

**A Macro-Micro System Architecture Analysis Framework  
Applied to Smart Grid Meter Data Management Systems**

By

**Sooraj Prasannan**

M.S., Electrical & Computer Engineering  
Worcester Polytechnic Institute, 2002

B.S., Electronics Engineering  
Bombay University, 1999

Submitted to the System Design and Management Program  
in Partial Fulfillment of the Requirements for the Degree of  
Master of Science in Engineering and Management  
at the

**Massachusetts Institute of Technology**  
**May 2010**

**© 2010 Sooraj Prasannan. All rights reserved**

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Signature of Author

---

Sooraj Prasannan  
System Design and Management Program  
May 2010

Certified by

---

Anas Alfaris  
Thesis Supervisor  
Research Scientist, Engineering System Division

Accepted by

---

Patrick Hale  
Director  
System Design & Management Program

THIS PAGE IS INTENTIONALLY LEFT BLANK

# ABSTRACT

## **A Macro-Micro System Architecture Analysis Framework Applied to Smart Grid Meter Data Management Systems**

By

**Sooraj Prasannan**

Submitted to the System Design and Management Program in Partial Fulfillment of the  
Requirements for the Degree of Master of Science in Engineering and Management

This thesis proposes a framework for architectural analysis of a system at the Macro and Micro levels. The framework consists of two phases – Formulation and Analysis. Formulation is made up of three steps – Identifying the System Boundary, Identifying the Object-Process System levels using the Object-Process Methodology (OPM) and then creating the Dependency Matrix using a Design Structure Matrix (DSM). Analysis is composed of two steps – Macro-Level and Micro-Level Analyses. Macro-Level analysis identifies the system modules and their interdependencies based on the OPM and DSM clustering analysis and Visibility-Dependency Signature Analysis. The Micro-Level analysis identifies the central components in the system based on the connectivity metrics of Indegree centrality, Outdegree centrality, Visibility and Dependency. The conclusions are drawn based on simultaneously interpreting the results derived from the Macro-Level and Micro-Level Analyses.

Macro-Analysis is vital in terms of comprehending system scalability and functionality. The modules and their interactions influence the scalability of the system while the absence of certain modules within a system might indicate missing system functionality. Micro-Analysis classifies the components in the system based on connectivity and can be used to guide redesign/design efforts. Understanding how the redesign of a particular node will affect the entire system helps in planning and implementation. On the other hand, design Modification/enhancement of nodes with low connectivity can be achieved without affecting the performance or architecture of the entire system. Identifying the highly central nodes also helps the system architect understand whether the system has enough redundancy built in to withstand the failure of the central nodes. Potential system bottlenecks can also be identified by using the micro-level analysis.

The proposed framework is applied to two industry leading Smart Grid Meter Data Management Systems. Meter Data Management Systems are the central repository of meter data in the Smart Grid Information Technology Layer. Exponential growth is expected in managing electrical meter data and technology firms are very interested in finding ways to leverage the Smart Information Technology market. The thesis compares the two Meter Data Management System architectures, and proposes a generic Meter Data Management System by combining the strengths of the two architectures while identifying areas of collaboration between firms to leverage this generic architecture.

Thesis Supervisor:  
Anas Alfaris  
Research Scientist, Engineering Systems Division

THIS PAGE IS INTENTIONALLY LEFT BLANK

## ACKNOWLEDGEMENTS

This thesis culminates my journey of the last two years. I would like to take this opportunity to thank everyone who has contributed in creating this thesis and for making the last two years something to cherish forever.

To my thesis adviser Dr. Anas Alfaris, I greatly appreciate your guidance every step of this thesis. Thank you for your time, patience and trust in me. Your passion and motivation kept me going through some very tough times. This thesis would not be possible without your expertise and I hope to rely on your guidance beyond this thesis.

To my manager Bruce Zimmerman, thank you for your support and understanding for the past two years. To Rob Masson, for helping scope out the thesis topic and for putting me in touch with the right people. To my friends at work Aviv Kaufmann, Steve Alves and Michael Leone for keeping me entertained during many stressful days.

To everyone at SDM for making this such a wonderful learning experience. To Pat Hale and the SDM staff, especially Chris Bates and Bill Foley for answering my never ending questions.

To all my great friends in the SDM program, it was a privilege to work with all of you on all the projects. Thank you for providing me with different perspectives that broadened my academic knowledge. I have made many lifelong friends here and I dearly hope to keep in touch with all of you till we meet next time.

To my parents and my brother, thank you for all your love, support and prayers throughout the years.

To my wife Asia, thank you for all your support, motivation, love and understanding. This degree would not be possible without you.

THANK YOU  
Sooraj Prasannan

THIS PAGE IS INTENTIONALLY LEFT BLANK

---

## TABLE OF CONTENTS

<b>Abstract.....</b>	<b>3</b>
<b>Acknowledgements.....</b>	<b>5</b>
<b>Table of Contents .....</b>	<b>7</b>
<b>List of Figures.....</b>	<b>8</b>
<b>List Of Tables .....</b>	<b>10</b>
<b>Chapter 1. Introduction.....</b>	<b>11</b>
1.1 Background .....	11
1.2 Motivation.....	12
1.3 Thesis Overview .....	12
1.3.1 Framework .....	12
1.3.2 Application.....	13
1.3.3 Layout .....	13
<b>Chapter 2. System Architecture.....</b>	<b>15</b>
2.1 Importance of System Architecture .....	15
2.2 Tools.....	17
2.2.1 Object-Process Methodology (OPM).....	17
2.2.2 Design Structure Matrix (DSM).....	20
2.2.3 Networks .....	22
<b>Chapter 3. Smart Grid.....</b>	<b>25</b>
3.1 Smart Grid Taxonomy .....	25
3.2 Smart Grid Data Flow .....	27
3.2.1 Advanced Metering Infrastructure (AMI).....	28
3.2.2 Meter Data Management Systems (MDMS) .....	30
3.2.3 Business Applications/Operations .....	32
<b>Chapter 4. System Architecture Analysis Framework.....</b>	<b>35</b>
4.1 Formulation.....	37
4.1.1 Identifying the System Boundary .....	38
4.1.2 Identifying Object-Process System Levels .....	38
4.1.3 Creating the Dependency Matrix .....	42
4.2 Analysis.....	44
4.2.1 Macro-Level Analysis.....	44
<i>Modularity from OPM.....</i>	<i>45</i>
<i>Dependency Matrix Analysis.....</i>	<i>45</i>
<i>Visibility-Dependency Signature Analysis .....</i>	<i>47</i>
4.2.2 Micro-Level Analysis.....	48
<b>Chapter 5. MDMS Case Studies .....</b>	<b>53</b>
5.1 Product 1: Formulation .....	53
5.1.1 Product 1: Identifying the System Boundary .....	53
5.1.2 Product 1: Identifying the Object-Process System Levels.....	54
5.1.3 Product 1: Creating the Dependency Matrix.....	65
5.2 Product 1 Analysis .....	67

---

---

5.2.1	Product 1: Macro-Level Analysis .....	67
	<i>Modularity from OPM</i> .....	67
	<i>Dependency Matrix Analysis</i> .....	67
	<i>Visibility Dependency Signature</i> .....	70
5.2.2	Product 1: Micro-Level Analysis .....	71
	<i>Connectivity Analysis</i> .....	71
	<i>Analysis Summary</i> .....	74
5.3	Product 2 Formulation .....	75
5.3.1	Product 2: Identifying the System Boundary .....	75
5.3.2	Product 2: Identifying the Object-Process System Levels.....	75
5.3.3	Product 2: Creating the Dependency Matrix.....	87
5.4	Product 2 Analysis .....	89
5.4.1	Product 2: Macro-Level Analysis .....	89
	<i>Modularity from the OPM</i> .....	89
	<i>Dependency Matrix Analysis</i> .....	89
	<i>Visibility Dependency Signature</i> .....	92
5.4.2	Product 2: Micro-Level Analysis .....	93
	<i>Connectivity Analysis</i> .....	93
	<i>Analysis Summary</i> .....	96
<b>Chapter 6. Inferences.....</b>		<b>98</b>
6.1	Comparing the Architectures .....	98
6.2	Proposing a Generic MDMS Architecture.....	100
6.3	Leveraging the MDMS Architecture .....	103
<b>Chapter 7. Conclusion .....</b>		<b>106</b>
7.1	Proposed Framework .....	106
7.2	MDMS Analysis Insights.....	107
7.3	Future research.....	108
<b>References .....</b>		<b>109</b>

## LIST OF FIGURES

Figure 1 Role of Architecture (8).....	15
Figure 2 OPM Building Blocks (10).....	17
Figure 3 Structural Links (10).....	18
Figure 4 Tagged structural links (10).....	19
Figure 5 Procedural links (10) .....	19
Figure 6 DSM (11).....	20
Figure 7 Clustered DSM (11).....	22
Figure 8 Network Diagram for DSM in Figure 6 .....	22

---



---

Figure 9 Smart Grid (16).....	26
Figure 10 Smart Grid Data Flow (17).....	27
Figure 11 Old Meter (Left) vs. Smart Meter (Right) .....	28
Figure 12 Bulk Electricity Prices on a hot summer day (source: <a href="#">ISO-NE</a> ) (18).....	29
Figure 13 Exponential Growth of Data (19) .....	31
Figure 14 Example Smart Grid Data Flow Installation (19).....	34
Figure 15 Framework.....	36
Figure 16 Formulation .....	37
Figure 17 Scientific Calculator: Identifying the System Boundary .....	38
Figure 18 Scientific Calculator: Object Decomposition .....	39
Figure 19 Scientific Calculator: Process Decomposition.....	39
Figure 20 Scientific Calculator: Top Level OPD.....	40
Figure 21 Scientific Calculator: System Level OPD .....	41
Figure 22 Scientific Calculator: Level 1 “Number Clearing” Process Centric View .....	42
Figure 23 Network Graphs vs. OPM .....	43
Figure 24 Scientific Calculator: DSM .....	43
Figure 25 Macro-Level Analysis .....	44
Figure 26 Scientific Calculator: Level 1 Decomposition.....	45
Figure 27 Scientific Calculator: Results of Newman’s Clustering.....	46
Figure 28 Scientific Calculator: Clustered DSM .....	47
Figure 29 Scientific Calculator: Visibility vs. Dependency Graph.....	48
Figure 30 Micro-Level Analysis .....	49
Figure 31 Scientific Calculator: Indegree and Outdegree Centrality.....	50
Figure 32 Scientific Calculator: Higher Order Interconnections .....	51
Figure 33 Identifying MDMS System Boundary.....	54
Figure 34 Product 1: Object Decomposition Levels (UCE Object Zoom In).....	55
Figure 35 Product 1: Process Decomposition Levels (Authenticating Process Zoom In).....	58
Figure 36 Product 1: Top Level OPD .....	60
Figure 37 Product 1: System Level OPD.....	61
Figure 38 Product 1: Level 1 Processing Centric View .....	62
Figure 39 Product 1: Level 2 Interfacing Centric View.....	62
Figure 40 Product 1: Level 1 Authenticating Centric View .....	63
Figure 41 Product 1: Level 2 Estimating Centric View .....	64
Figure 42 Product 1: Level 1 Analyzing Centric View.....	65
Figure 43 Product 1: DSM.....	66
Figure 44 Product 1: Level 1 Decomposition .....	67

---

---

Figure 45 Product 1: Results of Newman’s Clustering Analysis.....	68
Figure 46 Product 1: Clustered DSM.....	69
Figure 47 Product 1: Visibility-Dependency Signature Analysis.....	70
Figure 48 Product 1: Indegree vs. Dependency .....	73
Figure 49 Product 1: Outdegree vs. Visibility .....	74
Figure 50 Product 2: Object Decomposition (Security Object Zoom In).....	76
Figure 51 Product 2: Process Decomposition (Loading Process Zoom In).....	79
Figure 52 Product 2: Top Level OPD .....	81
Figure 53 Product 2: System Level OPD.....	82
Figure 54 Product 2: Level 1 Loading Centric OPD.....	83
Figure 55 Product 2: Level 2 Meter Read Data Loading Centric View.....	84
Figure 56 Product 2: Level 1 Processing Centric View .....	85
Figure 57 Product 2: Level 1Configuring Centric View.....	86
Figure 58 Product 2: Level 1 Application Processing centric view.....	87
Figure 59 Product 2: DSM.....	88
Figure 60 Product 2 Modularity from OPM .....	89
Figure 61 Product 2: Results of Newman Clustering Analysis.....	90
Figure 62 Product 2: Clustered DSM.....	91
Figure 63 Product 2: Visibility vs. Dependency Signature.....	92
Figure 64 Product 2: Indegree vs. Dependency .....	95
Figure 65 Product 2: Outdegree vs. Visibility .....	96
Figure 66 Comparing DSMs .....	99
Figure 67 Generic MDMS Architecture top level OPD .....	101

## LIST OF TABLES

Table 1 Taxonomy of System element interactions (11) .....	21
Table 2 Quantification Schemes (11).....	21
Table 3 Product 1: Indegree Centrality .....	71
Table 4 Product 1: Outdegree Centrality .....	72
Table 5 Product 2: Indegree Centrality .....	93
Table 6 Product 2: Outdegree Centrality .....	94
Table 7 Areas to Collaborate.....	103

---

# CHAPTER 1. INTRODUCTION

## 1.1 BACKGROUND

A system is an arrangement of entities that interact with each other to function as a whole and by design it is expected to perform better than the sum of its parts. Systems have form (structure), function (behavior) and interconnections. System Architecture defines the form, function and interconnections between the different parts of a system.

The architecture of man-made systems gives us insights into the past, the present and the possible future of the design. By analyzing the entities that make up the present architecture of a system we can identify the legacy of the system and suggest different ways to evolve the system. The architecture of a system also has a big influence on the firm's organizational layout, the firm's strengths and weakness, and the firm's strategy (1).

Understanding the architecture of an external product gives a firm valuable insight into how to leverage the architecture for collaborating with that external firm. Similarly, understanding the architecture of a competitive product enables the firm to understand the weakness and strengths of the opponent's offering. The firm can then use the knowledge to attack the competitor's weak spots and make internal product improvements to compete with the competitor's strengths.

This thesis proposes a framework to analyze the product architecture and then applies the proposed framework to two Meter Data Management Systems (MDMS). MDMS is a vital cog in enabling the Smart Grid Information Technology layer and numerous technology firms are looking for ways to apply their domain knowledge to gain a share of the large Smart Grid Information Technology market.

Smart Grid is an upgrade to the current residential and commercial electrical grid. Its purpose is to make it more reliable, efficient and environmentally friendly. The current electricity grid has not changed significantly since the early 1900's. Deployment of the computer and data communications-based Smart Grid will enable better management of the grid assets; in particular, facilitating optimal routing of power to respond to varying conditions, improving quality of service (QoS) for the end consumers, and enabling two-way communication between the consumers and the utilities. This will in turn increase system reliability, increase security and reduce overall costs. The two way communication between the consumer and the utility is particularly important since it changes the entire electricity delivery supply chain. The Smart Grid enables the end consumer to also become a supplier by selling back generated power (e.g. solar roofs on houses) to the Grid. This way, the consumers may become active players in the electricity supply chain. The consumers can also change electricity demand by changing their usage pattern based on electricity prices or other signals. Thus adding Smarts to the current grid will lead to an explosion of data flowing through the system. The Smart Grid information technology layer is tasked with managing and leveraging this data in order to deliver on the promise of a smarter, greener grid.

The Smart Grid Information technology layer consists of three entities the Advanced Metering Infrastructure (AMI), the Meter Data Management System and the Business Application/Operations System. The AMI is responsible for collecting the meter reads from various collection systems and feeding it into the MDMS. The MDMS is the central repository of this meter data and it manages and processes the data before it can be used by the Business Application/Operations system. The Business Applications/Operations system leverages the MDMS data to deliver the required business service such as Billing, Outage Management and Workforce Management.

## **1.2 MOTIVATION**

Understanding the MDMS architecture and suggesting ideas/areas to leverage that architecture was the main motivation behind this thesis. To achieve this goal one has to understand the system at both the top level and the low level, and then use that combined knowledge to draw conclusions to guide strategic design as well as business decisions.

Previous research on system architecture has either focused on the product level (2-4) or at the component level (5). Sharman and Yassine (6) have analyzed the architecture at different levels starting at the bottom, intermediate and the top but their research is more qualitative than quantitative whereas Sosa and Eppinger (5) offer metrics to understand the architecture at the component level. This thesis aims to understand the higher level Macro architecture and the lower level Micro architecture of a system using both qualitative and quantitative metrics.

The MDMS is the owner of all the meter information in a utilities Smart Grid offering. Data Management is typically not a core competency of the players in the Smart Grid business ecosystem. The utilities are actively looking for partners to enhance their offerings in this space. This creates an opportunity for the established technology companies to use their computer systems expertise to establish a foothold in this market. The MDMS is a central repository for the grid's meter data. With the active deployment of millions of smart meters, the amount of data generated is expected to increase exponentially. Storing these huge amounts of data while maintaining the same quality of service is a huge challenge. The MDMS has to be designed and maintained as one large system which presents many challenges but at the same time also provides many business opportunities.

## **1.3 THESIS OVERVIEW**

The framework, application and thesis layout are discussed in brief in the following section.

### **1.3.1 Framework**

The thesis proposes a framework to comprehend the architecture of a system in order to make well informed design/redesign decisions and to guide the strategy of the firm. The framework proposes looking at the system at two levels; the Macro-Level and the Micro-level when making the decisions.

The thesis framework consists of two steps - Formulation and Analysis. Formulation is the process of understanding the system by expressing it in simpler terms. Formulation consists of three steps - Identifying the System Boundary, Identifying the Object-Process System levels and Creating the Dependency Matrix. Analysis consists of two steps – Macro-Level Analysis and Micro-Level Analysis. Macro-Level Analysis is further divided into Modularity Analysis and Visibility-Dependency Signature Analysis whereas the Micro-Level Analysis is based on understanding the connectivity metrics of Indegree centrality, Outdegree centrality, Visibility and Dependency.

The thesis uses two forms of representation for the Formulation and the Analysis – Object-Process Methodology (OPM) and the Dependency Matrix. The OPM is primarily used for Formulation and Macro-Analysis but the tacit knowledge gained during the OPM process is vital when drawing conclusions from the analysis. The Dependency Matrix is used for both the Formulation and the Analysis and it provides the quantitative metrics to understand the architecture.

### **1.3.2 Application**

The framework will be applied to two industry leading Meter Data Management Systems (MDMS). The two systems will be analyzed separately at the Macro-Level and the Micro-Level. The Macro-level Analysis will identify the modules in the system and should quantify how each of the modules affects the system design whereas the Micro-Level Analysis should identify the important components in the systems based on quantitative metrics.

The thesis will then analyze the two MDMS simultaneously using the Macro-Micro Analysis findings - to compare the individual systems, to propose a generic architecture and to suggest areas of collaboration with a MDMS vendor.

### **1.3.3 Layout**

The thesis is organized as follows:

Chapter One: Introduction of the thesis, stating the background, motivation and overview.

Chapter Two: System architecture overview. This chapter also presents the various tools used in this thesis to formulate and analyze the architecture.

Chapter Three: Introduction to the Smart Electricity Grid and explanation of the Smart Grid Information Technology.

Chapter Four: Thesis Framework, with a detailed explanation of the Formulation and Analysis phases.

Chapter Five: Application of previously presented concepts to the two leading MDMS proposals. This section draws important macro and micro-level conclusions for both architectures.

Chapter Six: The two MDMS systems architectures are compared based on their macro-level modularity and micro-level connectivity. This chapter also proposes a generic MDMS

architecture that combines the strengths of both architectures and suggests possible areas of business collaboration for the vendors.

Chapter Seven: A summary the major points and the thesis's conclusion with suggestions for future research areas.

## CHAPTER 2. SYSTEM ARCHITECTURE

Crawley defines System Architecture as “The embodiment of concept, and the allocation of physical/informational function to elements of form, and definition of interfaces among the elements and with the surrounding context.” (7) Form is what the system is and function is what the system does.

Architecture of a system defines the behavior and the structure of a particular system. System architectures satisfy a set of primary functions plus other properties which are referred to as “ilities” such as adaptability, flexibility, robustness and scalability. Architectures evolve over time as the technologies mature and the life of a system is defined by the “ilities” of the underlying architecture. The primary functions of a system have immediate value whereas the “ilities” have long term value and the architect of the system has to resolve tradeoffs between the short terms vs. long term goals while designing the system.

