. POST also does not have the character transmission limit inherent with GET. To perform this functionality, there is an additional overhead in the TCP/IP network transmission for POST requests. If the features of POST are not necessary, then page contents can be retrieved from cache with GET requests. Setting the request method is done within HTML for tags as illustrated in Code Figure 4-14:

```
<form id=form1 method=GET>
```

Code 4-14. HTML form request tag

None of the techniques available to increase web site performance will work if web sites are located in geographically distant locations from clients. Even if users with high speed broadband access are browsing pages from a fast web server, response times will be poor if data has to travel from the other side of the planet. To get around this limitation, Content Delivery Networks (CDN) can be utilised to ensure users are being served internet content from a local source. The CDNs are created by placing web servers geographically close to where users are requesting the site. The server selected for delivering content to a user is based on a measure of network proximity. For example, the server with the fewest network hops or quickest response time is chosen transparently to the user so that they can use the same web address. Building a CDN can be outsourced to 3rd parties (Akamai 2010) so that service providers can focus on their core business and take advantage of technology specialists.

In addition to these proven methods for web performance optimisation, there are also a number of experimental technologies being developed. One notable protocol is that of SPDY by Google. SPDY has been tested to give 62% quicker response rates than traditional HTTP (Google 2011). While this technology is not currently mature enough for use within MES, it provides a good indication of future optimisations for web applications.

4.10 Semantic Web

The Semantic Web is an extension of the World Wide Web (WWW), whereby it is possible for machines to understand web based data. Traditional web information cannot be used consistently or reliably because it can only be searched as a raw string, requiring highly advanced algorithms, such as those devised by Google when indexing web pages. Semantic web endeavours to contextualise and tag relevant web based data so that it can be properly searched and utilised by computers. Semantic Web is not a technology but a guiding set of principles. Some of the notable underlying technologies & protocols are: Web Ontology Language, Resource Description Framework, SPARQL Protocol and RDF Query Language, Rule Interchange Format.

4.11 Network Load Balancing and Web Farms

The maximum user base capacity of a web server depends on the application design and various factors affecting the amount of computing resources used. The demands of large enterprise based systems mean that a single server cannot meet performance requirements. Scaling up to meet these demands with midrange and super computing power is prohibitively costly. In response to this limitation, a number of technologies have been developed that enable scaling out, with multiple standard servers that can perform transparently as one. Depending on application needs, load balancing can be performed by high-powered specialised load balancing hardware and network infrastructure, or by cheaper and less effective software load balancing functionality on traditional server platforms. While network load balancing refers to supporting all application types, Web Farm technology pertains to the load balancing of web pages only.

Although the process of creating a Web Farm is not a specialised skill set, administration of multiple farmed servers can be significantly more demanding. As computing power grows, the requirement for web farms will diminish. There will however always be a use for Web Farms even if single server computing meets application demands. This is because of the other benefits of load balancing like the redundancy of hardware. Clustering technology is similar in that mirrored hardware is available but the application becomes unavailable temporarily as a passive node is taking the place of the failed live node. In comparison to clustering, farming enables a transparent failover so that the client is unaware of the failure of a participating node. In this manner, rolling server upgrades can also be performed.

4.12 Database connection pooling

The concept of database connection pooling is not specific to any particular database or development platform. As can be seen from the following Figure 4-11, the connection pool is a cache of database connections maintained by the web server and database so that the connections can be reused when the database receives future requests for data:

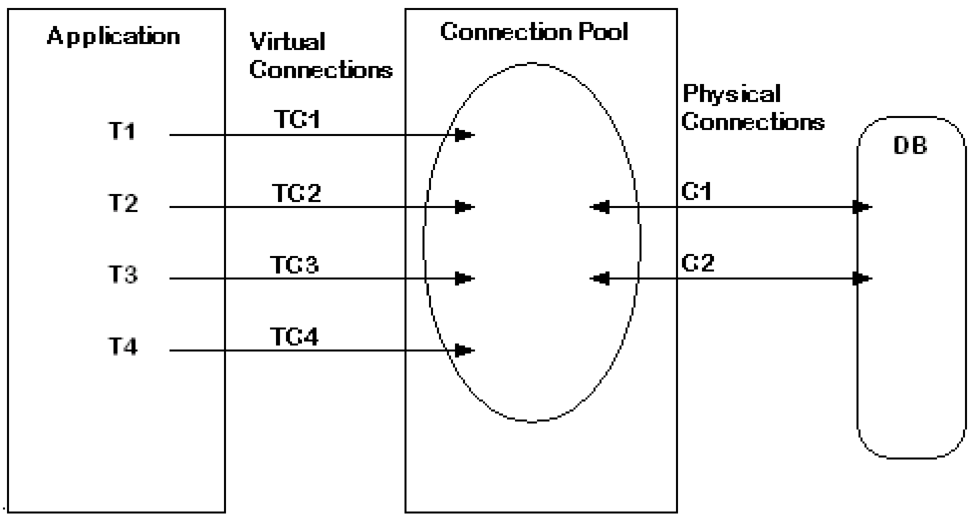

Figure 4-11. Database connection pooling (Oracle Documentation 2011)

This technology leads to significant performance gains for the web but is not limited solely for use on web architecture. Due to the stateless nature of web applications, each page load makes a separate request from the database. For database intensive websites, this constant opening and closing of connections would consume a large amount of computing resources. Most modern development languages inherently cater for connection pooling however. Code Figure 4-15 illustrates how a .NET connection string connecting to an Oracle database only needs a parameter to set the pooling configuration:

```
Max Pool Size=100
```

Code 4-15. Database connection pooling parameter for .NET

4.13 New programming languages such as ASP.NET & Java

Prior to ASP.NET and Java, dynamic content web programming was limited to CGI scripting with languages such as PERL and server side scripting with Active Server Pages. While the older languages were powerful, they were not suited to building fully functional web applications and lacked object-orientated capabilities. With the introduction of Java and ASP.NET, the delineation between desktop development and the web has blurred. A lot of the techniques to overcome the statelessness of HTTP are transparent to the user. The languages are hosted in extremely rich development environments that enable drag and drop of GUI controls that are rendered out as HTML and JavaScript on the client side. Because the languages are the same for desktop and web development, considerably less work is required to transition desktop apps to the web. Good design practices such as MVC prescribe separation of business logic and GUI elements which make it easier to port applications to a different platform. As newer development languages evolve, continually less code is required to perform tasks, with intrinsic libraries becoming more comprehensive.

The organic nature of this growing functionality is not without a downside however. ASP.NET for example can often have five ways to perform a single task. It can be confusing to developers as to which method is the most optimal and which ones are in the process of being depreciated. Java is also similar whereby multiple variants of a package provide the same functionality. In addition, packages can be inherited from in a daisy chain manner so that it becomes unwieldy to debug when tracking errors.

4.14 Web caching

Web caching is the storing of frequently used objects closer to the client through browser, proxy, or server caches. By storing these objects closer to the client, unnecessary HTTP requests and DNS hops are minimised. This reduces bandwidth consumption and server load with resultant improved response times. The objects concerned can be anything from HTML pages, web service calls, or result from some type of computation. A simple HTML tag to tell browsers to cache a web page is shown in Code Figure 4-16:

```
< meta http-equiv ="expires" content="Mon, 22 Sept 2009
13:00:00 GMT">
```

Code 4-16. META tag for page caching

Not all web content is suitable for caching. For example pages that serve dynamic content may want to turn caching off as illustrated by Code Figure 4-17:

```
<meta http-equiv="Pragma" content="no-cache">
```

Code 4-17. META tag for no page caching

Depending on how a computer is connected to the internet, web content may be cached on the web server, internet proxy, gateway or clients browser cache. There are also technology specific methods to cache objects. For example, a weather widget that is on the front page of a website can be set to expire every 30 minutes, and could be set in ASP.NET as illustrated with Code Figure 4-18:

```
System.Web.HttpContext.Current.Cache.Insert('weatherwidget',
objWeatherWidget, Nothing, DateTime.Now.AddMinutes(30),
TimeSpan.Zero)
```

Code 4-18. ASP.NET caching

There are also 3rd party cache implementers that enable web application caching without having to modify any application code (Squid Cache 2010).

4.15 Virtual private networks

Transmitting data over the internet is inherently insecure. The following Tracert command output as illustrated in Code Figure 4-19, is generated from an online tool (Vistatec 2011) and highlights how many hops and packet transfers are required to reach the Google homepage:

```
traceroute: Warning: www.google.com has multiple addresses;
using 216.239.59.147
traceroute to www.l.google.com (216.239.59.147), 30 hops max,
38 byte packets
 1  vlan120.switch.deg.vistatec.ie (85.159.16.65)  1.337 ms  1.174
ms  2.113 ms
 2  ge0-1.router.deg.vistatec.ie (85.159.16.25)  0.446 ms  0.255
ms  0.388 ms
 3  xe-0-1-0-119.dub20.ip.tiscali.net (77.67.66.213)  0.685 ms
1.052 ms  1.032 ms
 4  xe-5-0-0.lon10.ip.tiscali.net (89.149.187.21)  12.281 ms
11.955 ms xe-3-3-0.lon10.ip.tiscali.net (89.149.187.17)  11.982 ms
 5  74.125.51.173 (74.125.51.173)  11.963 ms  12.188 ms  11.925
ms
 6  209.85.255.175 (209.85.255.175)  11.994 ms  12.472 ms
12.050 ms
 7  209.85.250.216 (209.85.250.216)  14.014 ms  13.806 ms
13.802 ms
 8  64.233.174.185 (64.233.174.185)  14.173 ms  14.234 ms
72.14.232.241 (72.14.232.241)  14.201 ms
 9  216.239.49.114 (216.239.49.114)  16.481 ms  20.599 ms
17.233 ms
10  gv-in-f147.google.com (216.239.59.147)  12.726 ms  12.907
ms  14.570 ms
```

Code 4-19. TRACERT output

As illustrated, internet traffic is routed through multiple routing devices and servers to reach its destination. Depending on the destination, the route can go in different paths. Therefore, sending unsecured data can give rise to leaked information or malicious code injected by compromised intermediate paths to the requested destination. As business models are evolving, increasingly more systems are designed to openly share part of their systems to the outside world. SOA and other models mean businesses are sharing data and performing functions. Traditionally businesses set up extranets with dedicated access configured to let specific IP addresses access company networks in a secure manner. In comparison, Virtual Private Network (VPN) technology enables a much cheaper way to implement the same thing and also has a much greater degree of flexibility. VPN was developed to enable secure access from point A to point B over potentially unsecure channels. As illustrated in Figure 4-12, it essentially creates a secure tunnel across a public network through the use of a suite of underlying technologies and protocols:

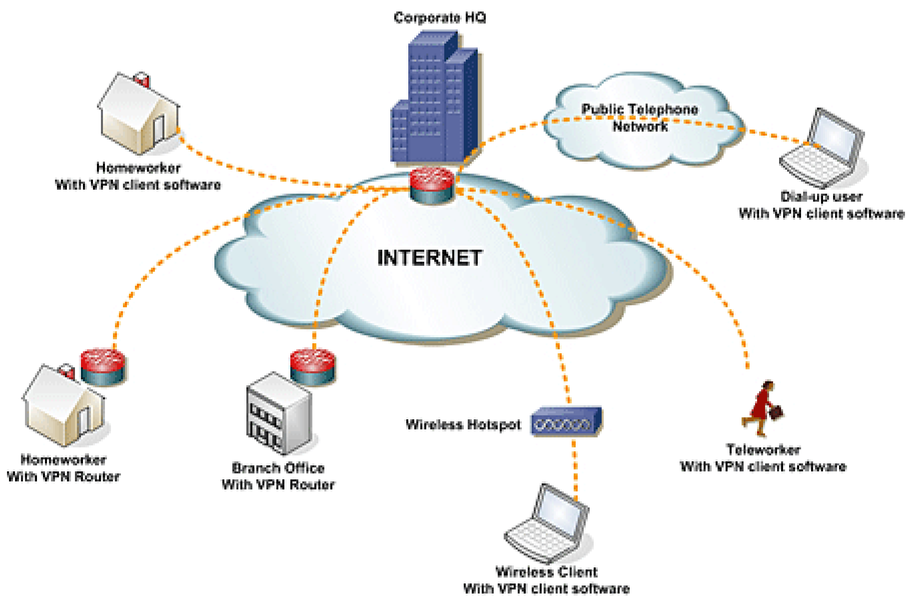

Figure 4-12. VPN overview (Networked Elements 2010)

4.16 Grid computing

Grid computing entails the use of several computers simultaneously working on a single task at the same time. To form a grid there can be two or more nodes participating together. Grids are mostly for performing heavy scientific calculations or technical problems that requires a great number of computer processing cycles. As in all distributed systems, the service that grid computing delivers is fault tolerant to individual nodes failing and is also transparent to the end user. The benefit of using a grid is that they can scale up to the processing power of a supercomputer but can be done at a fraction of the cost by utilising low cost hardware. In the context of the web, a form of grid computing has been developed to deliver highly scalable and fault tolerant database back ends for very large websites. Oracle provides a database grid architecture called Real Application Cluster (RAC), as illustrated in Figure 4-23:

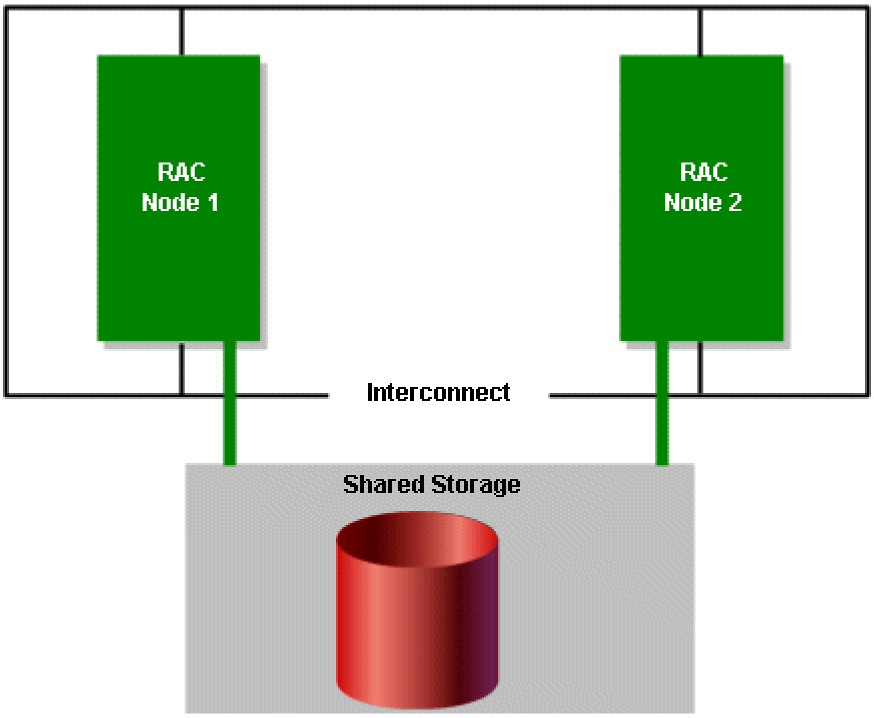

Figure 4-23. RAC overview (Oracle Documentation 2010)

With databases being a very critical element of web infrastructure, grid computing is another tool to maintain high performance and uptime. As with all technology, it does not operate in isolation and requires the use shared disk storage systems such as a SAN to facilitate multiple servers hosting the same database (Hart, Matthew and Jesse, Scott, 2004). While this can become a powerful method to maintain a highly available and scalable database, it can be quite complex and expensive to implement and administer. In Oracles version of grid computing, RAC can be very difficult to administer and requires highly specialised staff to maintain it. Also costs can be prohibitive with a three CPU server costs €69,000 to initially buy RAC technology and a €15,000 annual support fee (Oracle Technology Global Price List, 2008). To maintain a ROI the cost of database downtime has to be a significant factor.

4.17 Web Services

Web services are similar to standard API's in that they can be plugged into code to perform various actions. They differ in that standard API's usually conduct the processing locally whereas Web Services are usually processed on a remote host over the Internet. The Web Services Description Language (WSDL) language enables web services to be published and available for use by clients. Through defined interfaces, web services can be used to process requests on a remote host and return data. The underlying framework for web services can be either SOAP or REpresentational State Transfer (REST): both of these protocols utilise XML and HTTP at a lower level. Web services are extremely powerful because they can be implemented seamlessly in applications without them having to know anything about the remote client that services the requests. Because it is a defined standard, they also make it easy for dissimilar systems to interoperate. Web services currently available range from the Department of Homeland Security Threat Advisory information to weather updates in users locality. Most modern development environments enable web services to be plugged into custom code without any further development. This enables remote calls to be performed transparently as if they were being processed locally.

4.18 Service Orientated Architecture

SOA is a business model that extends from web services whereby systems can be integrated and tailored to meet individual companies' needs. These services can communicate with each other by either simple data passing or also involve two or more services coordinating some business activity. Historically, systems had to be purchased in their entirety regardless as to whether all of the functionality was needed. SOA enables customers to only pay for software services and functionality that they need. They can also use "best in class" services from different vendors, rather than being tied into one particular service provider. Through the use of standardised communication protocols, SOA also enables integration of dissimilar and hybrid systems. Figure 4-14 outlines the basic flow of requests SOA:

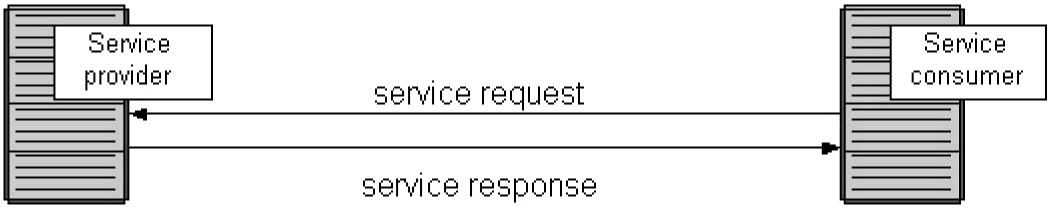

Figure 4-14. Basic SOA flow overview (Service Architecture 2011)

A downside of this technology is that customers are often locked in with particular service providers once their architecture has been implemented. Service providers tend to offer very low initial fees that increase dramatically as customers become locked in with the technology. As SOA products become much more flexible and customisable, alternative providers become harder to find or more expensive to configure. This makes it costly to change so service providers have the potential to increase their fees. As a long term strategy, price guarantees and set limits should be obtained when utilizing SOA services. At the outset, plans should be devised for an exit strategy to transfer to another preferred service provider if needed.

4.19 Cloud computing

Cloud computing is a relatively new paradigm whereby potentially unlimited computing services can be provided over the internet. Many cloud computing services have adopted the utility computing model, analogous to traditional utilities such as electricity and gas, where clients can pay on a subscription basis. It is a further extension of grid computing and SOA so that as much or as little computing power required can be obtained via “the cloud”. In this manner, customers do not own the infrastructure and simply rent services. This business model can be an enabler for smaller businesses to make use of super computing power for a fraction of the cost. Computing capacity rises dramatically, as customers do not have to engineer for peak loads and can easily scale up if required. The relevant service providers can become specialists in their distinctive field and leverage the benefits of economy of scale.

Similar terminology used is utility computing and Software as a Service (SaaS). Some notable providers of cloud computing are Amazon, Google and Salesforce.com. A lot of the initial cloud computing providers typically performed specific tasks such as data storage or processing, but as this business model is maturing more providers are delivering fully functioning business solutions. This type of business model is not suitable for customers that have highly sensitive data that cannot be accessible to a 3rd party.

4.20 Clustering

While clustering technology is not used solely in web architecture, its use helps provide a much more resilient infrastructure for web applications. Redundancy features include hardware and software components, including applications. In the case of failure of a hardware or software module within a server, users are shifted from the failed server another server in the cluster (Clark 2003: 12). The collection of several server computers into a single unified cluster makes it possible to share a computing load without users or administrators needing to know that more than one server are involved. For example if any resource in the server cluster fails, the cluster as a whole can continue to service user requests by using a resource on one of the other servers in the cluster, regardless of whether the failed component is hardware or software (Microsoft Windows clustering white paper 2003:5). One such example of this clustering middleware software is that of Microsoft Clustering Services. As illustrated in the Figure 4-15, only one server can be active at any one time:

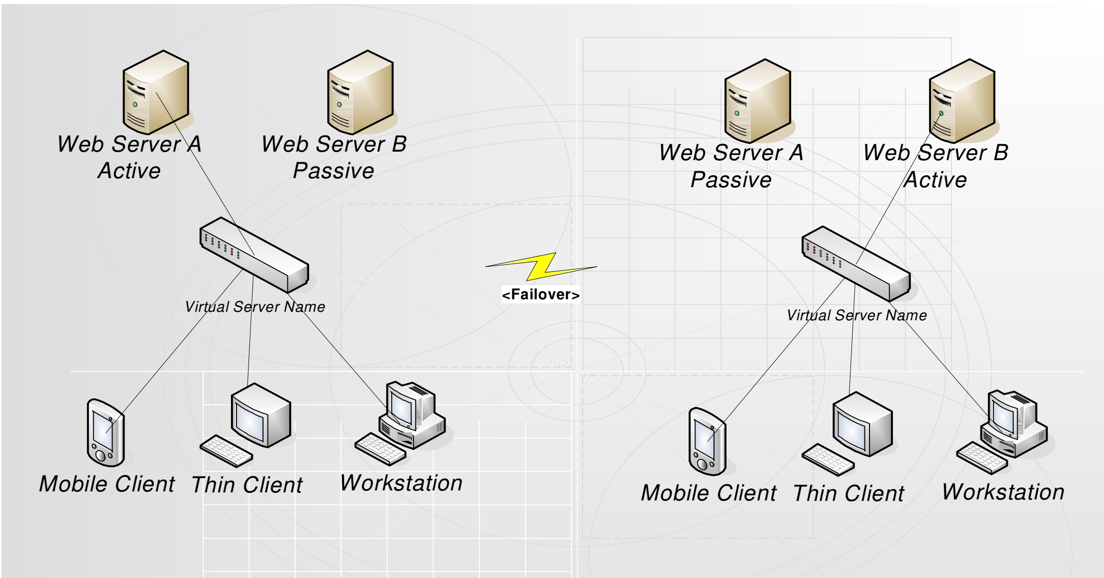

Figure 4-15. Use of clustering for provision of fault tolerant service

In the event of a hardware/software error on the active node, the cluster middleware software will activate the passive node to become the active one – this is called performing a failover. In this manner, the service can be resumed without manual intervention. While there is an overhead of always having one or more unused servers lying idle, clustering allows a quick service recovery. For critical systems cluster technology is essential.

4.21 Storage Area Networks

Due to the virtual design of server clusters, Storage Area Networks SANs must be used to provide sharable disk storage that can be accessed by multiple servers. SAN technology enables extremely fast disk access with automatic load balancing of frequently used disk sectors to maximise performance. This feature combats performance issues where a few disk heads are being over burdened with constant activity while other disks are lying idle. SAN architecture often involves the replication of data to a remote geographic site for the purposes of disaster recovery. In the event of a catastrophic failure, no data is lost because the data is replicated block by block to the mirrored storage. In regulated environments where data loss is not acceptable, SANs help businesses to maintain compliancy. SANs are now recognised as the preferred solution for fulfilling data-storage requirements for mission critical systems as they far exceed the capabilities of traditional storage access methods (Clark 2003:1).

4.22 Web analytics

The development of a web site does not necessarily end when the site goes live. Tweaking will often be required to ensure the application is performing as expected. To obtain automated feedback, web analytics are used for collection, analysis and reporting of web data. The data can then be used to understand how the site performs for users without the need for manual feedback. The analytics are typically either incorporated silently into each web page or compiled from the web server logs that are generated. Both methods have pros and cons that make them more suitable depending on the business model and how the web is being hosted. All data is reported back to a central repository, recording items such as client browser type, IP address, page accessed, load time, errors raised, visit frequencies, the sites that link in to the current web site etc.

Compiled information is surprisingly powerful and can show things like slow loading web pages, popular pages & sections, hits per hour/page, geospatial breakdown of users and other advanced segmentation. Often unhandled errors are unknown on web pages as users either workaround or stop using the website. With analytics it is possible to get detailed and useful information regarding unhandled errors such as error frequency by IP address and page navigation flow.

Web analytics even has the potential to perform Intrusion Detection System (IDS) functionality by highlighting attempted attacks. Some of the common hack attempts can be determined from repeated abnormal scanning of particular pages and rogue IP addresses. Other detectable attacks can come from high amount of page not found errors (401) or CGI execution errors (404). Most website hacking attempts try to uncover vulnerabilities in URLs that lead to SQL Injection attacks on the website database. This type of attack leaves the whole infrastructure wide open for malicious behaviour. Therefore it is critical that publicly accessible websites such as MES systems not only have its attack surface minimised but are also frequently monitored for attempted attacks and actual breaches. Data integrity is of the utmost importance because any leaked or missing data regarding manufactured product could give rise to business failure. Out of necessity, governing regulatory bodies stipulate that product genealogy is fully traceable. Tampering with configuration data stored in a database could lead to defective product being delivered to customers. With growing sophistication and numbers of web based security attacks (Symantec Internet Security Threat Report, 2008), more effort needs to be put into securing websites. In the case of pharmaceutical products, defective product has the potential to kill its customers. The Boston Scientific heart stint manufacturing issue is an unfortunate example of how lethal incorrectly manufactured product can be. The surgical company reportedly caused the death of three patients and another 47 injuries because of manufacturing glitch that resulted in faulty stints being used in surgery (BusinessWeek, 2004).

Microsoft has an extremely versatile tool called Logparser that can among other things, perform web server log analysis to interrogate and store web server log data. The tool can either be integrated into Common Object Model (COM) compatible development language such as VB or run from the command line.

Google analytics is an alternate page tagging method of website analytics. It has a variety of ready-made reports and can be easily integrated into an existing website with minimal code.

4.23 Conclusion

The underlying specifics of technology become quickly outdated due to the rapid aging of technology. Each section in this chapter refers to an area of technology that is under continual growth and enhancement. However it is expected that the underlying technology concepts will remain relevant for a much longer duration.

Upon review of this chapter, and in conjunction with the literature review, the potential benefits of convergence in different disciplines have become apparent. By combining technology with other areas, MES can gain advanced capabilities and continue to evolve. For example MES can utilise the cognitive psychology principle of Millers magic number. Through the use of applied AJAX techniques, the cognitive burden of the end user can be reduced, resulting in an easier to use application with less user generated errors.

A common theme can be seen with each of the technology aspects covered in this chapter. They all contribute to creating software functionality, to enhancing system responsiveness and performance. Adherence to these aspects give a good guide for the future adoption of technology. Unless there are quantifiable business benefits, the use of a particular technology cannot be justified and in fact, may have the downside of over complication and creation of inflexibility.

5 Industry standards and regulations

There are a number of different architectural models and functional requirements for MES design. Depending on the relevant industry sector and level of automation sophistication, there can be a number of capabilities that MES needs to support. These capabilities generally focus on what data is recorded and the type of controls around how product is manufactured. Some requirements are good practice guidelines, whereas others are mandatory and must be implemented if the MES application is to be used in particular industries.

5.1 Regulatory governance and compliance

There are varying industry regulations that MES must be compliant with. These requirements are non-negotiable and must be adhered to so that the company can sell manufactured product. The rationale behind this mandatory adherence to regulation is to ensure software behaves as expected and gives a reasonable degree of certainty that product manufactured under the control of MES is safe and performs as purported to. It is not possible for a product to be always perfect when reaching the customer. Because of potential product quality issues, another important factor for compliance is to be able to identify batches of product that have been identified as faulty after release to the customer. This necessitates recording all the details of product manufacture and the materials used to make them. When all of a product's genealogy is correctly recorded, faulty items can be isolated and withdrawn from the market, and the exact cause of failure can be ascertained.

Regulatory governance on MES functionality is dependent on industry type and the geographic region of its intended usage. One of the more common regulations called "21 CFR Part 11" is for any company (or subcontractors) that are either based in or do business in America. This regulation is set out by the Food and Drug Administration (FDA) of America and prescribes how MES should record electronic records for access control and digital signatures when signing for performing particular actions. The Sarbanes Oxley act of 2002 (SOX) is another of the more notable regulatory influences on MES design. While SOX was introduced to combat financial fraud, there are some elements of MES that must comply with the control and security prescriptive. The following highlights the regulatory bodies that software may need to comply with, depending on where it is being used and the nature of the business: ISO 17799, Cobit, Sarbaines-Oxly, HIPPA, Payment Card Industry, GLBA. NERC, PIPEDA (Symantic 2006).