Architecture of a system also defines the complexity of the system. The behavior of a complex system can be very different than the subsystems of that system. The interactions amongst the subsystems lead to certain unplanned behavior. This emergent behavior can lead to some desirable traits and some undesirable traits. The above discussion is summarized in Figure 1

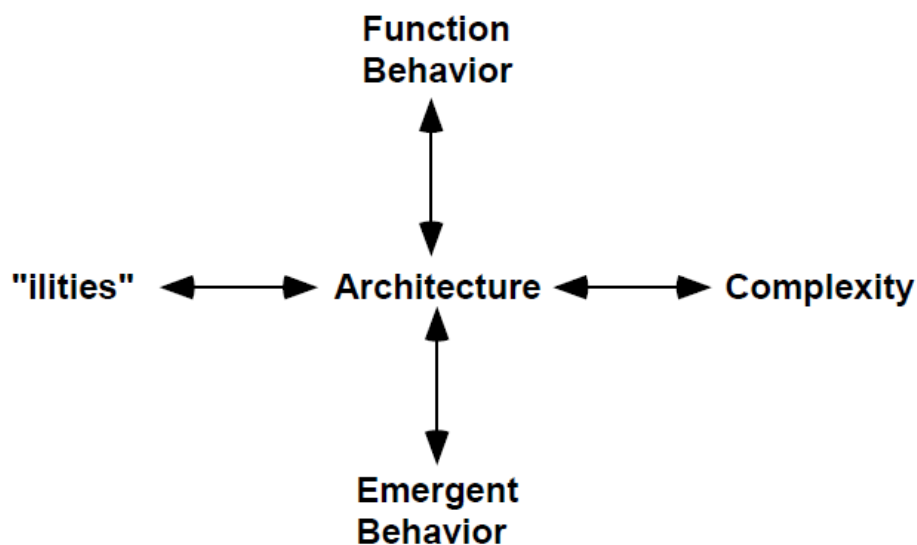


Figure 1 Role of Architecture (8)

### 2.1 IMPORTANCE OF SYSTEM ARCHITECTURE

“The importance of architecture is framed in three domains of importance: as a way to understand complex systems, to design them, to manage them and to provide long term rationality by means of standards.”(8)

Architecture is a representation of the components of a system and their relationship to achieve the desired functionality. The linkages and the interdependencies of the components influence the

behavior of the system when it's operating as expected. At the same time the architecture gives us an idea of the failure modes of the system. Identifying the components and interconnections give us a better understanding of the subsystems of a complex system.

Architects can design systems as a modular architecture or as an integral architecture; both have their advantages and disadvantages. Modular systems have a lot of interconnection within the module and minimal interactions between the modules whereas an integral architecture has high level of interaction amongst all the components.

Modular architecture design puts a lot of pressure on the interfaces in terms of scalability and functionality but in return it enables distributed development and production. Modular architectures might also be easier to replicate and hence protecting intellectual property will be a challenge. Modular architecture design helps in accommodating future uncertainty and makes complex designs more manageable. Most real world systems are hybrid of both modular and integral architectures.

Standards and protocols are required for the long term evolution of a set of systems. Standards often apply to interfaces which are a prime focus of architecture. Standards facilitate the evolution of a system and foster innovation (8) by allowing the designer to focus on immediate challenges of the system design.

Every system has to satisfy the primary need of the user; how well the system satisfies that need is dependant on the internal modules of the system and their interactions. Architectures with similar functionality can be compared in terms of the modules of the system and their dependencies. These architecture characteristics influence "ilities" such as scalability and changeability of the system.

Scalability is the property of a system that maintains or increases its performance and functionality when the scale of the system is increased. This can be achieved by increasing the number of elements or links within the system. An ideal system will scale linearly for every additional component or link but systems can hit a certain bottleneck such that any addition of extra components will cause a decrease in performance. This happens because of the unplanned interactions between the added components and the original system.

Changeability is the property of a system to adapt to changes in order to satisfy a particular need or to evolve the system while still satisfying the primary need. Planned change in certain modules of the system can lead to unplanned changes in other parts of the system. The larger the number of unplanned changes, the lower the changeability of the system. To achieve a system with high changeability understanding the interconnections of the different modules and their interdependencies is extremely important.

Managing a complex system is hard because systems are made of multiple subsystems and their interactions. Some system behaviors are expected while others are not. Also, the end users of the system might use the system in a different way than the architect had anticipated. Architects have to design the system so that the complexity of it is hidden from the end users while making the system adaptable to emergent behavior.



## 2.2 TOOLS

There are many tools to decompose/represent a system and other tools available to analyze a system. Some of the tools used for System Analysis are SysML, Pugh, System Dynamics and QFD analysis. This thesis uses the Object-Process Methodology to decompose the system and the Dependency Matrix analysis leveraging DSM and Network Theory methods to evaluate the architecture.

### 2.2.1 Object-Process Methodology (OPM)

OPM is an approach to design/decompose a system in terms of objects and processes. Object-process analysis combines both the structural and procedural aspects of a system in one coherent frame of reference (9). OPM is expressed textually via the Object-Process Language (OPL) and graphically via the Object-Process Diagram (OPD).

#### *Building Blocks*

OPM consists of object, process and state. Objects are things that exist and they define form/structure, while processes are things that act on objects and can define function or behavior. States are situations in which objects can be. A process can transform the state of an object. There are three types of transformations possible: generating, consuming or affecting.

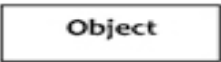
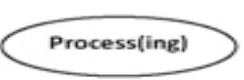

	Visual Representation	Textual Form	Definition	Description
Entities		Nouns; capitalized first letter in every word; if ending with "ing", "Object" is place as a suffix	An <b>object</b> is a thing that has the potential of stable, unconditional physical or mental existence.	Static things. Can be change only by processes.
		Nouns in gerund form; capitalized first letter in every word; if not ending with "ing", "Process" is place as a suffix	A <b>process</b> is a pattern of transformation that an object undergoes.	Dynamic things. Are recognizable by the changes they cause to objects.
		Nouns, adjective or adverbs: non-capitalized	A state is a situation an object can be at.	States describe objects. They are attributes of objects. Process can change an object's state.

Figure 2 OPM Building Blocks (10)

Objects are depicted as rectangles and processes are depicted as ellipses in an OPD. States are shown as rounded rectangles.

#### *Links*

Links are used to represent the connections between objects, processes or objects and processes. The links that connect the objects and processes in the OPM are very important and they have a wealth of information. There are two types of links in OPM - Structural links and Procedural links. A structural relation is a connection or an association between things that holds irrespective of time whereas a procedural link identifies the flow of information.

The Four Fundamental Structural Relations

Shorthand Name	Aggregation	Exhibition	Generalization	Instantiation
Symbol				
Meaning	Relates a whole to its parts	Relates an exhibitor to its attributes	Relates a general thing to its specializations	Relates a class of things to its instances

A fundamental structural relation can have many descendants.  
The different OPL sentences and OPD pictures are listed below.

Structural Relation Name and Shorthand Name	Number of Descendants						Description
	One		Two		Three or more		
	OPD	OPL	OPD	OPL	OPD	OPL	
Aggregation-Participation		A consist of B.		A consists of B and C.		A consists of B, C and D.	B, C and D are part of the whole A.
Exhibition-Characterization		A exhibits B.		A exhibits B and C.		A exhibits B, C and D.	B, C and D are attributes of A. If B is process, it is an operation of A.
Generalization-Specialization		B is an A.		B and C are As.		B, C and D are As.	B, C and D are types of A.
Classification-Instantiation		B is an instance of A.		B and C are instances of A.		B, C and D are instances of A.	B, C and D are unique objects of the class A.

The four fundamental relations are also applicable to process. Only exhibition can link objects with processes. Instantiation cannot generate a hierarchy while the other three can. Any number of things can be linked to the root.

	Aggregation	Exhibition	Generalization	Instantiation
Object				
Process				

Figure 3 Structural Links (10)

There are four kinds of structural relations (Figure 3):

- Aggregation – Participation relation: specifies the part to whole relationship for both objects and processes
- Generalization – Specialization relation: identifies specific example of a certain object or process
- Featuring – Characterization relation: represents the connections between things and their features
- Classification – Instantiation relation: connects things to their instances.

General tagged structural links provide additional “user-defined” links with specified semantics. (Figure 4)

**Tagged Structural Links**

Generally used between objects, but may also be used between processes.  
 Cannot be used to link an object to a process.

Link Name	Object Process Diagram (OPD) Symbol	OPL SENTENCE	Description
Tagged		R Object refers to S Object.	Relation from source object to destination object; relation name is entered by architect, and is recorded along link.
(Null)		R Object relates to S Object.	Relation from source object to destination object with no tag.
Bi-directional Tagged		R Object precedes S Object. S Object follows R Object.	Relation between two objects; relation names are entered by architect, and are recorded along link.
(Null) Bi-directional		R Object and S Object are related.	Relation between two objects with no tag.

Figure 4 Tagged structural links (10)

There are three types of procedural links:

- Enabling links that connect a process with an enabler of that process
- Transformation links that connect a process with an object transformed by the occurrence of that process
- Event links denote process executions due to internal and external events

**Procedural Links**

These links are generally used between an object and a process. They cannot be used to links objects together.

Link Name	OPD Symbol	OPL Sentence	Description
Consumption		Processing consumes Object.	Process uses object up entirely during its occurrence.
Result		Processing yields Object.	Process creates an entirely new object during its occurrence.
Effect		Processing affects Object.	Process changes the states of the object in an unspecified manner.
Input and Output		Processing changes Object from input state to output state.	The object is at input state prior to the process occurrence, and at output state as a result of its occurrence.
Agent		Object handles Processing.	Object is a human that is not changed by the process; process needs the agent object in order to occur.
Instrument		Processing requires Object.	Object is a non-human that is not changed by the process; process needs the instrument object in order to occur.
Invocation		X Processing invokes Y Processing.	First process directly stars up a second process, without an intermediate object.

Figure 5 Procedural links (10)

A system can be represented as a set of inter-related hierarchically organized OPDs. By zooming in and out, the system and the interconnections can be viewed at different levels of detail. A detailed diagram showing all the OPDs and the interconnections gives us a view of the architecture of a system.

### 2.2.2 Design Structure Matrix (DSM)

DSM is an information exchange model and was primarily designed for optimizing the product development process. It has since being used to represent the interactions between the elements of a decomposed system. The OPM tool described in section 2.2.1 gives a detailed graphical representation of the interconnections between the different elements of a decomposed system.

#### *DSM Construction*

DSM is a matrix representation of a directed graph. The columns headings and the rows headings correspond to the nodes of the graph and the marks in the matrix specify the relationship between the nodes. The diagonal elements of the DSM are ignored because they have no information (11).

	A	B	C	D	E	F	G
A		●			●	●	
B				●			●
C		●		●			●
D		●	●		●		●
E				●		●	
F	●				●		
G		●	●	●			

Figure 6 DSM (11)

For example in the DSM in Figure 6 the dots in the matrix signify the relationship between the nodes A to G. The outputs are represented by going down the columns – node C sends information to node E and node G. The inputs are represented by going across the rows – node C receives information from node B, node D and node G.

A systematic taxonomy and a quantification scheme to define the relationships between the nodes are very helpful in analyzing DSMs (11). The system taxonomy classifies the interaction in a physical DSM as shown in Table 1. Each cell can be split into 4 quadrants and the different interaction can be marked using a different color code.

Interaction	Description
Spatial	Associations of physical space and alignment
Energy	Needs for energy exchange
Information	Needs for data or signal
Material	Needs for material exchange

**Table 1 Taxonomy of System element interactions (11)**

The marks within the DSM can be assigned specific weights to differentiate the strength of the connection (Table 2) between the nodes as proposed by Pimmler and Eppinger (12). Adding the interaction and weight information makes the DSM representation much richer than the original DSM shown in Figure 6.

Strength	Title	Explanation
3	High	Significant flow of three or more of the interaction types
2	Medium	Significant flow of two of the above
1	Low	Significant flow of one of the above
0	Zero	No significant relationship

**Table 2 Quantification Schemes (11)**

### ***DSM Analysis techniques***

Sequencing algorithms are used to minimize the off-diagonal dependencies within the DSM. Once sequencing is done Clustering algorithms are applied to optimize the ordering of the nodes of the DSM and their aggregation into groups. Clustering algorithm is used in architectural analysis which is the focus of this thesis.

DSM clustering is the process of finding subsets of DSM elements that are mutually exclusive or minimally interacting. The elements within a cluster have high levels of interaction but the interaction between clusters is minimized or eliminated if possible (13,14).

The DSM in Figure 6 can be clustered in the following two ways with most of the interactions within two separate blocks (AD and EBCG) (Figure 7 a). It can also be clustered as shown in (Figure 7 b) where component E is part of two clusters and will act as an interface between the two groups. The decision of how to cluster a particular DSM will depend on the underlying technical as well as non-technical factors (11)

	A	F	E	D	B	C	G
A		●	●				
F	●		●				
E		●		●			
D			●		●	●	●
B				●			●
C				●	●		●
G				●	●	●	

(a) Clustered DSM

	A	F	E	D	B	C	G
A	A	●	●				
F	●	F	●				
E		●	E	●			
D			●	D	●	●	●
B				●	B		●
C				●	●	C	●
G				●	●	●	G

(b) Alternative Clustering

Figure 7 Clustered DSM (11)

2.2.3 Networks

Depicting data in the network form and applying network metrics has a number of conceptual benefits for evaluating complex systems. Networks analysis focuses on connectivity at a micro-macro level to identify important elements and relationships within a system.

The basic element of a network is the Adjacency matrix that defines the relationships between the different elements. The network map for the DSM in Figure 6 will look as shown in Figure 8

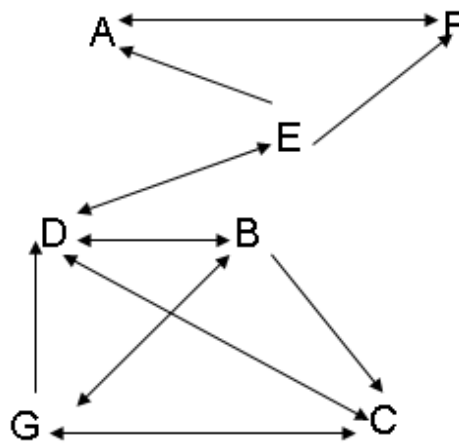


Figure 8 Network Diagram for DSM in Figure 6

Networks consist of nodes connected to each other via links. The links can be bidirectional or unidirectional and this leads to directed graphs and undirected graphs. The links between the nodes can have weights assigned or can just have binary values: 1 if the link is present and 0 if the link is absent. The weight plus the direction leads to four different kinds of graphs – Binary Undirected, Binary Directed, Weighted Directed and Weighted Undirected.

Adjacency Matrix is a matrix representation of the dependency in Figure 8 and is identical to the DSM in Figure 6. By replacing the marks in the DSM by 1 and the blanks by zero we get the Adjacency Matrix. The Adjacency Matrix is also referred to as the Dependency Matrix. DSM, Adjacency Matrix and Dependency Matrix are different names of the matrix used to represent the relationships between the components.

### ***Network Metrics***

Once we have the Adjacency matrix, network analysis has multiple techniques to decipher information from the matrix. Connectivity is one of the basic concepts of graph theory and is a representation of the flow of information/material in the system. The number of interconnections is a measure of the robustness of the network. Some nodes in a network have many connections while others have very few connections and this information is very important in understanding their attributes and behavior. In a directed graph, a node with many outgoing connections exerts a lot of influence on the other nodes and thereby on the system whereas the other nodes influence a node with many incoming connections. Identifying the highly connected nodes and their location in the structure is critical in understanding the underlying design.

Connectivity metrics might also influence redesign efforts. Understanding how the redesign of a particular node will affect the entire system helps in planning and implementation. On the other hand design modification/enhancement of nodes with low connectivity can be achieved without affecting the performance or architecture of the entire system. Identifying the highly central nodes also helps the system architect understand whether the system has enough redundancy built in to withstand the failure of the central nodes.

Metrics such as Indegree Centrality, Outdegree Centrality, Visibility and Dependency help in understanding and quantifying the connectivity of a system.

### **Indegree Centrality**

Indegree centrality measures the number of inputs to each node in the network. The nodes with high Indegree centrality receive or request a lot of information from the other nodes in the system. From an architectural standpoint, these nodes have to interface with the other nodes in the system to get the required information. The performance of the system might be to a large extent dependant on the nodes with very high Indegree centrality because these nodes will have to process all the information they receive or request in a timely and orderly fashion.

### **Outdegree Centrality**

Outdegree centrality measures the number of outputs from each node in the network. From an architectural standpoint, these nodes have to interface and satisfy the requirements of the other nodes in the system or push information to other nodes in the system. Knowledge of the Outdegree centrality along with the domain knowledge of the system is very useful in identifying the underlying component requirement of the system.

### **Dependency**

Indegree Centrality measures the first degree effects of the input connectivity whereas dependency measures the effect of higher order connections. Suppose a node A is dependent on B

and node B is dependent on C hence A is indirectly dependant on C via a second order connection. The Dependency metric is calculated by raising the Adjacency matrix to the  $n^{\text{th}}$  power, where n is the number of nodes in the matrix and then summing up the columns (columns represent inputs) for each node.

### Visibility

Visibility is similar to dependency but it works on the outputs for each node. For example if node D outputs information to node E and node E outputs information to node F, then node E is indirectly outputting information to node F. The Visibility metric is calculated similar to Dependency but instead of summing the columns we sum the rows of the  $n^{\text{th}}$  powered Adjacency Matrix



## CHAPTER 3. SMART GRID

The current US national electric power infrastructure is rapidly reaching its limits. Today's electricity system is 99.97% reliable, yet it still allows for power outages and interruptions that cost Americans at least \$150 billion each year (15). The current electricity grid cannot keep up with the ever growing demands of electricity in an exceedingly digital world. Since 1982 growth in peak demand for electricity has exceeded transmission growth by almost 25% every year (15). Efficiency, reliability and environmental impact of the grid are the other major areas of concern. If the grid was just 5% more efficient, the energy savings would equate to permanently eliminating the fuel and greenhouse gas emission from 53 million cars (15). Smart Grid is the way forward to building a more efficient, sustainable and 21<sup>st</sup> century Grid.

### 3.1 SMART GRID TAXONOMY

There are many definitions of the Smart Grid and it means different things to different people. In the very basic sense one can think of Smart Grid as adding intelligence (Smarts) to the current 100+ year old electrical grid. Frost and Sullivan define Smart Grid as “consisting of a web of technologies aimed at automating, improving efficiency, and increasing availability of the electric grid ranging from generation, transmission, and distribution levels. Automation also includes tools to conduct predictive, preventative and supply analysis based on data collection that is conducted at the transmission and distribution level.”

GTM Research (16) correctly identifies Smart Grid as the convergence of three sectors – Electric Power which is the physical power layer, Telecommunication Infrastructure which is the communication and control layer and Information Technology which is the application and services layer. Expertise from all three layers is required for a complete end-to-end Smart Grid Infrastructure. Each layer has its own set of challenges that needs to be overcome.

Clean renewable energy sources are needed to make the current grid more sustainable and green. At the physical power layer the biggest challenge is adding clean renewable energy sources to the existing grid. The middle of the country is abundant in renewable energy sources such as wind and solar but the largest demand for electricity comes from the east and west coast. The loss of energy when transporting via existing transmission lines over such large distances makes it unfeasible. New extra-high voltage transmission lines are required to make it viable and the funding needed to make this possible is the biggest obstacles to achieving a greener grid. The other challenge with renewable energy sources such as solar and wind is that they are dependant on the sun and wind and the demand for electricity will not match the supply of energy, hence the energy will have to be stored in batteries to match the demand-supply requirement. Also the utilities will need multiple energy producing sources to meet the demands of the end user since the supply from any single source might not be sustained. Hence technology that enables smart switching between the different energy sources is needed to enable a renewable electrical grid.

Upgrading the current transmission lines to facilitate this long-distance transmission of renewable energy, and adding technology to store that energy and to enable seamless switching between these different sources of energy is a vital for a future sustainable, green and efficient Smart Grid.

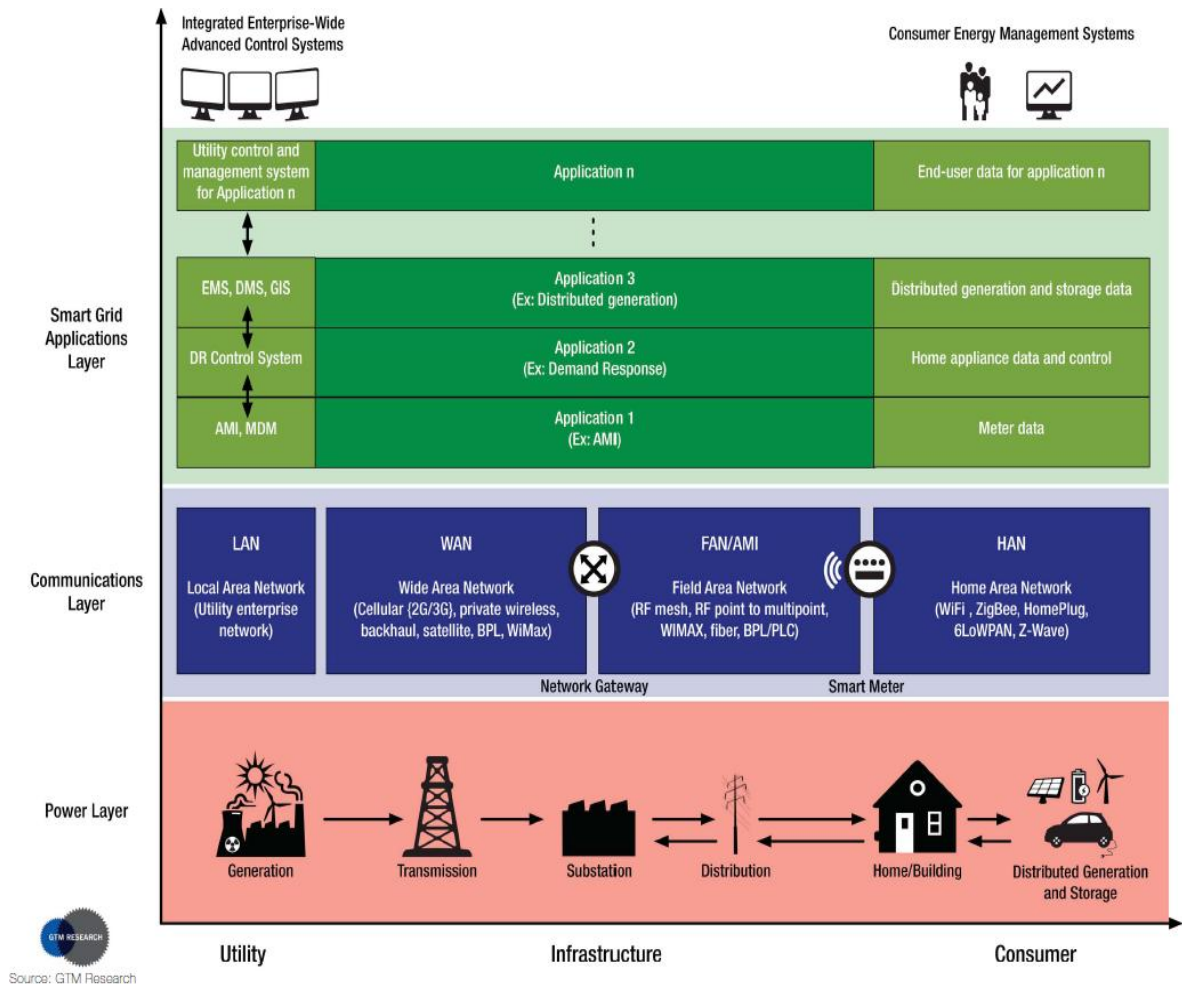


Figure 9 Smart Grid (16)

The telecommunication infrastructure of the current grid facilitates only one-way flow of information. The Smart Grid will enable two-way communication between the consumer and the utility company which will lead to a more responsive and reliable grid. There are numerous technologies to make this possible but the biggest hurdle is to define a standard set of communication rules to foster collaboration and innovation.

The two-way communication between the end consumer (Smart Meter) and the utility companies leads to the generation of large amounts of data. The data contains a wealth of information that will enable and enhance many business applications and customer service. The information can be used for effective outage management response, resource planning, asset allocation and

workforce management. The two-way communication enables demand response strategies which will keep the price of electricity from sky rocketing.

In this thesis the focus of this case study will be primarily the Information Technology layer and more specifically, the flow of data. The following discussion explains in detail the flow of data through the Smart Grid Infrastructure – Collection of Data → Cleansing and Storage of Data → Utilization of Data.

### 3.2 SMART GRID DATA FLOW

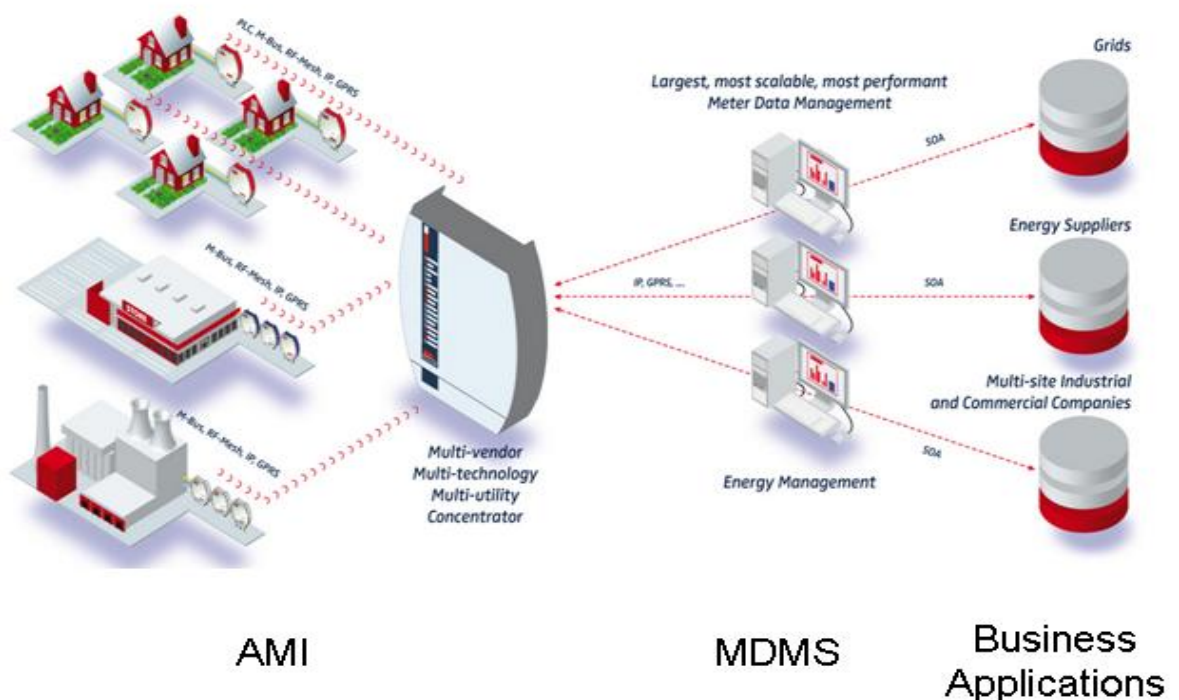


Figure 10 Smart Grid Data Flow (17)

Based on Functionality the Smart Grid Information layer is broadly classified into three sections:

- Advanced Metering Infrastructure (AMI) collects data from various Meters and facilitates the two way communication between the meters and the utilities
- Meter Data Management System (MDMS) authenticates, verifies and stores the data collected from AMI systems
- Business Applications/Operation System leverages the data from the MDMS to detect problems in the system and to operate the system more efficiently

### 3.2.1 Advanced Metering Infrastructure (AMI)

AMI is vital in enabling the many potential benefits of the Smart Grid and, as a result, a big chunk of the federal energy is allocated to installing smart meters. Installing the AMI is also the largest cost component of a large Smart Grid Information layer deployment.

AMI consists of the following components:

- Smart Meters to timestamp electricity usage and to enable net metering which facilitates selling energy back to the Grid
- Front-end two way Communication interface to collect data from individual smart meters and to send signals from the utility back to the meter
- Back-Haul Communication interface to aggregate the data from the millions of individual Smart Meters and to transport the data to the Central Repository System, which is the Meter Data Management System (MDMS)

#### *Smart Meters*



(image source: Chicago Tribune <http://www.chicagotribune.com>)

**Figure 11 Old Meter (Left) vs. Smart Meter (Right)**

A smart meter is an advanced metering device that timestamps the electricity consumption, supports automated meter reading and interprets signals from the utility. Currently, consumers pay a flat fee for electricity irrespective of the time of usage while the price of electricity varies over the time of the day, as shown in Figure 12.

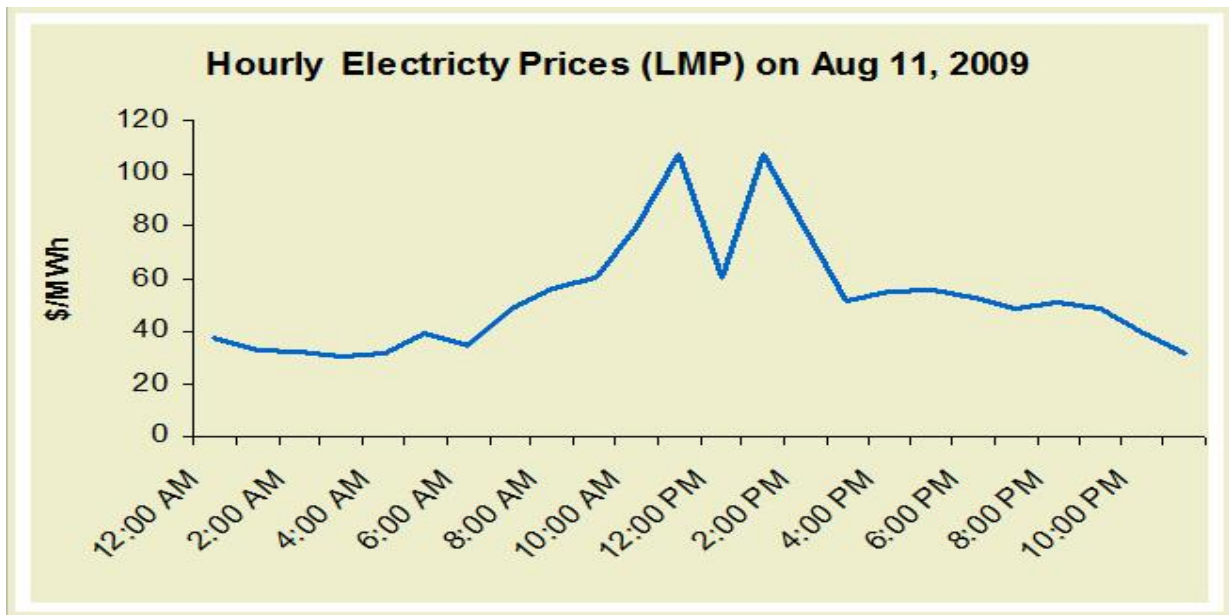


Figure 12 Bulk Electricity Prices on a hot summer day (source: [ISO-NE](#)) (18)

The time stamping of electricity consumption and the two-way communication enable the utilities to charge the end consumer variable pricing based on usage. The customer will be able to monitor their energy usage and change their behavior which will lead to a smoother electricity demand curve and reduction of prices of electricity. This lower peak demand will prevent utilities from bringing highly pollutant power plants online and make the grid much greener. It will also make the current grid much more reliable because the chance of demand overshooting supply will be less likely.

#### ***Front-end Two way communication interface***

The communication interface between the smart meters and the Head End System (HES) can be either wireless or wired. HES combines data from various smart meters and transports it over the back-haul communication interface. There is no set of standards for the front-end communication interface so different meter vendors use different technologies ranging from RF to WiMax to Fiber (Figure 9).

The front-end two communication interface helps the utility gather billing and event information automatically. It reduces cost of manpower and increases privacy for the end-consumer because the utility meter readers will not need to enter the property to gather the information. It will also expedite problem detection and mitigation and will result in increased customer satisfaction.

#### ***Back-Haul Communication interface***

The back-haul communication interface transports data from the HES to the central repository. The Wide area networks used for this communication are shown in Figure 9. These are existing networks that the utilities will have to lease from communication companies. The back-haul infrastructure and the relationship between the communication companies and the utilities are already in place. However, the amount of data being transported by the utilities over the back-haul will increase exponentially and the utilities will require better service levels in order to

leverage time-sensitive data. Communication companies and utilities will need to sign new contracts to satisfy the above stated needs.

### **3.2.2 Meter Data Management Systems (MDMS)**

The Meter Data Management system is the central repository where the meter data collected from different AMI installations and Legacy installations is stored, processed and leveraged for business applications.

The MDMS is responsible for the life-cycle management of the aggregated meter reads. It interfaces with the multiple external AMI collections systems to import data into the system. The utilities are migrating from Legacy collection system to AMI collection system and the MDMS should have the capability to facilitate this migration. The migration can happen over a period of time because the utilities might want to pilot some programs before going for a full AMI deployment and the MDMS should be able handle such a staged migration. The meter reads from the collection system can have missing data, incorrect data or corrupted data. It is the responsibility of the MDMS to validate the incoming data before loading it into the database. The MDMS also handles the estimation and editing of the data if required. Since the Smart Grid enables two-way communication the MDMS is able to request on-demand meter reads from the AMI systems in response to a business application request.

MDMS enables multiple methods to securely access the meter reads such as file transfer, thin clients and GUI interface. The data stored in the MDMS is made available to the business intelligence application in the requested format. Since all the business intelligence units access and use the same dataset the results are consistent and accurate. The MDMS also sets the policies regarding backup and recovery, disaster recovery, and data archiving and restoration.

MDMS consists of the following core components:

- AMI, Legacy and Future Collection System Integration
- Meter Meta Data and Meter Consumption Data Management
- Versioned Data Storage
- Two-way communication interface between business applications/operations and AMI
- Interface to enable AMI applications and business operations.

The amount of data captured by the utilities is expected to grow exponentially (Figure 13). MDMS should be designed to be extremely scalable to handle such large amounts of data.

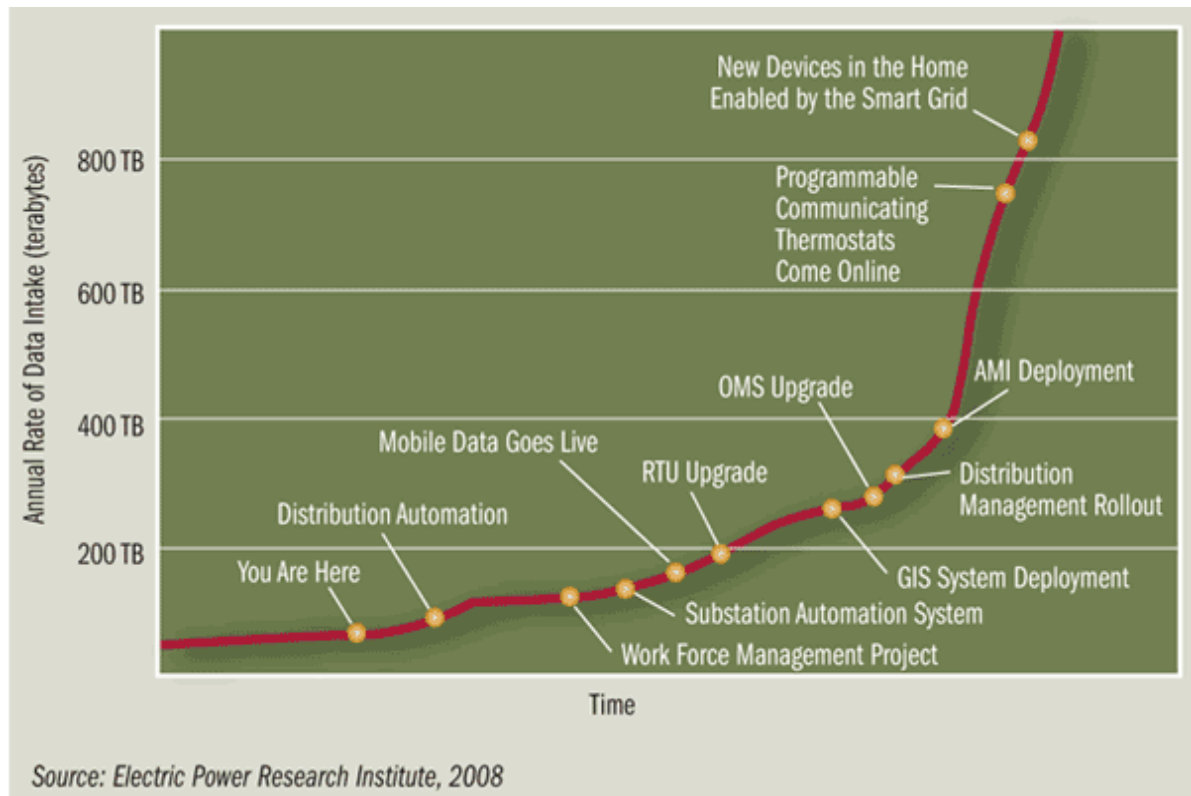


Figure 13 Exponential Growth of Data (19)

### ***AMI, Legacy and Future Collection System Integration***

Utilities are in the process of migrating from legacy metering systems to AMI installations. A typical utility has a mix of legacy collection systems and AMI collection systems. The MDMS solution has AMI and Legacy metering interface adapters to gather data from the various collection systems. The MDMS also provides the opportunity to integrate future collection systems when they come online.

The interface adapters enable the business intelligence processes to request information from AMI systems and push information over to the end consumers. The interface adapters also help in the migration of legacy systems to AMI systems without effecting core business applications such as billing.

### ***Meter Meta Data and Meter Consumption Data Management***

MDMS is collecting meter information from millions of meters and some of this data will be questionable. The data has to be validated before it can be used by other business intelligence applications because questionable data will lead to incorrect results.

A Data Validation, Estimation and Editing (VEE) module can identify questionable data and processes the data before it reaches the business intelligence applications. The Validation process flags every read valid or invalid based on a set of checks specified by the utilities best practice rules. Invalid data that can be estimated are processed by the estimation engine using historical data and statistical modeling. Data that cannot be estimated can be edited by the utility based on business rules and regulations and the record of such transaction is logged for future auditing.

MDMS should provide utilities with mathematical tools to calculate complex loads and billing determinants. The Common mathematical interface also mitigates unit and time conversion errors that result from data being collected from multiple AMI and legacy vendors.

#### ***Versioned Data Storage***

Versioning is an important feature of MDMS. Each meter read is time-stamped and stored which help in maintaining data integrity across multiple business applications. MDMS keep detailed logs of each change and the user or business application that made the change. Versioning of data is legally important to resolve billing disputes. Accounting standards such as the Sarbanes-Oxley Act require companies to store data for a long time and reproduce the data when required.

#### ***Two-way communication interface between business applications/operations and AMI***

Two-way communication interface opens the possibility of multiple business applications and services. For e.g. the utility can remotely diagnose problems, send pricing information to the consumer and request meter reads for billing.

#### ***Provide an interface to enable AMI applications and business operations***

MDMS allows for easy integration of AMI applications via a set of APIs and adapters. MDMS enables the applications to leverage the data and use it for vital business intelligence applications such as Outage Management System, Asset Management System and Demand Response System

### **3.2.3 Business Applications/Operations**

Business Applications and Operations leverage the meter data to deliver business services. The data mining opportunities of the MDMS are endless. Some of the core and optional business services are as follows:

- Billing System
- Outage Management System
- Workforce Management System
- Asset Management System
- Demand Response System
- Transformer Load Manager
- Revenue Protection and Theft analysis system
- Customer Information System

#### ***Billing System***

Billing Systems are vital for the survival of the utility company. The Bills have to be accurate and timely. Smart meter systems will allow the utilities to charge the customer time of use (TOU) pricing – higher prices for peak period usage and lower prices for off-peak period usage.



***Outage Management System (OMS)***

When the Legacy Grid experiences power failure it is the end consumer who reports the failure and triggers the service request. With the Smart Grid it's the OMS that alerts the utility when the power failure occurs and allows the problem to be quickly diagnosed and isolated. The work crews are sent out only if the problem needs physical fixing and everything else is solved remotely which increases responsiveness and customer satisfaction.

***Workforce Management System (WMS)***

WMS enables automation and optimization of the work-order process. Smart Grid data facilitates the optimization of processing orders and scheduling field crews which allows for better use of all the assets. For e.g. the automated meter reading reduces the number of workers needed to go out on rounds to collect meter readings.

***Asset Management System (AMS)***

AMS manages meters, data communication modules, communication nodes and the physical power layer.

***Demand Response System (DRS)***

The U.S Department of Energy's official definition of demand response (DR) is as follows: "Changes in electricity usage by end-use consumers from their normal consumption pattern in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized." Smart grid technology enables DR.

***Transformer Load Manager***

Transformer Load Manager monitors the aggregated meter reads and calculates peak loads on individual distribution transformers to identify potential overload and underutilized situations.

***Revenue Protection and Theft analysis system***

The Revenue Protection and Theft analysis system utilizes the meter read to flag potential cases of theft. This method is far more efficient, reliable and cost-effective than sending the utility's workforce out to the field for theft detection.

***Customer Information System (CIS)***

Customer information System keeps the entire customer account information. Data mining of CIS leads to finding of valuable customer profile information and usage information which can be used for resource planning and market analysis.

Figure 14 illustrates an example Smart Grid installation with all the components and the interactions.