With particular focus on MES, the following table lists some of the governing bodies MES must adhere to:

Standard	**Regulatory body**	**Applicable industry**	**Primary focus**
Sarbaines-Oxly (SOX) Act 2002	American legislation	Any company (or subcontractor) that is either based in or conducts business in America.	Prevention of financial fraud
Irish Data Protection Act 1998	EU directive	Any Irish company that saves personal information	Data protection
PIPED Act 2000	Canadian legislation	Any company (or subcontractor) that is either based in or conducts business in Canada.	Data protection
Food and Drug Administration (FDA)	American department of health	Any company (or subcontractor) that deals with food or medicine, and is either based in or conducts business in America.	Product quality

Table 5-1. Regulatory bodies governing MES

5.2 Standards adherence

While regulations are mandatory and specify MES outcomes, industry standards are more concerned with software functionality and design. These standards are not mandatory and purely specify guidelines on how MES should function and connect with interrelated systems. This prescription aids in communicating with other products and enables seamless integration in to the business supply chain with a lower potential for complications. It also helps in reducing cost by running production in a streamlined manner. One of the fundamental MES standards is the internationally recognised ISA-95 which specifies how MES should integrate with various interrelated systems in quality, finance, planning etc. (Gifford 2007: 20 - 22*)*. Another related standard is ISA-88 for batch processing. This specifies a common language for the design, operation and supervision of batching systems and attempts to prescribe enough flexibility so that it caters to the varying needs of different production environments and workflows. While the ISA body prescribes the core design of MES, there are also governing bodies that recommend functional designs such as ISO, ANSI, OAGIS, IEEE etc. The following table 5-2 lists some of the standards that provide guidance on MES software design and also the design process itself. While most of these standards are not specific to MES design, they contribute in creating a robust, future-proofed product:

Standard	Regulatory body	Primary focus	Latest revision
Control Objectives for Information and Related Technology (COBIT)	ISACA	IT governance framework	2012
ISO/IEC 27000	ISO/IEC	Information security management	2009
MESA Model	MESA	Manufacturing plant business operations	2012
SCOR	SCC Organisation	Supply chain development	2008
ISA-95, ISA-88	ISA	MES integration and batch processing	2005
ISO 9001:2008	ISO	Quality management systems	2008
DiRA	Microsoft	Reference architecture for discrete manufacturing	2011

Table 5-2. Applicable standards for MES

5.3 Conclusion

The compliance landscape is complicated. There are numerous regulations and standards in addition to those mentioned in this chapter. In general, a lot of these influences are good practices and provide benefits for MES to help improve its capabilities. There can also be drawbacks to the implementation of other more restrictive features. While all regulatory directives must be implemented, standards should only be utilised where they do not negatively impact the intended MES implementation. If nessesary, the gains from standards adherence should be evaluated so that extensive work is not conducted purely to implement a salient feature that may never be used.

While it may appear to counteract the benefits of standards, sometimes their implementation may need to be modified for suitability so that they fit within a particular MES design. Although this may appear as bad practice, it can be necessary on occasion because of how standards are devised. Often, governing bodies can take an inordinate amount of time to create and modify standards. HTML 5 for example has been under development for a number of years and is still not completed as of mid-2011. If web browser developers were to wait for it to be fully ratified, web advancement and innovation would be greatly hampered. Because of this, the standard has been implemented ahead of its completion. Another example of this would be the wireless protocol 802.11n standard for 300+ Mbit/s wireless communications. That has also been implemented and tweaked by manufacturing vendors due to the fact it has been in draft for so many years.

6 Results and Discussion

This chapter highlights the notable observations from industry standards and regulations review, web based device communication prototyping, MES analysis, MES related surveys and technology review.

6.1 Industry standards and regulations

Where possible, industry standards should be incorporated into MES development to make use of proven design models. Implementation of standards in MES design may seem unimportant when in the development phase, but the benefits become apparent when the need arises for integration between MES and adjoining systems. There may be scenarios where it is appropriate to omit certain features prescribed by standards. These omissions may be necessary when different standards conflict. Omissions may also be necessary for ancillary features that are very difficult to incorporate or develop and have no perceived future benefit.

The intended industry designation for MES under design should be clearly defined out from inception. This will enable the relevant regulatory requirements to be pre-determined, facilitating the appropriate MES design. Retrofitting functionality after a system has been built can be costly and be problematic.

6.2 Web based device communications

The capability to communicate to industrial devices via different web browser types was fundamental to this research. The coupling between MES and industrial devices increases as production methods becoming more advanced. Without this device communication ability, a web based MES would be severely limited in where it can be utilised. Because no COTS MES currently has this capability, it was necessary to prototype this functionality as proof that it was possible. Lee et al. (2007) subsequently developed a prototype of a Java based framework that could perform cross browser and cross OS device communication for MES. As part of the prototype, RS232 communication was conducted on an industrial printer from different browser types and versions.

6.3 Technical Survey Findings

This section highlights the significant findings of the technical survey. It was aimed at contributors to the design of MES either in the development, architecture or testing functions. While the participant opinions may not be representative of typical industry and also be focused on one particular proprietary MES product, they nonetheless give invaluable insight into MES design concepts. Not all of the survey findings are covered in the following sections as some questions were Bausch + Lomb specific with no relevance to the research. In addition, the geographic location of manufacturing plants has not been included with actual findings due to the potentially sensitive nature of the information.

6.3.1 Choices on competing design attributes

In MES design there is a conflict between two opposing aspects. In order to meet the challenging demands of business logic, MES needs to be highly configurable so that it can be quickly customised as required without having to modify code which would result in extensive validation on anything that could be affected by the changes. Unfortunately this high degree of configurability comes at a cost as the application becomes much more complex, needing extensive configuration when deploying to new sites. Also the process of code modification becomes an arduous task, with the need to perform extensive testing on all possible configuration permutations. When asked to choose between these two aspects, the technical survey participant list clearly preferred a high degree of configurability, as illustrated in the following Figure 6-1. A small percentage was even willing to completely sacrifice having an easily reusable generic product for a highly configurable product.

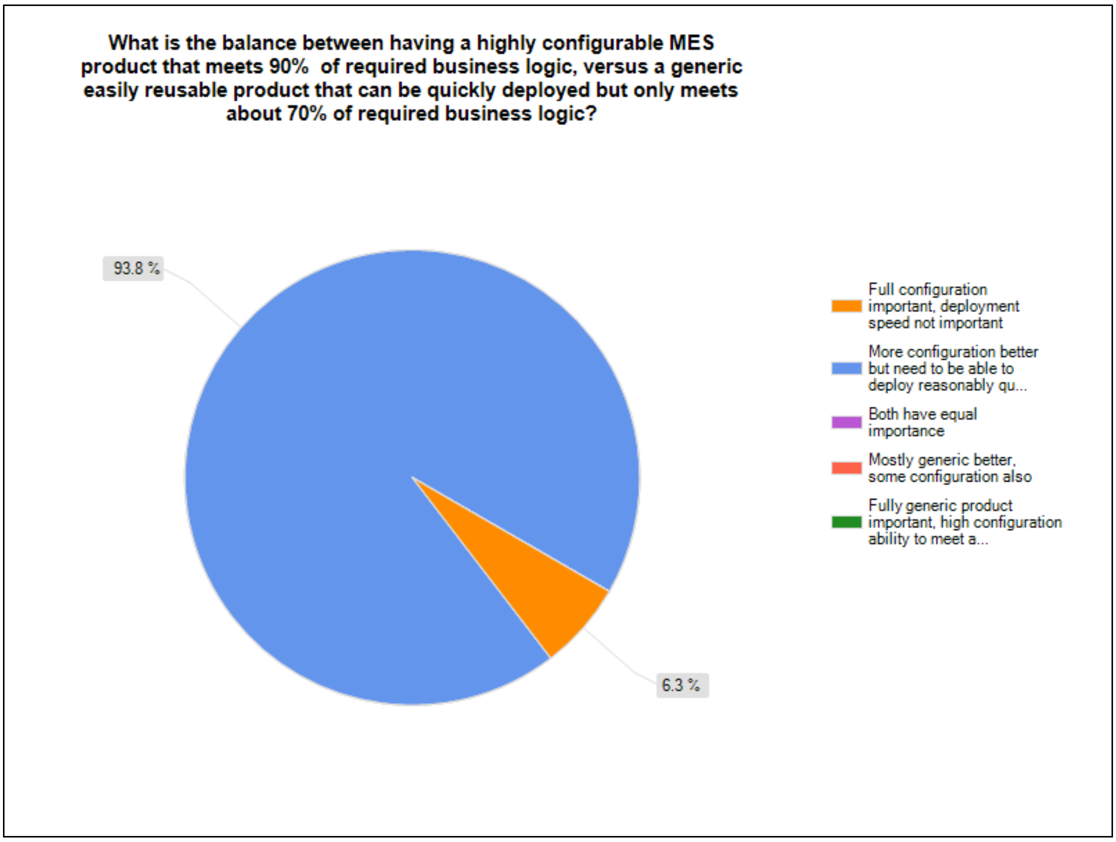

Figure 6-1. Balance of configurable MES versus generic product

The findings were not so clear cut when asked to choose between speed of application response versus facilitation of business logic. A significant percentage of participants chose the requirement for an even balance, with another 25% slightly favouring response times. Based on open ended feedback, a number of participants indicated that this would have to be determined on a case by case basis. Unfortunately when dealing with a single configurable product, priority can only be given to one aspect. On further analysis of the survey data, it became apparent that different professions give a slightly different priority, with development staff choosing slightly more facilitation of business logic and people more aligned with production having more emphasis on response times. As a result the findings from this question are inconclusive as can be seen from the following Figure 6-2. While production requires greater facilitation of response times, they are also the drivers of business logic requirements.

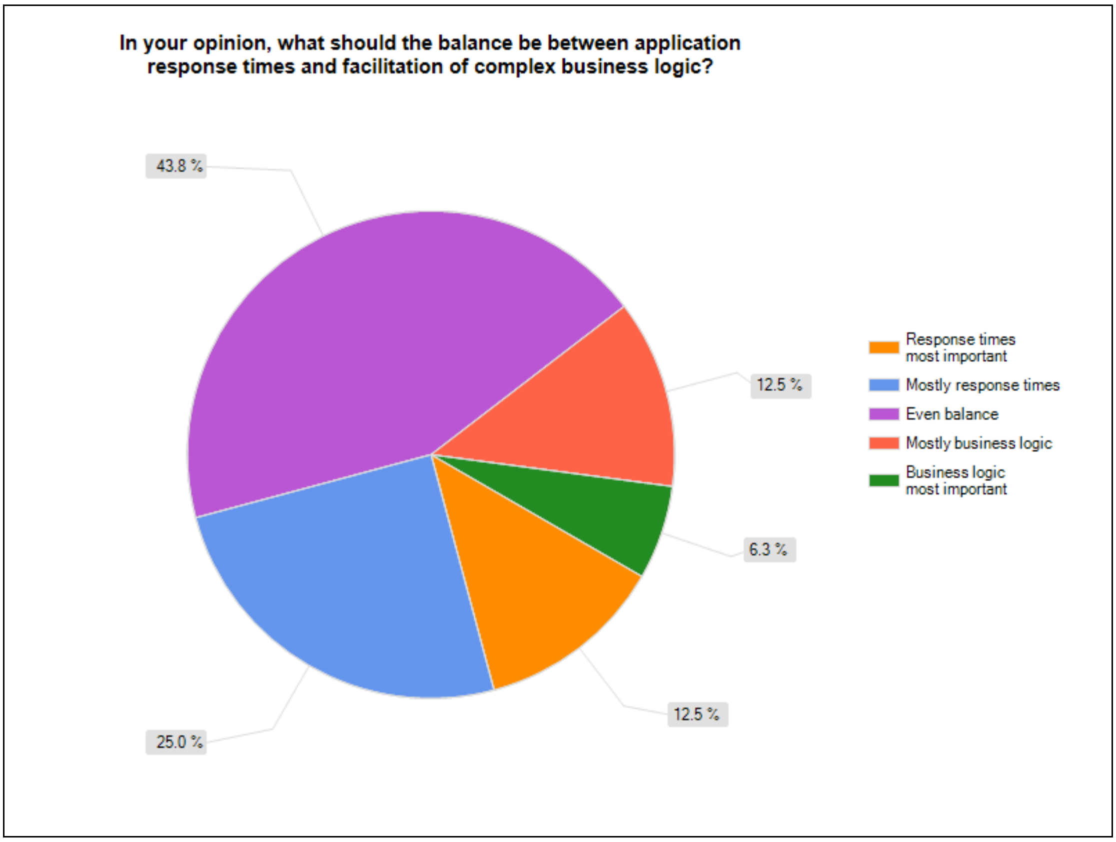

Figure 6-2. Balance of response times v business logic

This conflict of requirements is also present for the core MES design factors, whereby satisfying one requirement minimised the amount of support given to another. In theory, all requirements can be equally satisfied. In reality however, constraints such as time, cost, scope and quality impose restrictions that diminish the feasibility of all requirements being implemented. This concept of attribute trade-off is well defined by the project triangle model in the following Figure 6-3:

Figure 6-3. Project triangle model (MIS Services 2010)

As a means to force participants to sacrifice opposing design factors, a ranked matrix was devised. Various design factors could be ranked from one to nine, where one is most important. The key to this question was that participants could not give the same ranking to more than one factor.

Looking at the overall results of the attribute ranking, as can be seen in the following Figure 6.4, it became clear that handling complex business logic versus response times are seen as equally important.

At the forefront of the ranked stacked bar chart, compliance to industry regulations is deemed to be the most critical factor. This corresponds with the fact that MES must be fully compliant to the regulations that govern particular markets it's being used in. At the other extreme, ability for product localisation was seen outstandingly as the least important. Interestingly, geographic sites that had poor English comprehension gave a higher emphasis on localisation. This indicates that generally, sites already have a localised MES package and are satisfied on that product attribute. End users could not operate an MES package in a language they do not understand; therefore although localisation was ranked the lowest, it is still critical for MES.

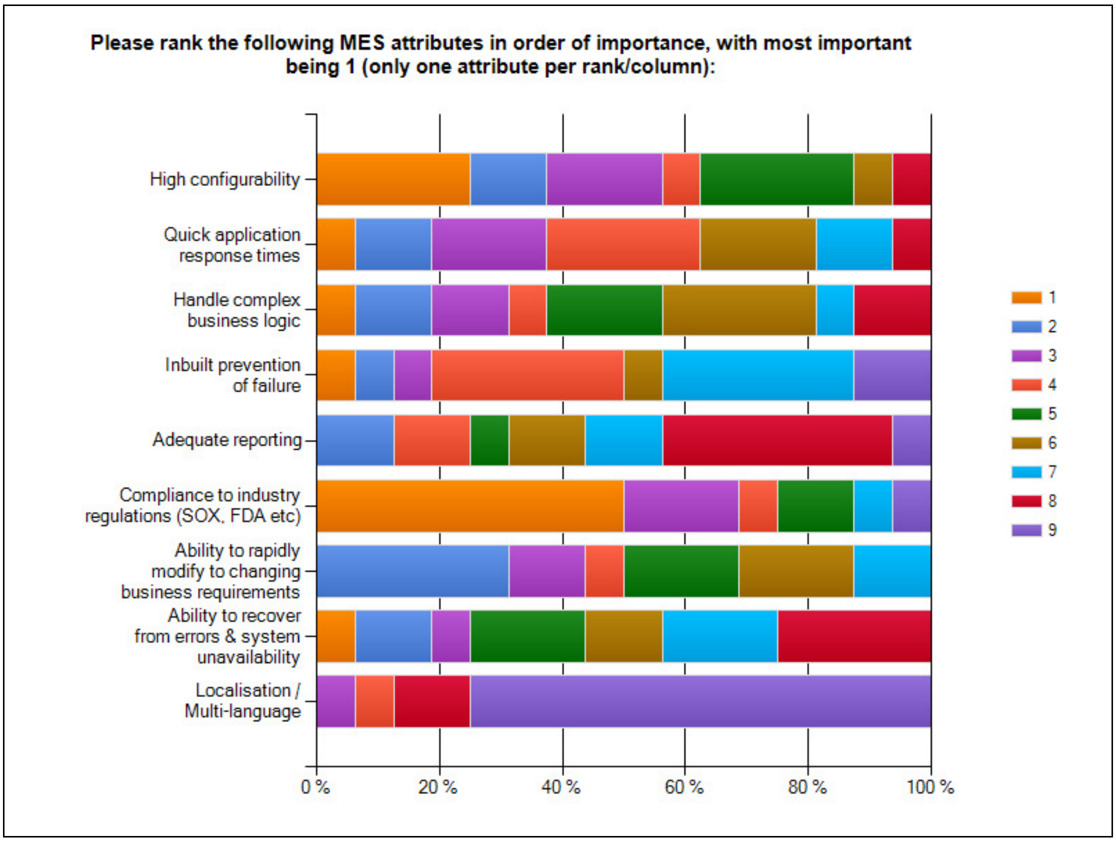

Figure 6-4. MES attribute ranking

6.3.2 Perceived reliability of remotely hosted web based MES

The views on a remotely hosted MES solution were not surprising. Due to the critical nature of MES, even minor issues could cause significant financial loss. As can be seen from the following Figure 6-5, all areas scored low with poor to neutral being the major proportion. In the early survey draft, this question pertained to web based remote MES hosting. The Bausch + Lomb survey team requested that the reference to web was removed due to it being perceived as not viable at all. This view represents industry opinion that the client-server model is the only reliable architecture for MES. There is still resistance even though the web based model has been proved by various remotely hosted applications such as Salesforce CRM.

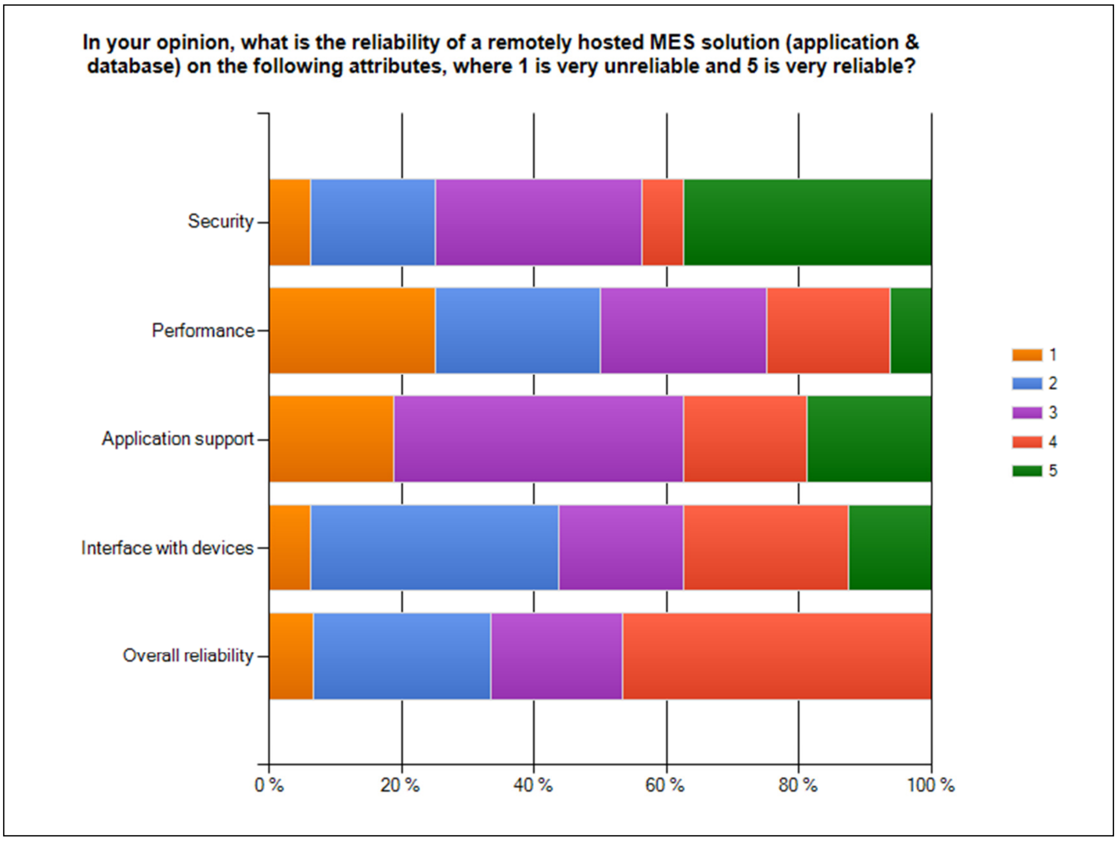

Figure 6-5. Remotely hosted MES reliability

Feedback from a question asking for potential issues surrounding web based MES highlighted concerns over application response times and capacity for rich functionality via the web. As highlighted in the literature review, web technology has progressed to the point of being able to match desktop functionality and deliver reasonable levels of performance. Other participant concerns related to questionable ability to perform reliable web based device communications. Additionally, other participants envisaged that maintaining code for device communication in web based architecture would be overly complicated. These concerns have been addressed by the research paper published by Lee et al. (2007) under the title "Web Browser Device Communication for Shop floor Information Systems". Further concerns were raised with plugins, toolbars, patches etc. which can and often do cause problems viewing web content in Microsoft Internet Explorer. These problems can be mitigated against by ensuring the web application is cross browser compliant, enabling use of alternative browsers if needed.

Performance degradation due to internet and ISP issues was also reported to be a potential issue for web based MES. These performance issues could be caused by a variety of issues and all have the potential to cause a loss of service. This is a valid point but the risk can be minimised by choosing a reliable ISP and ensuring an adequate Service Level Agreement (SLA) is in place. Other areas for potential loss of service and degradation of performance can be mitigated through utilization of web farming architecture.

6.3.3 Compliance to standards and industry regulations

When asked what industry standards their MES product supported, as can be seen in the following Figure 6-6, half of the participants responded that they either did not know or were not fully sure:

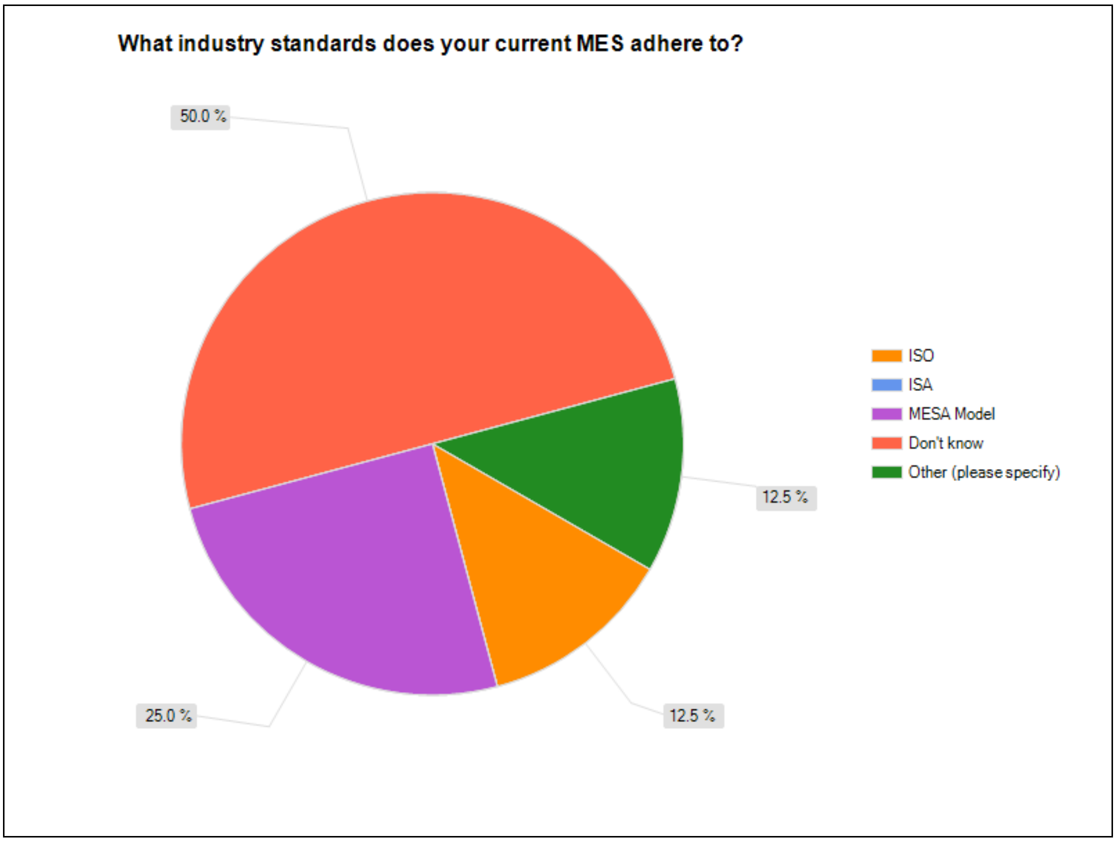

Figure 6-6. Standards compliance

From the supplemental text based feedback, it became clear that while support to industry standards was deemed important, it was not a critical factor and could not be warranted purely for the sake of standard adherence. Benefits of standards compliance are clear, with greater capacity for interoperability between connecting systems and reuse of well-defined models that have proven efficacy. If the cost of implementing standards compliance outweighs the benefits however, then the extra overhead cannot be justified.

In contrast, compliance to relevant regulatory bodies is viewed with much greater importance. When asked if their MES was compliant with SOX and FDA, the vast majority of participants reported affirmatively, as can be seen in Figure 6-7:

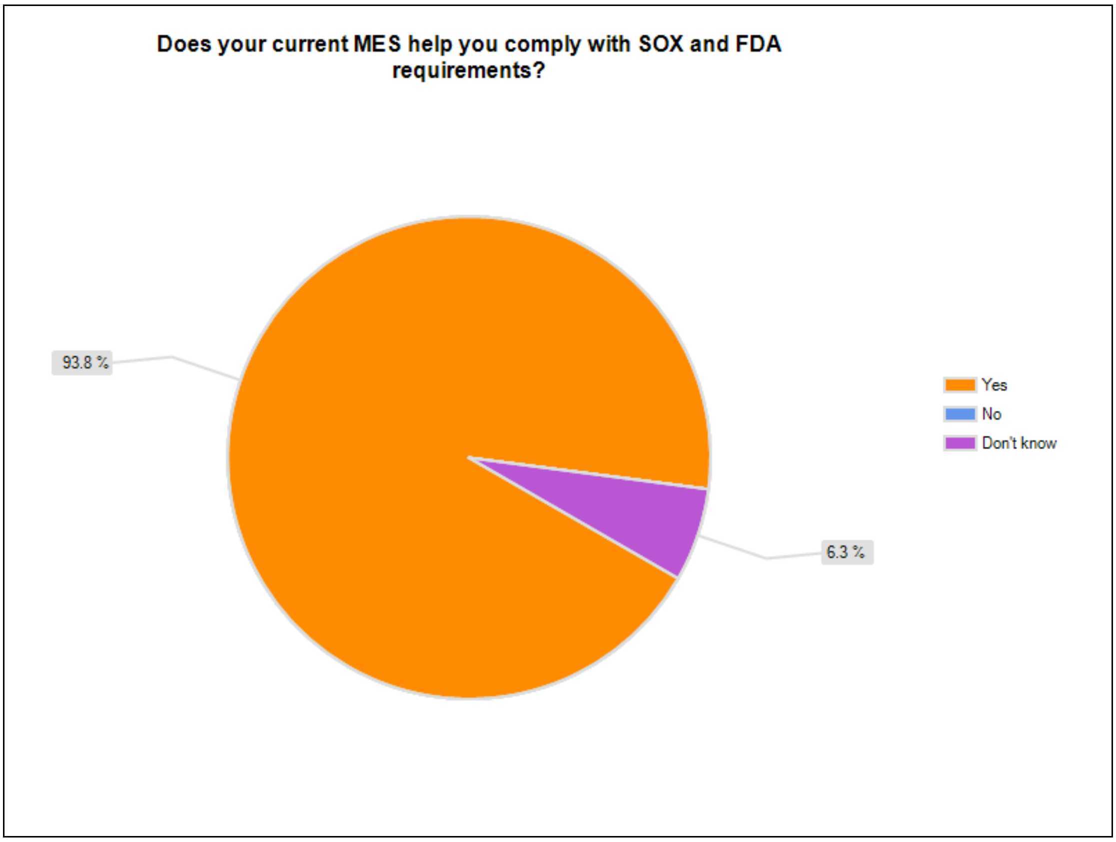

Figure 6-7. Regulatory compliance

This reflects the fact that adherence to relevant regulatory bodies is mandatory in order to conduct business and sell products in their respective markets.