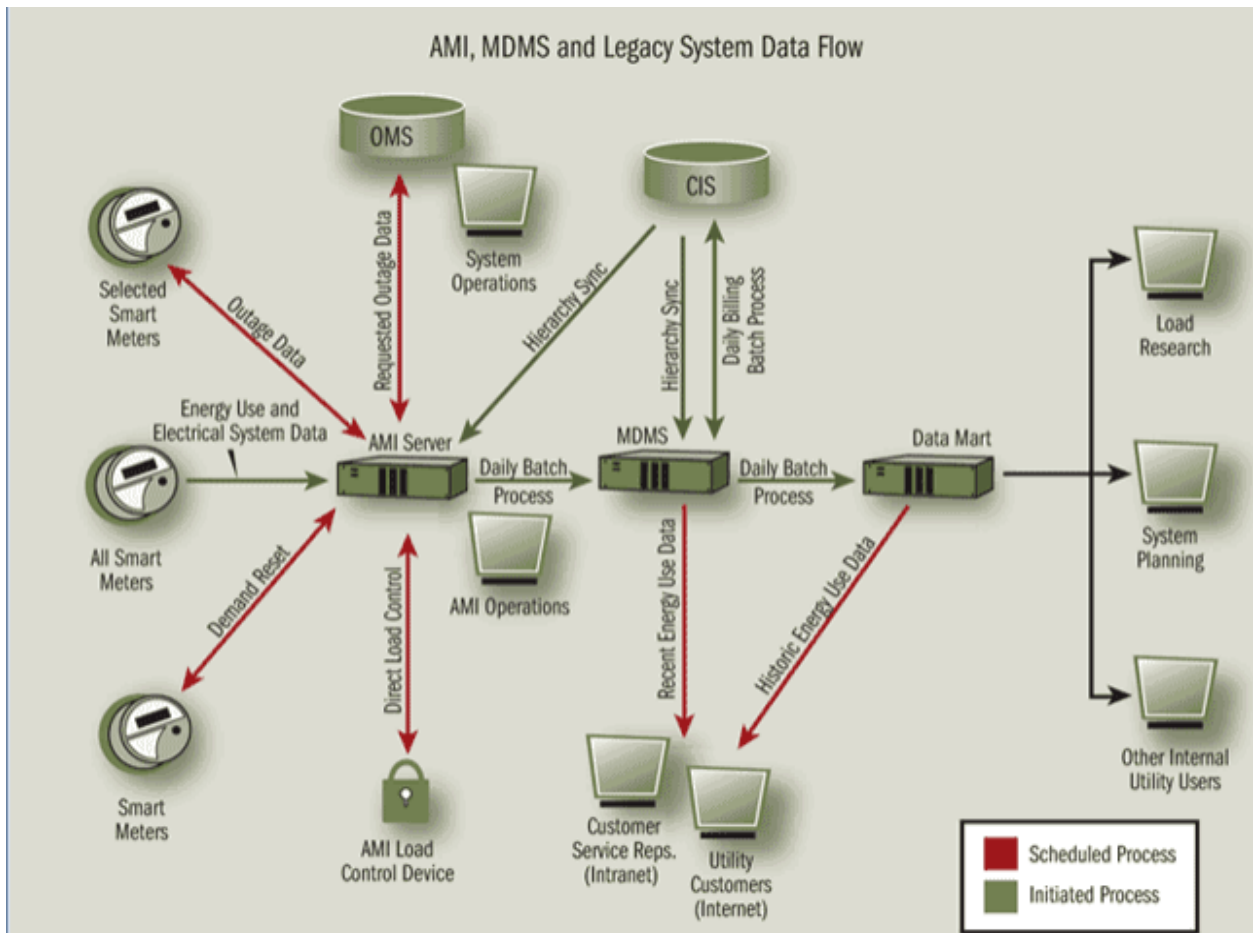


Figure 14 Example Smart Grid Data Flow Installation (19)

## CHAPTER 4. SYSTEM ARCHITECTURE ANALYSIS FRAMEWORK

Previous research on system architecture has either focused on the product level (2-4) or at the component level (5). Sharman and Yassine (6) have analyzed the architecture at different levels starting at the bottom, intermediate and the top but the research is more qualitative than quantitative whereas Sosa and Eppinger (5) offer metrics to understand the architecture at the component level. This thesis aims to understand the higher level and the lower level architecture of a system using both qualitative and quantitative metrics.

This thesis proposes a two phase approach to understand the architecture of a system – Formulation and Analysis. The first step to understanding anything complex is to break it down iteratively into smaller chunks. Formulation decomposes the system into smaller chunks and identifies the interconnections of these smaller chunks. The Analysis step analyzes these smaller chunks at a macro level and micro level in order to comprehend the architecture of a system.

Macro-level analysis identifies the different groups in the system and their interactions and it influences strategic decisions such as product development organization structure (2), product life cycle(20), supply chain (21-23) and product portfolio definition (24-26). Macro-Analysis is also vital in terms of comprehending system scalability and functionality. The modules and their interactions influence the scalability of the system while the absence of certain modules within a system might indicate missing system functionality.

Micro-Analysis classifies the components in the system based on connectivity. Decisions such as component design/redesign, complex product development, component-level outsourcing and mitigation of component obsolescence (5,27-29) are based on understanding the micro-level architecture of the system. Understanding how the redesign of a particular node will affect the entire system helps in planning and implementation. On the other hand, design Modification/enhancement of nodes with low connectivity can be achieved without affecting the performance or architecture of the entire system. Identifying the highly central nodes also helps the system architect understand whether the system has enough redundancy built in to withstand the failure of the central nodes. Potential system bottlenecks can also be identified by using the micro-level analysis. Many macro-level design decisions depend on the micro-level product architecture and vice versa and hence it is important to comprehend the architecture at macro-micro levels to get a broad and in-depth understanding of the system.

Figure 15 gives a pictorial representation of the framework. Formulation is made up of three steps – Identifying the System Boundary, Identifying the Object-Process System levels and Creating the Dependency Matrix. Analysis is composed of two steps – Macro-Level Analysis, which identifies the system modules and their interdependencies based on Clustering Analysis and Visibility-Dependency Signature Analysis, and Micro-Level Analysis, that identifies the central components in the system based on the connectivity metrics of Indegree centrality, Outdegree

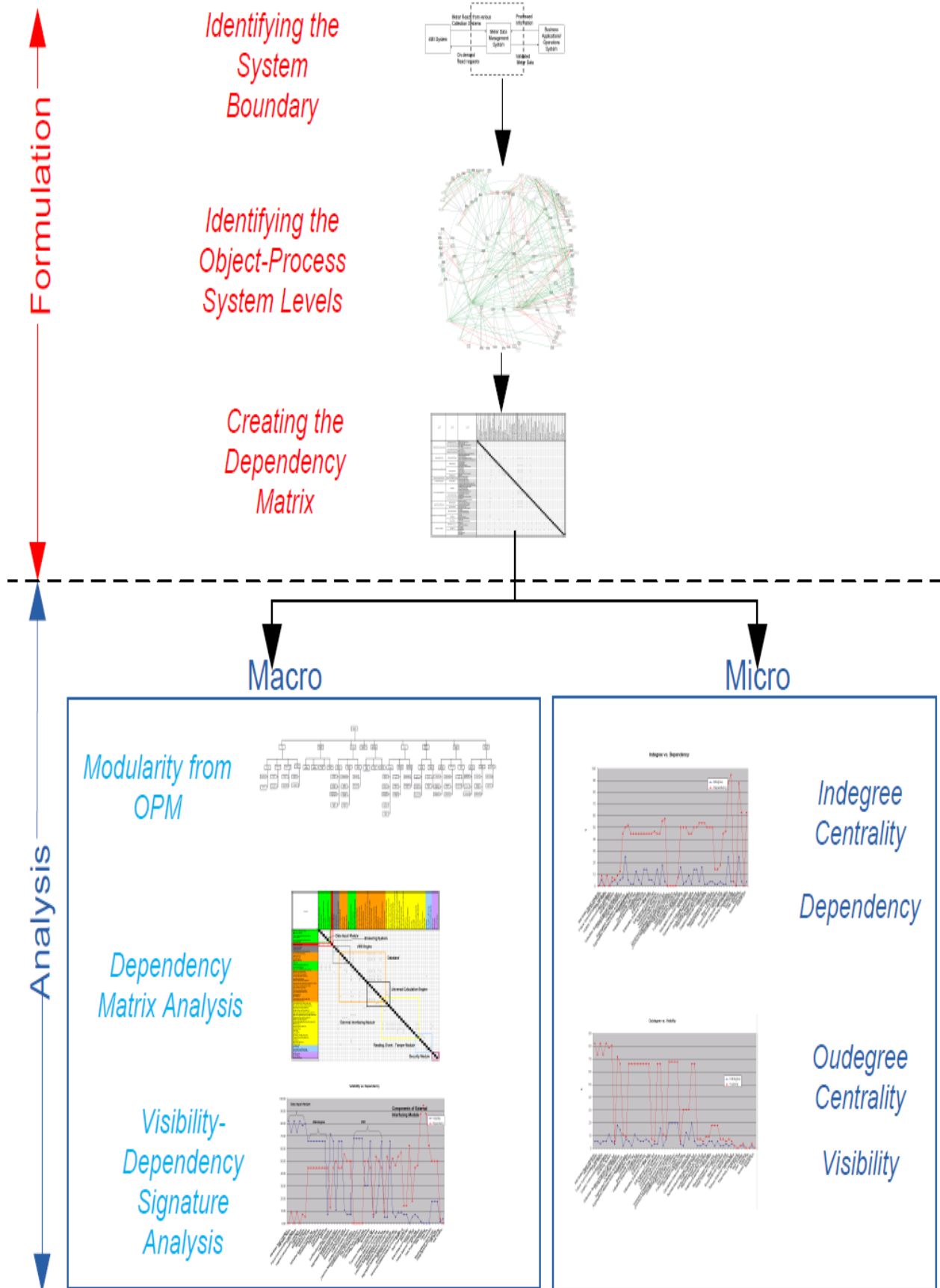


Figure 15 Framework

centrality, Visibility and Dependency. The conclusions are drawn based on simultaneously interpreting the results derived from the Macro-Level and Micro-Level Analyses.

The System Architecture Analysis Framework is explained by applying it to the architecture of the Scientific Calculator Tool available on all Windows computers.

### 4.1 FORMULATION

A thorough understanding of the system is the necessary first step in order to analyze the system. Formulation (Figure 16) of a system’s architecture involves steps necessary to comprehend the system that consists of:

- Identifying the System Boundary: Identifying the primary purpose of the system and the inter-system interactions required to fulfill that purpose
- Identifying the Object-Process System Levels: Decomposing the system into smaller entities and identifying the relationships between these entities.
- Creating the Dependency Matrix: Presenting the interrelationship in an easy to analyze format.

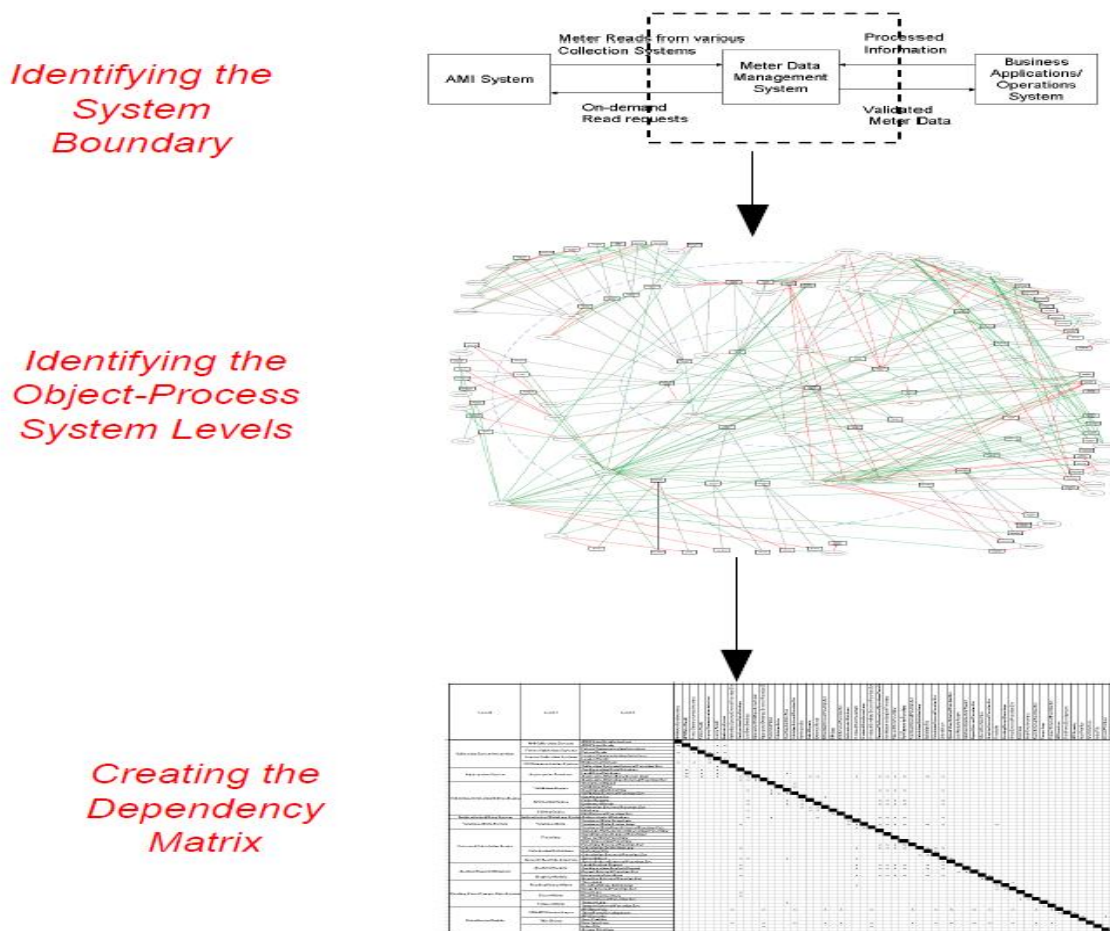


Figure 16 Formulation

### 4.1.1 Identifying the System Boundary

Identifying the boundaries of a system and its interactions with other systems in the environment is the first and most important step in understanding the system architecture. Almost all real-world systems are made of multiple subsystems that interact with each other making the task of defining boundaries very challenging.

System boundary is defined based on structure or function. It involves understanding the environment in order to identify other systems within that ecosystem and to determine their primary purpose and their interaction with the system of interest.

The Scientific Calculator system boundary is as shown in Figure 17. The keyboard is used for inputting the numbers and the Monitor displays the numbers. The Calculators primary function is to process the numbers received from the keyboard and generate the desired output to be displayed on the Monitor.

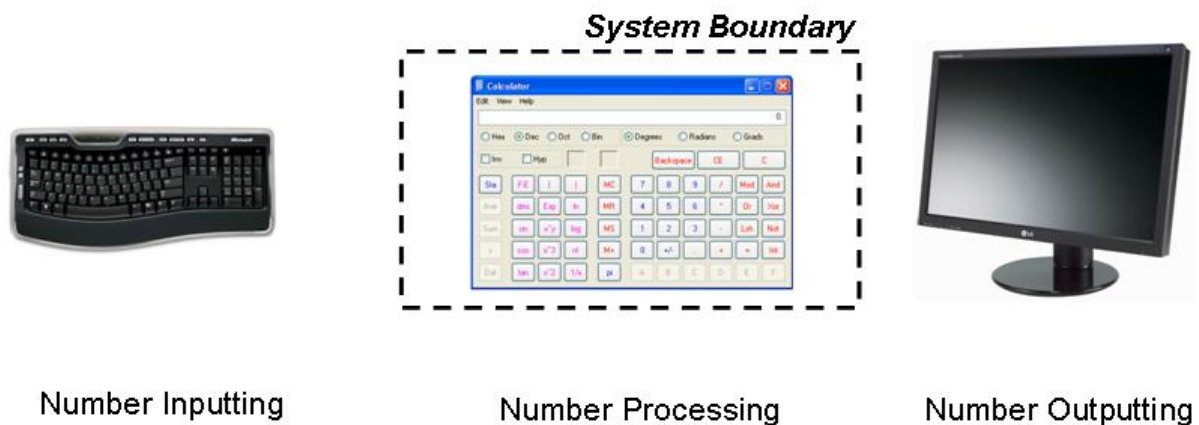


Figure 17 Scientific Calculator: Identifying the System Boundary

### 4.1.2 Identifying Object-Process System Levels

Once the system boundary is identified, the next step is to break down the system into individual components. We start with the system as a whole and then break it down to the next layer. Then we take this next layer and break it down further and so on. Each layer is referred to as a level. Simon (30) states that such hierarchical decomposition helps us analyze the complexity of a system without specifying the content of that complexity.

First we decompose the system based on form components while simultaneously decomposing the same system based on functions. We then combine the form decomposition and function decomposition to derive the single system decomposition which depicts the interrelations between the individual components. OPM is an ideal tool to decompose the system because objects capture form and the processes capture function. The form OPD and the process OPD are combined to create the system OPD.

The Scientific Calculator's object decomposition is as shown in Figure 18 and the process decomposition is as shown in Figure 19. The object decomposition in Figure 18 can be broken

down to another level. For example the Arithmetic Operators in Level 2 can be decomposed into addition, subtraction, division and multiplication but our primary goal is to understand the basic architecture of the Scientific Calculator so identifying that there is an Arithmetic Object to handle all the arithmetic functions is all that is relevant. The extra level decomposition does not add any substantial information and therefore may not be necessary. Similarly for the process we could have broken Function Selecting into different functions such as trigonometric selecting or statistic selecting. The number of levels we need to decompose the architecture depends on the depth of understanding required for a particular analysis and it is left to the discretion of the user decomposing the system.

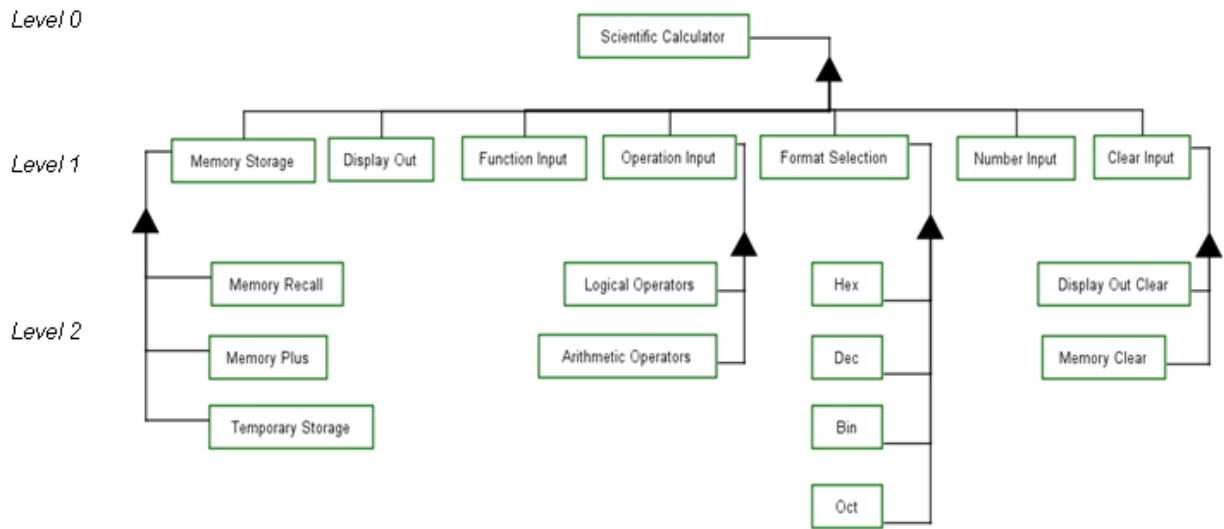


Figure 18 Scientific Calculator: Object Decomposition

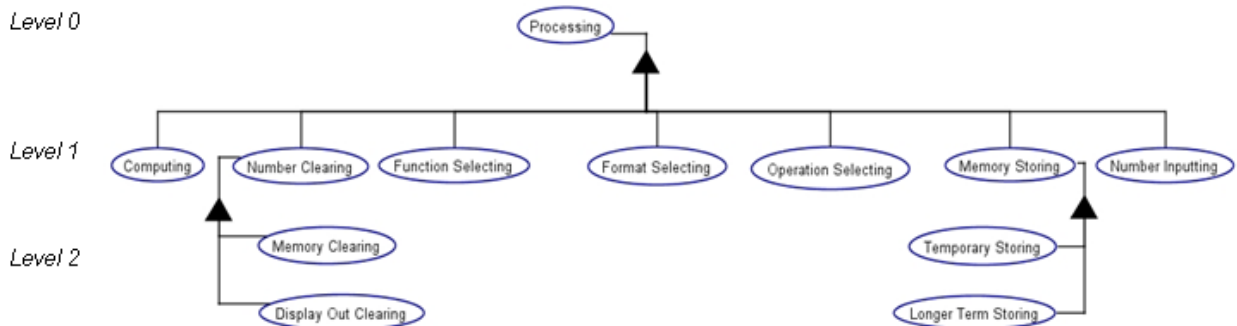
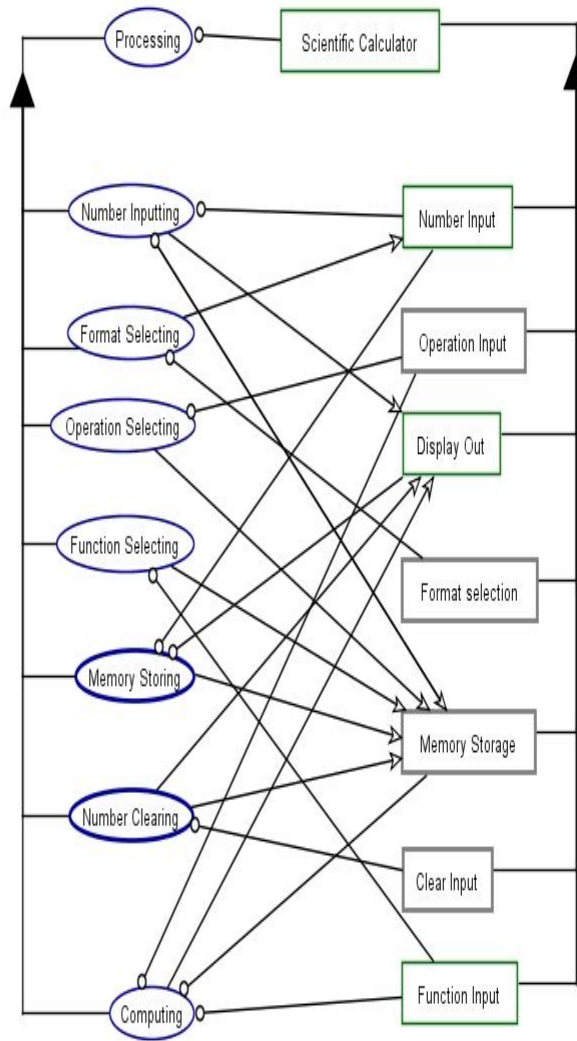