6.3.4 MES usability improvements

The following Figure 6-8 shows that approximately three quarters of the technical participants responded favourably when asked if increased GUI prompting could make it easier for users of the system and potentially reduce user originated errors:

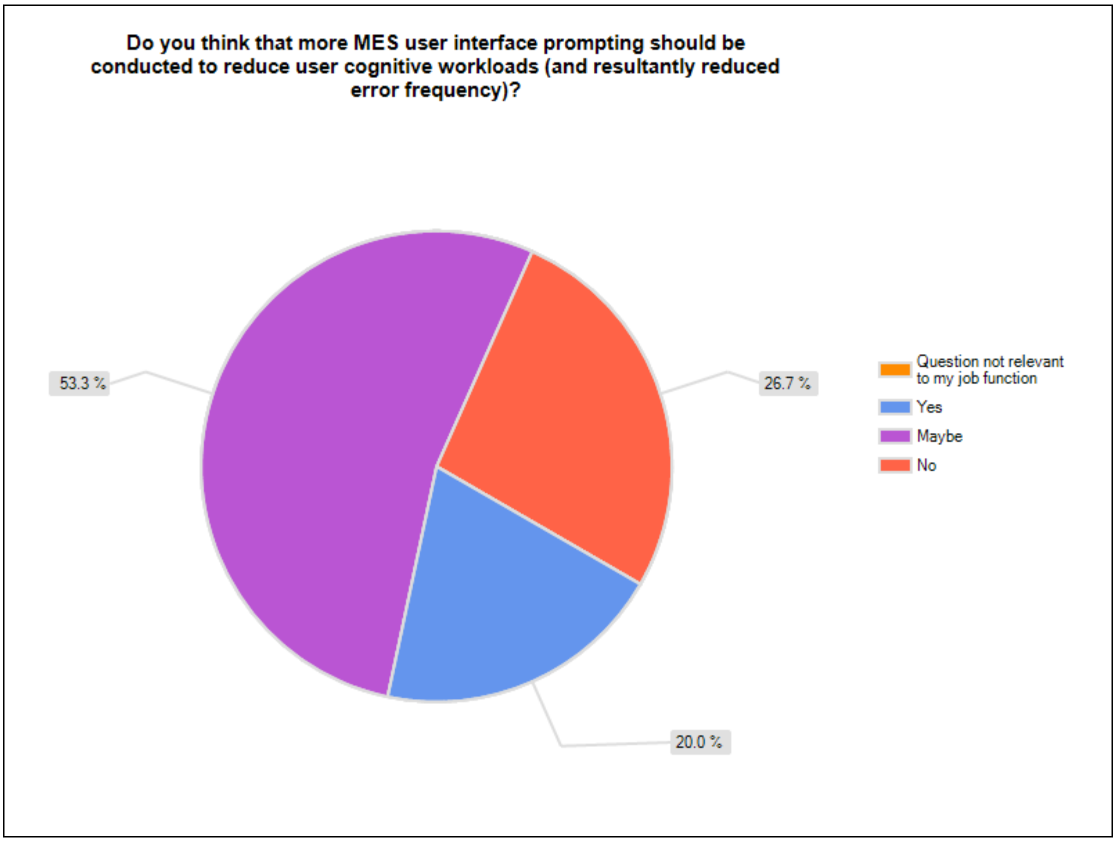

Figure 6-8. Increasing interface prompting

Of those that indicated a “maybe” response, text based feedback highlighted that not all implementations of CAMS needed this. Also multiple participants believed that user prompting would only provide a marginal initial benefit; when users become somewhat familiar with a software product they tend to ignore prompts and just proceed as normal. In order to prevent this reduction in prompting effectiveness, on screen prompting would need to be correctly implemented.

6.3.5 MES Interfacing and reporting

With MES situated between various application layers of interfacing systems, it is no surprise that the ability to integrate with these related systems is rated very high, as seen in the following Figure 6-9. More effective standards compliance is seen as one way to achieve better integration, as often excessive work is required to enable two pre-existent systems to communicate effectively.

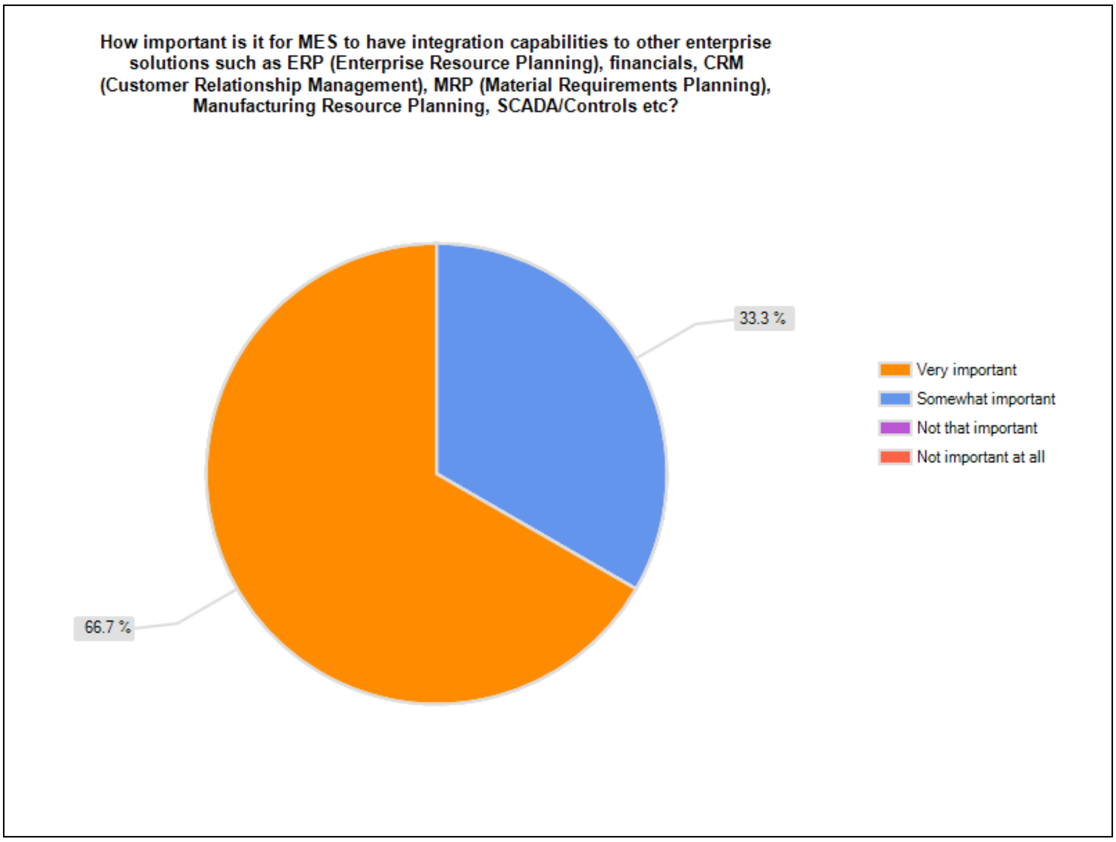

Figure 6-9. MES integration capabilities

With regards to reporting tools, as seen in the following Figure 6-10, the survey participants recommend that reporting should be designed with very little or no integration. This type of reporting architecture is intended to reduce MES complexity and enable plugging in different reporting tools as needed.

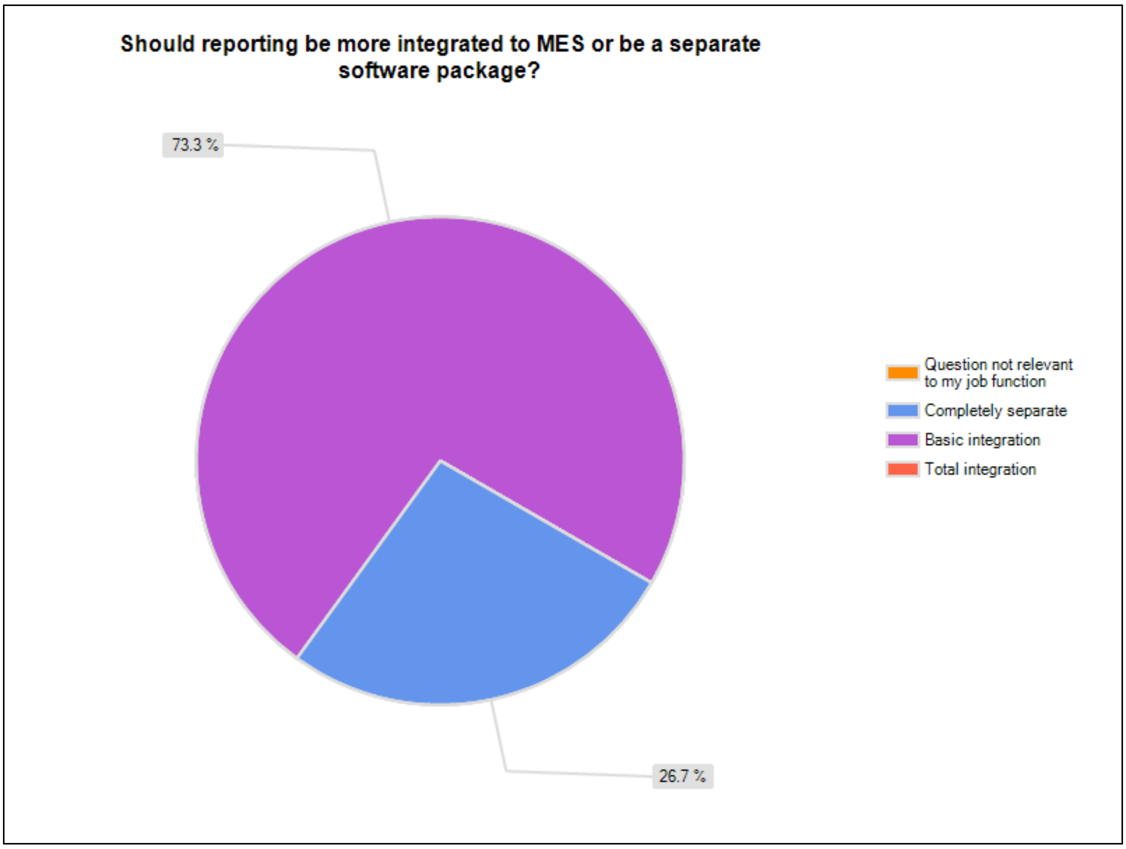

Figure 6-10. MES reporting integration levels

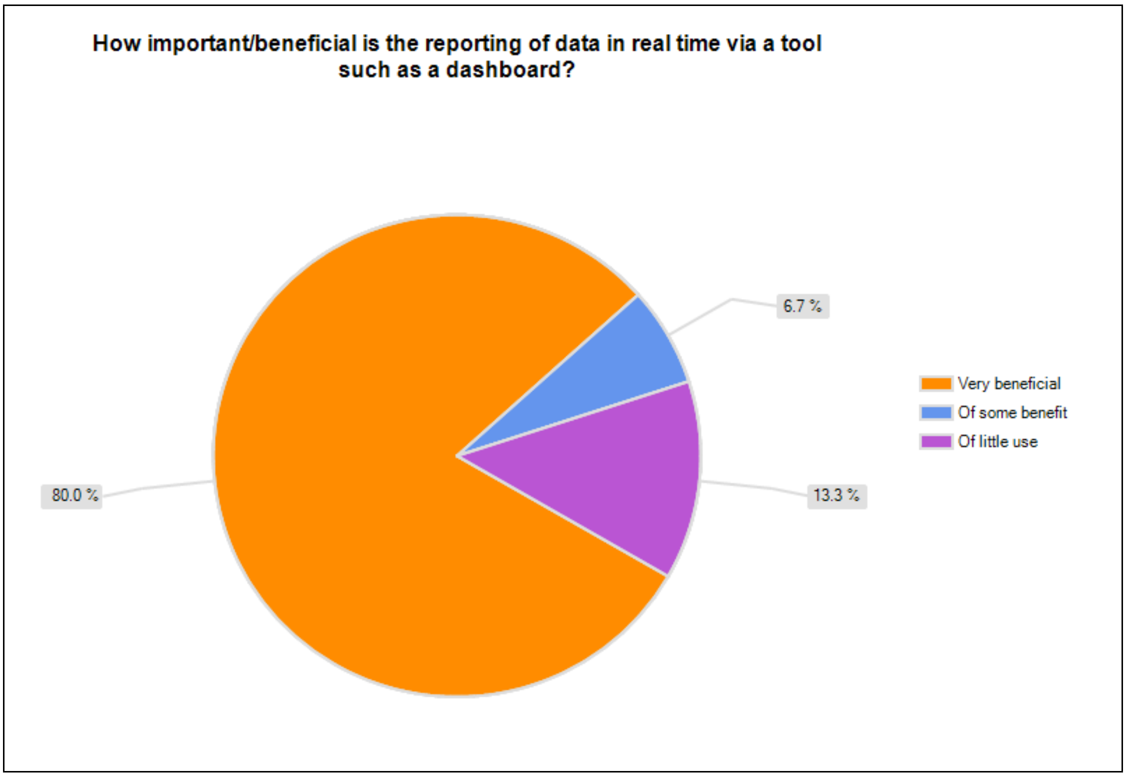

Figure 6-11. Real-time MES reporting tools

Access to real-time information is also deemed as a critical aspect of reporting, as illustrated in the preceding Figure 6-11. This is seen as an area that has scope for considerable improvement. As manufacturing becomes leaner, personnel within the organisation need to have easily accessible information to enable quick decisions based on accurate data. Another area for improvement perceived by the technical participants is that of data mining, as illustrated in Figure 6-12:

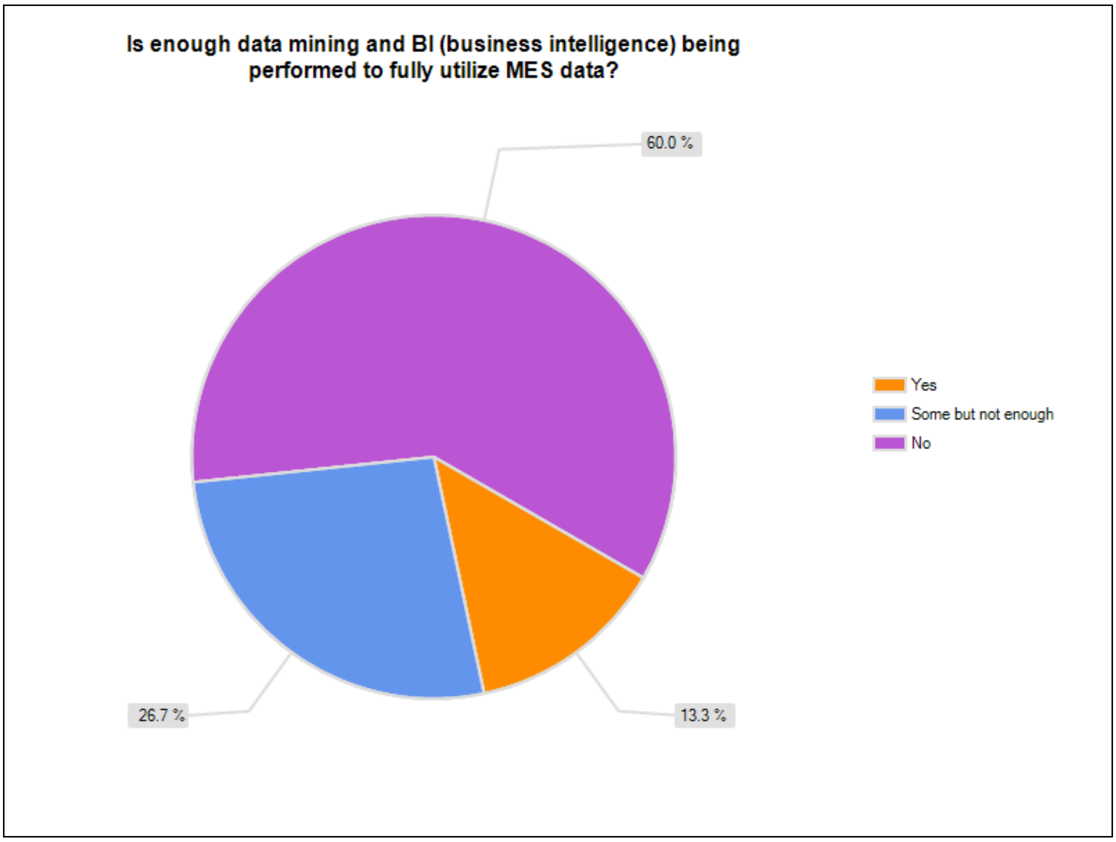

Figure 6-12. MES data mining

Often systems are developed purely to business requirements, without the consideration of non-essential functionality that has the potential to yield significant business benefits. While the initial installation and configuration of data mining is an IT project, business personnel ultimately need to drive data mining usage as they are the ones who use the elicited information. This lack of ownership can be a reason for failure with traditional analytical data retrieval projects.

As illustrated in the following Figure 6-13, the opinions for potential amalgamation of geographic MES data were mixed:

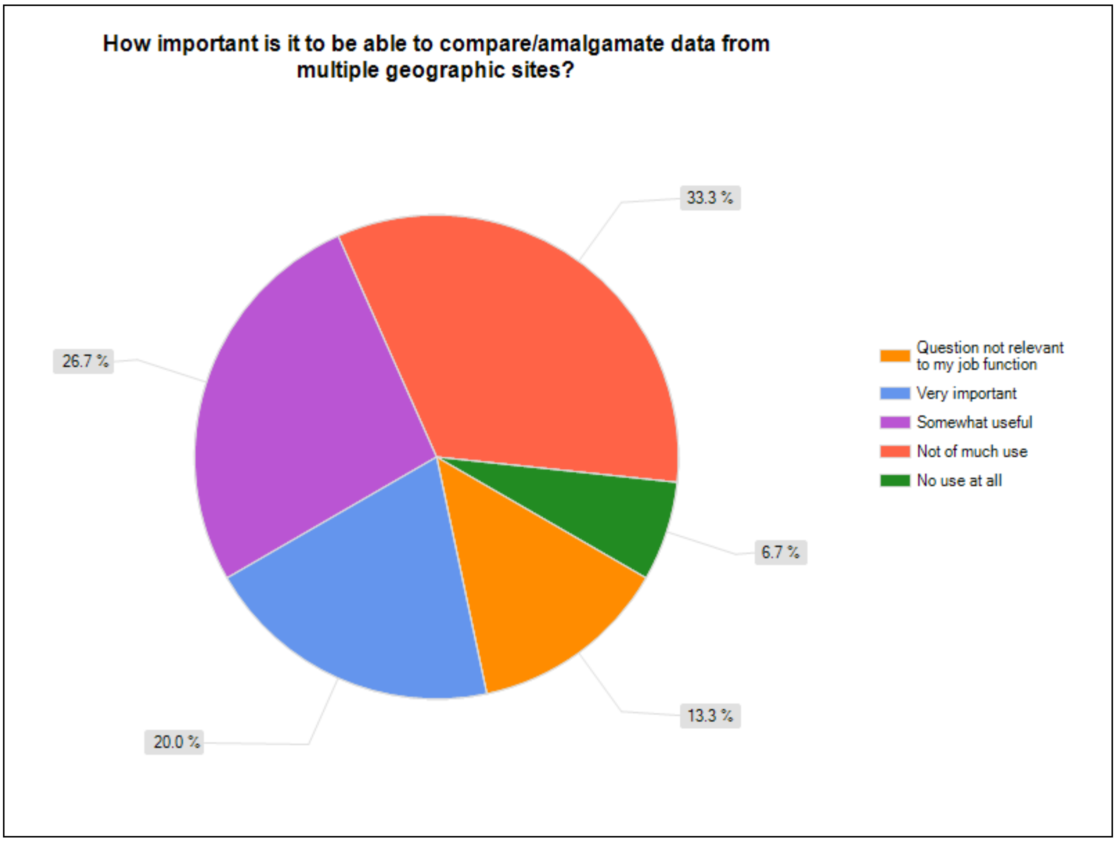

Figure 6-13. Multi-geographic data amalgamation

Further analysis of this questions responses revealed that this answer depended on job function. Higher level managers consider it very important to compare data from other sites whereas personnel with site specific responsibilities and a narrower job function gave a much lower priority. Also from looking at the data, it can be observed that there may be competitive political forces in existence between geographic sites. Some sites may potentially resist being compared with another manufacturing plant in case it reflected badly.

6.3.6 General MES enhancements

A wide variety of suggestions were made in response to the question of "If no barriers existed (time, money etc.), what could be done to improve MES design?". Refactoring of CAMS was a recurring suggestion. The need for refactoring is more prominent for software that has grown over time through feature enhancements and add-ons. While organic growth is unavoidable for large enterprise based systems, this tends to negatively impact the overall code base and impair performance and scalability. This was a very suitable answer to the question, considering there would be little to no financial benefit to performing.

Another suggestion for MES improvement was for the incorporation of RFID tracking on product containers used in manufacturing. Realistically this is an unnecessary overhead in the cost and complexity of manufacturing. While it is suitable for certain industry types and applications, it is not really suitable for the container management facility of MES in a high automation environment. RFID is triggered by proximity and so could not be used to accurately track product being processed. Traditional barcode scanning still performs its task of ensuring the right container is in the right place during processing.

6.4 User Survey Findings

The following subsections highlight the significant findings of the user survey.

6.4.1 Usage of computer terminal input devices

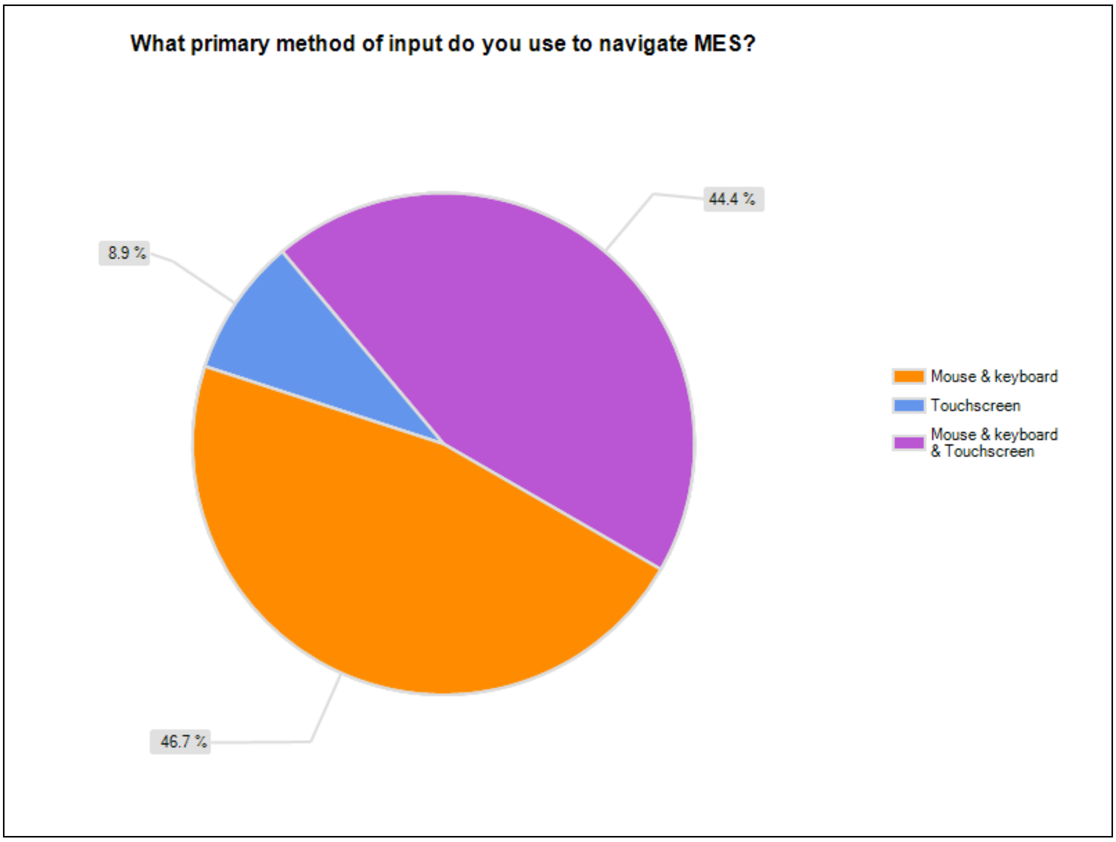

Figure 6-14. Primary input method

When asked about computer terminal input devices, as seen from the preceding Figure 6-14, it was found that the mouse & keyboard was the primary method utilised, where touch screens are used mostly to supplement navigation. With the majority of computer terminals in manufacturing being touch enabled, such dominant use of keyboard and mouse was not envisaged. In a manufacturing environment, operators are not always in an ergonomic position for using traditional input devices such as mouse & keyboard and can often be standing or crouching over a work bench. In the course of working in such an environment, input devices may be inaccessible or become hidden around machinery. Regardless of the benefits of touch screens, they still did not appear to be utilised fully.

The question on input preference was constructed so that the user would be forced to choose between keyboard & mouse versus touch screen. In contrast with actual usage, Figure 6-15 shows a greater number of users would like to primarily use touch screens when interfacing:

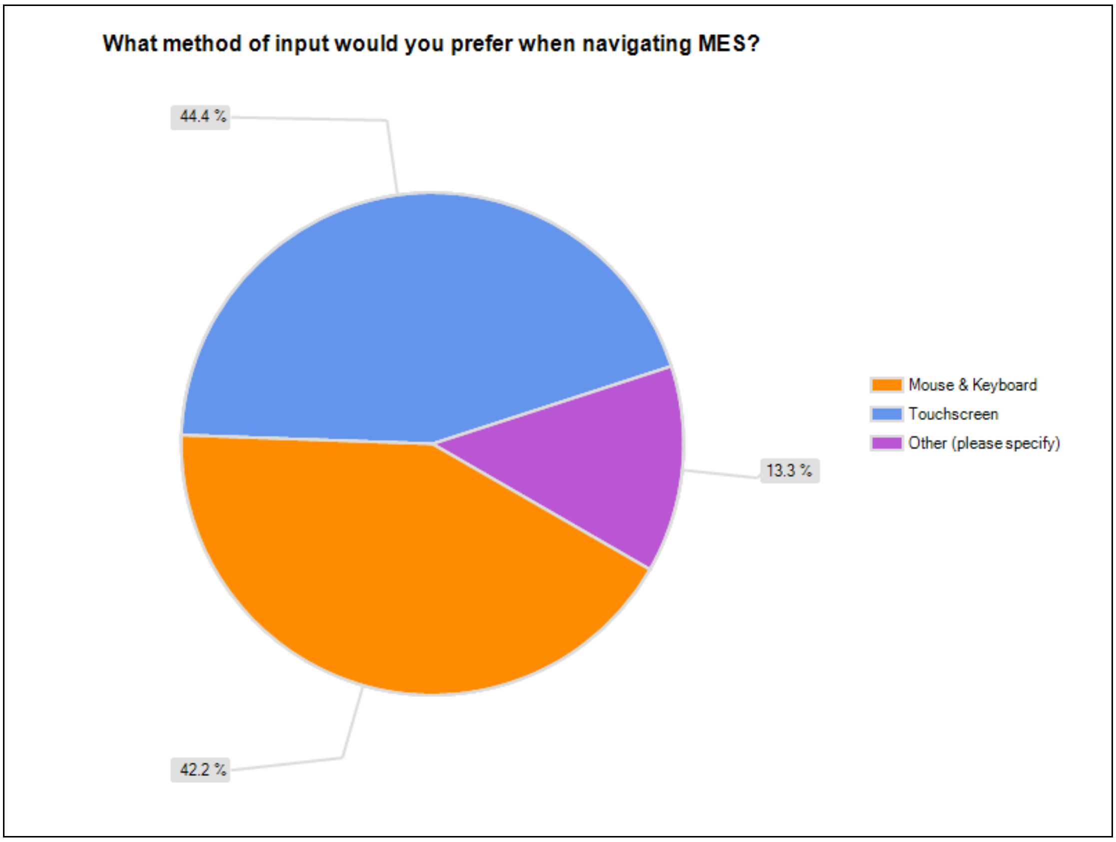

Figure 6-15. Preferred input method

A lot of text based feedback indicated that the GUI was not designed for text and numerical entry and therefore were forced to use the keyboard & mouse. This was an interesting discovery as the CAMS GUI had always been developed in a touchscreen friendly design, with reasonable sized controls for ease of access, and an onscreen keypad for numerical data entry. A major deficiency was the lack of ability to enter textual data in addition to numerical entries.

6.4.2 Users perspective of error prevention enhancements to MES

A number of responses were received to the question "What changes do you think could be made to improve MES error prevention i.e. reduce frequency of errors that prevent you from doing your work?". The top recurring response was that of system speed. The request for faster system response was widespread but in general it did not cause any problems apart from a lag of a few seconds from the user requesting an action to when it is completed. This is often a trade-off between implementing complex business logic and also investing in cost effective IT infrastructure that performed satisfactorily, but did not have abundance of excess capacity. An excessively poor system response can cause issues such as lost productivity, operator frustration and related timeout errors that require technical support intervention. Through grouping of responses by site, it became apparent that two separate manufacturing sites had particularly bad MES response times. Upon further investigation it was uncovered that the first site had a sub-optimal database configuration that could be tuned to give a significant performance boost. The second site used old computers that could not perform as needed for the client MES application. Personnel in the two problem sites had come to accept that the MES application was very slow and so the issues were never escalated. As a result, data from the survey could be used for justification for implementing the improvements.

An interesting nuance of GUI design was also highlighted with some users requesting final confirmation dialog boxes before processing user requests. Operator fatigue can exist due to uncomfortable manufacturing environments and shift work. Memory and concentration levels may not always be optimal so additional prompting for confirmations before performing significant tasks may alleviate errors in human judgment. Additional steps to counteract fatigue related issues were also suggested with the request for more undo facilities. Often users realise their mistake immediately after they have performed an incorrect action. Direct modification of data to the correct values would not be acceptable to regulatory bodies such as FDA and HIIPA etc. Facilitating the user to undo would be an acceptable action, but only as long as all before and after values are recorded to an audit trail.

Manufacturing operators clearly communicated in the survey that they wish to up skill and be empowered to fix as many problems as possible. There is a balance however as providing too much access to users can be deconstructive, where they attempt to resolve a small issue and end up causing a much larger one. An optimal balance could be reached by providing online user driven learning content that is contextually relative to their MES implementation. Another suggestion to have designated "super users" conceptually has more potential for success. These users could respond much more rapidly than technical support as they are already in the vicinity of the issue. Upon drilling down into the user survey data, these requests appear to have been made from sites that do not employ the "super user" strategy that has been proven effective in other manufacturing sites.

6.4.3 MES performance requirements

As seen in the following Figure 6-16, there was a wide range of answers when asked what the maximum acceptable response time was for performing actions within MES. The even spread of answers was indicative of the varied business logic implemented by the different manufacturing processes and also by the level of machine integration and automation being performed. Batch processing gave the greatest tolerance for performance because MES was only being utilised periodically over a shift.

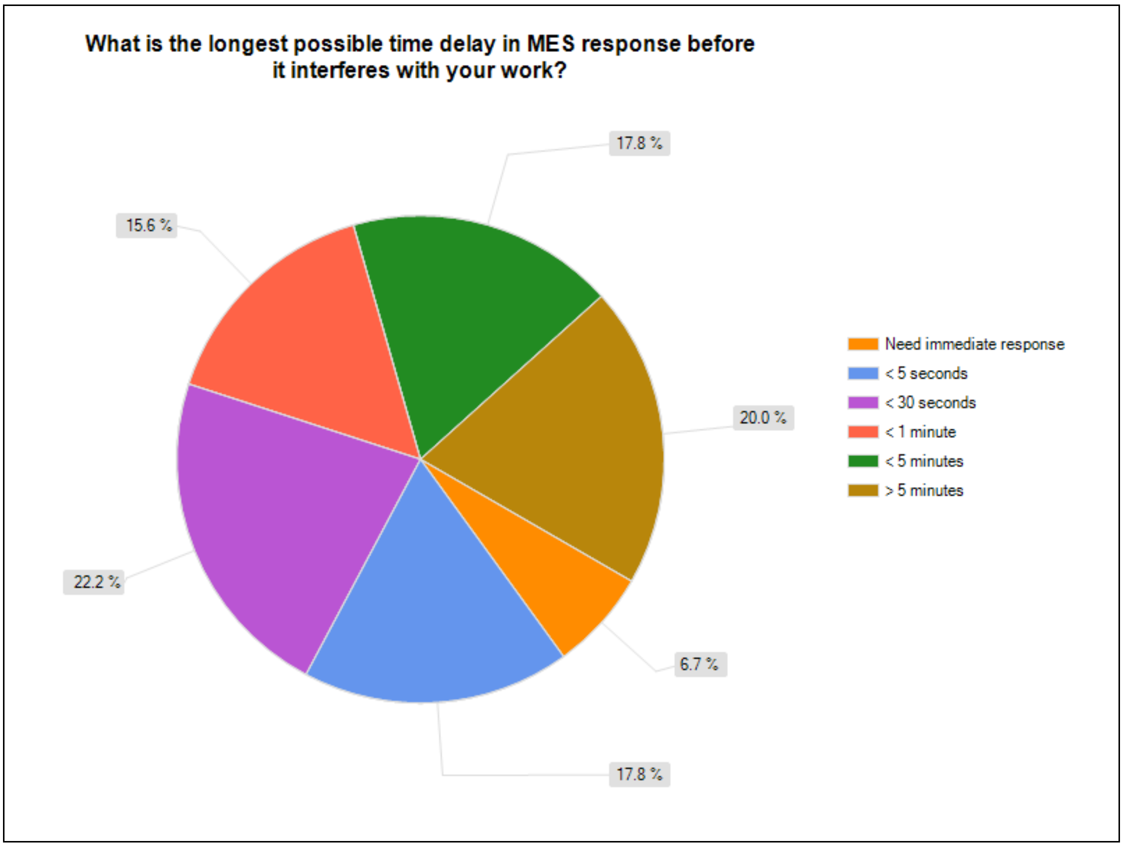

Figure 6-16. Acceptable MES response times

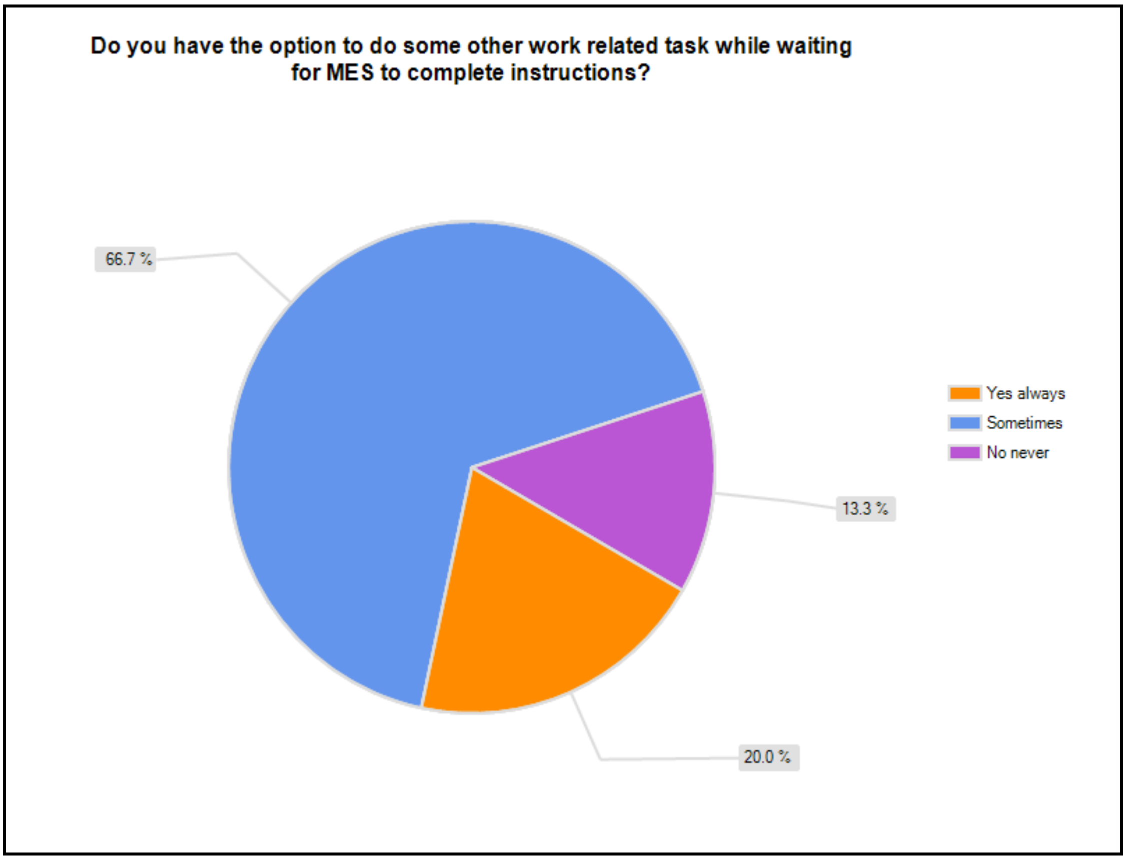

Figure 6-17. Lost productivity due to slow MES response

Possible delays do not necessarily interfere with productivity however, as indicated in the preceding Figure 6-17, where often there are other tasks to be performed while waiting for the system to process. Unfortunately with increasing levels of machine integration and automation, slowdowns in system response do negatively impact productivity. Upon further analysis of this questions response, the users that cannot complete other tasks are all from high automation stages in manufacturing.

6.4.4 MES usability and training from a users perspective

When asked the frequency of having to make guesses when using MES, users responded with a surprisingly high percentage, as seen in Figure 6-18:

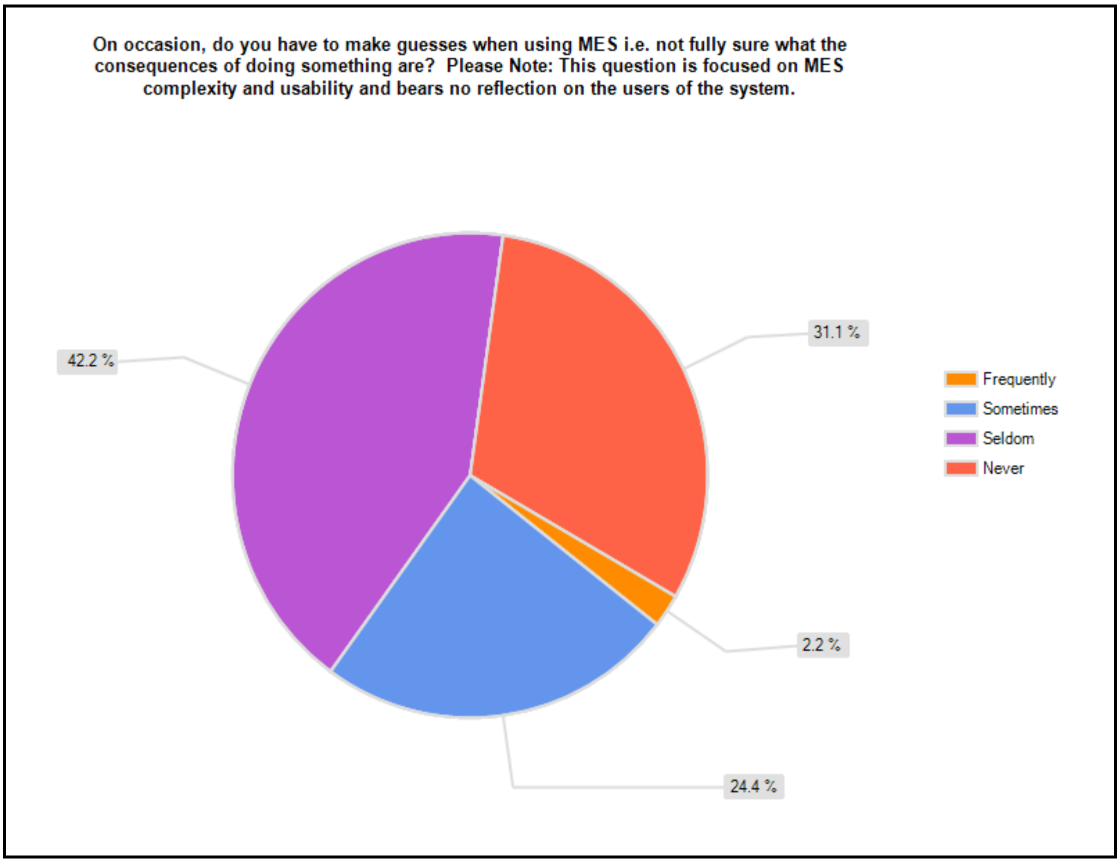

Figure 6-18. MES training needs

Although users all undergo defined training courses relevant to their jobs, there were quite a number of users not entirely sure of the actions they are taking. Based on feedback, it became clear that this happened sometimes when the MES was modified to execute a requested change in business logic. While these changes manifested in only small perceivable alterations in MES behaviour, it was enough to create uncertainty for the users. In addition to this, although operators undergo adequate training, they sometimes have difficulty remembering due to fatigue and other factors related to working in a manufacturing environment. This indicates a potential need for on-demand self-based training that could be used by operators as required to refresh their skills.

When asked of preferred methods of training, users predominantly chose instructor based training, as illustrated in Figure 6-19:

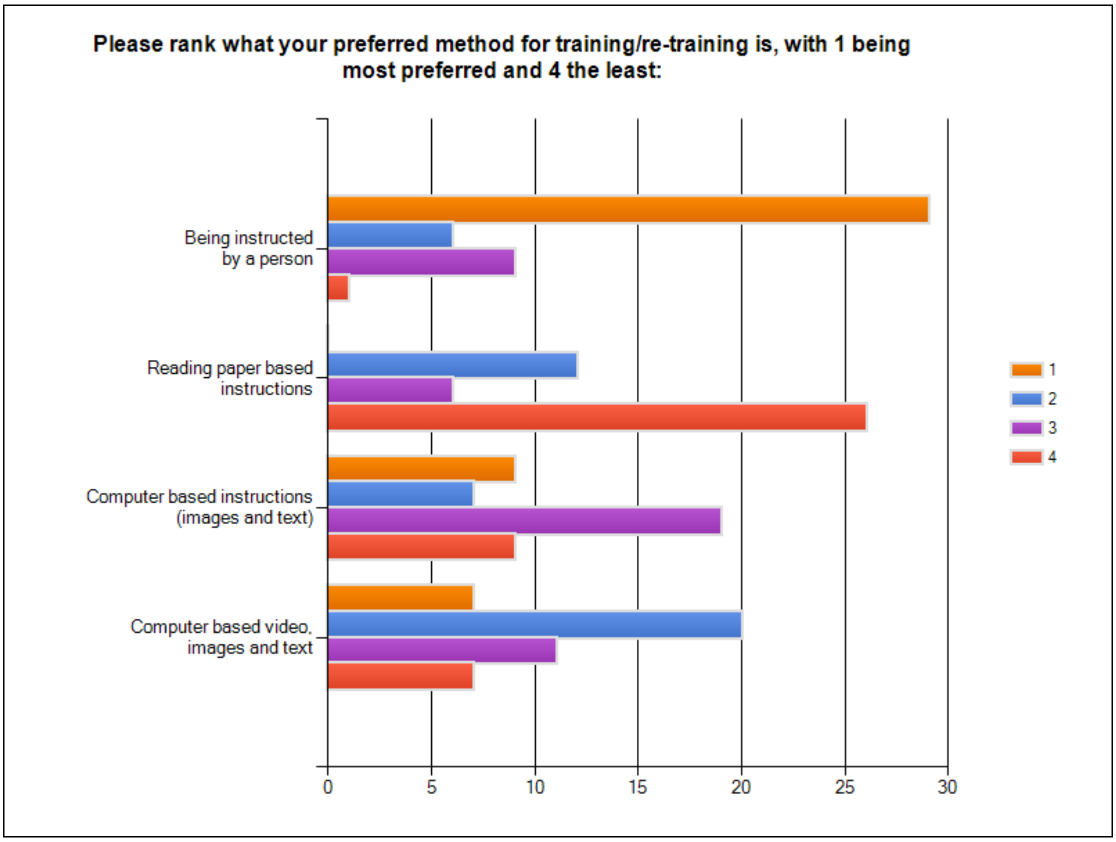

Figure 6-19. Preferred training methods

This is unfortunate as one-to-one training is very labour intensive and is limited to availability of instructors. Text based feedback from users highlight that this is what they are most familiar with. Also in geographic sites where English is not the primary language, users feel more comfortable with a human instructor and can ask for clarification in their native language if they feel the need.

In second place the next preferred training method was with computer based text and video. This really is the best method, as people tend to retain more information when conveyed via a rich interactive media. It is highly available for on-demand use and is significantly more cost effective than one to one training or even paper based training.

When asked if more online based training would be of help, over 90% of user survey participants responded favourably, as seen in Figure 6-20. This indicates that although users currently prefer one-to-one training, there is scope to alter the primary training method. Often users just become familiar with a particular work practice. Having more computer based training available could with a reasonable certainty, help users adopt the optimal computer based training methods. It could be adopted as a learning aid initially with the goal of shifting to the primary learning method eventually.

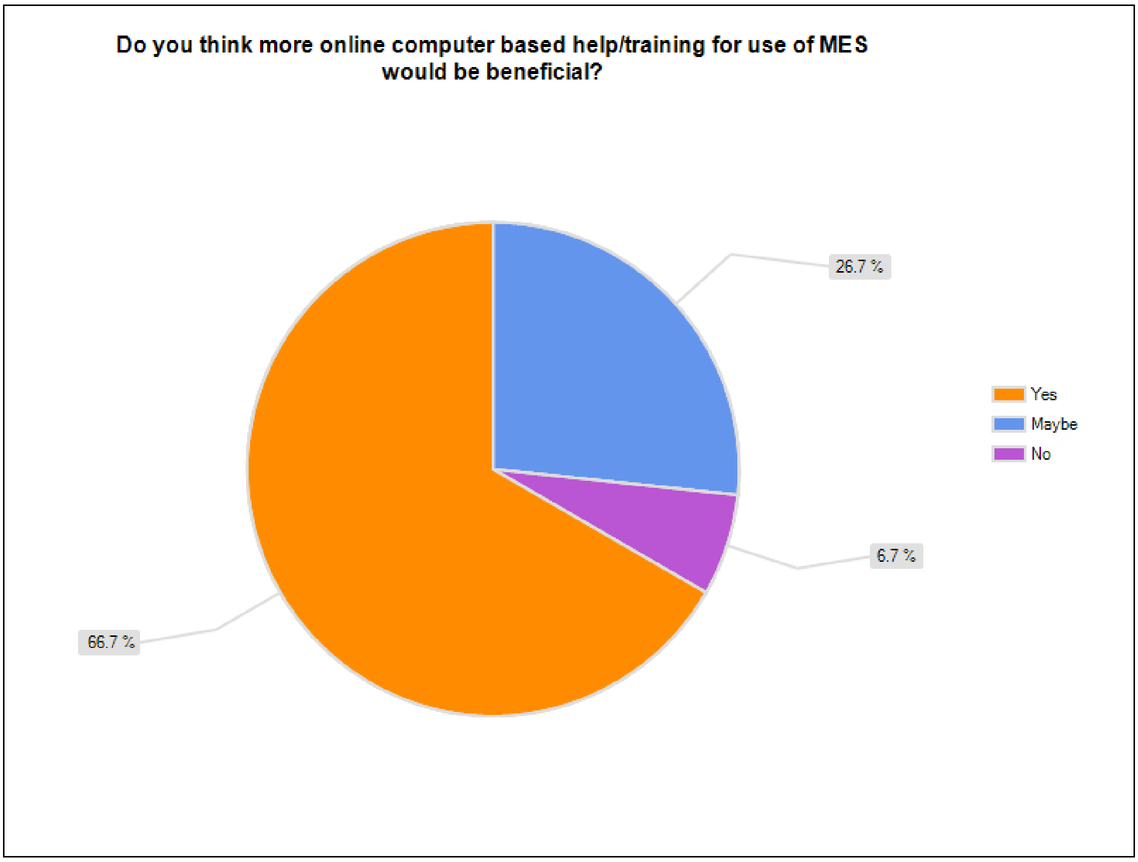

Figure 6-20. Openness to more computer based training

The question of perceived MES purpose helped highlight the differences in attitudes between different cultures. As illustrated in the following Figure 6-21, the use of MES is widely welcomed by operators as they appreciate how it reduces the complexity of their jobs.

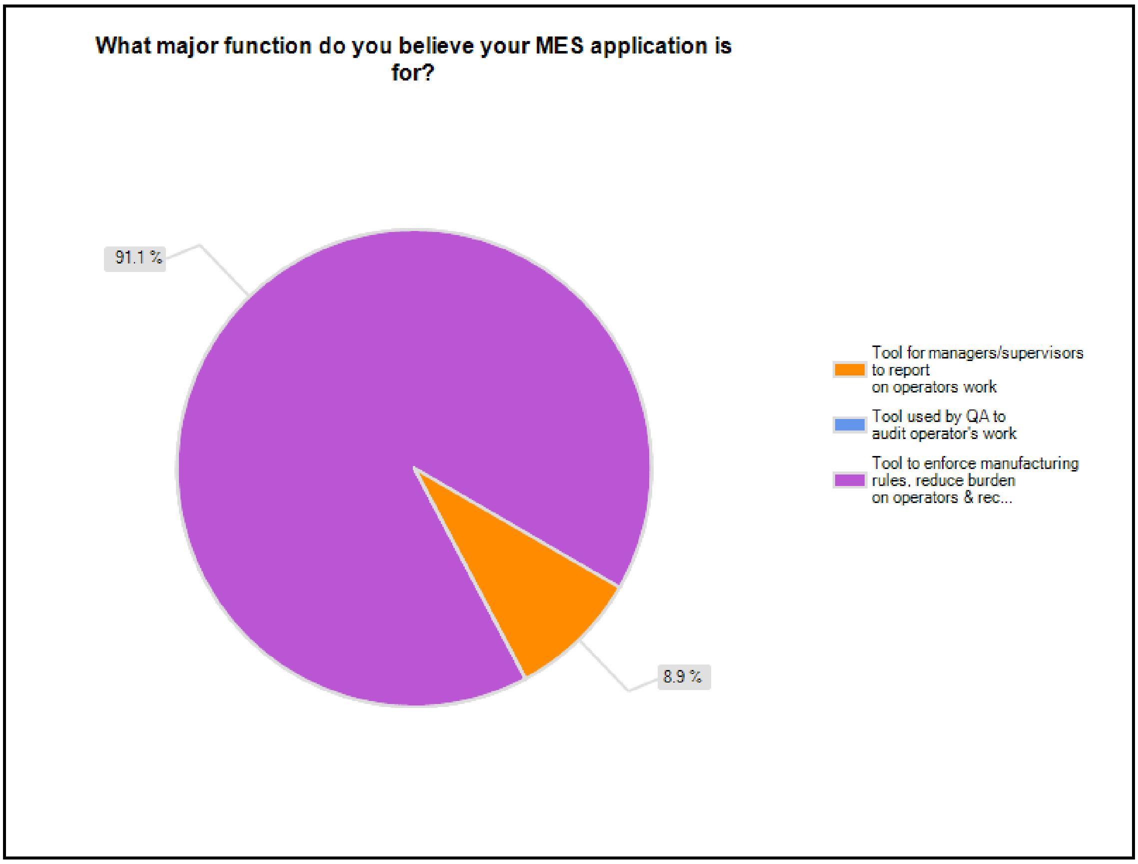

Figure 6-21. Perception of MES function

In one particular geographic site, MES was viewed cynically as a tool for reporting and controlling operator behaviour. It is important to overcome this view as resistance to MES reduces its effectiveness. Although it is not feasible to attempt to alter the norms of cultural opinion, users can be enlightened as part of their MES training. So in addition to just training on how to use the tool, users should ideally be informed as to how it makes their jobs easier and all of the other associated benefits.

6.5 Review of other MES applications

There are numerous COTS MES available for use in manufacturing. This list is constantly changing with the discontinuance of some products and the entrance of others. It is apparent that the more sophisticated MES offerings have very clearly defined segments targeted. Some of the lower end MES have a wide range of capabilities spanning a business and are intended to be used as the single software tool driving the whole company. While this may suit smaller businesses, these MES types do not scale well and cannot react quickly to change. One such example would be that of changing taxes and levies in the Irish annual budgetary alterations. Rather than attempting to have a MES that caters for the breath of the supply chain, MES should be specialised for its core role. As needed, it can have interlinking modules such as reporting, ERP etc. These extensions of MES should be sophisticated enough so that they provide more than just the bare essentials. As has been seen with some COTS MES, the synergies between MES and custom integrated software can yield even greater benefits than utilization of "best in class" products, such as seen with Camstar and its integrated LIMS capabilities. For building a truly scalable MES, integration capability is key, with adherence to industry standards highly advisable. This interoperability involves the ability to share information and perform processing with adjoining system types. While tools such as Microsoft's Biztalk exist to connect disparate systems, its use is not ideal as it brings additional complexity, creates another point of system failure and gives rise to increased application costs to customers. It is also worth noting that industry standards may not be sufficient for guidelines on interoperability. Potential need for interaction with related systems should be kept in mind when designing all MES interfaces. One way to build accessible interfaces into MES is to construct it in a modular design. Modular design also facilitates the ability to turn certain pieces of functionality on or off as needed. This capability can be used to customise the product for client needs and enable flexible pricing structure so that clients with low budgets can still avail of the baseline MES package. Resulting from this, designing for standards compliance fits in coherently with the SOA business model that can be used to maximise the usefulness of MES.

To drive true benefit to the business, MES needs to transcend its core function of just running a manufacturing process. Ideally it should have Artificial Intelligence (AI) integrated to automatically react to undesirable changes in manufacturing. Rather than reacting when the error state has occurred, it should be pre-emptive in nature and strive to prevent issues from occurring. This type of MES AI can at a basic level alert to appropriate staff to take action. With greater levels of sophistication, MES could autonomously make changes in manufacturing to maintain optimal levels of quality and output. For example, in a manufacturing process where a tablet being manufactured is going out of defined specification, MES could autonomously cause changes in the tablet recipe so that it remains within tolerance. The traditional function of MES in this scenario would have rigidly guided manufacturing tolerances and just highlight an error after product goes out of specification.

There are a number of 3rd party tools that integrate with MES to perform the lower level SCADA functionality. These products specialise in automation integration on the factory floor. While SCADA tools can be integrated into MES to perform these lower layer activities, the integration can be complex and give rise to increased costs due to additional software licensing. This coupling of layers can also impose restrictions on application architecture. A cross browser web based device communication model would not be possible through the integration of 3rd party SCADA integration. Even for traditional client server models there are restrictions on how the application is designed.

There is a balance with COTS on the amount of customization that is optimal. On one side, there should be enough configurability to tailor the application for a particular company's requirement. On the other side, the application need to be relatively easy to configure, maintain and modify. For this reason, the larger MES providers have separate MES products for different industry types so that there are no conflicting design requirements or overly complex configurations. Defined API's can be provided to enable development of custom code that is very particular to a customer's specific requirements. Custom development should be a last resort as this goes against the concept of COTS.

Adequate reporting is essential for all MES because without visibility into process performance, even the features of a highly sophisticated MES are useless. Greater benefits are to be gained through full understanding of manufacturing processes and where improvements can be made. MES reporting functionality should ideally be integrated into the application, instead of having to customise a separate reporting tool to the particulars of the MES data structures.

Security aspects of COTS MES are perceived with a much lower priority than needed. All of the commercial applications list security as purely an access control restriction and product functionality limitation method. Vendors generally do not go past the basic security concepts. This is not sufficient with the year 2010 seeing an explosion of threats, like the STUXNET worm and numerous other vulnerability postings for SCADA related software. In March 2011, multiple exploits were publically released for SCADA systems such as Siemens, Iconics, 7-Technologies, Datac, and Control Microsystems (The Register 2011). A publically available website (Shodanhq 2011) lists all of the public facing SCADA systems. The site uses a scanning technique to detect the signature of systems on the internet to determine their industrial nature, and proves the possibility to detect internet facing industrial systems through a general internet search. This is another reason why combining a 3rd party SCADA system to custom MES can have negative consequences, whereby any vulnerability that may exist in them can be used to compromise the whole MES system.

6.6 Web Technology review

A general overview of applicable technology has been listed in chapter 3. As with all technology, there is a limited lifespan on the technical specifics, due to it becoming outdated and irrelevant. Innovation is constantly being driven through new technologies and the embellishment of existing ones. Where possible this research has highlighted the technological concepts in relation to MES as opposed to repeating the detailed workings of the relevant technology.

Adherence to technology standards bodies' prescriptions is heavily encouraged for web based applications. W3C for example governs HTML and how browsers should interpret web pages. In around the year 1997, version four became the latest HTML standard. This version was often found to be lacking by browser developers, which in turn encouraged the developers to inject their own little quirks to get the results they were looking for. Back then, standards were just guidelines providing a starting point for browser development. The need for strict HTML compliance is now understood, as the benefits of interoperability are realised between the numerous browser models and versions available. As we move forwards through this period of transition to strict adherence, it is more important than ever to ensure web pages comply with the standards. By sticking to the guidelines, websites stand a much greater chance of operating correctly in a number of different browsers. Also of equal importance, compliant websites will continue to operate correctly in newer browser versions that are developed in the future.

Implementation of reusable frameworks is an excellent method for RAD construction of MES; rather than developing from the ground up, sophisticated functionality can be readily integrated with little work. Another benefit from use of these frameworks is that they generally get updated and enhanced periodically. This brings a range of benefits for future MES modifications. However, care should be taken on which frameworks are chosen as some may be discontinued which could give rise to issues around future MES modification. Also from a security perspective, any framework vulnerability can be used to exploit the entire MES application. This is even more applicable for web based software.

One very notable emergent technology covered through the course of this research is that of HTML 5. As of July 2011, this technology is still in draft. Even though it has not been officially standardised, it has been widely adopted by the latest versions of web browsers. The power and flexibility of HTML 5 means that a lot of workarounds and circumvention with hybrid technologies are no longer required. HTML can now host a wide range of capabilities such as embedded video, audio and dynamic graphical content. Possibly one of the most powerful features is that of the highly efficient web socket. Being part of a supplemental WC3 standard to the HTML 5 specification, web sockets facilitate bi-directional TCP communication. This totally removes the stateless limitation inherent in previous HTML versions. While web sockets are not fully supported by all browsers and there is no backwards compatibility for older versions, it may at some point in the future rival the abilities of AJAX.