Figure 19 Scientific Calculator: Process Decomposition

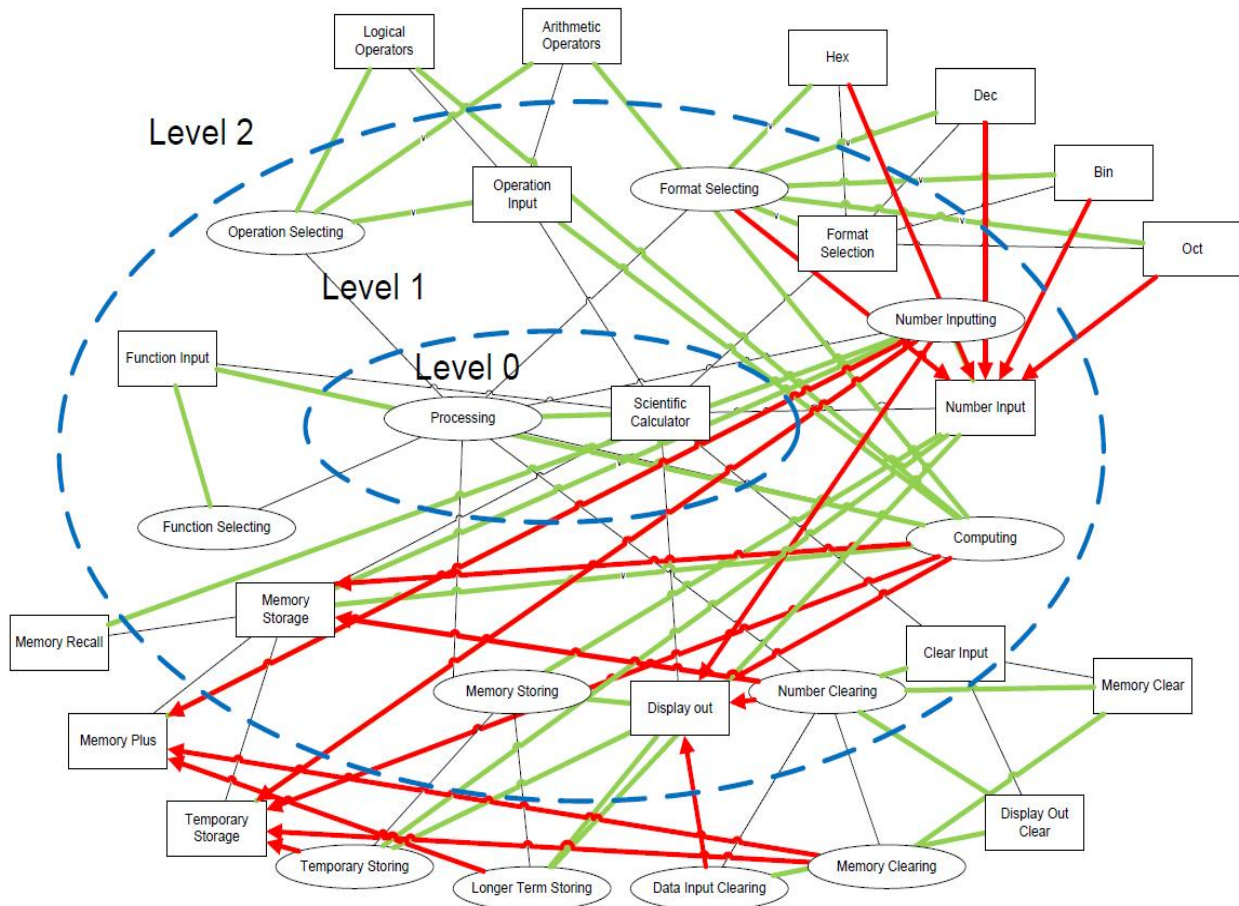
Creating the combined Object-Process Diagram involves establishing links between objects using the processes. The resultant Level 1 OPD is as shown in Figure 20. Figure 20 shows the interconnections between the objects and processes. For example, the Number Inputting process can accept inputs from the Number Input Object and the Memory Storage object while its outputs information to the Display Out Object and also the Memory Storage objects. We can see that the Number Inputting process inputs and outputs information to the Memory Storage object and to understand those interconnections we will have to analyze the interconnections at Level 2.



**Figure 20 Scientific Calculator: Top Level OPD**

By connecting all the objects and process at all levels we get the system level OPD as shown in Figure 21.





**Figure 21 Scientific Calculator: System Level OPD**

The OPD as shown in Figure 21 is very informative at the higher levels of decomposition and gives us a clear and general understanding of the system functionality. However for complex systems with multiple components and interconnections we need to zoom in using either an object centric or process centric OPD view to help identify the interconnections between the components and processes, and the type and direction of these interconnections. For example, the process centric view of the Number Clearing process is as shown in Figure 22. The Number Clearing process consists of two processes Data Input Clearing and Memory Clearing. Data Input Clearing is responsible for clearing the calculator display whereas the Memory Clearing clears the numbers stored in Memory.

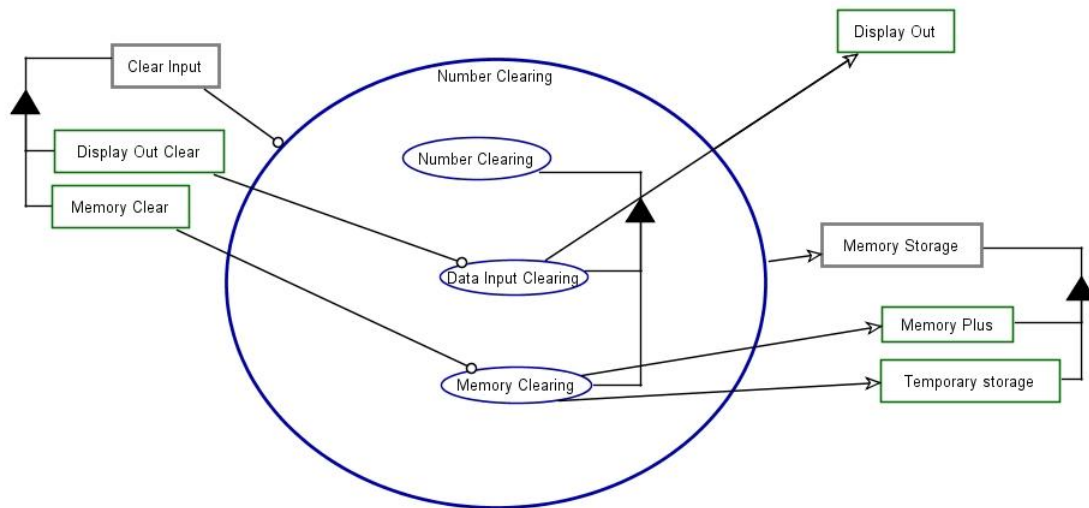


Figure 22 Scientific Calculator: Level 1 “Number Clearing” Process Centric View

### 4.1.3 Creating the Dependency Matrix

For complex systems drawing an OPD is very informative because of the tacit knowledge gained while constructing an OPD cannot be obtained from other tools. OPDs are very rich in representation however some aspects of system architecture are very difficult to analyze from the OPD alone. Therefore, we use a Dependency Matrix to transform the information from the OPD into a matrix representation. Although some information is lost when moving from the OPD to the Dependency Matrix, the ability to analyze additional information is gained. This Dependency matrix can be seen as a Design Structure Matrix (DSM) or the Network Adjacency Matrix.

The DSM is a very valuable tool to represent the interconnections in the OPM in a matrix format to perform detailed analysis, draw conclusions and suggest enhancements. DSM for the purpose of this thesis will be a matrix representation of the relationships between the objects of the system. The columns and rows of the DSM are made up of the objects of the OPD. For each column entry in the matrix the outputs to the corresponding rows are identified from the OPD and marked accordingly. The end result is a matrix which captures the main relationships between the objects and/or processes. The rows of the matrix identify the inputs to each node and the columns identify the outputs from each node. To apply network theory, the DSM is converted to a Network Adjacency Matrix by replacing the marks by 1's and the blanks by 0's.

Network analysis is widely used to analyze social networks (31-34) and to some extent in engineering design (35-38). Using Network analysis opens up the opportunity to leverage the vast amount of research to quantify and measure some of the metrics of system architecture (5).

An OPD is in essence a network. Network theory therefore can be used to identify nodes and the linkages between them. Figure 23 illustrates the similarities between the two methodologies.


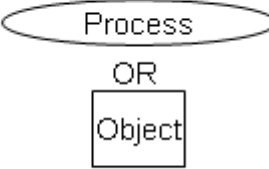
	<i>Network Graph</i>	<i>OPD</i>
<i>Components</i>		
<i>Links</i>	Directional or Non-directional	Structural or Procedural

Figure 23 Network Graphs vs. OPM

The Scientific Calculator DSM derived from the OPD in Figure 21 is as shown Figure 24. For example by analyzing the Number Clearing process centric view in Figure 22 we can see that the Display Out Clear object is clearing the Display Out object via the Data Input Clearing process, so we have a mark for the column Display Clear Out in the DSM that corresponds to the row Display Out.

	Number Input	Logical Operations	Arithmetic Operations	Display Out	Hex	Dec	Bin	Oct	Memory Recall	Memory Plus	Temporary Storage	Memory Clear	Display Out Clear	Function Input
Number Input					x	x	x	x						
Logical Operations														
Arithmetic Operations														
Display Out	x	x	x						x		x		x	
Hex														
Dec														
Bin														
Oct														
Memory Recall														
Memory Plus	x			x							x	x		
Temporary Storage	x	x	x	x					x			x		x
Memory Clear														
Display Out Clear														
Function Input														

Figure 24 Scientific Calculator: DSM

Network Adjacency Matrix derived from the DSM is binary directional in nature. The matrix is binary because no weights are assigned to the strength of the relationships between the nodes

whereas the matrix is directional because the DSM does convey the inputs and outputs for each node.

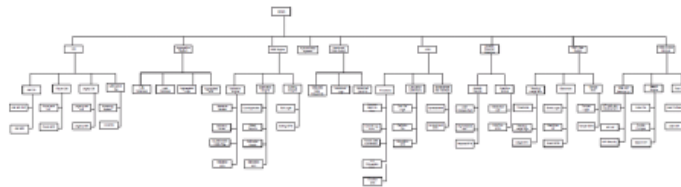
Representing the information in the Dependency Matrix format facilitates the use of DSM (2.2.2) and Network (2.2.3) analysis techniques to understand the system.

## 4.2 ANALYSIS

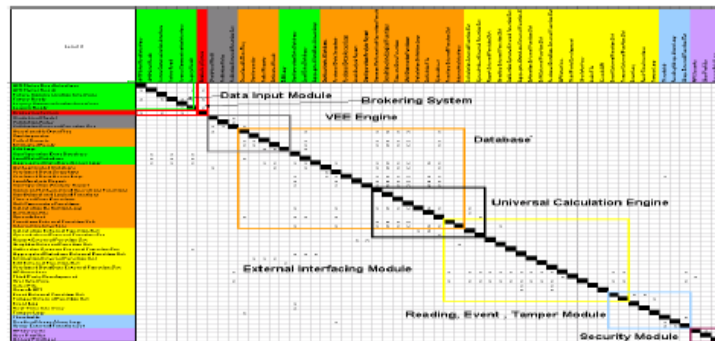
After producing the OPD representation and the Dependency Matrix representation of the system architecture in the Formulation phase, the next phase is Analysis which is done at two levels - the top level – Macro and the lower level - Micro. The following sections explain Macro-Level Analysis and Micro-Level Analysis in more detail.

### 4.2.1 Macro-Level Analysis

*Modularity from OPM*



*Dependency Matrix Analysis*



*Visibility-Dependency Signature Analysis*

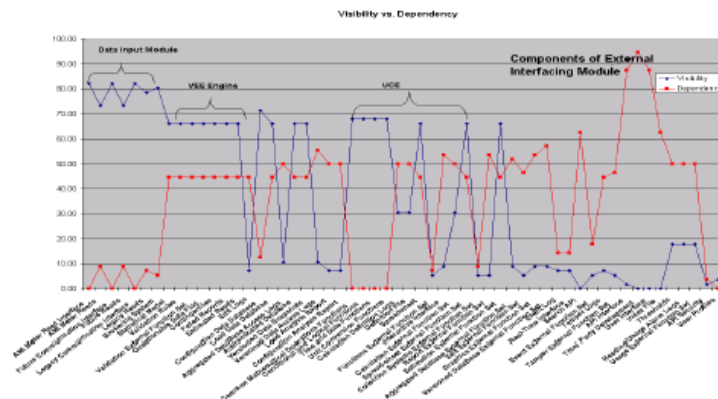
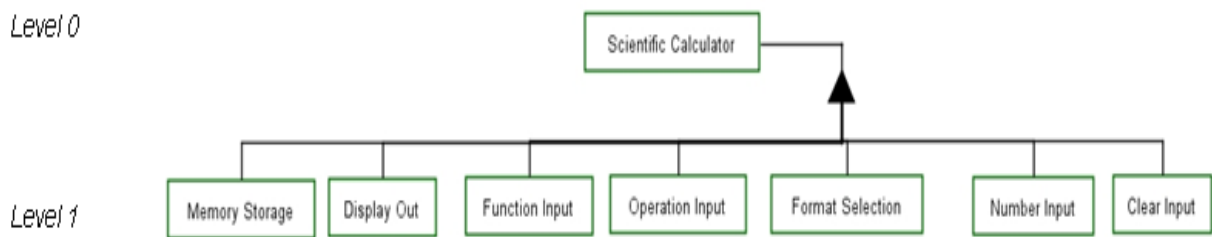


Figure 25 Macro-Level Analysis

A key characteristic of a system architecture is the extent to which it is modular or integral (39). Baldwin and Clark argue that modularity adds strategic value by creating options that enable the evolution of both designs and industries (40). Previous work has focused on identification of modules using decomposition methods of complex systems using, graphs, trees and matrices (37,38). DSM representation has also been used to analyze architectures at the product level (6,41). The Macro-Level Analysis identifies the modules and its characteristics based on product decomposition using OPDs, Dependency Matrix analysis and Network metrics.

### ***Modularity from OPM***

OPM has been applied in various fields such as real-time systems (42) and Web applications (43). The OPM decomposition assumes that the system has certain modules based on function, structure or organization. An initial modular structure decomposition of the system can be initiated at level 1. This may be a good starting point since this modularity is based on the understanding of the system when the OPD was created and hence is a high level representation of the system. This decomposition is then refined further in lower levels. The Level 1 Decomposition of the Scientific Calculator is as shown in Figure 26.



**Figure 26 Scientific Calculator: Level 1 Decomposition**

### ***Dependency Matrix Analysis***

The Dependency Matrix as explained in 4.1.3 can be derived from the OPD to help analyze the system architecture further. To sequence and cluster the Dependency Matrix we use the methodology proposed by Leicht and Newman (44).

Newman's clustering methodology groups' nodes into communities (modules) to attain the highest value of the benefit function  $Q$ , called the modularity which is defined by:

$$Q = (\text{fraction of edges within communities}) - (\text{expected fraction of such edges}).$$

Edge is a link between two nodes and nodal degree is the sum of the number of input and output connections from a node. "The expected fraction of edges is evaluated using the configuration model, a random graph conditioned on the degree sequence of the original network, in which the probability of an edge between two vertices  $i$  and  $j$  is  $k_i k_j / 2m$ , where  $k_i$  is the degree of vertex  $i$  and  $m$  is the total number of edges in the networks. The modularity can be written:

$$Q = 1/2m \sum_{ij} [A_{ij} - k_i k_j / 2m] \delta_{i,c_j}$$

Where  $A_{ij}$  is an element of the Adjacency Matrix  $\delta_{icj}$  is the Kronecker delta symbol and  $c_i$  is the label of the community to which vertex  $i$  is assigned. The algorithm then maximizes  $Q$  over possible divisions of the network into communities to derive the clusters in the network” (44).

The Newman Clustering Analysis for the Scientific Calculator DSM in Figure 24 is as shown in Figure 27.

	Number Input	Logical Operations	Arithmetic Operations	Display Out	Hex	Dec	Bin	Oct	Memory Recall	Memory Plus	Temporary Storage	Memory Clear	Display Out Clear	Function Input
Number Input	■				x	x	x	x						
Logical Operations		■												
Arithmetic Operations			■											
Display Out	x	x	x	■					x		x		x	
Hex					■									
Dec						■								
Bin							■							
Oct								■						
Memory Recall									■					
Memory Plus	x			x						■	x	x		
Temporary Storage	x	x	x	x					x		■	x		x
Memory Clear												■		
Display Out Clear													■	
Function Input														■

Figure 27 Scientific Calculator: Results of Newman’s Clustering

Once we get the results of the Newman’s clustering analysis we further manually cluster the Matrix based on our understanding of the OPM. As suggested in 2.2.2 some nodes might be part of two clusters and only way to identify this is to use the knowledge gained during the OPM analysis. The two stage clustering process utilizes all the tools at our disposal to identify the different modules in the system architecture.

The clustered DSM for the Scientific Calculator with the modules is as shown in Figure 28. The initial Level 1 Decomposition had seven modules (Figure 26) but the clustering analysis reveals the tight interconnections between the initial seven modules that result in two large modules – Data Input Module and Processing Module.

	Number Input	Hex	Dec	Bin	Oct	Logical Operations	Arithmetic Operations	Display Out	Memory Recall	Memory Plus	Temporary Storage	Memory Clear	Display Out Clear	Function Input
Number Input	x	x	x	x	x									
Hex		x												
Dec			x											
Bin				x										
Oct					x									
Logical Operations						x								
Arithmetic Operations							x							
Display Out	x					x	x	x		x			x	
Memory Recall									x					
Memory Plus	x									x	x			
Temporary Storage	x					x	x	x	x		x			x
Memory Clear												x		
Display Out Clear													x	
Function Input														x

Figure 28 Scientific Calculator: Clustered DSM

### Visibility-Dependency Signature Analysis

The OPM and the DSM modularity analysis are based on the first order (direct) interconnections between the components. Network analysis' reachability metrics of visibility and dependency (2.2.3) can be used to understand how the higher order interconnections influence those modules and how these modules in turn influence the system. This framework builds on the idea of the Visibility-Dependency graph idea proposed by Sharman and Yassine (6).