There will more than likely be a never ending battle between the major development languages for dominance in software development. Java and .NET are two very powerful languages under continuous growth and enhancement. At any point in time, one development language may beat the other at certain aspects. As has been seen with previous releases, the advantage is then lost as the competing language is modified to meet if not surpass the others abilities. Choice of language therefore should not go just on the contemporary capabilities in existence. Instead, other factors should be appraised such as how the language fits in with the intended system architecture. Also some languages have better interoperability in certain designs than others. A lot of choice can come down to personal preference and the availability of relevant skills. However some thought should be given to future technology trends and potential of existing technology to be depreciated by newer developments. One such example of this would be the Flash and Silverlight technologies. While they will more than likely have a place in web development for quite some time, it is widely envisaged that they will eventually have to concede to the advanced capabilities of HTML 5.

As bandwidth capacity, disk and processing power is continually increasing, there is less emphasis given to developing an optimised system that uses as little resources as possible. While it may not seem that important in the early stages, scalability can be considerably impeded by system architecture. MES may reach a point where it is not able to cope with the performance demands being placed on it. SANs, Web farms and database grid technology greatly assist in the potential for web applications to meet growing demands. However this technology cannot be relied upon as the sole avenue to meet performance and scalability needs. MES should be designed from the outset so that it can cater for all potential customer transaction levels not just now, but also in the future. Unlike standard websites, the industrial nature of MES means that any service outage incurs a considerable financial loss. For this reason it is critical that unnecessary limitations do not exist in application design.

One of the aims of this research was to assess the feasibility for remotely hosted MES. Cloud computing capabilities are maturing with the adoption continually rising and more investment being made. While the Cloud model is certainly capable of performing this remotely hosted MES functionality, its reliability and security has yet to be proven. Amazon is one of the prominent Cloud hosts with some high profile customers. Confidence in the Cloud has diminished after Amazon suffered a number of outages (V3 2011). Apart from the reliability perspective, Amazon has also suffered a number of hacks that either directly targeted Amazon, or used it to perform further hacking, such as happened with the Playstation Network hack (Internet Security 2011). While this was a 3rd party Cloud service, it highlights the potential for these types of issues to occur for a MES solution.

7 Conclusions

MES covers a broad landscape of technology. In order to be of sufficient academic and industrial value, the scope of this research was limited to the core MES technology with particular emphasis on a web based architecture. Through the course of this thesis, various insights were uncovered. It is envisaged that this research will provide benefit to those concerned with both MES design and also web technologies. While all aspects of this research were relevant to MES design, there could have been potential benefit to break down this research into smaller papers. This strategy would have enabled in-depth research into the various aspects and also reduce the research publish time from years to months. The quicker publication of research is most appropriate to areas of technology so that the findings are not outdated by the time they are made public.

Manufactured product quality, process efficiency, speed of change and low cost consumer products, are all underlying market drivers of MES design. These drivers create the need for a software product that is highly usable and minimises errors as much as possible. In essence, manufacturing software plays a direct role in the survival of organisations in the global economy. The following sections build upon the individual chapter findings and seek to deliver further insight through synergistic combination of the research as a whole.

7.1 Considerations on design effort

Investigation into MES design factors may appear to be a low value adding activity of system design. These design factors however can be compared analogously to Hertzberg Hygiene Factors theory of management. Extensive work into design factors may not appear to yield many benefits, but failure to properly design these aspects will negatively impact MES and its capabilities. In a similar manner, adherence to standards and relevant governing bodies is of the utmost importance. MES sits within an application stack where interoperability is essential. While it may be possible to deliver a fully functional product without the incorporation of prescribed best practices, scalability and interoperability will be severely affected. With a growing need for visibility into real-time business performance, disjointed and unconnected applications create barriers and reduce organisational effectiveness. What may seem like the quicker route to product development, will ultimately create an arduous task to rectify the shortcomings experienced later on in the lifecycle of the MES product.

In the initial stages of MES design, a number of aspects need to be defined such as how critical the uptime of the MES will be, the nature of the target industry it is to be deployed in, the desired commercial business model and other such aspects. The intended MES usage will change the importance of the various design factors. However, in general, the core design factors such as HCI, scalability, interoperability and security will always be of high importance.

Upon consideration of pre-existent client-server MES packages, it may not be feasible to port to a web based alternative. Some newer development technologies are capable of easier architecture change, whereas the older technologies would warrant a complete rewrite. For low customer base COTS MES packages, there may not be a ROI from architecture redesign. There is still a potential for enhancement though various techniques to overcoming client-server limitations. For example, to facilitate rapid application deployment, the .NET feature of "ClickOnce" can be utilised to automatically download the correct version of the desktop client application each time it is executed. As another example of bringing web architecture benefits to client-server, desktop virtualization could be employed. This would facilitate rapid deployment, and also provide the additional ability to utilise low cost terminals in manufacturing due to the processing being performed on the server side. As a result, redesigning the architecture of an existing MES package may not be the most appropriate enhancement. However for all new MES endeavours, the superiority of the web based architecture over its client-server counterpart is evident.

There is also an emergence of hybrid MES architecture that utilises web services to overcome the limitations of client server systems. This architecture of client-server with SOA can circumvent the inherent limitations of traditional design. While this does not completely mitigate the client-server shortcomings, it may be a path of least effort to bring some benefits of the web to legacy COTS MES.

7.2 Research survey collaboration with Bausch + Lomb

While the survey findings provide insights and clarity to MES design, it became apparent that there are various aspects warranting further investigation. A lot of the survey results while beneficial, could not be taken to be fully conclusive due to the subjective nature of the Bausch + Lomb specific survey. It also became apparent that some survey insights did not give a true representation of necessary MES functionality, such as the low weighting for the need to localise MES for different cultures and languages. Attempting to look objectively at the data, when surveying experts in their various fields, it appears experts give a lower emphasis for the need to meet a requirement if it is already being satisfied. If however those needs could not be met, such as having MES supporting a particular language, then importance of that requirement would become critical. Surveying experts about a new implementation of MES could remove these biases and give a greater representation of the real MES requirements.

Even by focusing on the core MES design in isolation from the many integrating systems, there were a number of areas such as security, HCI etc. that warranted a research undertaking all of their own. This was not a surprise however due to the many complexities and specialisations of MES design.

The collaboration with Bausch + Lomb was not without its downside. Control was lost by the research undertaker, and it became an on-going project within Bausch + Lomb. The Bausch + Lomb MES team had some conflicting priorities and goals that were subjective and were not particularly suitable to the research goals. What had begun as a scheduled four week task eventually took over one year. The survey construction was in a constant state of flux, with requested modifications taking place on a regular basis. Due to key personnel being on vacation or tied up with work commitments, the scheduled timeframe quickly slipped. The process of multi-geographic change agreement also added to the delay, with lengthy cycles of sending the surveys out for review, compiling all the requested changes, and resending out for subsequent review. Ultimately, the overall reasons for a thirteen month schedule slippage were due to the socio-political factors surrounding collaboration between the geographically dispersed team, and also the low priority of the survey project due to organisational and economic changes that were happening in the same period. Even with the benefit of hindsight, there was no avoidance of the one year delay to this research undertaking, as collaboration was critical to getting relevant data from Bausch + Lomb. Discontinuing the survey was also not an option, not just because of the amount of work that went into it, but also for the invaluable information it could potentially yield.

While the findings were derived from only one proprietary MES product, it still had the potential for invaluable insight into an optimal MES design. To effectively target the various MES stakeholders, two surveys were devised. The first survey was primarily targeted at the users of MES. It was designed to be understood by as wide a range of people as possible; for some of which English was not their primary language.

A lot of tweaking took place with the wording of questions, the sequencing of questions and also the phrasing of possible answer selections. The primary goal was to obtain useful and well-structured data that accurately represented the truth. Interestingly, some subtle changes were required to ensure meanings were consistent between different nationalities. This intimate tailoring of the survey to the Bausch + Lomb culture would not have been possible with inter-company participation.

7.3 MES product insights

Some of the less established MES packages possessed a wide array of capability ranging from Time and Attendance to Documentation control. While this large feature set may appear more attractive to customers, it also brings about the challenge of concurrency to the MES developer. Through industrial, technological and even legislative change, MES could quickly become outdated and in need of extensive maintenance. The more business functions that the MES package caters for, the more frequent and extensive is the need for product updates. Wide feature sets may be of most benefit to smaller organisations that do not have a fully comprehensive application portfolio. For larger organisations that need best in class applications and high scalability, the core MES functionality needs to be kept separately to the adjoining application stack that makes up the supply chain.

While it would be unethical to intentionally lock in customers to a MES package, the better fitting to a manufacturing process, the harder it would be for product substitution threats. Rapid customisation through configurability is key to tailoring a COTS MES package to client's individual manufacturing environments. Higher levels of adaptation to customers' needs make it increasingly harder to move to a competitor product. However, the risks of a cumbersome design with limited scalability need to be mitigated in the goal for extensive configurability.

There are numerous technology toolkits in existence that can be used to rapidly construct software. These toolkits facilitate the plugging in of pre-built logic and GUI elements. While this enables RAD and the reutilization of powerful and pre-tested functionality, there are also some potentially negative aspects. Integration of 3rd party tools needs to be carefully managed so that they do not cause problems such as limiting future scalability, substantially increasing the MES product cost or even creating security vulnerabilities.

Cloud computing offers the potential for massively scaled applications. It is an evolution from ASP and SOA models of utility computing. While this is a very promising business model, it may bring a major benefit to remotely hosted MES, due to the fact that MES usage is always limited to the physical footprint of the manufacturing floor. For this physical limitation, the demands of MES probably will not need to fully adopt the Cloud model for scalability purposes. In fact, with the numerous Cloud outages suffered in 2011 by the large providers such as Apple and Microsoft, it highlights that the cloud model may not yet be mature enough. 2011 has also seen other threats to Cloud and web technology through what can be called a hacking epidemic. These two factors indicate that technology is not yet at a point that a web based MES can be used for critical manufacturing operations. It may still be suitable for smaller less automated environments where constant connectivity is not essential. Ultimately it should be decided on the savings made with going to a Cloud MES solution and contrasting that with the financial loss incurred from temporary loss of service or potential data loss.

7.4 Recommendations for the future

Upon reflection, the goal of eliciting optimal design factors for MES has been met. Ideally however, the research would have been completed in a shorter timeframe so as to have a greater impact and provide more insight. This part-time research was conducted over a number of years. In this timeframe some technical sections had to be refreshed to cater for newer developments. In particular, continually more applications are migrating to web architecture. Salesforce.com was one of the first enterprise level web based applications. This paradigm has continued to the point that now even Microsoft is moving its Office suite, having a desktop architecture since the very start of Windows OS, into a web model by the name of Office 365. Through the prevalence of CSS3, HTML5 and next generation development technologies, the general computer user is starting to have higher expectations of web application capability. The perceived difference between web and desktop applications has further diminished.

While there have been some interesting findings resulting from this research, it also serves as a guide for the construction of manufacturing based applications and in particular, to web based MES. Because the scope was still relatively wide even after limiting it to MES utilised in American multi-national healthcare companies, many of the areas could only be covered at a higher level. In hindsight, it would have been beneficial to perform further scope reduction, to make a more in-depth study possible. In fact, a full research paper could be dedicated to any number of areas covered in this research such as HCI, security etc.

While the survey yielded interesting findings, its results were particular to the custom MES developed for Bausch + Lomb. The results may not be truly representative of COTS MES requirements because the questions were tailored to a specific MES implementation. Because of this, the importance of some MES functionality and aspects will be skewed towards Bausch + Lomb.

7.5 Future recommendations

Arising from the survey data, it became apparent that there is an opportunity for further investigation into determining the most effective training methods that are employed in manufacturing environments for people with varying cultures, abilities and language comprehension. This could lead to significant gains as traditional training is a labour intensive task, whereas computer based training methods enable quicker self-driven knowledge transfer. The constant availability of online training can also assist in reducing the frequency of user based errors as they can refresh their skills as needed. The benefits of computer based training can also magnified through the implementation of effective HCI concepts. So by coupling a study into online training methods and analysis of HCI, it is potential that significant gains may be accomplished for MES design.

Another recommendation would be for an in-depth analysis and potential prototype of a security framework for MES and related systems. 2011 has seen an explosion of security issues through a variety hacking techniques. This is a persistent and advancing threat as malicious code is being developed with growing sophistication. As has been seen with Stuxnet, designer worms are now being written with specific intent. Through a greater understanding of potential security flaws in manufacturing systems design, this growing threat can be mitigated.

Html 5 has a very strong potential to replace Flash and Silverlight in the foreseeable future. This powerful emerging technology could be a viable method to replace the Java applet method of device communication that was prototyped in support of this research. Removing the need for Java is desirable so that there are no pre-requisite software installation requirements apart from a browser, in supporting an optimal web based MES.

References

- Adaptive Path (2009), *HTTP Versus AJAX Comparison*. [IMAGE ONLINE] Available at: http://www.adaptivepath.com/images/publications/essays/AJAX-fig1.png [Accessed: 12 June 2009].
- Akamai (2010), *Content Delivery Networking*. [ONLINE] Available at: http://www.akamai.com/html/about/index.html [Accessed: 13 January 2010].
- Bilconference (2010), *The Humane Interface.* [ONLINE] Available at: http://bilconference.pbworks.com/w/page/7143783/The%20Humane%20Interface [Accessed: 10 June 2010].
- Bowden, R. et al. (1994), *Shop Floor Control Systems: From Design to Implementation*, Chapman and Hall
- Brown, A. and Hellerstein, J. (2005), 'Reducing the Cost of IT Operations.Is Automation Always the Answer?', *IBM Thomas J. Watson Research Center, Hawthorne*, New York, 10532
- Business Week (2004), 'Medical Devices: The Wrong Way To Run A Recall', *August 30 Issue*
- Chiu, Shean-Shyong and Luh, Yih-Ping (2001), 'Architecture Design of Shop Floor Information System', *Department of Mechanical Engineering National Taiwan University, Taipei, Taiwan, ROC*
- Clark, T. (2003), *Designing Storage Area Networks: A Practical Reference for Implementing Fibre Channel and IP SANs*, Second Edition, Addison Wesley
- Computer Science and Telecommunications Board (2005) *Information Technology for Manufacturing: A Research Agenda*, National Academy Press
- Crockford, D. (2009), *The JavaScript Minifier*. [ONLINE] Available at: http://crockford.com/JavaScript/jsmin [Accessed: 18 July 2009].
- CWE Institute (2009), *2008 CWE/SANS Top 25 Most Dangerous Software Errors*. [ONLINE] Available at: http://cwe.mitre.org/top25 [Accessed: 12 March 2009].
- ELO Touchsystems (2011), *Touchscreen Application Tips*. [ONLINE] Available at: http://www.elotouch.com/Resources/10tips.asp [Accessed: 19 Jan 2011].
- Erbschloe, Michae. (2005), *Trojans Worms and Viruses*, Elsevier Inc.

- Esposito, D. (2005), 'Take Advantage of ASP.NET Built-in Features to Fend Off Web Attacks', *MSDN ASP.NET Technical Article*
- Feng, S.C. and Zhang, Y. A. (1998), 'A Manufacturing Planning and Execution Software Integration Architecture', *Globalization of Manufacturing in the Digital Communications Era of the 21st*, Springer; illustrated edition
- Forfás (2010), *Review of Labour Cost Competitiveness*, Forfás
- Frontend (2011), *Designing For Touch-screen Kiosks*. [ONLINE] Available at: http://www.frontend.com/products-digital-devices/designing-for-touch-screen-kiosks.html [Accessed: 16 October 2011].
- Galitz, W. (2007), *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*, John Wiley & Son
- Gifford, C. et al. (2007*), The Hitchhiker's Guide to Manufacturing Operations Management: ISA-95 Best Practices Book 1.0*, International Society of Automation
- Google (2011), *SPDY: An experimental protocol for a faster web*. [ONLINE] Available at: http://www.chromium.org/spdy/spdy-whitepaper [Accessed: 01 December 2011].
- Griffiths, B. and Cecelja, F. (2002), *Manufacturing Information and Data Systems: Analysis, Design and Practice*, Penton Press
- Hart, M. and Jesse, S. (2004), *Oracle Database 10g: High Availability with RAC Flashback & Data Guard*, McGraw-Hill/Osborne
- Hayes, J. H. (2001), 'Affects on Maintenance of Web Software Applications'*, Laboratory for Advanced Networking Computer Science*
- Hayes, J. H. and Cai, H. (Hui) (2001), 'Software Engineering and Building Multi-Tier E-Business Applications Using Web Services Technology: Forget Everything You Know?', *Laboratory for Advanced Networking Computer Science*
- IBM (2000), *'Honeywell's Award-winning Approach to Web-based Development'*, IBM Corporation Whitepaper
- Internet Security (2011), *Amazon EC2 Cloud Servers Used to Hack PlayStation Network*. [ONLINE] Available at: http://www.internet-security.ca/internet-security-news-archives-031/amazon-ec2-cloud-servers-used-to-hack-sony-playstation-network.html [Accessed: 12 August 2011].

- Johns, M. and Winter, J. (2006), 'RequestRodeo: Client Side Protection against Session Riding', *OWASP Europe Conference*, Leuven, Belgium, May 2006
- Kahl, B. and Vasovski, S. (2004), 'The August 2003 Blackout', *Seminar in Electric Power Systems*
- Kasvi, J.J. and Vartiainen, M. (2000), 'Performance Support on the Shop Floor', *Performance Improvement,* 39 (6) 2000
- Kletti, J. (2007), *Manufacturing Execution System - MES,* Springer
- Koch, A. and Amrehn, E. (2005) 'Linux Software Management with Aduva OnStage', *IBM Redbooks paper*
- Lee, D. et al. (2007), 'Web Browser Device Communication for Shop Floor Information Systems', *24th International Manufacturing Conference*, Waterford Institute of Technology
- Leveson, N. G. and Turner, C. S. (1993), 'An Investigation of the Therac-25 Accidents', *Computer*, July, 18-41, 1993.
- Lewis, T G (2004), *Critical Infrastructure Protection in Homeland Security: Defending a Networked Nation*, Wiley-Interscience
- Lidwel, W. et al. (2010), *Universal Principles of Design*, Rockport Publishers Inc
- Litchfield, D. (2006), *Dangling Cursor Snarfing: A New Class of Attack in Oracle,* NGSSoftware Insight Security Research (NISR) Publication
- Loftus, E. (2003), 'Perception', *Nature Reviews, Neuroscience*, March 2003 volume 4
- Managing Automation (2011), *The MES technology Landscape*. [ONLINE] Available at: http://www.managingautomation.com/techmatch/mes-technology-landscape.aspx [Accessed: 21 October 2011].
- McConnell, S. (2004), *Code Complete,* 2nd Edition, Microsoft Press
- MESA International (1994), *The Benefits of MES: A Report from the Field,* MESA Whitepaper #1
- MESA International (1997a), *MES Functionalities & MRP to MES Data Flow Possibilities,* MESA Whitepaper #2
- MESA International (1997b), *MES Explained: A High Level Vision,* MESA Whitepaper #6

- Meschkat, S. et al. (2007), 'Keeping the Web in Web 2.0, An HCI Approach to Designing Web Applications', *CHI 2007 Course Notes from Google*
- Microsoft (2008), *Standards compliance updates in Internet Explorer 8*. [ONLINE] Available at: http://msdn.microsoft.com/en-us/library/dd433047%28v=vs.85%29.aspx [Accessed: 19 Feb 2009].
- Microsoft Windows Clustering White Paper (2003), 'Server Clusters: Architecture Overview', *Microsoft Windows Server 2003 Technical Article*
- MIS Services (2010), *The Quality Triangle*. [IMAGE ONLINE] Available at: http://www.mis-services.com/_images/services/triangle.gif [Accessed: 11 Mar 2010].
- MSNBC (2010), *Malware Implicated in Fatal Spanair Plane Crash*. [ONLINE] Available at: http://www.msnbc.msn.com/id/38790670/ns/technology_and_science-security/#.TsJ7RVZ8E80 [Accessed: 19 Feb 2011].
- Networked Elements (2010), *VPN Overview*. [IMAGE ONLINE] Available at: http://www.networkelements.co.uk/media/vpn.gif [Accessed: 12 May 2010].
- Nielsen, J. (2010), *Ten Usability Heuristics.* [ONLINE] Available at: http://www.useit.com/papers/heuristic/heuristic_list.html [Accessed: 10 June 2010].
- Oracle (2010), *Strategic Acquisitions*, [ONLINE] Available at: http://rxtx.qbang.org/wiki/index.php/Main_Page [Accessed 15 November 2010].
- Oracle Documentation (2010), *Oracle RAC Overview*. [IMAGE ONLINE] Available at: http://download.oracle.com/docs/cd/E12840_01/wls/docs103/jdbc_admin/wwimages/jdbc_oracle_rac.gif [Accessed: 16 November 2010].
- Oracle Documentation (2011), *Database Connection Pooling*. [IMAGE ONLINE] Available at: http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28427/img/connection_pooling.gif [Accessed: 12 May 2011].
- Oracle Technology Global Price List (2008), *Software Investment Guide 2008*

- Particletree (2010), *Fitt's Law Target Area Acquisition*. [IMAGE ONLINE] Available at: http://particletree.com/examples/fitt/f1.gif [Accessed: 25 June 2011].
- Particletree (2011), *Fitt's Law*. [IMAGE ONLINE] Available at: http://particletree.com/examples/fitt/f0.gif [Accessed: 25 June 2011].
- Passin, T. B. (2004), *Explorers Guide to the Semantic Web*, Manning Publications Co.
- Patterson, D. A. (2002), ‘A Simple Way to Estimate the Cost of Downtime’, *LISA Conference 2002*
- PC Advisor (2009), *Security Firm Kaspersky's Website Hacked*. [ONLINE] Available at: http://www.pcadvisor.co.uk/news/security/110584/security-firm-kasperskys-website-hacked/ [Accessed: 23 March 2009].
- Pinheiro, E. et al. (2007), ‘Failure Trends in a Large Disk Drive Population’, *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST’07)*
- Privacy Rights Clearinghouse (2011), *Chronology of Data Breaches*. [ONLINE] Available at: http://www.privacyrights.org/data-breach [Accessed: 12 March 2011].
- Project Fondue (2009), *CSS Sprite Generator*. [ONLINE] Available at: http://spritegen.website-performance.org/ [Accessed: 10 August 2009].
- Ricci, I. (2009), *CSS Sprite Sample*. [IMAGE ONLINE] Available at: http://www.ignacioricci.com/tutorials/sprites/images/mySprite.png [Accessed: 10 August 2009].
- Riehle, D. and Züllighoven, H. (1996), *Understanding and Using Patterns in Software Development*, Theory and Practice of Object Systems volume 2 issue 1 1996.
- Roessingh, J.J.M and Hilburn, B.G. (2000), *The Power Law of Practice in Adaptive Training Applications*, National Aerospace Laboratory NLR-TP-2000-308
- RXTX (2011), *Main page*, [ONLINE] Available at: http://rxtx.qbang.org/wiki/index.php/Main_Page [Accessed: 25 August 2011].

- SANS Institute (2011), *2011 CWE/SANS Top 25 Most Dangerous Software Errors*. [ONLINE] Available at: http://www.sans.org/top25errors/ [Accessed: 11 November 2011].
- SAP Design Guild (2000), *Interaction Design Guide for Touchscreen Applications*. [ONLINE] Available at: http://www.sapdesignguild.org/resources/tsdesigngl/TSDesignGL.pdf [Accessed: 19 June 2010].
- Sarter, N.B., Woods, D.D and Billings, C.E. (1997), Automation Surprises, *Handbook of Human Factors and Ergonomics second edition*, Wiley
- Scholten, B. (2009), *MES Guide for Executives: Why and How to Select, Implement, and Maintain a Manufacturing Execution System*, International Society of Automation
- Schroeder, B. and Gibson, G. A. (2007), 'The Computer Failure Data Repository (CFDR)', *Proceedings of the 2006 ACM/IEEE conference on Supercomputing*
- Science aid (2010), *Atkinson and Shiffrin Memory Construct*. [IMAGE ONLINE] Available at: http://scienceaid.co.uk/psychology/cognition/images/msm.png [Accessed: 12 May 2011].
- Simply Psychology (2013), *Ebbinghaus's Serial Position Effect*. [IMAGE ONLINE] Available at: http://www.simplypsychology.org/serialposition.jpg [Accessed: 21 March 2013].
- Security Focus (2003), *Slammer Worm Crashed Ohio Nuke Plant Network*. [ONLINE] Available at: http://www.securityfocus.com/news/6767 [Accessed: 02 May 2009].
- Security Focus (2004), *Tracking the Blackout Bug*. [ONLINE] Available at: http://www.securityfocus.com/news/8412 [Accessed: 03 December 2011].
- Service Architecture (2011), *Web Services and Service-Oriented Architectures*. [IMAGE ONLINE] Available at: http://www.service-architecture.com/images/web_services/service-oriented_architecture_basics.jpg [Accessed: 16 November 2011].
- Shalloway, T. (2001), *Design Patterns Explained: A New Perspective on Object-oriented Design (Software Patterns Series),* Addison Wesley

- Shaw, M. J. (2001), *Information-Based Manufacturing: Technology, Strategy and Industrial Applications*, Kluwer Academic Publishers
- Shaw, W. T. (2006), *Cybersecurity for Scada Systems*, PennWell Books
- Shneiderman, B. (2004), *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison-Wesley
- Shodanhq (2011), *Expose Online Devices*. [ONLINE] Available at: http://www.shodanhq.com/ [Accessed: 11 August 2011].
- Silicon Republic (2010), *Cyber attack - Stuxnet Worm Hits Iranian Nuclear Plant*. [ONLINE] Available at: http://www.siliconrepublic.com/strategy/item/17911-cyber-attack-stuxnet-worm [Accessed: 11 November 2010].
- Smith, P. G. (1999), *From Experience: Reaping Benefit from Speed to Market*, Elsevier Science Inc.
- Softpedia (2011), *Cyber Windows 8 Touch Interactions*. [ONLINE] Available at: http://news.softpedia.com/news/Windows-8-Touch-Interactions-225740.shtml [Accessed: 01 November 2011].
- Spera, J. (2005) 'Shop Floor Information Systems Emerging as Commercial, Off-The-Shelf Technology', *Aegis Industrial Software Corporation Philadelphia, Pennsylvania, United States*
- Squid Cache (2010) *Optimizing Web Delivery*. [ONLINE] Available at: http://www.squid-cache.org [Accessed: 10 January 2010].
- Sullivan, W. G. and Ahmad, M. M. (1997), *Flexible Automation and Intelligent Manufacturing,* Begell House Publishers, Inc
- Swift, M. M et al. (2005),'Improving the reliability of commodity operating systems', *ACM Transactions on Computer Systems*, Volume 23 Issue 1, February 2005
- Symantec Internet Security Threat Report (2008), *Trends for July–December 07 Volume XII*
- Symantic (2006), *IT Controls Reference 2006*
- Teach PE (2010), *Hick's Law*. [IMAGE ONLINE] Available at: http://www.teachpe.com/images/psychology/hicks_law.jpg [Accessed: 25 August 2011].

- Teunis, Gerri. et al. (1998), 'A New Architecture for Flexible Shop Control Systems', *The European Conference on Integration in Manufacturing, Göteborg, Sweden, October 1998*
- The Economist (2008), *Economies of Scale and Scope*. [ONLINE] Available at: http://www.economist.com/node/12446567 [Accessed: 23 August 2001].
- The Register (2011), *Dozens of Exploits Released for Popular SCADA Programs*. [ONLINE] Available at: http://www.theregister.co.uk/2011/03/22/scada_exploits_released/ [Accessed: 11 August 2011].
- United States Congress (2004), 'Virtual Threat, Real Terror', *Technology, and Homeland Security United States. Congress. Senate. Committee on the Judiciary. Subcommittee on Terrorism*, For sale by the Supt. of Docs., U.S. G.P.O
- Usability First (2011), *Meyer's Law*. [ONLINE] Available at: http://www.usabilityfirst.com/glossary/meyers-law/ [Accessed: 19 Jan 2011].
- V3 (2011) *Amazon Web Services Suffers Another Outage as Irish Datacentre Woes Continue*. [ONLINE] Available at: http://www.v3.co.uk/v3-uk/news/2100342/amazon-web-services-suffers-outage-irish-datacentre-woes-continue [Accessed: 11 August 2011].
- Van Der List, E. et al. (2007), *Professional Web 2.0 Programming*, Wiley Publishing Inc.
- Vestring, T. et al. (2005), *Making the Move to Lowcost Countries*, Bain & Company Inc.
- Vijayan, J. (2000), *Manufacturing Execution Systems*. [ONLINE] Available at: http://www.computerworld.com/s/article/47639/Manufacturing_Execution_Systems?taxonomyId=133&pageNumber=2 [Accessed: 09 Feb 2008].
- Vistatec (2011), *Online Traceroute*. [ONLINE] Available at: http://benjamin.deg.vistatec.ie/cgi-bin/trace [Accessed: 10 January 2011].
- W3C (2004), *W3c Technology Governance*. [IMAGE ONLINE] Available at: http://www.w3.org/2004/10/RecsFigure-Smaller.png [Accessed: 15 November 2011].

- W3C (2009), *CSS Validation Service*. [ONLINE] Available at: http://jigsaw.w3.org/css-validator [Accessed: 13 July 2009].
- W3C (2009a), *CSS Validation Service*. [ONLINE] Available at: http://jigsaw.w3.org/css-validator [Accessed: 13 July 2009].
- W3C (2009b), *eXtensible Markup Language (XML)*. [ONLINE] Available at: http://www.w3.org/XML/ [Accessed: 14 Feb 2009].
- W3C (2011), *Mission Statement*, [ONLINE] Available at: http://www.w3.org/Consortium/mission [Accessed: 12 June 2011].
- Walker, J. M. (1996), *Handbook of Manufacturing Engineering*, Marcel Dekker, Inc.
- WaSP (2009), *Acid2: The Guided Tour.* [ONLINE] Available at: http://www.webstandards.org/action/acid2/guide/ [Accessed: 15 July 2009].
- WaSP (2010), *What are web standards and why should I use them.* [ONLINE] Available at: http://www.webstandards.org/learn/faq/ [Accessed: 28 June 2010].
- Web Designish (2011), *14 Best JavaScript Frameworks*. [ONLINE] Available at: http://www.webdesignish.com/14-best-javascript-frameworks.html [Accessed: 03 August 2011].
- Website Optimization (2009), *Web Page Analyzer*. [ONLINE] Available at: http://www.websiteoptimization.com/services/analyze/ [Accessed: 20 August 2009].
- Wikipedia (2008), *Software Design Pattern*. [ONLINE] Available at: http://en.wikipedia.org/wiki/Software_design_pattern [Accessed: 09 Feb 2008].
- Wikipedia (2009), *Web 2.0*. [ONLINE] Available at: http://en.wikipedia.org/wiki/Web_2.0 [Accessed: 05 December 2009].
- Wikipedia (2010), *Fitt's Law*. [ONLINE] Available at: http://en.wikipedia.org/wiki/Fitts%27s_law [Accessed: 10 June 2010].
- Wikipedia (2010b), *Power Law of Practice.* [ONLINE] Available at: http://en.wikipedia.org/wiki/Power_Law_of_Practice [Accessed: 10 June 2010].
- Zakas, N. C. et al. (2006), *Professional AJAX*, Wiley Publishing, Inc

- ZDNet (2008), *SSL broken! Hackers Create Rogue CA Certificate Using MD5 Collisions*. [ONLINE] Available at: http://blogs.zdnet.com/security/?p=2339 [Accessed: 12 March 2011].
- Zhang, I. X. (2007), 'Economic consequences of the Sarbanes–Oxley Act of 2002', *Journal of Accounting and Economics 44* (2007) 74–115

Appendices

A. CWE/SANS top 25 security flaws

The following sections list the 25 CWE/ SANS error types (SANS Institute 2011) and discusses how they relate to MES architecture.

A.1 Improper Input Validation

Incorrect input validation can lead to vulnerabilities when attackers modify their inputs in unexpected ways. All user-entered information should be checked and cleaned when submitted. For example if the system is expecting a number and a user enters a character, indeterminate results may arise. Bounds should also be put in place so that abnormal values can be detected and handled accordingly. For example, a user entering the number of parts at a stage in manufacturing should not be a negative value. Many of today's most common vulnerabilities can be eliminated, or at least reduced, with strict input validation. Input validation is quite possibly the most import error type as a gateway to leverage the other errors in this listing. Fortunately, most of the modern development languages have ready built components that can be quickly and easily utilised in code.

A.2 Improper Encoding or Escaping of Output

Improper output encoding is at the root of most code injection attacks. An attacker can potentially modify the commands that are sent to parts of an application. The application expects basic text and handles it as such, but when there is the facility to enter text/data, newly specified code can be injected maliciously. The resulting problems can range from unhandled errors that cause crashes, to the whole system being compromised. This type of attack predominantly affects web architectures. As a basic example Code Figure A-1 shows what could be entered into a form textbox – when a web page goes to print out what has just been entered, the script is executed:

```
<script>alert('This is a basic hack demonstration');</script>
```

Code A-1. Hack on improper escape encoding

Similarly to the Improper Input Validation error type, many of the modern development languages have ready built components available to easily prevent this type of problem, such as the following example A-2:

```
Page.Validate=True
```

Code A-2. .NET method to prevent improper encoding

A.3 Failure to Preserve SQL Query Structure

In this type of exploit, 3rd parties alter SQL being sent to a systems database. Innocuous lookup queries that list a small set of non-sensitive data and print on-screen can be modified to list out database structure. Gaining this information then enables a hacker to subvert security controls and perform malicious acts for monetary gain, system disruption and other motivations. More commonly known as SQL Injection attacks, this can be performed to steal sensitive information, delete and modify data. URLs are the most common entry point for SQL Injection because of the statelessness architecture of HTTP. The querystring can be manually edited easily. Textbox content is another common entry point. As an example the Code Figure A-3 illustrates a typical query behind a logon screen that gives access to a protected section of a website:

```
Select username, password from system_users where username =
'X' and password = 'Y'
```

Code A-3. Typical SQL used for website login

The value of X will be the user entered text of the user textbox. The value of Y will be the user entered text of the password textbox. Code Figure A-4 illustrates a hack that can be entered into the user textbox:

```
' or 1=1 and rownum < 2 --
```

Code A-4. SQL Injection hack

This then changes the behind the scenes SQL as illustrated in Code Figure A-5:

```
Select username, password from system_users where username =
'' or 1=1 and rownum < 2 -- and password = 'Y'
```

Code A-5. Final SQL after SQL Injection hack

The modified SQL will allow the attacker to logon to the system under the permissions of the first entry in the database without knowing their password. This is just a basic technique and can be modified to perform a range of actions. To protect against this type of exploit, manually entered text should be encoded for inverted commas before SQL is sent for execution. In addition, the database account used should adhere to the minimum permissions principal.

A.4 Failure to Preserve Web Page Structure

This exploit is more commonly known as XSS (Cross-site scripting) and is again made possible because of the stateless nature of HTTP. Attackers can inject JavaScript and other executable content into legitimate hosted web sites. Then as clients visit the altered page, they become unwitting participants in propagating further malicious activities. This type of attack accounted for 80% of attack methods in 2007 (Symantec Internet Security Threat Report, 2008). It is a particularly attractive method of attack because it is very hard to trace due to the multitude of legitimate users that are stealthily used to perform the actual attack. This enables the orchestrators to conceal themselves as they do not play a part in the attack itself. In contrast, worms and botnets create a lot of unusual network traffic that can be detected by analysing network traffic signature and composition.

The clients may also become victims themselves with the data theft of personal information that could be used for profit such as credit card details. It is also possible to fool the user into thinking they are in legitimate websites by redirecting them to a site that looks the same as the one they requested. This method is called phishing where the user is fooled into entering personal details.

There is no single way to prevent attacks like these as it can be implemented through exploitation of a gateway error type that is used to give initial entry. The best way to mitigate against XSS is to minimise the surface area of attack and periodically verify that the code in place is the original specification. ASP.NET has a good defence by having web code in compiled DLL files that makes it harder for attackers to implement XSS. Ideally the ASP.NET DLLs should be obfuscated for maximum security.

A.5 Failure to Preserve OS Command Structure

This error type is more commonly known as OS Command Injection. While applications can use the power of dynamic scripting to interact with the Operating System, it also makes it possible for an attacker to inject malicious code that can alter the OS commands being executed. This method enables the system to be fully compromised through the escalation of permissions. Developers should be particularly careful of code they write when it involves running host commands. Like most error types, there are multiple points of attack and no single method for prevention. For example the Code Figure A-6 lists a script to send an alert message is fully dynamic:

```
Shellexecute('net send somepc "The shift has ended please log out of your system"')
```

Code A-6. Dynamic script execution

An attacker could potentially modify the script to do something malicious as illustrated in Code Figure A-7:

```
Shellexecute('format c:\')
```

Code A-7. Malicious dynamic script execution

To lock down the ability of the dynamic code, constants should be used as illustrated in Code Figure A-8, so the scope to make changes is limited:

```
NET_SEND_START_COMMAND = 'net send'
NET_SEND_END_COMMAND= '"The shift has ended please log out
of your system"'
Shellexecute(NET_SEND_START_COMMAND & 'somepc' &
NET_SEND_END_COMMAND)
```

Code A-8. Dynamic script execution

A.6 Cleartext Transmission of Sensitive Information

Most applications have some form of security control that enables users to gain escalated privileges by providing credentials. There are also other types of personal information such as credit card details used on systems. The transmission of this private information is passed from the client to the remote server. As can be seen with a Tracert command (Code Figure 2-24), information travelling between nodes over the internet can pass through a number of points and networks. These intermediate points are controlled by 3rd parties and cannot be guaranteed secure. Only one point in the infrastructure chain needs to be compromised for unintended parties to gain access to the passing information; this is commonly known as a Man In the Middle (MIM) attack. The unencrypted transmission of credentials can easily be intercepted by malicious parties, as the snooping of traffic is possibly one of the easier hacking activities. So as a general rule, all personal information should be encrypted before transmission. Different forms of encryption have varying strengths and weaknesses so the appropriate encryption algorithm should be chosen for specific tasks. Cryptology is a specialised field so custom developed encryption methods should not be used as they could be easily cracked.

A.7 Cross-Site Request Forgery

Cross-site request forgery (CSRF) is similar to XSS in that users are tricked into executing 3rd parties code. It is slightly different as it exploits the trust a website has for a user whereas XSS exploits the trust a user has for a website. In essence, an attacker masquerades as a legitimate user and gains all the potential access that the user has. It is desirable for malicious activities to gain as much access as possible so gaining administrator privileges could lead to the system being completely compromised. If a user is logged into a particular website, there is potential for an attacker to trick the user's browser into making a request to a specific URL from a page that has been compromised by the attacker. In this manner a task is performed and recorded as having been requested by the user, even though an attacker was responsible. To do this, an attacker must construct tags that match what would otherwise be a normal transaction. A very basic CSRF exploit could be crafted by having an HTML image perform some hidden action on the client side. Code Figure A-9 illustrates code that performs a transfer to an attackers bank account:

```
<img
src="http://somebank.com/transferfunds?account=me&amount=10
00000&for=attacker">
```

Code A-9. Simple CSRF exploit

While GET HTTP requests are easier to modify, they are not the only attack vector as POST commands can also be crafted, as illustrated in Code Figure A-10:

```
<script>
 var post_data = 'account=me&amount=1000000&for=attacker ';
 var xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
 xmlhttp.open("POST", ' http://somebank.com/transferfunds',
true);
 xmlhttp.onreadystatechange = function () {
 if (xmlhttp.readyState == 4)
 {
 alert(xmlhttp.responseText);
 }
 };
 xmlhttp.send(post_data);
 </script>
```

Code A-10. POST CSRF request

A commonly suggested method of preventing this exploit is to check browser refer headers. Unfortunately this offers little protection as headers can be spoofed to whatever is expected by the legitimate site. Often this type of attack is made possible by malicious parties first exploiting other security weaknesses, so that they can then proceed to compromise legitimate web pages. Some protection is afforded by newer generation browsers that perform blacklisting of malicious websites. This is an effective method to prevent unseen threats being conducted on user's browsers. Using randomly generated form numbers is also a strong method to prevent this attack type (Johns and Winter 2006:6).

A.8 Race Condition

A race condition occurs when multiple processes access and manipulate the same data concurrently, and the outcome of the execution depends on the particular order in which each process accesses the data. This error type is usually exploited after other gateway error types are taken advantage of, such as SQL Injection. If an attacker can compromise one of the interacting processes then it may be possible to interrupt the correct processing sequence and cause system errors, complete system failure and even data corruption. Attackers typically use this error type to conduct Denial Of Service (DOS) and disruption, as the potential for monetary gain is limited. Multithreaded applications such as multi-user web apps are particularly prone to potential race condition abuse as there are multiple concurrent activities taking place. In addition to threading race condition, there can also be other forms whereby one process is half way through an execution task when another process changes some shared variables or data that case the execution of the initial process to become invalid. There is no one method for mitigation of this type of exploit as developers need to be watchful for the potential of invalid sequencing of interacting processes. Sound development techniques such as locking and synchronising threads may also help avoid this issue. While the ramifications of this error type can be induced by attackers, it can also happen due to unusual circumstances, as was the case in the 2003 North American power blackout (Security Focus 2004).

A.9 Error Message Information Leak

Unwittingly, error messages designed to provide detailed debugging information for easier problem resolution could disclose invaluable information to an attacker. The information shown such as system architecture, file locations, software versions, sensitive database information and even server configuration would otherwise be hidden. Armed with very useful information, malicious 3rd parties could potentially circumvent security restrictions. While this information may have been invaluable in debug stages of testing, it should not be presented to the user in production systems. When errors do occur, the user should be notified with basic information and the detailed data stored in a database for administrator review. Attackers may also try induce errors by malforming input and URL querystrings. Mitigation against other error types such as input validation and character escaping will help reduce the attack vector for this error type. The following Figure A-1 demonstrates how an error provides information to path structure and web server software:

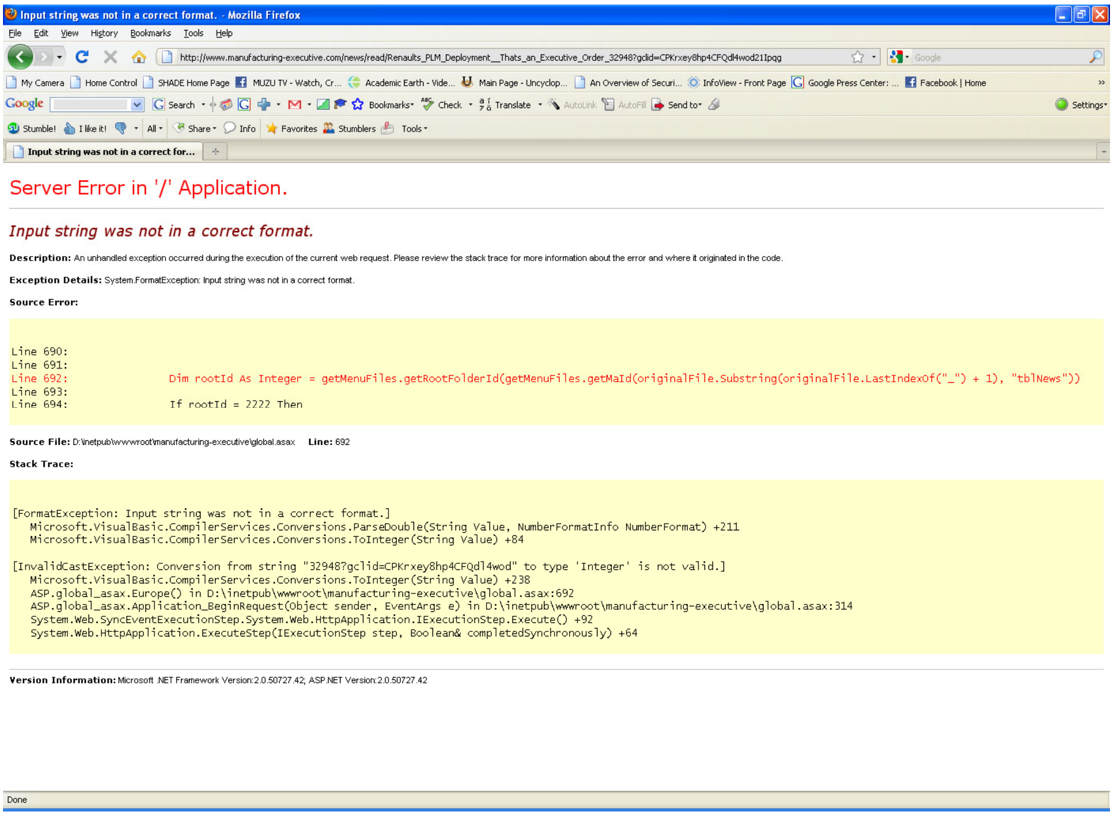

Figure A-1. Unfiltered error details transmitted to client web page

A.10 Failure to Constrain Operations within the Bounds of a Memory Buffer

Otherwise known as buffer overflow, this type of attack primarily afflicts lower-level programming languages such as C where more focus was traditionally given to programming efficiency than to security. Code developed in higher-level languages is also susceptible because they interact with lower level software such as the OS and other API's and interfaces. This is a gateway error type that can be used on its own to compromise systems. It enables attackers to run arbitrary code on a users system by injecting a longer buffer string of memory than the code caters for. The extra data when inserted overlaps in another area of the programs running memory. While not being an easy flaw to exploit, it enables a users system to become completely compromised. A malicious 3rd party can send these invalid commands to vulnerable installed software on the remote user. Vulnerable software can range from OS, browsers or any other executable software that resides on the client. To minimise threat from this type of exploit, the attack vector should be minimised by having only necessary software installed and also staying up to date with relevant security patches. Software programmers should also ensure they maintain memory buffer checks when writing and reviewing code. Microsoft OS versions have an intrinsic function called

DEP (Data Execution Protection) that protects against most forms of buffer overflows.

A.11 External Control of Critical State Data

All dynamic websites need to give user specific content that tracks user selections and preferences. In order to overcome the statelessness of HTTP, data needs to be transmitted on each page posting, with the unfortunate effect of opening up to potential hacking. This can be performed with hidden HTML form fields, database records, configuration files, querystrings, input parameters, registry keys and cookies. Any alteration of this post data could lead to attacker privilege escalation so should be encrypted to prevent tampering. As a basic example of how data can be changed, Figure A-2 depicts a Microsoft page analysed by firebug:

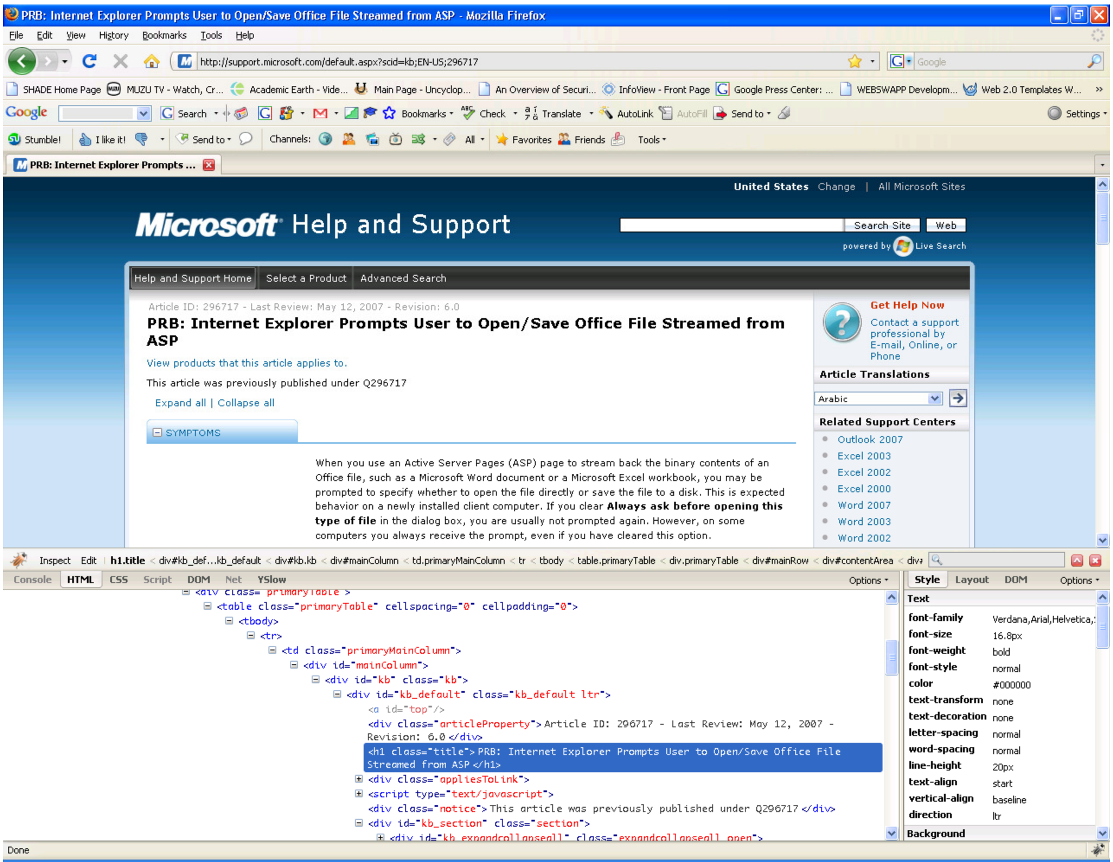

Figure A-2. Using Firebug to view with 3rd party web site code

The firebug tool is invaluable for web developers in debugging and optimising their code. It also has the power to modify rendered output in the browser. As a visual example, Figure A-3 illustrates where a text field has been changed in the following figure to red with the wording "Hacked content using debugging tool":

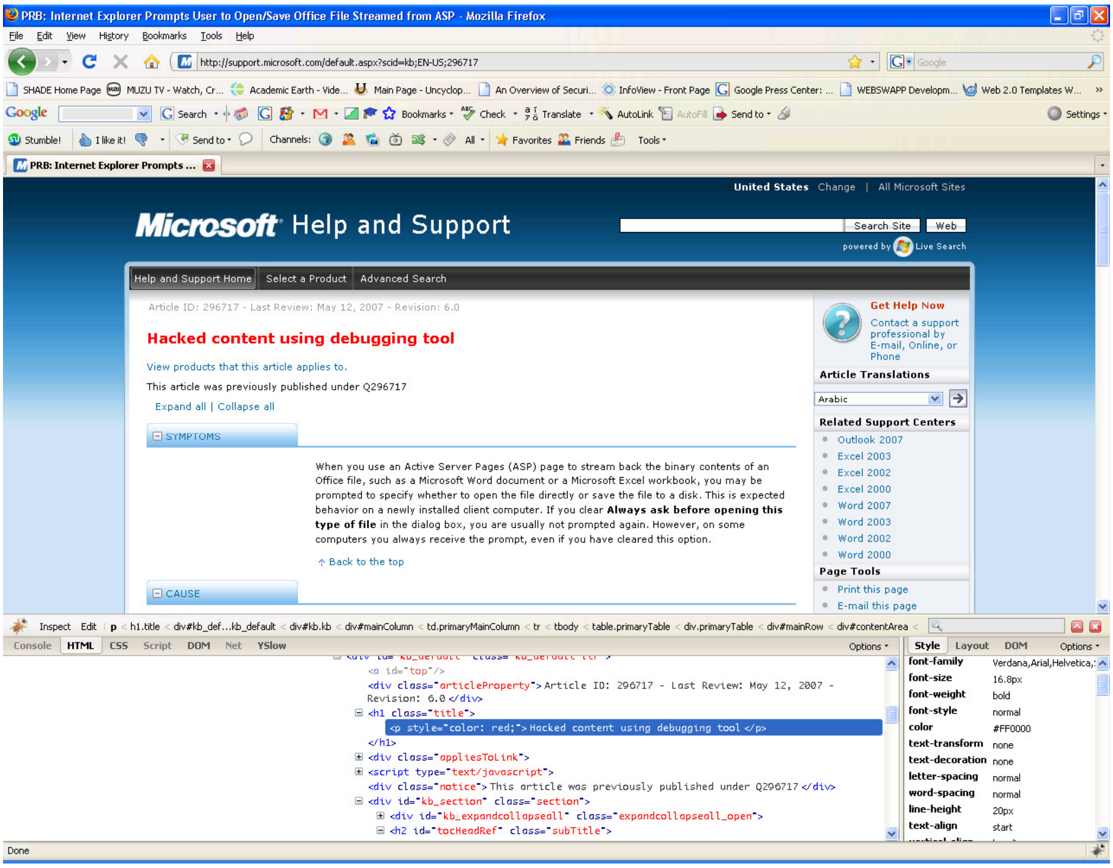

Figure A-3. Using Firebug to tamper with 3rd party web sites

Although this was just modification of a HTML text field, a hardcoded hidden HTML control can be changed in the same manner and posted back to the server. Code Figure A-11 illustrates a hardcoded hidden field setting a users access type:

```
<input type='hidden' name='USER_ACCESS_TYPE' value='basic' />
```

Code A-11. HTML code before alteration

Code Figure A-12 demonstrates what the hidden HTML field could be updated to using freely a available tool:

```
<input type='hidden' name='USER_ACCESS_TYPE'
value='administrator' />
```

Code A-12. HTML code after alteration

Technologies such as ASP.NET inherently maintain Viewstate in an encrypted format so are ideal for providing protection against modification of critical state data. Nevertheless developers should perform additional checks when using Viewstate data to display data and grant access. Mitigating against other exploit types such as input validation and proper escaping of data will also reduce the potential for this vulnerability type to be exploited. While server side validations increase the processing load on web servers, it gives a much greater level of security.

A.12 External Control of File Name or Path

Due to the dynamic nature of websites, paths to file locations are often constructed from multiple parts in code. It is important to implement safeguards to prevent gaining access to unintended directories. By simply adding ".." into a file path, a hacker would be able to navigate to a lower directory structure and potentially browse sensitive directories and files on the web server. There is even potential to construct Universal Naming Convention (UNC) links to shares on other servers that the webserver has access to. This error type is also vulnerable to URL manipulation. To minimise the threat of this exploit, the webserver application should run under a lower privilege OS account that adheres to the least privilege rule so that it does not have any unnecessary permissions. All entered data should be properly validated and special characters escaped.

A.13 Untrusted Search Path

Reusable libraries enable RAD and robust web applications. These libraries can be in the form of Server Side Includes (SSI), configuration files, JavaScript libraries or CSS styles. They are then plugged into web pages such as the Code Figure A-13 for a JavaScript resource:

```
<script language='JavaScript' src='SomeCodeFile.js' />
```

Code A-13. Server Side Include

There is potential for an attacker to modify the library and point to a maliciously crafted file. To prevent this, as many sections of search paths should be hardcoded as possible. Similar practices should be adhered to as the ones to defend against the "Failure to Preserve OS Command Structure". In reality, if an attacker was able to perform this type of exploit, they have already compromised the webserver and so have a multitude of methods to propagate anyhow.

A.14 Failure to Control Generation of Code

Dynamically generated code is a powerful and efficient way to create interactive, rich and customisable websites. Unfortunately this also opens up the potential for a form of malicious exploitation by 3rd parties, commonly known as Code Injection. HTML, JavaScript and VBScript are commonly generated and are the technologies that hackers can leverage. Again the reality of this error type would only occur if an attacker was already after performing some form of malicious action on the webserver and so would have a large choice of actions to perform further propagation. One of the key features of MES is in potential to configure to suit varying manufacturing processes without the need to recode. This requires code to be highly dynamic and so vigilance should be placed around these code elements. Dynamic code is often generated by issuing SQL commands to the database with substitutable variables. In this manner all manner of calculations can be performed at runtime. Some pseudo SQL for such an operation is illustrated in Code Figure A-14:

```
Select (birthday_date – now) as "days_remaining" from
person_information
```

Code A-14. SQL dynamic calculation example

A.15 Download of Code Without Integrity Check

Exploits surrounding the downloading executable code such as scripts and programs are not so relevant to the core MES architecture. However as the complexity of automated manufacturing grows there may be more of this type of activity taking place. Also when utilising 3rd party plug-ins such as Google's analytics, direct references point to vendors that could potentially be compromised to perform malicious activities. By openly using their services, the exploit could also be spread to the MES system. The exploit is centred around the possibility of a hacker modifying executable code while it is in transit so that malicious code is executed on the client. To prevent this from happening, a number of technologies such as VPN and digital signatures can be used to ensure there has been no tampering.

A.16 Improper Resource Shutdown or Release

System resources are transitive and should have a defined life span. They can consist of memory allocations, files, data structures, sessions, cookies and related constructs. Allocated resources exist to drive some dynamic function of an application or maintain a state for web applications. When these resources are no longer required, they should be disposed of so they can no longer be utilised. Otherwise attackers can take advantage of these resources to learn sensitive information or gain elevated privileges. For instance if database connections are not closed properly, a web application may start failing due to a limit being reached with the number of available connections. The following Java example provided by the SANS/CWE research study defines how to properly dispose of database connections:

```
try {
        Connection con =
        DriverManager.getConnection(some_connection_string)
}
catch ( Exception e ) {
        log( e )
}
finally {
        con.close()
}
```

Code A-15. Proper database connection closing in Java

An example of gaining elevated privileges can be demonstrated in Oracle database by what is called Dangling Cursor Snarfing (Litchfield, David 2006). In essence, code in Oracle PL/SQL does not close memory cursors properly in error situations or just in general. These cursors are left "dangling" in memory and are available to be "snarfed" by malicious 3rd parties. If a cursor was created by a user with high level access, a user with basic access could potentially gain access to privileged information contained within the cursor's recordset. As a sound development technique, performing timely garbage collection not only protects against malicious behaviour, but also ensures benefits application performance.