The Visibility vs. Dependency graph can be derived as follows

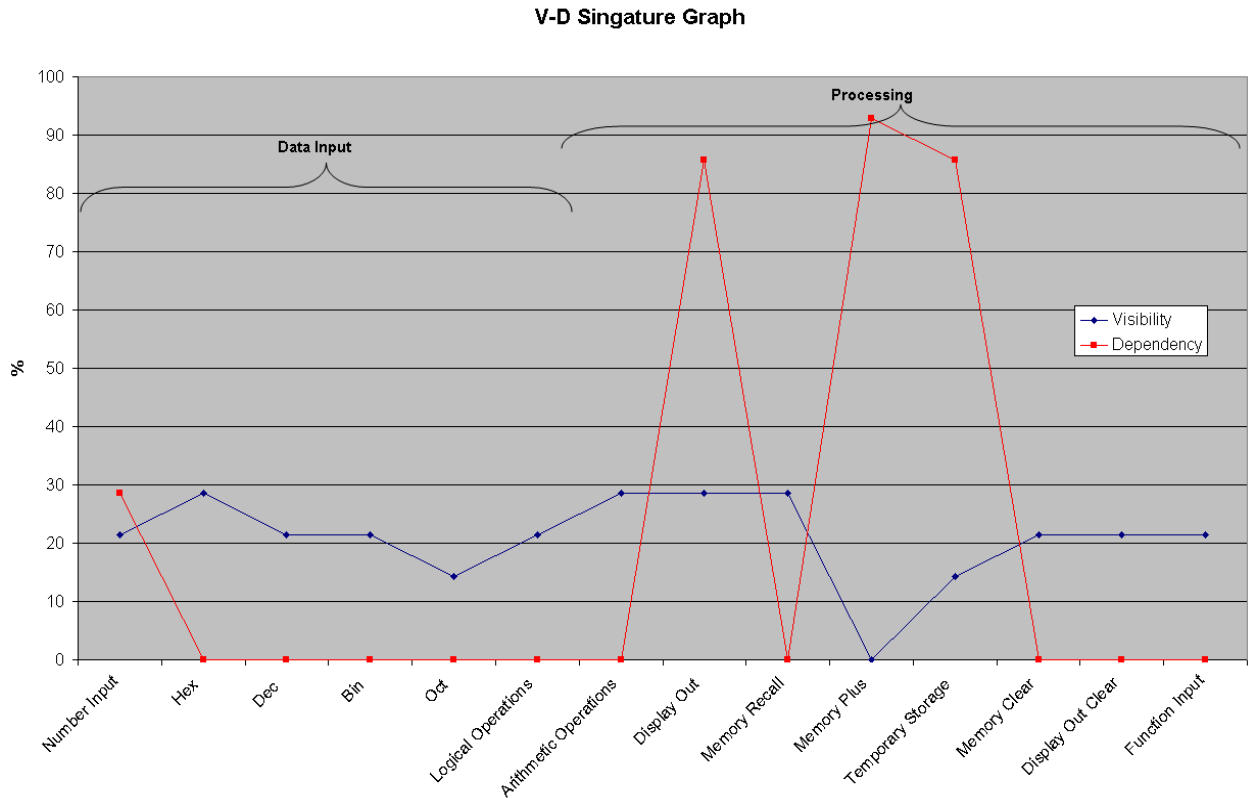
- Calculate the Visibility and Dependency (2.2.3) values for each component
- Calculate the percentage Visibility and Dependency:

$$\% \text{ Visibility for a node} = (\text{Visibility for node} \times 100) / (n-1)$$

$$\% \text{ Dependency for a node} = (\text{Dependency for node} \times 100) / (n-1)$$

Where n is the number of nodes in the graph.

- Plot the Visibility and Dependency values for each component – X axis is the component name



**Figure 29 Scientific Calculator: Visibility vs. Dependency Graph**

The Visibility-Dependency Graph for the Scientific Calculator is as shown in Figure 29. The Data Input Module has very low dependency on the processing module since it receives inputs from the external system (keyboard). Overall the system has low visibility which implies that the change in component level design will not have significant system wide impact.

#### 4.2.2 Micro-Level Analysis

Micro-level connectivity analysis of the architecture can be used to quantify component level modularity of complex products (5), to design a robust system, to identify product evolution opportunities and to recognize areas of possible collaborations with other firms.

Network theory argues that the behavior of a particular node is dependant largely on how that node is connected to the other nodes in the system (34,45). Micro-Level analysis looks at the connectivity of an individual node and identifies the central node in the system based on connectivity. (31,32,34).

Components in a system are not only directly connected but also indirectly connected to other components because dependencies can propagate through intermediary components and reach other distant components. Hence we analyze components based on the number of direct connection between the components and also on the number of second order connections and the number of occurrences on the path of two other components (31,34).

The metrics (2.2.3) that quantify the above mentioned measures are as follows:



- Indegree Centrality and Outdegree Centrality that measure the number of direct connections between the components
- Visibility and Dependency that captures the second order connections between the components



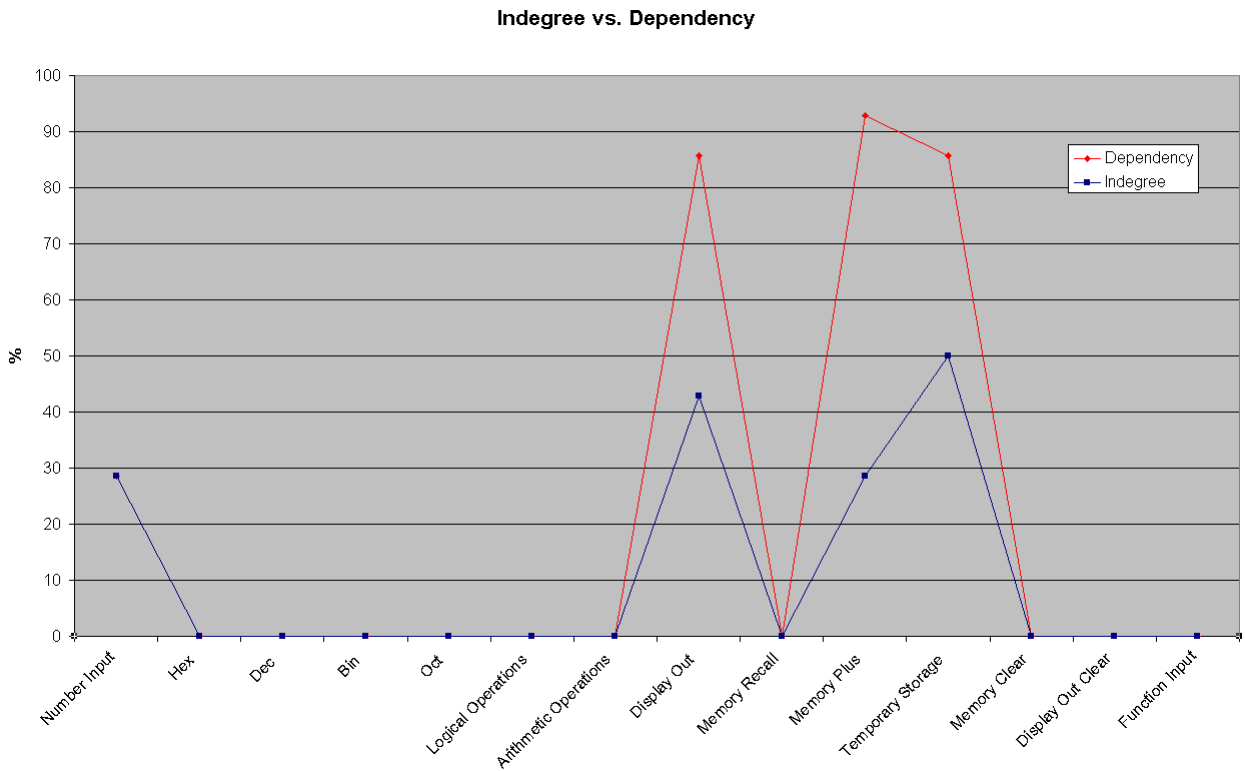
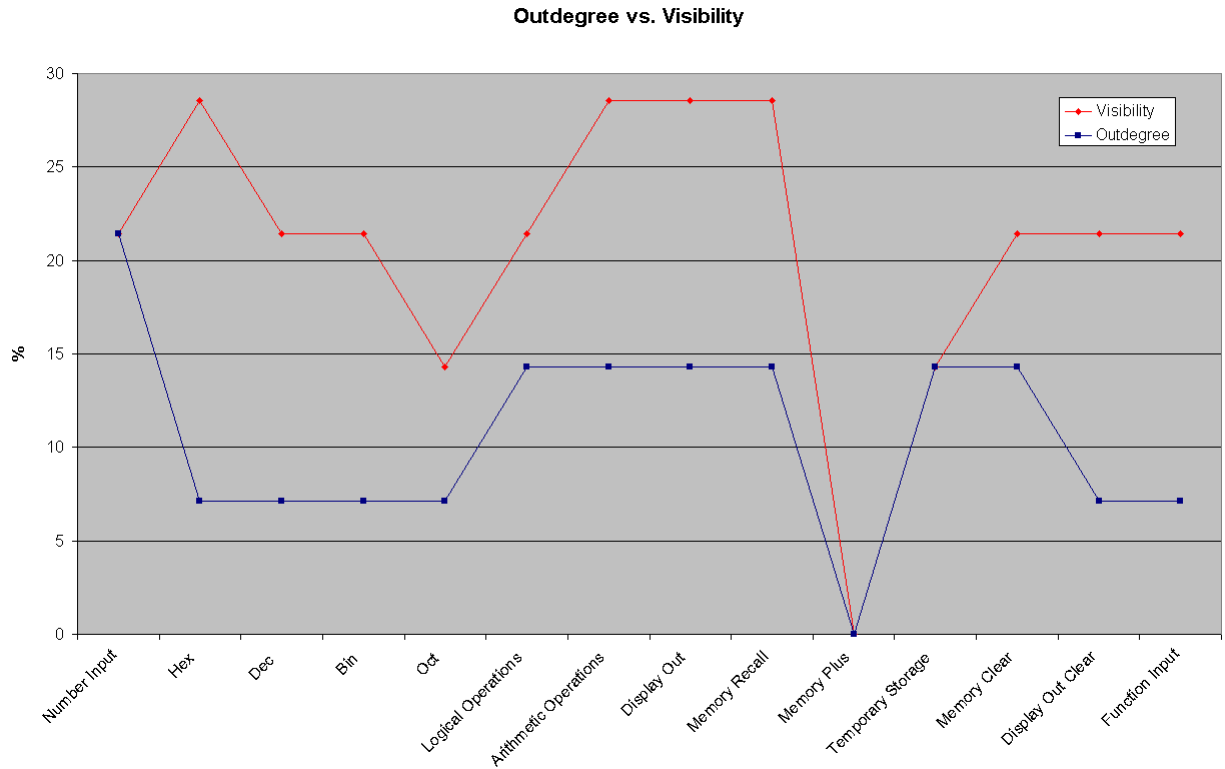
Figure 30 Micro-Level Analysis

The Indegree and Outdegree metrics for the Scientific Calculator are as shown in Figure 31. The central components are the Number input, Display out and Temporary Storage. Number Input has high Indegree and Outdegree Centrality and hence is identified as a central node. Display Out and Temporary Storage have high Indegree Centrality and hence are identified as central nodes.

	Number Input	Logical Operations	Arithmetic Operations	Display Out	Hex	Dec	Bin	Oct	Memory Recall	Memory Plus	Temporary Storage	Memory Clear	Display Out Clear	Function Input		Indegree Centrality	Indegree Centrality %
Number Input	0	0	0	0	1	1	1	1	0	0	0	0	0	0		4	29
Logical Operations	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
Arithmetic Operations	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
Display Out	1	1	1	0	0	0	0	0	1	0	1	0	1	0		6	43
Hex	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
Dec	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
Bin	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
Oct	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
Memory Recall	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
Memory Plus	1	0	0	1	0	0	0	0	0	0	1	1	0	0		4	29
Temporary Storage	1	1	1	1	0	0	0	0	1	0	0	1	0	1		7	50
Memory Clear	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
Display Out Clear	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
Function Input	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
Outdegree Centrality	3	2	2	2	1	1	1	1	2	0	2	2	1	1			
Outdegree Centrality %	21	14	14	14	7	7	7	7	14	0	14	14	7	7			

**Figure 31 Scientific Calculator: Indegree and Outdegree Centrality**

The second order connections of Visibility and Dependency are as shown in Figure 32. Visibility and Outdegree Centrality are drawn on the same graph to identify central components we might have missed in the Outdegree Centrality Analysis. Similarly Dependency and Indegree Centrality are drawn together to identify central components we might have missed in the Indegree Centrality Analysis. Since the Scientific Calculator is a small system, the Indegree and Outdegree Analysis are enough to identify the central nodes. The Visibility and Dependency metrics do not identify any new central components. We will see that when we analyze a more complex system (Chapter 5.) the Visibility and Dependency metrics become more valuable.



**Figure 32 Scientific Calculator: Higher Order Interconnections**

Identifying nodes/components with low component modularity is important for making redesign decisions. Components with high Indegree centrality and high Dependency are more likely to experience higher levels of unplanned redesign due to the impact from changes in other components whereas components with high Outdegree centrality and high Visibility levels are more likely to experience higher levels of planned redesign (5).

Micro-Level Analysis will reveal single point of failures. Once the central nodes are identified based on the connectivity metrics engineers can conduct failure scenario analysis to understand the robustness of the product architecture and if necessary add redundancy at the component level to protect against failures.

Design evolution decisions can be made by taking into consideration the connectivity of the nodes. For example by interfacing a new component to a particular node with high Outdegree centrality and high Visibility the design change can be propagated through the entire system. On other hand if the requirement is to minimize the effects of a new component to a certain module interfacing the component then a component with low connectivity measures might make the most sense.

Firms with strategic partnerships have product architectures that work very closely with each other. Identifying central nodes in the product architecture will facilitate efficient and seamless merging of the two product architectures.

## CHAPTER 5. MDMS CASE STUDIES

In this chapter the analysis framework proposed in Chapter 4. is applied to two industry leading Meter Data Management Systems. The analysis identifies the macro-level and micro-level characteristics of both products which can be used to compare their architectures or to enhance the functionality of the respective architectures. From here onwards P1 refers to Product 1 and P2 refers to Product 2.

Since both the products offer the same functionality the first Formulation step, “Identifying the System Boundary” is identical for both products and is covered in one section. For sake of consistency “Identifying the System Boundary” section of P2 will refer back to P1. The remaining framework steps are broken down for each product separately, starting first with P1 and then performing the same analysis on P2.

### 5.1 PRODUCT 1: FORMULATION

#### 5.1.1 Product 1: Identifying the System Boundary

The MDMS is a part of the Smart Grid Information System as explained in 3.2.2. The ecosystem consists of AMI, MDMS and Business Applications/Operations. The MDMS is the central meter data repository in the Smart Grid and hence its primary function is to store the meter data and to facilitate the leveraging of that data.

Identifying the flow of data is the first step in identifying the system boundaries in an information technology environment. There is bidirectional flow of data through the Smart Grid Information System and the MDMS sits right at the center of this data flow. It receives data from both the AMI and the Business Applications/Operations Systems and it also outputs data to both the AMI and the Business Application/Operation Systems.

The next step is to classify the nature of the information flowing from the MDMS to the AMI and the Business Application/Operations systems. This will clearly identify the functionality of each system in the environment with respect to the MDMS and thus define the MDMS system boundary. The information flowing in and out of the MDMS can be classified as follows

- Data flowing into MDMS
  - AMI system is responsible for collecting all the meter reads from different Meter collection systems and feeding it into the MDMS. The MDMS processes and stores the information
  - Business Applications/Operations feed processed information and reports into the MDMS. Other business units can access these stored data and report from the MDMS thus minimizing rework.
- Data flowing out of the MDMS

- MDMS provides data to the Business Applications/Operations systems. The data can be pushed out to the Business Applications/Operations systems or the Business Units can request it as and when required.
- MDMS has the ability to request data from the AMI system based on Business Applications/Operations systems needs.

The predominant flow of information through the system is asynchronous but some flow such as information regarding power outages can be synchronous. Based on the above information the System boundary can be defined as shown in Figure 33.

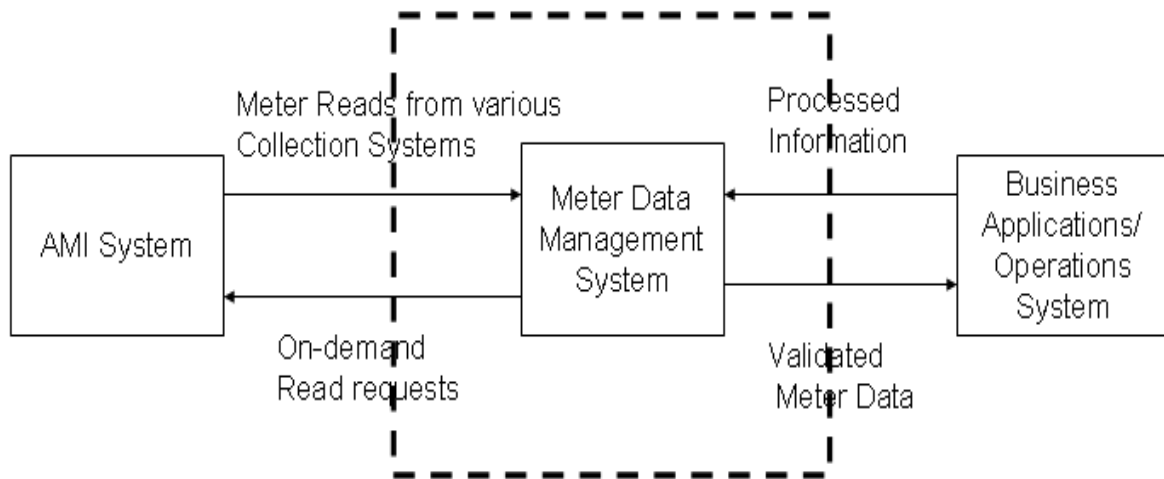


Figure 33 Identifying MDMS System Boundary

### 5.1.2 Product 1: Identifying the Object-Process System Levels

Here we decompose the MDMS into modules based on functionality. Identifying the Object-Process System Levels involves breaking down the system into levels and then identifying the interconnections as proposed in section 4.1.2.

Identifying the Object-Process System Levels consists of the following three steps

- Decompose based on objects
- Decompose based on processes
- Combine the object and process decompositions to establish the cross-functional linkages thus generating the complete multi-level system decomposition

#### **Object Decomposition**

The primary function of the MDMS is to accept information, store information, process information, generate reports and convey information. The objects that satisfy these needs are explained below. The Object decomposition is as shown in Figure 34.

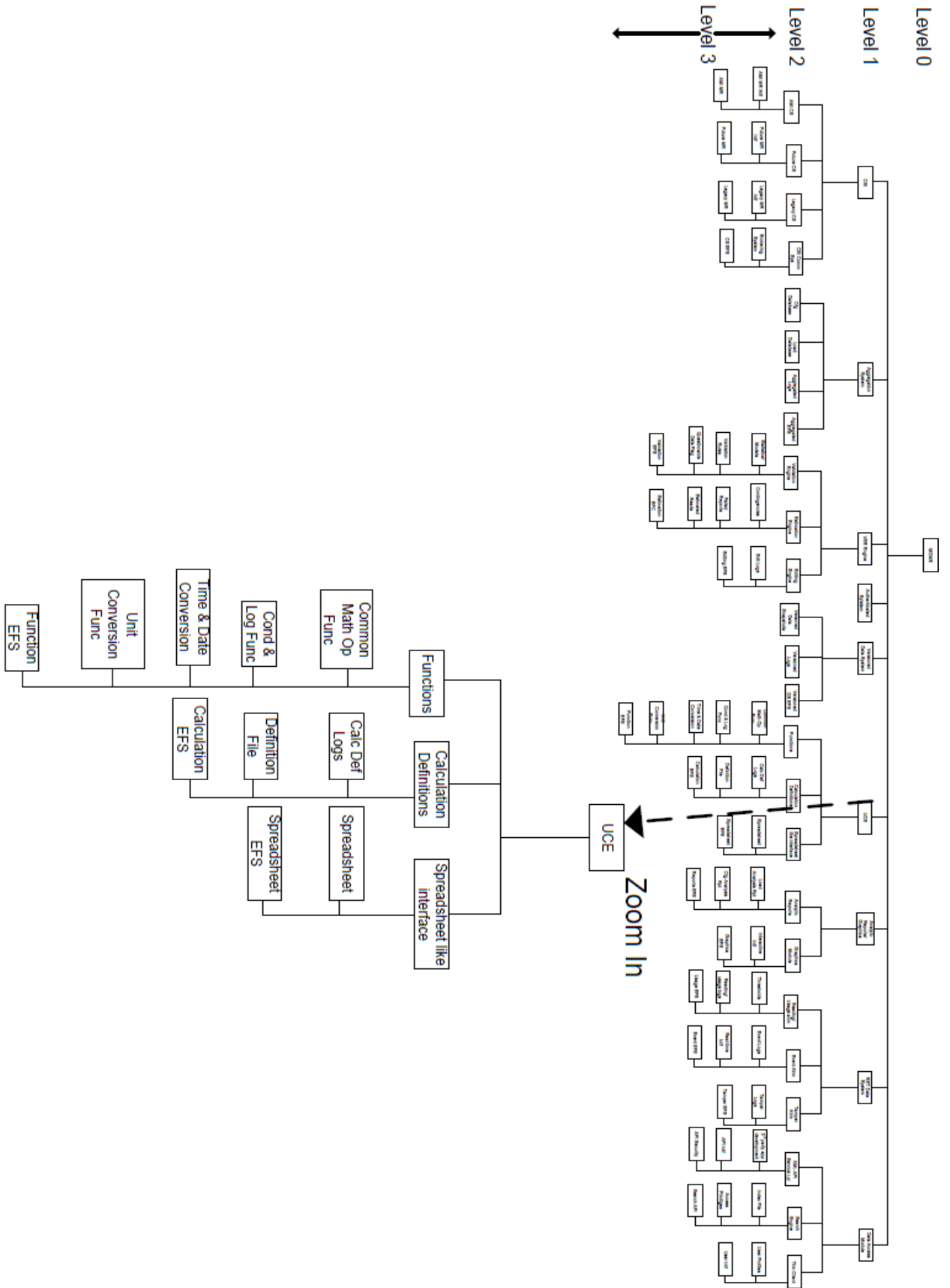


Figure 34 Product 1: Object Decomposition Levels (UCE Object Zoom In)

### Collection Systems Integration (CSI)

Collection System Integration is responsible for:

- Gathering data from various meter-reading systems and forwarding the data to the storage system.
- Requesting Meter reads from the respective external Collection Systems

It is made up of the Legacy Collection System, AMI collection System, Future Collection System and the CSI Communication System. All the individual collection systems have adapters that interface with the respective external collection systems. The CSI communication System module is responsible for brokering read requests between the MDMS and the external Collection Systems

### Aggregation System

Aggregation System is one of the many storage modules identified based on functionality. The reads collected from the CSI are stored in the Aggregation system. Meter reads can be broadly classified into two types- configuration reads and consumption (load) reads. The configuration reads contain the meter metadata information such as location, time zone, and contract id. The consumption reads consist of the data required to calculate the amount of electricity consumed by the meter user. Based on this understanding we decompose the Aggregation System into Configuration Database and Load Database. Any activity (read or write) to the two databases have to be logged for record-keeping purposes hence we identify two additional modules: the Aggregation Database Access Logs to track the activity and Aggregation Database external function set to facilitate external access of the Aggregation System Database.

### Validation, Estimation and Editing Engine (VEE)

The Validation, Editing and Estimation Engine is responsible for scrubbing the consumption data for future use. The consumption data in the Load Database of the Aggregation System might have incorrect or lost information and hence the data has to be cleansed before it can be used for analysis. The VEE Engine validates the data using the validating engine. The Validation Engine validates data based on a set of business rules and statistical models. If the data cannot be validated, the VEE engine marks it for estimation or editing. Data marked for estimation is calculated using the Estimation Engine. The Estimation engine calculates data based on pre-set business contingencies. If the estimation engine is unable to generate an estimate it generates a failure report. Data marked for editing is edited by an authorized user and the logs of each change are maintained to comply with legal requirements. Data cleared by the VEE engine is ready to be used by all other business applications.

### Authenticated Data System

Consumption Data that clears the VEE engine is stored in the Authenticated Database system. This data stored in this module is ready to be leveraged by all other internal and external systems.



### Versioned Data System

The Versioned Data System maintains snapshot of each meter read associated with a time reference. Each Business application/Operation unit might use a different snapshot of the data based on its need. MDMS Versioned Data System keeps multiple versions of the data stating when and who changed the data. It is important to reproduce versions of data from a particular date and time for auditing purposes and also for maintaining data integrity across the multiple business units

### Universal Calculation Engine (UCE)

Universal Calculation Engine provides a platform to run complex mathematical operations on the dataset without worrying about unit conversion and time zone issues. It also eliminates the need for each business application to copy the dataset back and forth to perform calculations thereby reducing a lot of unnecessary traffic across the systems and maintaining data integrity. The UCE enables the business applications to calculate any complex load as well helps in VEE processing.

### Analytic Reports/Graphics

The MDMS generates a lot of reports and graphs. Many of the reports are status reports that inform the system administrator on a daily basis on the loading of the database, failed validations and failed estimations. The external Business Application/Operations such as billing will also generate reports. Reports can also be user generated in response to a particular query to the underlying database.

### Reading, Event and Tamper Data System

The Reading, Event and Tamper Data System consists of three modules the Reading Module, the Event Module and the Tamper Module. The Reading Module performs a trend analysis on the consumption data and raises a flag if the data exceeds certain pre-set thresholds. The Event Module has a real-time interface to inform the appropriate business applications/ operations in case of power outages or transformer overload conditions. The Tamper module identifies potentially cases of theft and sets off the tamper alarm.

### Data Access Module

The Data Access Module facilitates the leveraging of the data by external applications. It provides a XML API service layer for third party development. It also provides Thin Client capability and search engine capability for external users of the system. It is also responsible for granting access rights to each of the external applications/users.

### **Process Decomposition**

The process decomposition (Figure 35) of MDMS identifies the sub processes that perform each of the individual tasks identified in the object decomposition - accept information, store information, process information, generate reports and convey information. Each individual process is explained in brief below:

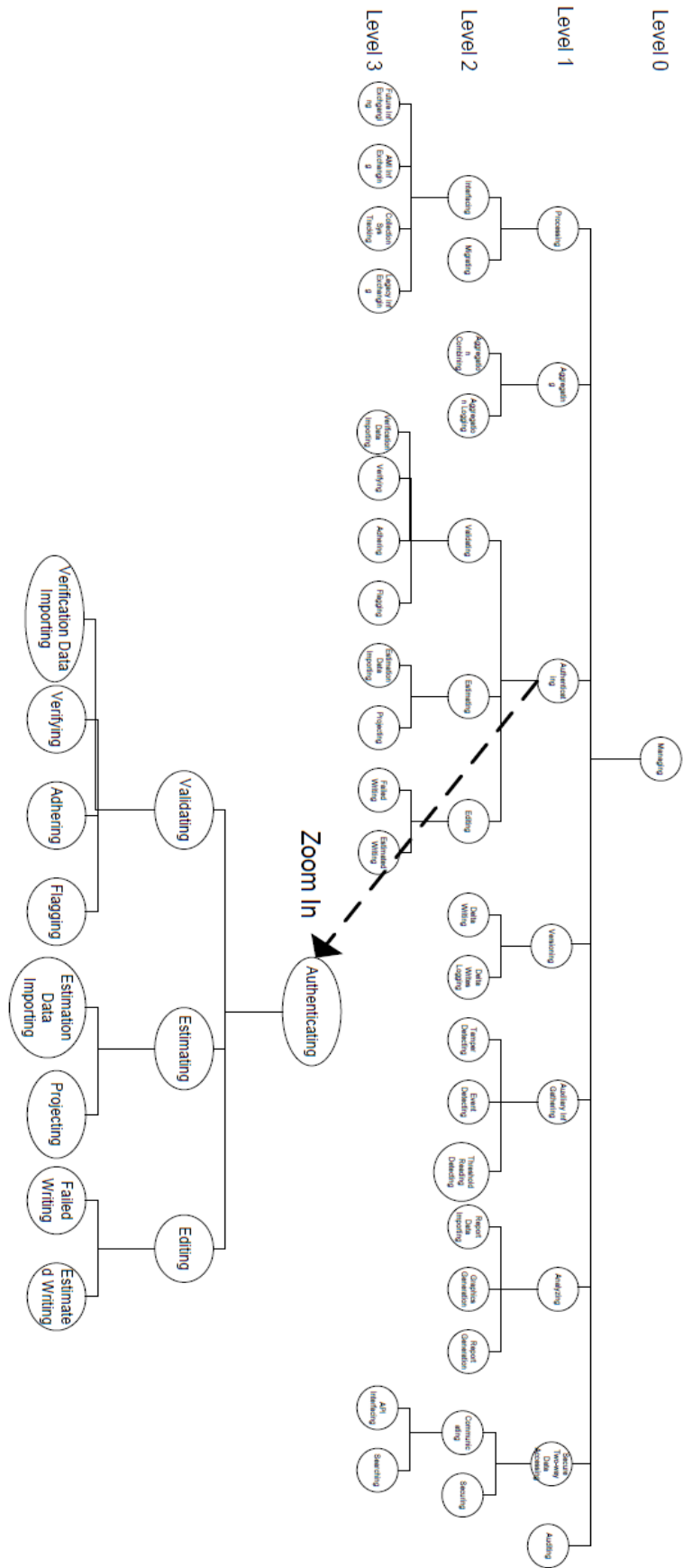


Figure 35 Product 1: Process Decomposition Levels (Authenticating Process Zoom In)

### Processing

This process is responsible for interfacing with all the external collection systems and it also facilitates the migration from a legacy meter read system to an AMI system.

### Aggregating

Aggregating process aggregates data from all the different collection systems and feeds it to the aggregated database. It also keeps a log of all the data loading activities in the aggregated database access logs.

### Authenticating

Authenticating runs the aggregated load data through the VEE engine and generates the authenticated database. It uses the UCE to perform the estimation and validation calculations.

### Versioning

Versioning is responsible for making changes to the database if and when required and for tracking those changes.

### Auxiliary information Gathering

Auxiliary information gathering process keeps tracks of the alarm detecting events and reports it to the Reading, Event and Tamper data system.

### Auditing

Auditing process is responsible for gathering data required for any sort of external or internal investigation.

### Analyzing

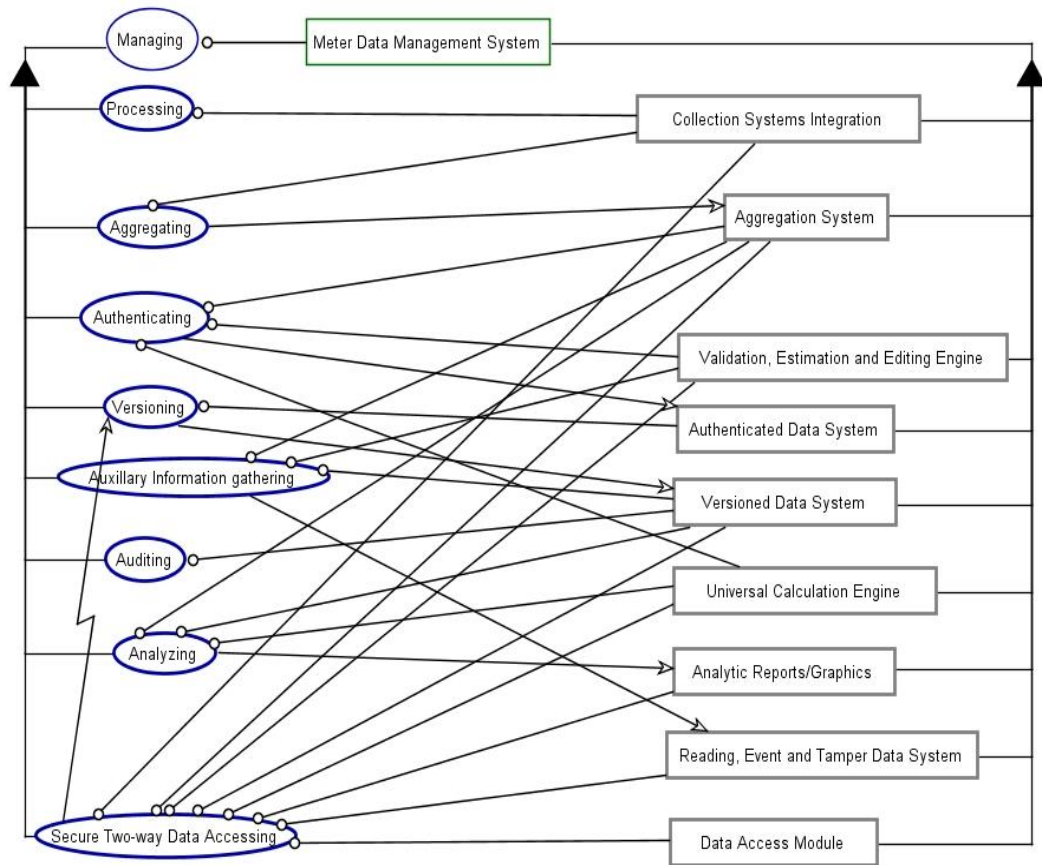
Analyzing process is tasked with generating reports and graphics. It requires access to the database and the UCE engine to produce these reports and graphs.

### Secure Two-way Data Accessing

As the name suggests this process is responsible for security and two way communication between the MDMS and the external applications. Security involves keeping track of user profiles, access profiles and API security.

### ***Combining the object and process decompositions and identifying the interconnections***

Combining the object and process decompositions starts at the top level. The input and output objects for each top level process are identified as shown in Figure 36



**Figure 36 Product 1: Top Level OPD**

The top level connections determine the linkages between the objects and processes at the levels below. This process is iteratively repeated for each level down that yields the complete system object-process diagram as shown in Figure 37. In Figure 37 the black lines represent aggregation-participation links, the green lines represent instrument links and the red lines represent result/consumption links (2.2.1).

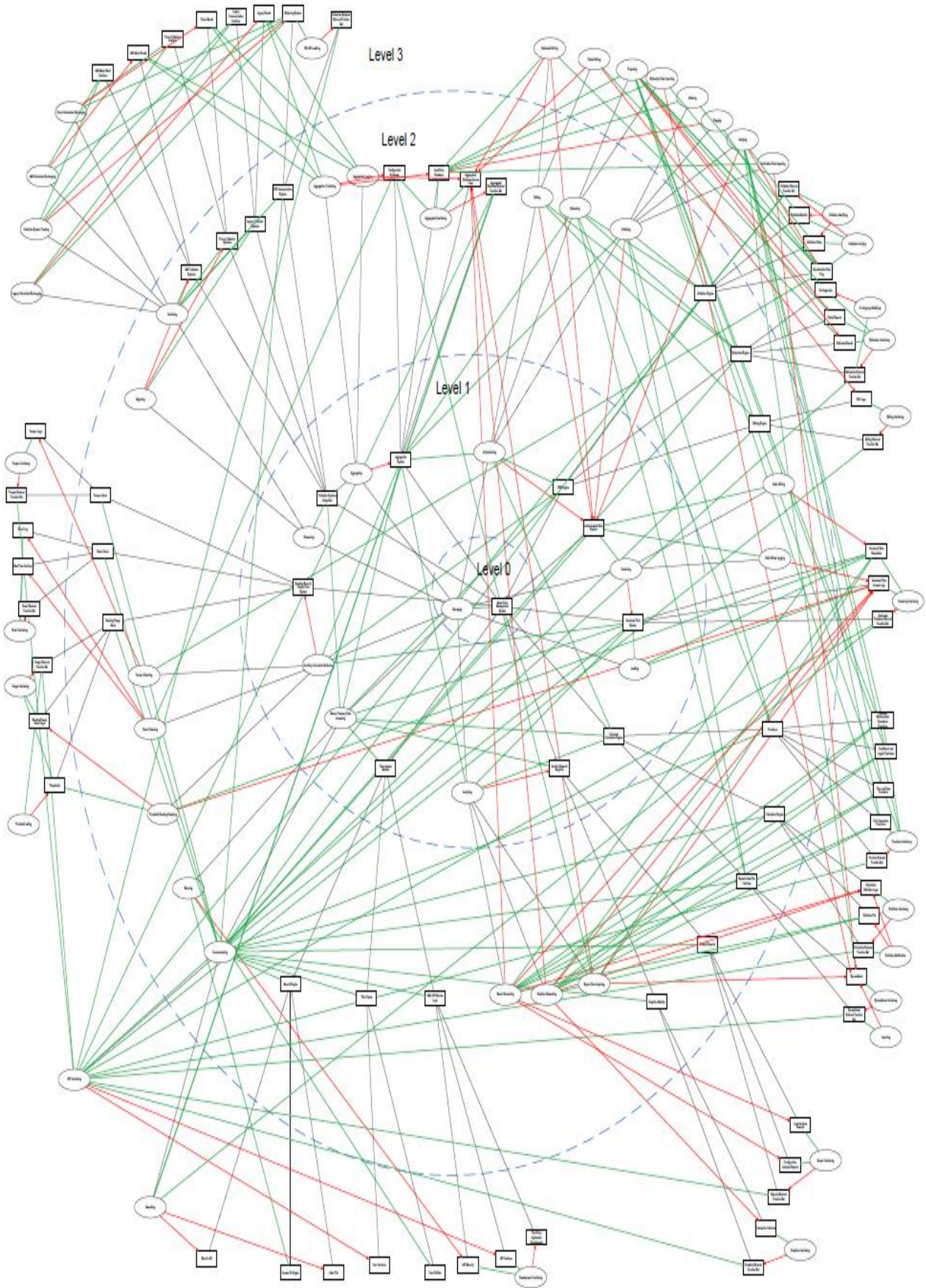
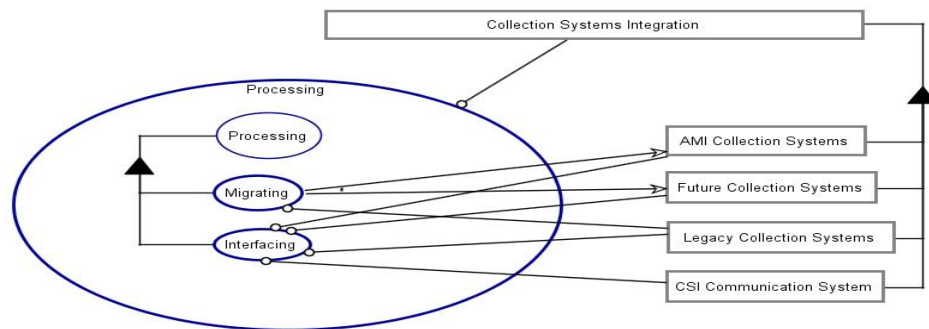


Figure 37 Product 1: System Level OPD

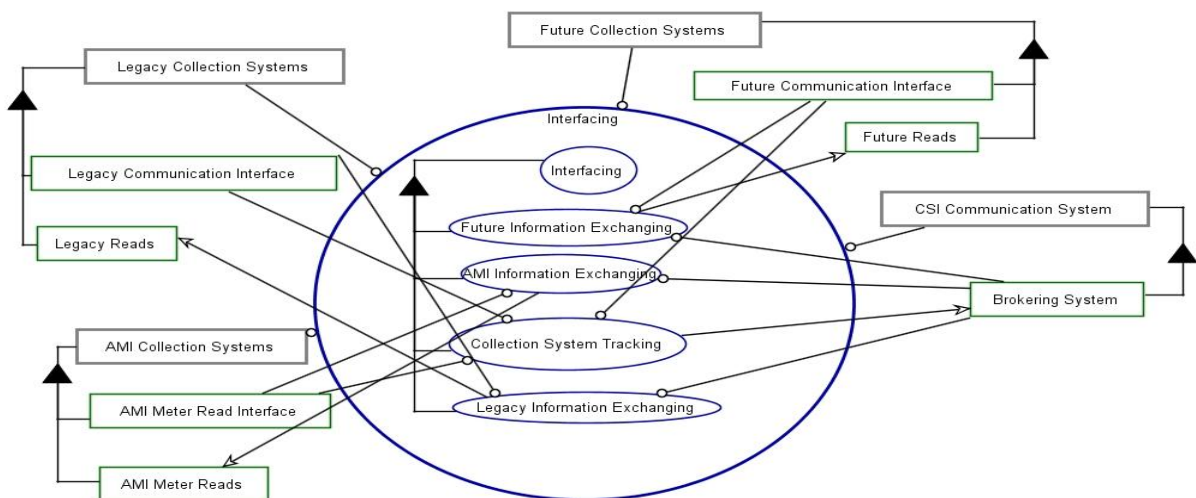
The OPD in Figure 37 is extremely complex when looked as a whole so we zoom into certain areas of the OPD to identify the relationships between the objects for further analysis. We will use a process centric view of the OPD to identify the interconnections and the flow of information in the system. The following discussion focuses on some of the important processes in the P1 MDMS architecture.

*Interconnections in the “Processing” centric OPD*



**Figure 38 Product 1: Level 1 Processing Centric View**

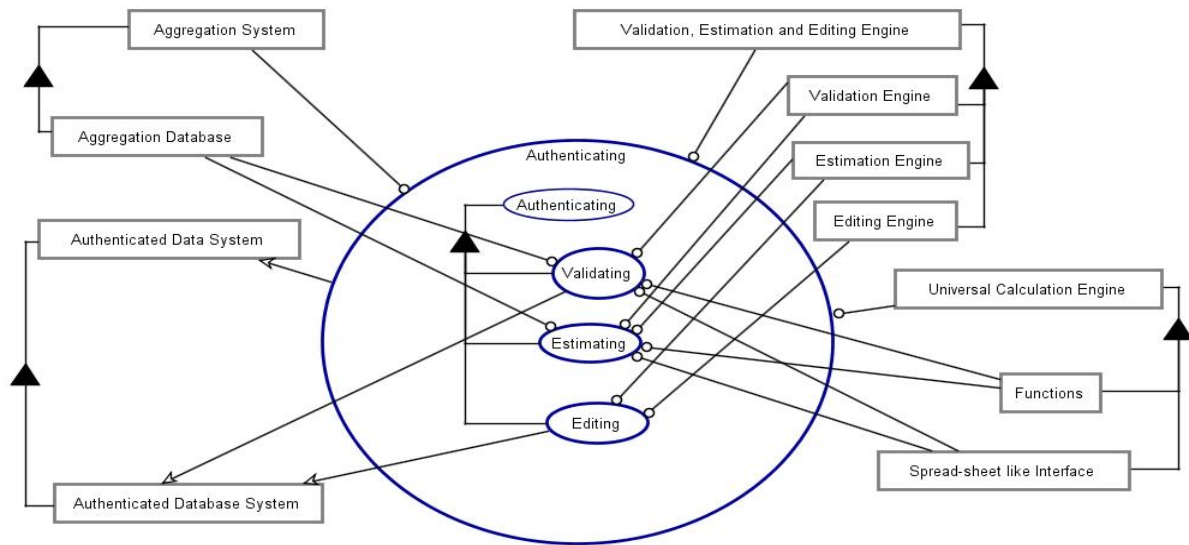
The Processing process as described in the previous section consists of two processes – the Migrating process and the Interfacing processes. The Migrating process is responsible for migrating Legacy Collection Systems to Future or AMI collection System as shown in Figure 38.