A.17 Improper Initialization

As with the "Improper Resource Shutdown or Release" exploit type, system resources are transitive and should have a defined life span. Before any of these constructs are used, they should be properly initialised by resetting to their default values, enabling the application to proceed as intended. If not initialised properly, there is potential for an attacker to inject malicious code and get the application to either crash or perform unintended operations. If constructs are used in security-critical operations, as in making an authentication decision, they could be modified to bypass security controls. The following Code Figure A-16 illustrates C declaration that does not set a default value for an object:

```
String objSecurityLogon;
```

Code A-16. Improper C variable declaration

Instead it should be initialised as shown in Code Figure A-17 so that if it is used before assignment, it will not take what was in memory previously or placed there by a malicious party:

```
String objSecurityLogon = null;
```

Code A-17. Correct C variable declaration

A.18 Incorrect Calculation

Attackers can manipulate input to make a program behave abnormally. A typical example would be to enter a value into a form that is not what was expected, such as entering a value greater than an integer can handle (i.e. over 32,767) creating an integer overflow. This seemingly innocuous action could at the very least cause a system error or worse make it conduct indeterminate actions like granting elevated access to the system. For MES systems this type of error could even be caused by innocent mistakes that lead to incorrect product quantities being recorded. There is also a possibility that business logic is violated causing a logical data corruption. Mitigation for this error type primarily depends on proper input validation. Actual defences will be development language specific but the underlying concept is the same.

A.19 Improper Access Control (Authorization)

System security control measures need to differentiate between different users and control what they are allowed to do. The principle of least access ensures that users only have the minimum privileges that enable to perform their required tasks and cannot perform any unauthorised actions. MES systems need to have fine grained security control to ensure only trained users can perform particular functions. Using well defined authentication frameworks such as Microsoft's Active Directory or OWASP ESAPI Access Control ensures the benefits are obtained from these proven effective tools.

In particular for access control, caching can be a security risk as copies of web pages are stored to client hard drives. A balance needs to be accommodated where sensitive information is not publicly accessible but also system performance does not suffer. If possible caching should be performed on the server and client browsers instructed to not cache such as illustrated previously in Code Figure 2-22. This error type is manifested by exploitation of a number of the items listed in the SANS/CWE top 25. By employing protection from these items, risk of unauthorised access will be minimised.

A.20 Use of a Broken or Risky Cryptographic Algorithm

On the outset, development of custom cryptographic algorithms may seem like a good idea as lots of arbitrary and unusual obfuscation can be implemented and because they are custom, not much may be know about it by 3rd parties. Unfortunately cryptography is a specialised field with many intricacies. What may seem like a good encryption method may be easily cracked. Even sophisticated methods of encryption like SSL can be subverted due to flaws in underlying encryption algorithms, such as happened with MD5 (ZDNet 2008). All modern development languages have ready-made encryption components that can be easily utilised in code. Currently SHA and its variants are trusted hashing algorithms and RSA and its variants are a trusted symmetric encryption algorithm, and therefore would be suitable for incorporation into MES where required.

A.21 Hard-Coded Password

Hard-coding passwords in an application are an extremely bad practice. This makes it difficult to change passwords as a recompilation and rollout of the updated version is required each time. Often passwords are not changed for convenience because of this, which then reduces the effectiveness of having a password. Over time passwords defuse into common knowledge so that it is no longer a secret for systems users. It is also possible to reverse engineer passwords from binaries. While obfuscation makes it harder to disassemble binaries, it is still possible uncover embedded passwords. Figure A-4 shows a VB.NET application with a hardcoded system password at the top of the code with a value of “SYSPASS99”:

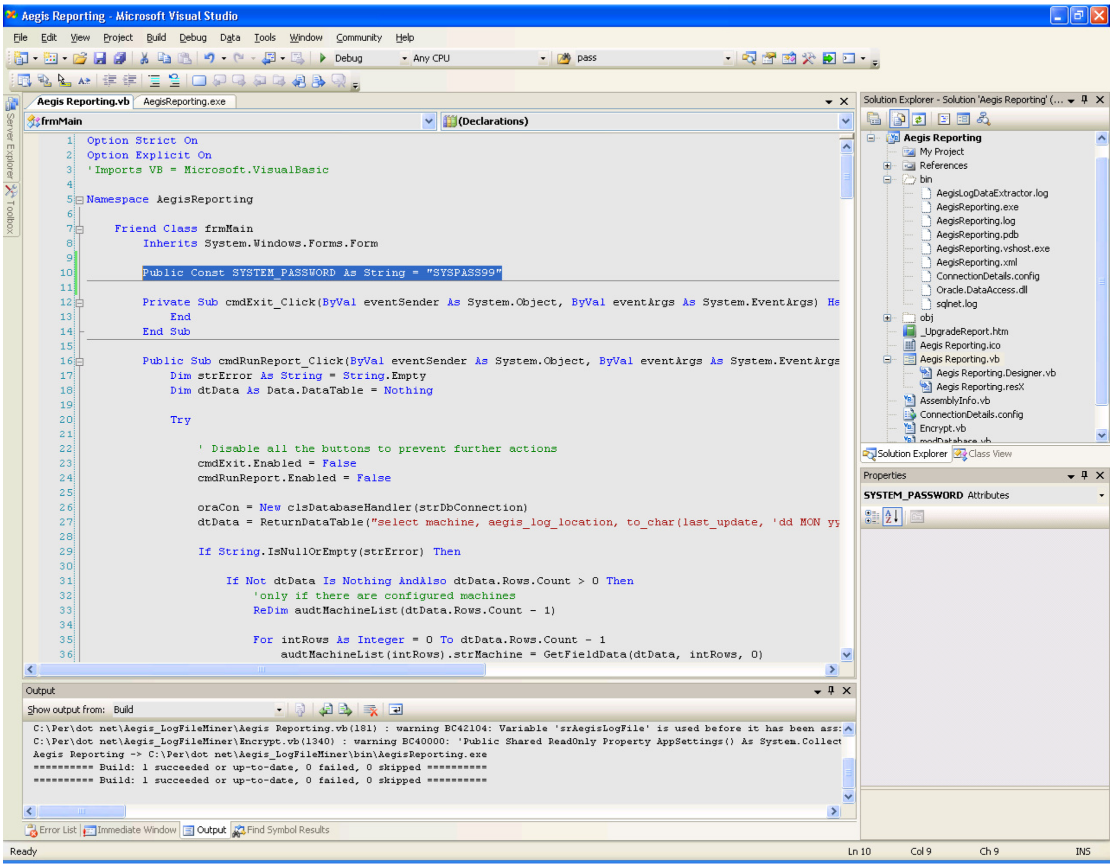

Figure A-4. Listing of a constant value hardcoded into a .NET project

By just peeking with a freely available binary editor it is possible to view what the hardcoded password is, as demonstrated in the following Figure A-5:

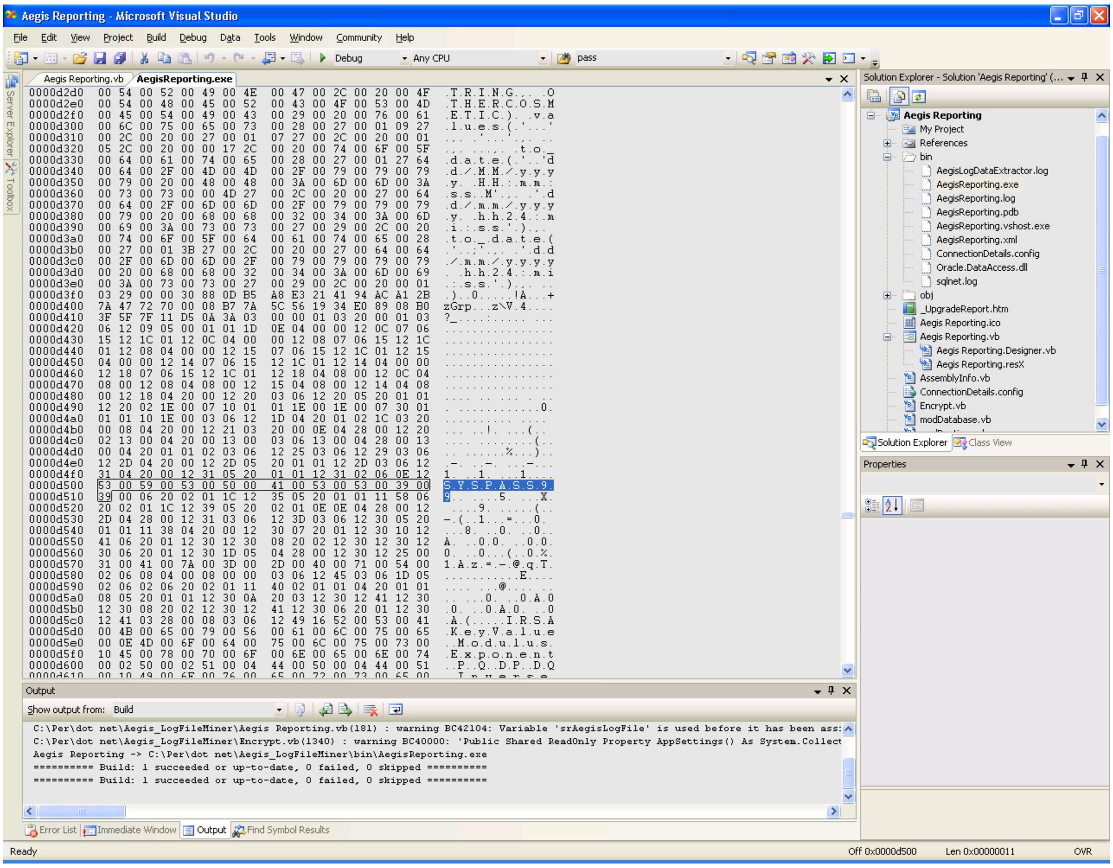

Figure A-5. Peeking of constant value hardcoded into a .NET assembly

As a solution to this issue, passwords should be held outside of the application and in an encrypted format. The RSA encryption algorithm would be suitable as it is infeasible to crack. RSA is also serial in that it enables the password to be decrypted, as opposed to hashing methods whereby the password cannot be retrieved.

A.22 Insecure Permission Assignment for Critical Resource

This vulnerability type is very similar to “External Control of Critical State Data”. The critical resource could be some configuration file, registry entry or database record holding information that enables access to system functionality, such as usernames and passwords. All configuration data should be encrypted to prevent malicious activity. As with most of the vulnerabilities listed, other gateway exploits need to be subverted before this can be taken advantage of. This vulnerability not only exists for systems and their configurations but also for its data. Since 2005, incidences of data breaches have escalated at a steady pace (Privacy Rights Clearinghouse 2011). Care should also be taken to protect data, reports and other form of information extracted from systems as these items could also be of benefit to potential hackers and even used directly for financial gain.

A.23 Use of Insufficiently Random Values

Random values are used to uniquely identify users and objects. The random value can then be used in a number of ways such as generating unique session IDs or temporary file names. These values are intended to be sufficiently random so that an attacker cannot guess them. Random number generation is not as simple as would be initially thought however. Devising a random number algorithm, known as Pseudo Random Number Generator (PRNG), may in fact yield a detectable signature that enables the guessing of numbers. Instead of attempting to create a new PRNG, there are a number of pre-existent frameworks in place. To ensure adequate protection from hacking, cryptographic measures should be taken to further prevent guessing of numbers. Cryptographically Secure Pseudo Random Number Generators (CSPRNGs) are therefore more secure than PRNG's, and should be used when looking to have uniquely identifiable numbers. Code Figure A-18 illustrates a C# example using the secure and random built-in RNG cryptographic provider:

```
byte[] bteRan = new byte[2];
RNGCryptoServiceProvider rng = new RNGCryptoServiceProvider();
rng.GetBytes (bteRan);
```

Code A-18. Correct random number generation

The use of Globally Unique Identifiers (GUIDs) is also prevalent in generating unique values. As governed by RFC4122, GUIDs are constructed alpha-numeric 128bit strings from a defined algorithm. While the generation GUIDs is sufficiently random, it is not as secure as CSPRNGs. GUIDs were designed to minimise the possibility of randomly generating the same value twice. While the uniqueness of each generated GUID cannot be guaranteed, the total number of unique keys (2 to the power of 128) is so large that the probability of the same value being generated twice is infinitesimally small. As can be seen from RFC4122 however, parts of the GUID are comprised of the time of generation. This increases the potential for a hacker guessing values and therefore should not be used for random values that enforce security elements. GUID generation should not be discounted though, as it is an excellent method of creating unique keys. These keys are very useful for creating unique primary keys in database tables. A VB.NET example of creating a GUID is illustrated in Code Figure A-19:

```
System.Guid.NewGuid().ToString()
```

Code A-19. GUID generation in VB.NET

An example of a GUID is illustrated in Code Figure A-20:

```
{a53e98e4-0197-4513-be6d-49836e406afa}
```

Code A-20. Generated GUID example

As computing power increases in line with Moore's Law, the implementation of CSPRNGs should be reviewed for effectiveness. In general, increasing the size of the key or seed can strengthen algorithms. With contemporary processing power, keys of 256bits are reasonably sufficient to prevent hacking.

A.24 Execution with Unnecessary Privileges

The recurring guidance throughout the top 25 list is that of granting the lowest amount of access necessary to perform a function. Hacking methodologies make use of leveraging even the smallest of foothold to gain greater access. The more access a hacker has control of, the more potential for malicious activities. By granting privileges on a finely grained basis, conducting unauthorised activities becomes a much harder task. In addition to unnecessary privileges, there may also be unnecessary services and functionality that should be turned off. For example, unused TCP/IP ports are often used as a backdoor. By turning off these unused services the surface area of attack is also minimised.

A.25 Client-Side Enforcement of Server-Side Security

Client-side validation of user input is a user friendly feature that gives instant feedback. JavaScript is the most predominantly used scripting language for client-side validation although some sites may opt for the use of VBScript. Client-side scripting cannot be depended upon for input validation however due to the fact it is being performed on a remote client and not the webserver. There are a number of ways in which the validation can be interfered with or turned off. An example of the Firefox browser option to turn off JavaScript can be seen in the following Figure A-6:

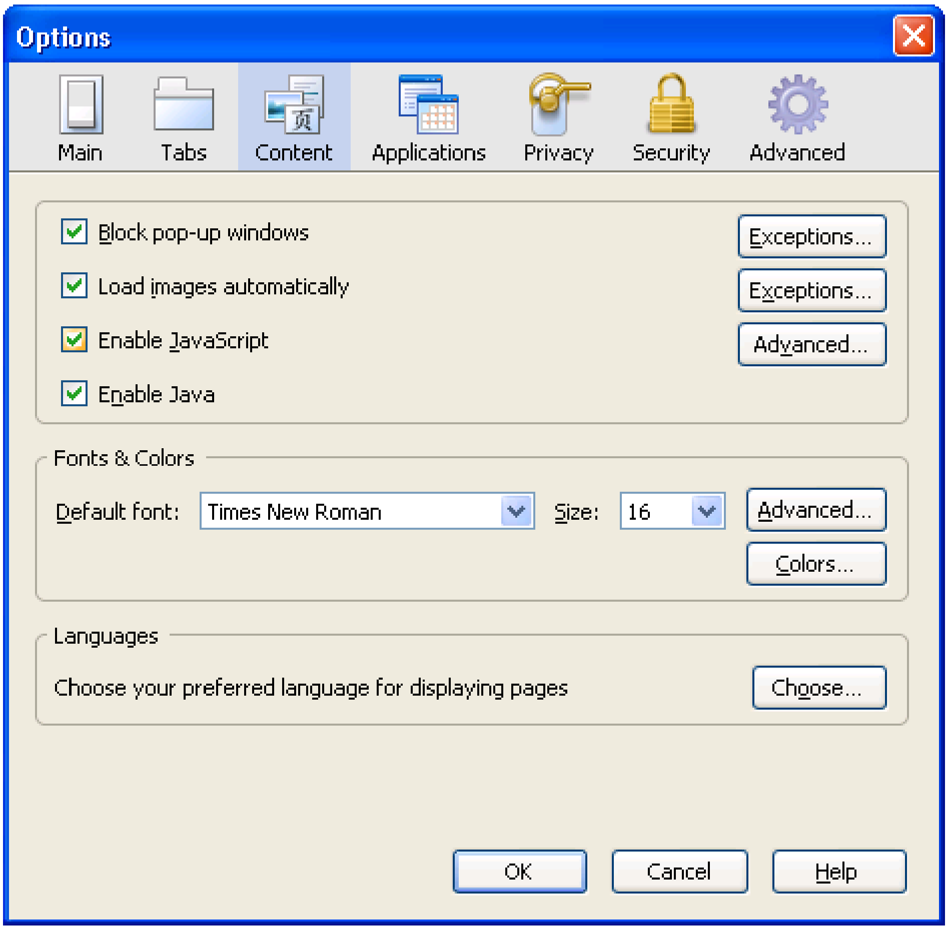

Figure A-6. Firefox option configuration page

Depending on what type of validation is being performed, it could be possible to bypass input validation and escaping of encoded characters. As a result, it is critical that server-side validation is performed to ensure controls cannot be maliciously subverted. It is still a good practice to maintain client-side validation for the purposes of usability and instant user feedback. Care should be taken if algorithms are the same on server and client, as can be done with using classic Application Server Pages and JavaScript. While this may be a good method to reduce the amount of code, attackers may be able to devise methods to circumvent controls as they can view the client-side JavaScript source. Minifying and obfuscation of client-side JavaScript should be performed in this scenario.

B. Commercial MES Functionality Compilation

B.1 List of Appraised MES Solutions

The following is a list of companies' websites providing information on their products. All websites were last valid as at 28th April 2008

Company/Product	WebSite
Primac	http://www.vercom.com/sfis.htm
GSInnovate - Shop Floor Control	http://www.gsinnovate.com/
M1 by B&G - Shop Floor Data Collection	http://www.2020software.com/products/M1_by_B&G_Shop_Floor_Data_Collection.asp http://www.bowen-groves.com/index.htm
Syspro	http://www.syspro.com
Sage MAS 500 - Shop Floor Control	http://www.2020software.com/products/Sage_MAS_500_Shop_Floor_Control.asp http://www.sagemas.com/products/sagemas500/manufacturing/
Werums PAS-X	http://www.werum.de/en/mes/pasx/overview/index.jsp?top=1
Made2Manage - Shop Floor Data Collection	http://www.2020software.com/products/Made2Manage_Shop_Floor_Data_Collection.asp http://www.made2manage.com/made2manage/
Camstar	http://www.camstar.com/
FoxFire	http://www.foxfiresoftware.com
E2 Shop System	http://www.shoptech.com/
Casco - ShopVue	www.cascodev.com
Intercim	http://www.intercim.com/
Honeywell POMS	http://www.poms.com/
Elan's XFP-MES	http://www.elansoftware.com
RS PMX MES & Factory talk - Life Sciences sector	http://www.rockwellautomation.com
SIMATEC	https://mes-simaticit.siemens.com http://www2.sea.siemens.com/Products/MES/Product/Simatic

	-IT-Framework.htm
Wonderwares' Factelligence	http://mes.wonderware.com
Aegis Industrial Software	http://www.aiscorp.com
COSS Manufacturing	http://www.coss-systems.com
CIMNET- Paradigm	http://www.cim-sys.com
ProductionACE	http://www.productionprocess.com
JD Edwards	http://www.oracle.com/applications/jdedwards-enterprise-one.html

B.2 Snapshot of MES Functionality

The following sections are a functionality listing for the 22 COTS MES applications shown in Appendix B.2.1. The information was either copied directly from source or paraphrased for conciseness. It is in a raw non-interpreted format and forms the foundation for sub chapter 2.13, Review of Other MES applications.

B.2.1 Primac:

- Mainly for graphic arts industry
- Collection of labour time, material usage, and job status information from all areas of the plant
- Approximately 20 different functional modules available including: financial, production scheduling, job costing, inventory control
- Touch screen design
- Real-time job review
- Data is validated as entered
- Barcode scanners can be used and labels can be printed
- High Degree of Configurability
- Windows based using ODBC – upgrade to web based in development
- Extensive reporting:
    - Drill down reports
    - Reports viewable via browser
    - Reports delivered by schedule to email
    - Security integrates with active directory
    - Export data to excel
    - Has dashboard with gauges for progress
    - Business intelligence functionality
    - Freeform-search capabilities
- Use of security based access
- Inventory application on handheld unit (PDA)
- Business Activity Monitoring (BAM), and Business Process Management (BPM) module
- Has an ASP business model - Hosted at a dedicated facility in Dallas
- Secure Network Access

- Active Directory Based Security
- Conversion underway to Web Service Data Access, SOA Compatibility
- Also has web based customer support – knowledge centre

B.2.2 GSInnovate

- Web based Uses N-tier architecture
- Real-time data
- On demand SOA services
- High amount of functionality
- Well-designed GUI
- ASP business model
- Various reporting
- Full control over WIP and visibility
- Additional functionality such as inventory, scheduling, job planning, finance, payroll, ERP, MRP, SCM etc
- BOM handling
- Work ordering
- Kanban for pull replenishment
- SCM integration
- JIT elements
- Lot tracability
- BI elements with alerting
- Has a dashboard display of production progress – colour codes for importance
- Automates and streamlines the manufacturing process
- Security control
- Integrated reporting & audit trailing
- Knowledge centre to support product
- SOX compliant
- Multi-lingual and currencies

B.2.3 M1 by B&G

- The interface has been designed to be user friendly and fast.

- Real-time information available
- BOM usage
- Scrap and rework analysis
- Designed for touch screen
- Light pen and scanner input
- Windows-based, client/server application
- Document Management integration
- Visual basic & SQL server design
- Designed to handle all manufacturing operations
- Business analysis and productivity tools, such as graphs, calendars, wizards
- Integrated Customer relationship management
- Security by user
- Audittrail and change log
- Preconfigured reports
- Inventory, labour, scheduling etc.
- Ms office integrated

B.2.4 Syspro

- Supports mobile applications (using PDA's, smart phones etc.)
- Integrate to best in class applications
- Integrate to scanning and RFID solutions
- XML is the communication medium
- Interact with Microsoft's BizTalk server
- Workflow management
- Web based using ASP.NET – some parts are desktop based
- Needs Internet explorer 6.0 or greater
- Real time data access and update
- Configurable product
- Localization to different languages
- Send Email notification
- Has user dashboard configurable per operator
- Very customizable reports for non programmers – output to excel or email

- Can perform product scheduling using Gantt charts
- SQL server database
- Many modules with sales, inventory, scheduling, quotations, accounts etc.
- Scalable and customizable to meet specific requirements
- Web Services for custom development
- Can access data through ODBC connection
- Handles BOM
- Job / traveler printing
- Print barcodes
- No device communication possible
- Integrated change control features
- Customizable user interface
- Lot tracability

B.2.5 Sage MAS 500

- Real-time interaction
- Designed for use with touch screen technology and handheld devices
- Built-in bar coding functionality to scan work order numbers and work order steps for automatic log-ins
- The log-in and log-out activities include easy-to-use look-ups and number pads for data entry
- Shop Managers can assign Job Tickets to Employees through the Labour Assignments screen
- Employees can then view their assigned tasks and report production using Start and Stop Job Entry
- Supports a paperless shop floor environment
- Easily customised to include virtually any activity, maintenance function, report or Inquiry you need to run your shop floor
- High level of security
- Reports can be generated and viewed through Microsoft Internet Explorer
- Export all report data to an external file – word, excel, html etc.
- Create customised, presentation quality reports through Crystal Reports software

- Can run as a stand-alone system installed on terminals or handheld devices
- Each screen includes touch screen-enabled look ups and number pads for data entry without the use of a keyboard.
- Handles scrap entry and prompts the employee for Scrap Reason codes
- May enter downtime with Downtime Reason codes
- The Complete Enterprise Solution – modules plug into full erp package
- Modules include MRP, costing, HR & payroll, WIP, BOM, accounts, CRM, distribution etc.
- Uses Microsoft SQL server, and .NET Visual Basic
- Paperless manufacturing

B.2.6 Werums PAS-X

- Primarily for pharma industry – integrated suite of modules
- Desktop based application
- PAS-X is a modularly structured standard product
- Compliance with 21 CFR Part 11
- Full audittrailing of actions
- All historical data can be accessed for life of product
- Security based access and control of system – different access levels
- Available in more than 10 languages
- MES integrates with lower level SCADA - Wonderware (InTouch), Intellution (iFIX), Siemens (WinCC)
- Electronic Machine Logbook (control logs)
- Structured Management of Master Batch Records through configuration
- Paperless manufacturing system
- Integrated planning & scheduling system
- Has planning & scheduling component
- Intuitive interface with user prompting
- Shows training documents online
- Displays relevant manufacturing information on screen to users.
- PDA and portable devices supported

- Deviation handling – compares current manufacturing data to Master Batch Records
- Real-time monitoring of the production process enables corrective user intervention
- Traffic-light Red/Yellow/Green alerting
- Barcode scanning of materials
- System changes stored in historian database
- Batch tracking reports to follow full lot/batch genealogy
- Container management functionality
- Guides the user safely and easily through manufacturing steps
- Integrated warehouse management system
- Integrates with ERP such as SAP (Certified)
- Integrated CAPA module – web based & customisable
- Automatic data collection based on equipment interfaces, e.g. PLC, SCADA interface on the basis of OPC
- Various reports – export to crystal etc
- Integrated tool to control label template layouts
- The software and its layout have been tested according to ergonomic standards (ISO 9241, section 10-17)
- Individual users only need to navigate through dialogs which are relevant to them – all else hidden.
- Colour coding simplifies the user‘s orientation within the complex packaging processes
- Graphical screen of traffic lights for current quality at each operation
- Uses 10g database
- KPI measurements

B.2.7 Made2Manage

- Primarily used in electronic and general hardware manufacturers
- Uses SQL server backend
- Real-time scheduling
- Intuitive reporting
- Enterprise resource planning

- Supply chain management
- Customer relationship management
- Business intelligence
- Web-based
- Built on vb.net
- Input/output tracking with barcode wands
- Product tracking
- Inventory control
- Has BOM and product routing
- Event notifiers for custom alerting
- Document management
- ASP hosting
- Integrates into supply chain using EDI
- Real-time visibility into product status

B.2.8 Camstar

- Suitable for wide range of industry – medical, bio-tech, electronics etc. – has tailored Package for different industries
- Performs data collection, audittrailing, WIP, Quality etc.
- Real-time operation
- Delivers complete manufacturing visibility and traceability
- Provides manufacturing control needed to help eliminate scrap, rework and errors
- Automates real-time feedback needed to quickly identify and resolve issues that inhibit process and product improvement
- Enables collaborative development of manufacturing processes
- Monitors, controls and synchronises global manufacturing
- Designed for SOA
- Designed for rapid deployment
- Drastically limit scope of an issue or recall with easy identification and isolation of affected material

- Easily integrates with ERP, Product Lifecycle Management and SCM, SCADA, and PLCs
- Compliant with standards such as OAG, S95/B2MML and OPC-UA
- FDA 21 CFR Parts 11 and 210-211 compliance
- Employing Web services and XML
- Highly configurable – can adapt to business process requirements
- Provides alerting to process/product issues
- Contains business intelligence elements
- Data can be aggregated from multiple sites for management reporting
- Paperless electronic DHR functionality
- Integrates with SAP
- Comprehensive product and process tracking
- Communicate to devices such as printers – extensive print support
- Multi-tiered web based application
- Has personalised dashboard – uses business objects enterprise portal
- Contains integrated CAPA management system to streamline issue handling – very powerful
- All product is fully traceable – built in quality and data recording – product designed to reduce manufacturing issues.
- Has operator skills tracking module – automatically enforces restrictions in real-time based on operator certification (what training performed, expirations etc.)
- Integrated maintenance management module for fault tracking-resolution
- Workflow Management
- Electronic Signatures, Records and Security
- Non-Conformance Management
- Statistical Process Control functionality
- Customization Without Programming

B.2.9 FoxFire

- Continuous or discrete manufacturing
- Quality control/quality assurance features

- Work centre production reporting
- Machine work centre scheduling
- Machine capacity & schedule compliance
- Lot tracking from raw materials to finished product
- Machine downtime reporting
- Production order tracking and control
- Raw material & WIP inventory management
- Production yield and scrap reporting
- Production history reporting
- Interfaces with Ink Jet Spray Systems and Robotic Label Applications
- Standard interfaces through the application
- Low cost network of data collection terminals
- Employee time and attendance
- Comprehensive quality assurance subsystem
- Barcode based tracking of production order status
- Extensive production backlog analysis by operation/work centre/production order
- Timedate stamped employee/operation/order/lot audit trail
- Interfaces with incentive payroll systems
- Incentive earnings capability – show on screen piecemeal earnings
- Extensive mechanic/machine repair subsystem
- Real time line balancing features
- Work order status analysis
- Multilanguage support with user defined dictionary
- Fasttrack rapid implementation capability
- User configurable with numerous system options Intuitive, easy to learn,
- Realtime MS Windows GUI
- Multiple virtual machines configured per pc – hardware savings

B.2.10 E2 Shop System

- Machine shop industry - made-to-order manufacturing companies
- SME usage

- Quoting functionality that can schedule and plan production
- Data collection for real-time data
- Uses barcodes to track and touchscreen for input
- Handles materials and shipping
- Has extensive accounts module (financials)
- Extensive reporting is fully integrated and uses crystal reports
- Time and attendance
- Audittrails all transactions
- Web view into stored data for customers and suppliers
- Wireless device used to view stored data also with limited data entry (PDA)
- Windows based only
- Integrated email messaging
- Convert output to various document types
- Printable job travellers
- Product yield analysis
- ISO compliancy
- Customisation per client while still using same framework
- Database – access (20 users or less) or SQL server
- Supports terminal services
- Gant chart scheduling graph

B.2.11 Casco - ShopVue

- Machine shop usage ranging from electronics to pharma industries
- .NET platform
- WIP tracking, time and attendance etc. all in one tool
- Online work instructions – operator help via multimedia
- Configurable without need for custom coding
- Thin client – reduces network load – good response time for remote plants
- Can hold historical data without slowdowns in queries
- Can collect data from wide range of devices
- Multi-lingual support
- Connects to ERP system

- Touch screen compatible
- Crystal reports
- Web interface
- Fault tolerant design – redundancy
- Customisable information display on each screen
- Real-time feedback
- Capacity planning with in-depth machine and labour availability tracking
- Monitors OEE
- Mobile device compatible
- Interface directly with machines - Uses OPC (only)
- Reduced operator workload
- Supports RFID
- Handles scrapping
- Supports fully automated data collection
- Filters out information to display to operators – AI to prevent overburdening of info
- Supports password encryption and Active Directory
- Sends alerts

B.2.12 Intercim

- For use in aerospace, automotive and pharma industries
- Paperless system
- Complies with government regulations
- Enforces product quality
- Real-time data collection
- Online, context sensitive work instructions
- Sends alerts
- Adjustable pattern matching process control
- Non-conformance is triggered when product goes out of tolerance
- Service orientated architecture
- Web interface – no custom coding
- Web farm servers for scalability

- Highly configurable
- User friendly
- Must use IE client
- Uses .NET technology
- Pattern matches data to proactively alert potential issues
- Easily integrates with ERP
- Six sigma and Kazan techniques built in
- Real time WIP status
- Records all data
- Dashboard performance tracker
- Reporting and ad-hoc data search capability
- SOA design
- XML-based and compliant with data exchange standards such as ISA-95 and ISA-88

B.2.13 Honeywell POMS

- Operations of FDA-regulated pharmaceutical companies
- Used at every step of the process, from materials receipt to the finished goods
- View, control and track all aspects of production on the plant floor,
- Guiding operators and equipment through the manufacturing process with interactive, Step-by-step dialog
- Order and Batch Management
- Materials Management
- Exception and Deviation Management
- Asset Tracking and Management
- Quality Data Management
- Data Input Source Discrimination
- FDA Electronic Signature Compliance (per 21 CFR Part 11)
- Complete Compliance Record
- KPI Measurements
- Materials Traceability and Genealogy
- Asset Utilization Tracking

- Easy to configure/change
- Module based allows easy integration with ERP, LIMS etc.
- Version and Revision control
- Uses barcode scanning
- BOM handling
- Direct interface to devices such as printers, scales etc.
- Implements ISA S88 and S95, S88 batch standards
- Integrated scheduling
- Mobile device compatible – pocket-pc part uses .NET
- Oracle 9i, developed with .NET, cluster load balancing
- Administration parts are web based, utilises web services
- Customizable reports output in HTML or Adobe Acrobat

B.2.14 Elan's XFP-MES

- Create, route, approve, manage and distribute all work instructions
- Guidance of operators by automatic calculations of quantity
- Inventory and container management
- Barcode technology for scanning items
- Paperless Manufacturing
- Traceability of critical process data
- cGxP software, records electronic signatures
- Advanced QA and QC compliance
- Full traceability and audit trail
- Entire production genealogy
- Work instruction merger
- Fully integrates ERP, LIMS, EDMS
- Real-time deviation flagging
- R/F, PDA, PC, WIFI terminal use
- 21CFR part 11, FDA, GMP compliant
- Creates master batch documents and manages versioning control
- OPC compliant
- Workflow functionality

- Client / Server architecture uses oracle database
- Client applications running on PC, Windows NT, 2000, 2003, XP, Unix and Linux platforms.
- Interfaces with ERP, LIMS, EDM, equipment.
- All the standard serial ports (balances, printers, etc.)
- Genealogy, audit trail, complete weighing actions traceability.
- Traffic light indicators of lot quality
- Records errors and deviations
- Manufacturing rules integrated (FIFO/FEFO, lot status, Date of Use Limit, sampling
- Labeling per entity: pallet, carton, unit, container, equipment, locations
- Control and enforcement of the sequence of tasks and events
- Access rights management
- Stock management
- Electronic Picking List
- Checks every operator order in real time
- Secures flows by matching materials, quantities and locations
- Enables multi-site warehouse management
- Makes all work order traces visible
- Configurable alerts - real-time flagging of deviations and review exceptions

B.2.15 RS PMX MES

- Developed for Life Sciences sector
- Real-time exchange of information
- Suite of highly scalable, modular software applications
- Full screen browser
- Label printing for all equipment can be defined and printed
- Enhanced equipment management
- Equipment status changes
- Improved logbook handling
- Manual barcode input triggers electronic signature
- Version controlled report management

- XML export
- Equipment flow control definition
- Chart embedding
- Limit check values are returned and create a deviation for exceptions
- Full product genealogy tracking
- Paperless production
- Minimum manual data entry
- Automatic material flow control
- Container management
- Implemented personnel qualification profiles
- Integrated environments for creating and modelling production processes
- Rapid process development, simulation and deployment
- Low level data collection system (integrates with other Rockwell software)
- Collect and analyse real-time production data
- Monitors and records manufacturing system changes.
- Publishes activity, status, warning and error messages
- Security controls for access restriction
- Modular system design supports incremental solution deployments, allowing you to Maximise legacy technology investments while improving your ability to incorporate Newer technologies
- Built-in diagnostics and centralised audit logs
- Reporting and analysis tools with management dashboards
- Organise, archive and integrate process and production data collected by automated or manual systems
- Product tracking and genealogy
- Interactive enforcement of manufacturing process control
- Design and Configuration applications to configure system
- KPI tracking functionality
- Role-based operator interfaces and dashboards
- Comprehensive diagnostics and device configuration
- Audit operator actions and device health
- Calibration and real-time monitoring

- Order scheduling and execution
- Multi-site production synchronization
- Personnel qualification, skill management and training scheduling
- Interface for ERP systems (SAP certified) and other connected systems such as LIMS etc.
- Order management and finite planning on short-term bases
- Audit trailed and version-controlled management of master data and production recipes
- Process visualization/monitoring and data acquisition and reporting
- Inventory and materials management from goods receiving to goods issue
- OPC compliant
- 21 CFR Part 11 compliant electronic signatures
- Archiving of data
- Enforcement of compliant workflow
- Electronic logbooks
- Open client/server architecture with defined interfaces to external systems (API)
- RFID capable
- Centralised access control and centralised management of system-wide policies
- Oracle or SQL server backend
- Portal with customised views
- Data historian capabilities

B.2.16 SIMATEC

- Detailed product genealogy and quality records are stored and maintained to ensure regulatory compliance
- KPIs are delivered to decision-makers through web Interfaces and wireless devices in real time
- Graphical representations of the execution logic – maps out flow of processes
- Designed to allow seamless integration of 3rd party components and legacy applications
- Everything is structured according to the ISA S95 standards,

- Plant-centric IT solution covering all manufacturing needs
- Integrates with ERP
- Various add-ons such as plant performance, material management, scheduling, report management etc.
- Automated data collection
- Real-time production monitoring
- Modular based design
- Has an extensible library that can be programmed with for additional functionality
- FDA compliance support
- KPI modelling and calculation
- SQL server based
- Web based reporting – ASP.NET
- Data access via COM/DCOM or ODBC
- Audit trailing with electronic signatures
- Data historian capability – data collection and aggregation
- OEE and Downtime Management and analysis
- Automated report scheduling and delivery
- Product extensible with vbscript
- Multi-language support

B.2.17 Wonderware Factelligence

- Suitable to a wide range of industries such as automotive and pharma
- Factelligence is a configurable, web-based product
- Twelve MES modules that plug into its core XML data conduit.
- Cater for discrete, continuous and batch processes
- Facilitate the introduction of practical manufacturing processes including lean and make-to-order strategies
- Complete Integration of factory operations with supply chain management, ERP and other manufacturing systems
- Provide external web based visibility to suppliers and customers
- Provide real-time visibility into plant operations

- Support a platform for continuous quality improvement including Six Sigma, ISO9000 and HACCP requirements
- Provide a live and accurate set of manufacturing KPIs including OEE, etc.
- Provide the ability to compare the results against standard or expected results
- Integrate with the most popular control level hardware including PLC's, DNC systems and HMI systems
- User configurable, yet support easy expansion or customization should it be required
- Web-based digital cockpit
- Supports 21 CFR Part 11 requirements of the US FDA
- Integrates with ERP
- Security with password management - all user actions tracked
- Real time visibility of data
- Complies with OPC and XML
- ISA S88 based batching capabilities
- Document management capabilities such as work instructions
- Provides production tracking
- .NET based
- Certification manager – ensure employees are trained up for production
- Various reports emailed
- Multi-lingual – based on user login
- Data historian
- KPI dashboards with vector graphics and animation
- Combine and visualise data from any data source
- Secure interface distribution enterprise-wide
- Customised end-user interfaces for HMI graphics, MES, and Manufacturing Intelligence analysis
- Secure "no-touch" deployment and long-term maintenance
- WIP and inventory tracking
- Paperless and Quality management
- Labour tracking and management

B.2.18 Aegis Industrial Software

- BOM and Process revision records.
- Process deviations released during the production of that unit.
- WIP history complete with logged operator at each station.
- Part number and lot records for all tools, chemicals, and consumables used in the processing of the unit.
- Pre-production operator checklist confirmation per station.
- False positive inspections records and station.
- Rework and repair to connection level records.
- Measurement data records for all measurements derived from any process or test equipment in the route.
- Parametric operator data collection (such as trim settings, adjustments, specially serialised components, etc.) from each station applicable in the route.
- Process alarms emerging from any machine in the route while the unit was within its transfer zones.
- SPC metrics of any measurement data recorded
- On-demand, schedule and event driven reporting
    - Pre-configured and scheduled reports are dispatched automatically via email to assigned parties, providing hands-off management visibility and updates.
    - Runtime parameter entry via the web to customise on-demand reports at time of generation.
    - Event or alarm based generation and dispatch of preconfigured reports.
    - HTML, XML, and PDF outputs.
    - Over 50 preconfigured reports provided standard.
    - Embedded graphical report layout editor with full-featured graphing tool.
- Statistical Process Control engine
    - Comprehensive server-side SPC engine with user-friendly GUI for configuring new reports and analyses.
    - Supports a wide variety of basic trends.
    - SPC methods can be applied to any attribute or measurement data set collected anywhere in-process.
    - Full SPC plotting and charting utility for inclusion in web reports.

  - Out of Control conditions may be mapped to any system alarm, enabling emails, pages, report generation, or line stops.
- Quality, performance and efficiency monitoring dashboard
  - View real-time machine and line states through a graphical representation of the lines.
  - Drill into detailed machine parameters specific to the nature of machine types such as reflow, reflow profiling such as KIC, placement, inspection, test, printing, etc.
  - Save the data set of any view or chart displayed in the browser as XML, Excel, or Text to the local PC for further use in other applications.
- Web based product and process release control
  - Completed process and product definitions initiate notification emails to appropriate parties depending on the nature of the product.
  - Recipients review the full scope of BOM and process via their web browser, provide comments, and submit their approval/rejection.
  - System intelligence manages rejections/approvals and eliminates risk of unapproved data sets or jobs reaching the factory floor.
  - The process of electronic approval is maintained in the system and recalled later as part of the total traceability output of the enterprise solution.
  - FDA 21 CFR part 11 compliant
- Technology used
  - Uses web services
  - Server farm
  - Xml based
  - SQL server or oracle
  - Multi-lingual – based on pc settings
  - Web centric multi-tiered

B.2.19 COSS Manufacturing

- Used for aerospace and various metalwork type shops
- Full ERP built-in
- Various modules such as: accounts, estimating, job costing, work orders, BOM, scheduling

- Can perform EDI
- Mobile client support such as PDAs and smart clients
- Supports barcoding and manual data entry
- Real-time information available
- Document control functionality
- 150 reports available for various information

B.2.20 CIMNET- Paradigm

- Mainly in electronics industry
- Various modules available such as sales, cycle times, costing etc
- Desktop based client/server app
- SQL Reports - includes a variety of pre-defined on-screen reports
- View/Report system - customizable reporting options available throughout the system
- ODBC compliant
- Screen customisation
- Multi-language, Multi-currency, Flexible tax structure ,Choice of date format
- Open database architecture enables interfacing with 3rd party tools
- EDI with ASCII and XML
- Alerting when collected data outside tolerances
- Export data to various formats
- Plant maintenance tracking – MTBF etc.
- Executive information system data modelling

B.2.21 ProductionACE

- Real-time OEE dashboard – visible via browser or PDA
- Automated reporting – both scheduled and alerting
- Coordinate production schedule at multiple plants.
- Can collect data via wireless system – need to install client agent on each pc
- Interfaces with ERP

B.2.22 JD Edwards

- HTML server and web client
- Design tools enable creation and customisation, customise, and deploy interactive and batch applications, workflow, and reports
- Integration and interoperability tools-enable integrations between 3rd party products
- Change management tools
- Supports multiple modes of manufacturing:
    - Discrete
    - Engineer-to-order
    - Batch process
    - Configure-to-order
    - Continuous process
    - Assemble-to-order
    - Make-to-order
    - Remanufacture
    - Make-to-stock
    - Push and/or pull
- Workflow control.
- Integrated full ERP system
- Online engineering change request
- Plant visibility dashboard
- Integrated BI with report authoring

C. Global Bausch + Lomb MES survey Questions (End user)

Page 1 - General MES details

This survey is being conducted by Global CAMS to gain a better understanding of current MES (Manufacturing Execution System) usage & capabilities. The survey goal is purely software focused and aims to gain a better understanding of current MES strengths, weaknesses and potential improvements.

MES is the computer based system that facilitates you to complete your activities in the manufacturing environment. All participant involvement in this survey is anonymous with no possible tracing of responses to individuals.

1. What manufacturing facility are you situated in?

2. What manufacturing area and operation do you mostly work at?
E.g. Purevision dialpack or Toric casting or RPIII audits etc.

Please Note: If you use work at different operations, please complete a separate survey for each.

3. What type of MES solution do you use?

Please Note: If you use multiple MES solutions, please complete a separate survey for each.

- [ ] CAMS
- [ ] BPCS
- [ ] Excel
- [ ] Paper based
- [ ] Other (please specify)

4. During your working day/shift, how many times do you access MES for the purposes of data entry e.g. shift start/end, every hour, every 5 minutes, continuously?

Page 2 - Interface

5. Does your MES have an easy to use screen layout (GUI)?

- Very easy
- Reasonably easy
- Average
- Poor
- Very poor

Additional comments (if any)

6. What primary method of input do you use to navigate MES?

- Mouse & keyboard
- Touchscreen
- Mouse & keyboard & Touchscreen

Additional comments (if any)

7. What method of input would you prefer when navigating MES?

- Mouse & Keyboard
- Touchscreen

- [ ] Other (please specify)

8. Do you find the MES system user friendly and intuitive?

- [ ] Yes
- [ ] Somewhat
- [ ] No

Other (please specify)

9. Does the MES on your PC interface with any of the following devices:

- [ ] Barcode scanner
- [ ] Magnetic card reader
- [ ] Printer
- [ ] PLC
- [ ] Vision system
- [ ] None
- [ ] Gague
- [ ] Weighing scales

Page 3 - Error handling

10. In general, does MES prevent you from making mistakes?

- [ ] Yes definitely
- [ ] Yes somewhat
- [ ] Not that much

- Not at all

Additional comments (if any)

11. How many times a week do you need technical intervention to resolve non-machine related issues in MES e.g. You cannot proceed with what you were doing until IT fix an issue?

- 0
- < 3
- < 5
- < 10
- < 20
- > 20

Additional comments (if any)

12. What is the average length of time before MES problems (as per the previous question) are resolved by IT?

- < 5 minutes
- < 30 minutes
- < 60 minutes
- > than 60 minutes

Additional comments (if any)

13. What changes do you think could be made to improve MES error prevention i.e. reduce frequency or errors that prevent you from doing your work?

Page 4 - Response speed

14. In general, what is the speed of system response from MES e.g. how long to do something when you press a button or enter information?

- Always fast
- Usually fast
- Sometimes faster, sometimes slower
- Usually slow
- Always slow

Additional comments (if any)

15. What is the longest possible time delay in MES response before it interferes with your work?

- Need immediate response
- < 5 seconds
- < 30 seconds
- < 1 minute
- < 5 minutes
- > 5 minutes

Additional comments (if any)

16. Do you have the option to do some other work related task while waiting for MES

to complete instructions?

- Yes always
- Sometimes
- No never

Additional comments (if any)

17. What issues does system slowness cause (if any)?

Page 5 - Usability and training

18. On occasion, do you have to make guesses when using MES i.e. not fully sure what the consequences of doing something are?

Please Note: This question is focused on MES complexity and usability and bears no reflection on the users of the system.

- Frequently
- Sometimes
- Seldom
- Never

Additional comments (if any)

19. Would MES on a mobile platform such as a portable handheld computer enable you to do your job better?

- Yes

☐ Maybe

☐ No

Additional comments (if any)

20. Do you feel that you have enough MES training in order to carry out your job?

☐ Yes

☐ Maybe

☐ No

Additional comments (if any)

21. Please rank what your preferred method for training/re-training is, with 1 being most preferred and 4 the least:

	1	**2**	**3**	**4**
Being instructed by a person	☐ Please rank what your preferred method for training/re-training is, with 1 being most preferred and 4 the least: Being instructed by a person 1	☐ 2	☐ 3	☐ 4
Reading paper based instructions	☐ Reading paper based instructions 1	☐ 2	☐ 3	☐ 4
Computer	☐ Computer	☐ 2	☐ 3	☐ 4

based instructions (images and text)	based instructions (images and text) 1			
Computer based video, images and text	☐ Computer based video, images and text 1	☐ 2	☐ 3	☐ 4

Additional comments (if any)

22. Do you think more online computer based help/training for use of MES would be beneficial?

☐ Yes

☐ Maybe

☐ No

Additional comments (if any)

Page 6 - General

23. Please rate your experience of MES on the following:

	Very good	**Good**	**Fair**	**Poor**	**Very poor**
Clarity of error messages	☐ Please rate your experience of MES on the following: Clarity of error	☐ Good	☐ Fair	☐ Poor	☐ Very poor

	messages Very good				
Ease of use	Ease of use Very good	Good	Fair	Poor	Very poor
Ease of learning how to use	Ease of learning how to use Very good	Good	Fair	Poor	Very poor
Available help to learn features	Available help to learn features Very good	Good	Fair	Poor	Very poor
Low frequency of errors	Low frequency of errors Very good	Good	Fair	Poor	Very poor
Speed of system response	Speed of system response Very good	Good	Fair	Poor	Very poor
Ability to recover from errors/mistakes yourself	Ability to recover from errors/mistakes yourself Very good	Good	Fair	Poor	Very poor

Additional comments (if any)

24. What major function do you believe your MES application is for?

Tool for managers/supervisors to report on operators work

☐ Tool used by QA to audit operator's work

☐ Tool to enforce manufacturing rules, reduce burden on operators & record manufacturing data that can help improve the process

Other (please specify)

25. Based on your experience of MES, are there any improvements that you could suggest?

D. Global Bausch + Lomb MES survey Questions (Technical)

Page 1 - General MES details

This survey is being conducted by Global CAMS to gain a better understanding of current MES (Manufacturing Execution System) usage & capabilities. MES is the computer based system or systems that facilitate the completion of work based activities in the manufacturing environment. The survey aims to gain a better understanding of MES strengths, weaknesses and potential improvements.

In addition to questions, text boxes are provided for participants to give their invaluable insight and expertise of MES, related software, manufacturing systems and industry in general. All participant involvement in this survey is anonymous with no possible tracing of responses to individuals. The participant sample was chosen to elicit knowledge and experience from the varied B&L technical community.

1. Please enter your general stakeholder classification with MES e.g. quality, development, architect, manager, support, engineering etc.

2. With regard to MES, what manufacturing facility do you work with?

PLEASE NOTE: If you work with multiple facilities, please complete a separate survey for each facility.

3. What type of MES solution do you use?

PLEASE NOTE: If you work with multiple MES solutions, please complete a separate survey for each one.

- CAMS
- BPCS
- Excel
- Paper based
- Other (please specify)

Page 2 - MES selection criteria

4. What is the balance between having a highly configurable MES product that meets 90%+ of required business logic, versus a generic easily reusable product that can be quickly deployed but only meets about 70% of required business logic?

- Full configuration important, deployment speed not important
- More configuration better but need to be able to deploy reasonably quickly
- Both have equal importance
- Mostly generic better, some configuration also
- Fully generic product important, high configuration ability to meet all business requirements not important

Additional Comments

5. In your opinion, what should the balance be between application response times and facilitation of complex business logic?

- Response times most important
- Mostly response times
- Even balance
- Mostly business logic
- Business logic most important

Additional Comments

6. Please rank the following MES attributes in order of importance, with most important being 1 (only one attribute per rank/column):

	1	2	3	4	5	6	7	8	9
High configurability	Please rank the following MES attributes in order of importance, with most important being 1 (only one attribute per rank/column): High configurability 1	2	3	4	5	6	7	8	9

Quick application response times	Quick application response times 1	2	3	4	5	6	7	8	9
Handle complex business logic	Handle complex business logic 1	2	3	4	5	6	7	8	9
Inbuilt prevention of failure	Inbuilt prevention of failure 1	2	3	4	5	6	7	8	9
Adequate reporting	Adequate reporting 1	2	3	4	5	6	7	8	9
Compliance to industry regulations (SOX, FDA etc)	Compliance to industry regulations (SOX, FDA etc) 1	2	3	4	5	6	7	8	9
Ability to rapidly modify to changing business requirements	Ability to rapidly modify to changing business requirements 1	2	3	4	5	6	7	8	9
Ability to recover from errors & system unavailability	Ability to recover from errors & system unavailability	2	3	4	5	6	7	8	9

Localisation / Multi-language	1 / Localisation / Multi-language 1	2	3	4	5	6	7	8	9

Additional comments (if any)

7. In your opinion, what is the reliability of a remotely hosted MES solution (application & database) on the following attributes, where 1 is very unreliable and 5 is very reliable?

	1	**2**	**3**	**4**	**5**
Security	In your opinion, what is the reliability of a remotely hosted MES solution (application & database) on the following attributes, where 1 is very unreliable and 5 is very reliable? Security 1	2	3	4	5
Performance	Performance	2	3	4	5

Application support	1 Application support 1	2	3	4	5
Interface with devices	Interface with devices 1	2	3	4	5
Overall reliability	Overall reliability 1	2	3	4	5

Additional Comments

8. Apart from reliability and security, what other potential issues do you envisage with a web based MES?

9. What industry standards does your current MES adhere to?

- ISO
- ISA
- MESA Model
- Don't know
- Other (please specify)

10. What elements of the MESA model does your MES application support (if any)?

- Quality Management
- Environment, Health and Safety Management

- Data Collection/Acquisition
- Document Control
- Resource management
- Labour management
- Equipment management
- Materials management
- Product tracking and genealogy
- Dispatching Production Units
- Operations/Detail Scheduling
- Material Storage & Transport
- Maintenance
- Performance Analysis
- Process Management
- Don't know

Additional Comments

11. What importance is there in your MES applications adherence to industry standards such as ISO, ISA, MESA Model etc?

- Question not relevant to my job function
- Very important
- Somewhat important
- Not that important
- Of no use at all
- Don't know

Additional Comments

12. Does your current MES help you comply with SOX and FDA requirements?

- Yes
- No
- Don't know

Additional comments (if any)

Page 3 - Interfacing

13. What integration capabilities does your current MES have to other enterprise solutions?

- ERP (Enterprise Resource Planning)
- Financials
- CRM (Customer Relationship Management)
- MRP (Material Requirements Planning)
- Manufacturing Resource Planning
- SCADA/Controls
- Reporting systems
- Other (please specify)

14. How important is it for MES to have integration capabilities to other enterprise solutions such as ERP (Enterprise Resource Planning), financials, CRM (Customer Relationship Management), MRP (Material Requirements Planning), Manufacturing

Resource Planning, SCADA/Controls etc?

- [ ] Very important
- [ ] Somewhat important
- [ ] Not that important
- [ ] Not important at all

Additional Comments

15. Do you think that your current MES user interface has enough prompting so that the user cognitive workload is minimised?

- [ ] Question not relevant to my job function
- [ ] Yes
- [ ] Maybe
- [ ] No

Additional Comments

16. Do you think that more MES user interface prompting should be conducted to reduce user cognitive workloads (and resultantly reduced error frequency)?

- [ ] Question not relevant to my job function
- [ ] Yes
- [ ] Maybe
- [ ] No

Additional Comments

Page 4 - Devices / Security

17. Do you think there are benefits to utilizing biometric authentication?

- (x) Yes
- ( ) Maybe
- ( ) No

Additional Comments

18. What access control does your MES application use?

- ( ) Question not relevant to my job function
- ( ) Badge & PIN
- ( ) Windows active directory logon
- ( ) PIN
- ( ) Biometric
- ( ) Other (please specify)

19. How important is it to consolidate equipment communication technologies & protocols to enable simpler device integration i.e. 1 protocol to communicate with all devices?

- ( ) Question not relevant to my job function
- ( ) Very important
- ( ) Somewhat important
- ( ) Not that important
- ( ) Totally unnecessary

Additional Comments

Page 5 - Reporting and information access

20. Do you currently have adequate reporting of your MES data?

- Yes
- Somewhat
- No

Additional Comments

21. How important/beneficial is the reporting of data in real time via a tool such as a dashboard?

- Very beneficial
- Of some benefit
- Of little use

Additional comments (if any)

22. Should reporting be more integrated to MES or be a separate software package?

- Question not relevant to my job function
- Completely separate
- Basic integration
- Total integration

Additional Comments

23. Do you think enough automated report generation and delivery is utilised to provid

useful MES information e.g. emailed reports or saved to common network drive?

- Yes
- Some but not enough
- No

Additional Comments

24. Is the ability to generate your own customised reports important?

- Yes
- Somewhat
- No

Additional comments (if any)

25. Do you believe that all the data you require is being captured by the MES system?

- Yes
- No

Additional comments (if any)

26. Is enough data mining and BI (business intelligence) being performed to fully utilize MES data?

- Yes
- Some but not enough
- No

Additional Comments

27. How important is it to be able to compare/amalgamate data from multiple geographic sites?

- Question not relevant to my job function
- Very important
- Somewhat useful
- Not of much use
- No use at all

Additional Comments

28. What kind of reporting tools does your MES solutions use?

- CAMS WebReports
- CAMS PA Reports
- Crystal reports
- Oracle discoverer
- Real-time reports
- InfoView
- Printed reports
- Custom reporting tool
- Other (please specify)

Page 6 - Design & functionality

29. What kind of system architecture does your current MES solution have?

- Client / Server
- Web
- Citrix / Terminal Services
- Standalone
- Other (please specify)

30. Does your current MES application facilitate electronic DHR / Batch Records?

- Yes
- Partially
- Don't know
- No

Additional Comments

31. Should a complete "paperless" design be integrated into MES even if it is not directly cost effective.

- Yes
- Maybe
- No

Additional Comments

32. Do you think SPC (statistical process control) functionality would be of benefit in an MES solution?

- Question not relevant to my job function
- Yes
- Maybe
- No

Additional comments (if any)

33. Should data historian functionality be integrated into MES applications or should there be a separate data historian solution/application?

PLEASE NOTE: A data historian is roughly defined as a system that can store and report on high volume real-time data from multiple sources.

- Fully integrated
- Integrate some elements
- Totally separate
- Don't need data historian functionality

Additional Comments

34. How much error prevention does your MES application provide?

- Question not relevant to my job function
- A lot of prevention
- Some but not enough
- Definitely not enough

Additional Comments

35. If no barriers existed (time, money etc.), what could be done to improve MES design?

36. Are you aware of all the functionality that exists within your current MES solution?

- [ ] Question not relevant to my job function
- [ ] Yes
- [ ] Unsure
- [ ] No

Additional comments (if any)

37. Do you regularly deploy the latest releases & enhancements?

- [ ] Question not relevant to my job function
- [ ] Yes
- [ ] Unsure
- [ ] No

Additional comments (if any)