**Figure 39 Product 1: Level 2 Interfacing Centric View**

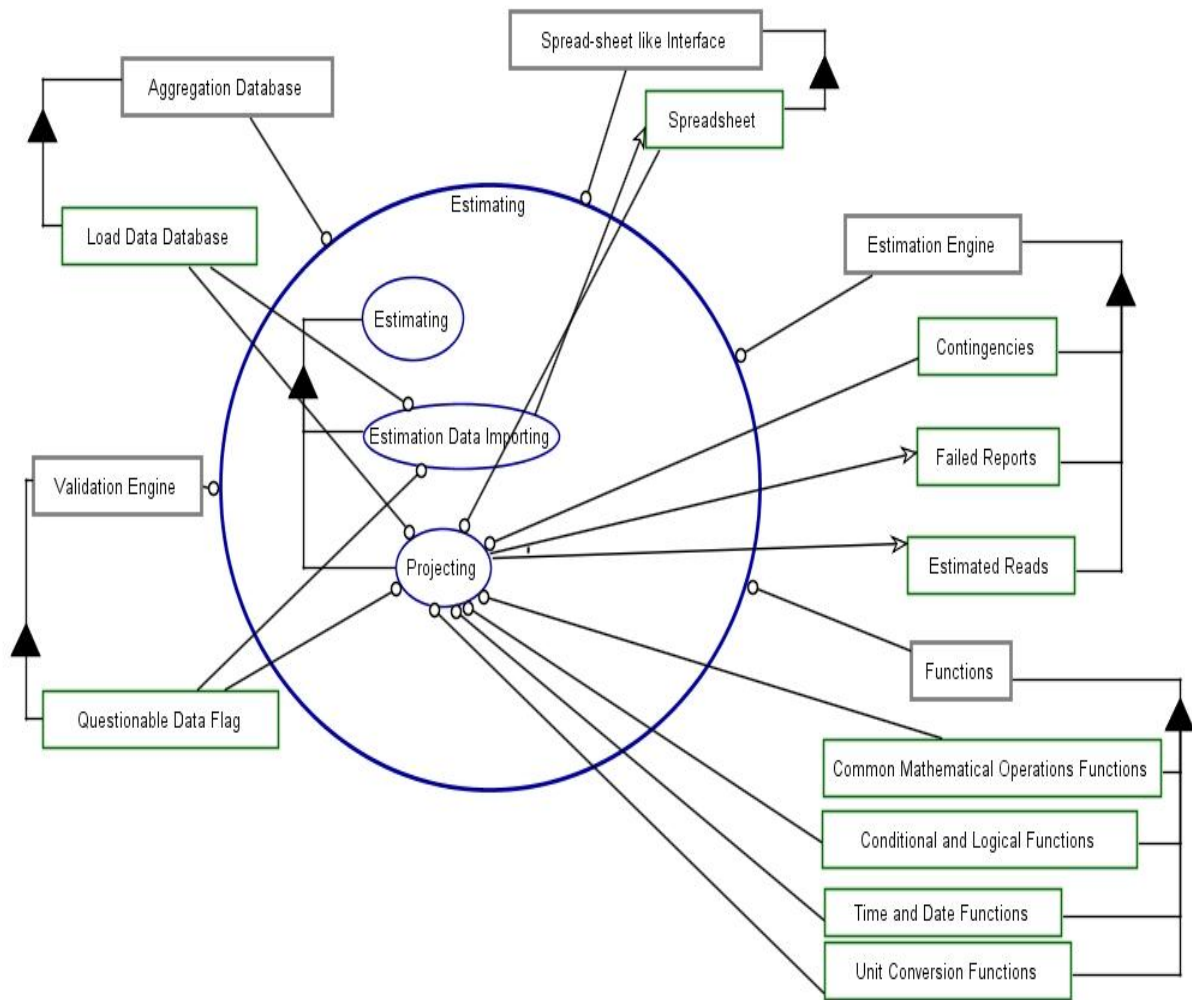
The Interfacing process handles the handoff of data from the various collection systems to the P1 MDMS and is shown in detail in Figure 39. The Interfacing process is decomposed into four processes; all the information exchanging processes (AMI, Future and Legacy) require the respective interfacing modules to yield the meter reads to be fed into the MDMS database whereas the collection system tracking process is responsible for recording the interface information of the individual collection systems (Future, AMI and Legacy) and feeding it into the brokering system to enable two-way communication between the P1 MDMS and the external collection systems.

*Interconnections in the “Authenticating” centric OPD*



**Figure 40 Product 1: Level 1 Authenticating Centric View**

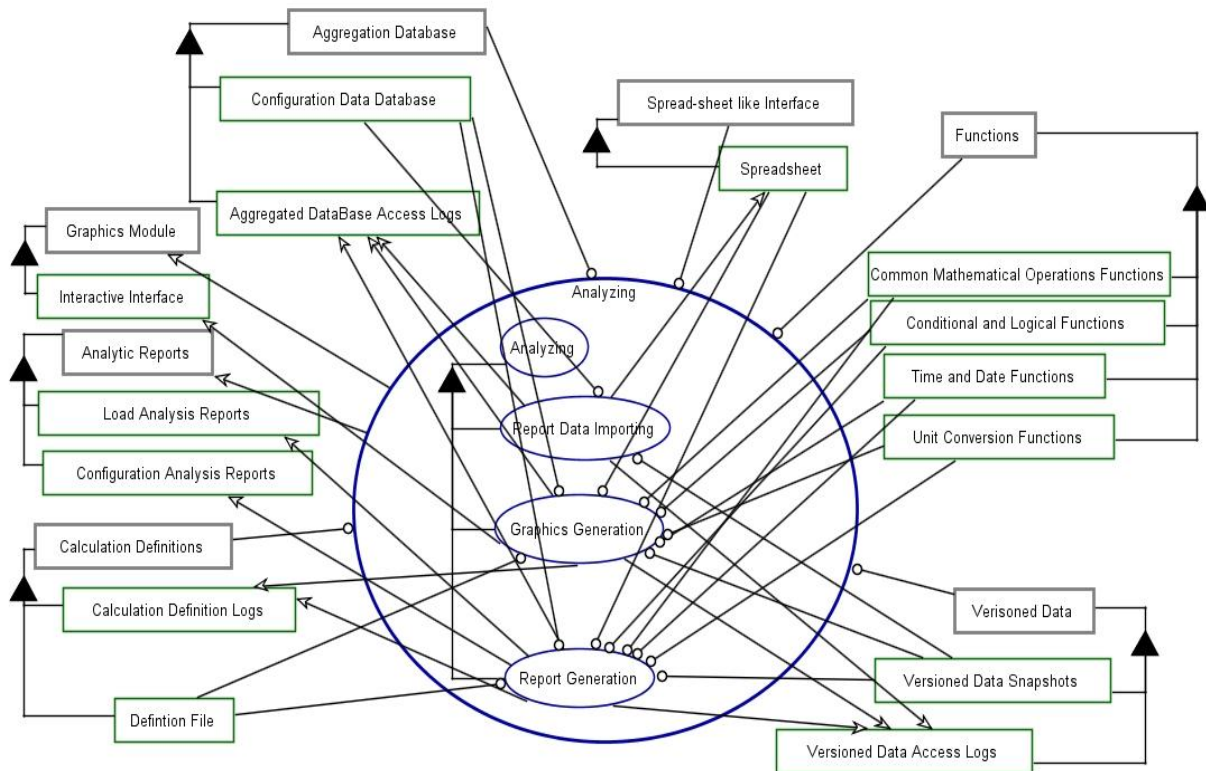
Authenticating is responsible for cleansing the data before it can be used for other decision making processes. It consists of Validating, Estimating and Editing process and it interconnects the UCE, the VEE Engine, the Authenticated Data System and the Aggregation System (Figure 40). The Aggregation System has the data that needs to be authenticated. UCE and VEE Engine are the systems have the tools to perform the authenticating process and the end result is stored in the authenticated Data System.



**Figure 41 Product 1: Level 2 Estimating Centric View**

Figure 41 zooms into the estimating process and shows the linkages at the lower level. The Estimation process consists of the Estimation Data Importing process and the Projecting process. The Estimation Data Importing process imports the data marked as questionable by the Validation Engine from the Load Database. This data is stored in a spreadsheet for estimating purposes. The Projecting process takes that data from the Spreadsheet and based on contingencies specified in the Estimation engine estimates the data yielding the Estimated reads if the estimation is successful or generating a failed report in case the estimation fails. The Projecting process uses the UCE Functions components as and when required for estimation purposes.



*Interconnections in the “Analyzing” Centric OPD*

**Figure 42 Product 1: Level 1 Analyzing Centric View**

Analyzing process consists of the Report Data Importing process, Graphics Generation process and Report Generation process. It leverages multiple objects in the MDMS to generate reports and graphs. The Report Data Importing module is responsible for collecting data from the Versioned Data Snapshot and Configuration Database and exporting them to the spreadsheet object. All data accessed by the Report Data Importing process is logged for future reference. The Graphics Generation and Report Generation processes use the data from the Spreadsheet to generate the reports/graphs. The Report Generation and Graphics Generation processes leverage the UCE components for complex calculations. All the calculations are logged in a Calculation Definition log object for reuse or reference purposes. The Graphical reports can be viewed using an Interactive interface; all reports are stored in the Analytics Module.

### 5.1.3 Product 1: Creating the Dependency Matrix

Transforming the P1 system level OPD (Figure 37) into a DSM enables analysis of the architecture. Using the techniques identified in section 4.1.3 the DSM is filled with marks based on inputs and output relationships between the components. The resultant DSM is shown in Figure 43. Comparing the two pictures (Figure 37 vs. Figure 43) we can easily see why the DSM is easier to analyze. But as stated in section 4.1.3 moving from the OPD to the DSM we lose specific relationship information. The DSM shows us all the inputs and outputs for a particular component but we cannot identify which inputs or outputs are active when delivering a specific



## 5.2 PRODUCT 1 ANALYSIS

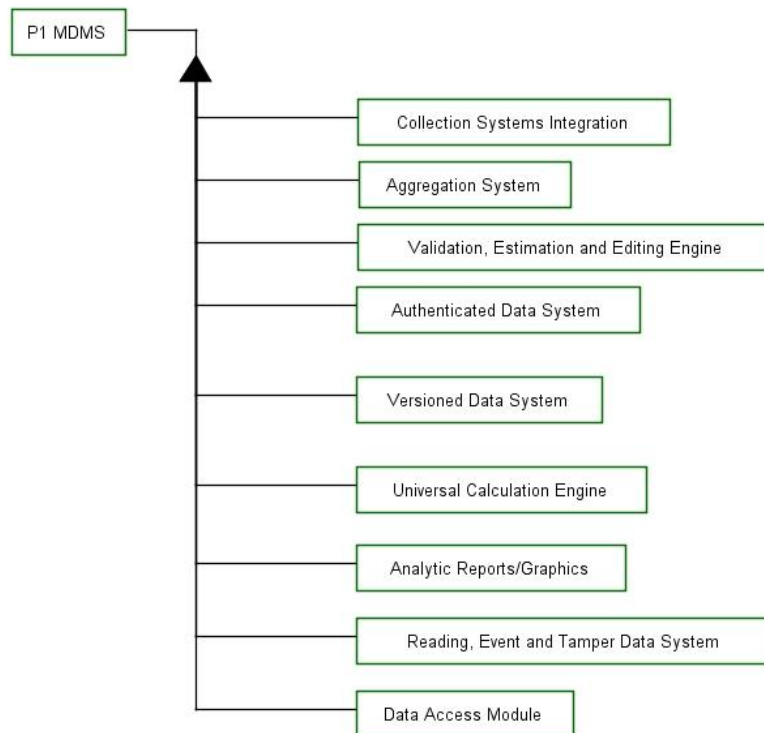
We now have both representations OPD and Dependency Matrix that we will use to analyze the architecture. The following sections leverage each of the representations as described in section 4.2 to come up with a system level understanding of the architecture.

### 5.2.1 Product 1: Macro-Level Analysis

First we take a look at the macro-level characteristics of the architecture.

#### *Modularity from OPM*

The Level 1 decomposition of the system is based on the initial understanding of the architecture, without any knowledge of the interconnections at the lower levels. The P1 MDMS based on Level 1 decomposition consists of the all the objects discussed in section 0. The decomposition is as shown in Figure 44. This is an initial Modularity from the OPM which consists of the following modules – Collection System Integration, Aggregation System, VEE Engine, Authenticated Data System, Versioned Data System, Universal Calculation Engine, Analytic/Reports Graphics, Reading Event Tamper System and the Data Access Module.



**Figure 44 Product 1: Level 1 Decomposition**

#### *Dependency Matrix Analysis*

Dependency matrix analysis refines the understanding of the modularity of the system derived from the OPM. Applying Newman's clustering algorithm (4.2.1) to the Dependency Matrix we

get a new understanding of the modules in the system. Using this knowledge combined with the knowledge gained from the OPM we can suggest a refined grouping of the components for the system based on DSM clustering as explained in the framework section (4.2.1).

Newman’s analysis reveals the clusters as shown in Figure 45; components with the same color fall in the same cluster.

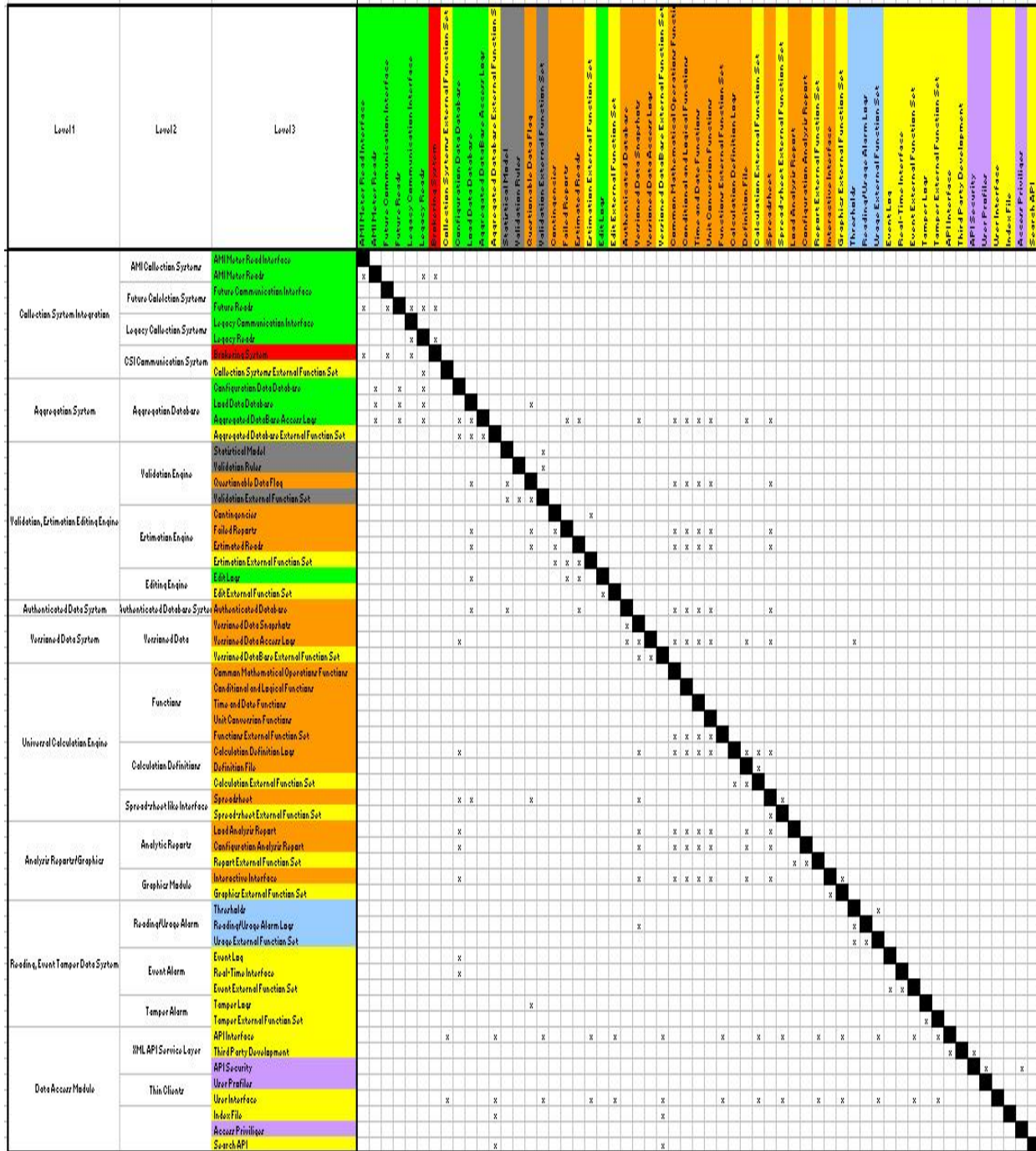


Figure 45 Product 1: Results of Newman’s Clustering Analysis

The components with identical colors are grouped together and the resultant clustered DSM is as shown in Figure 46.

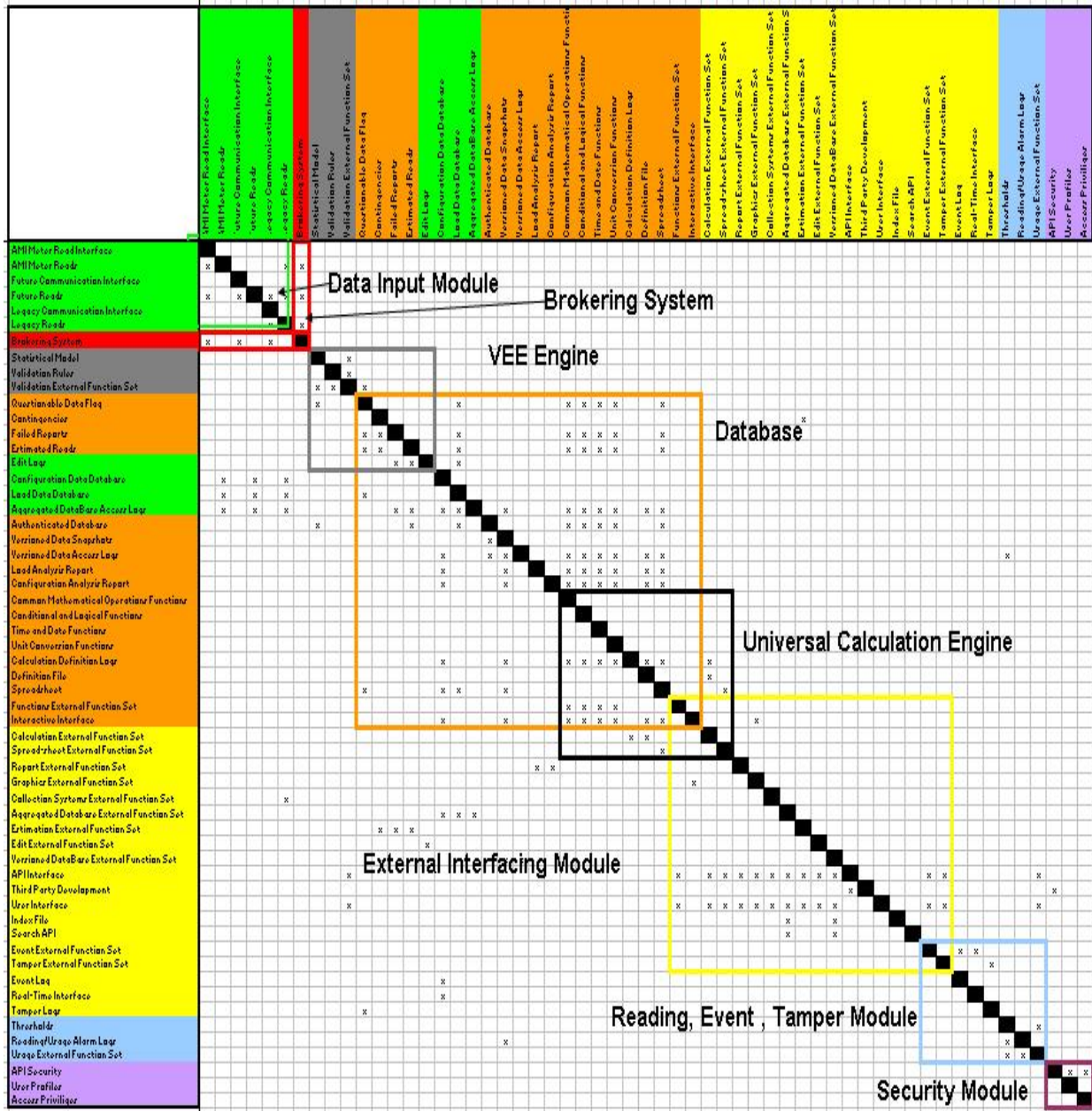


Figure 46 Product 1: Clustered DSM

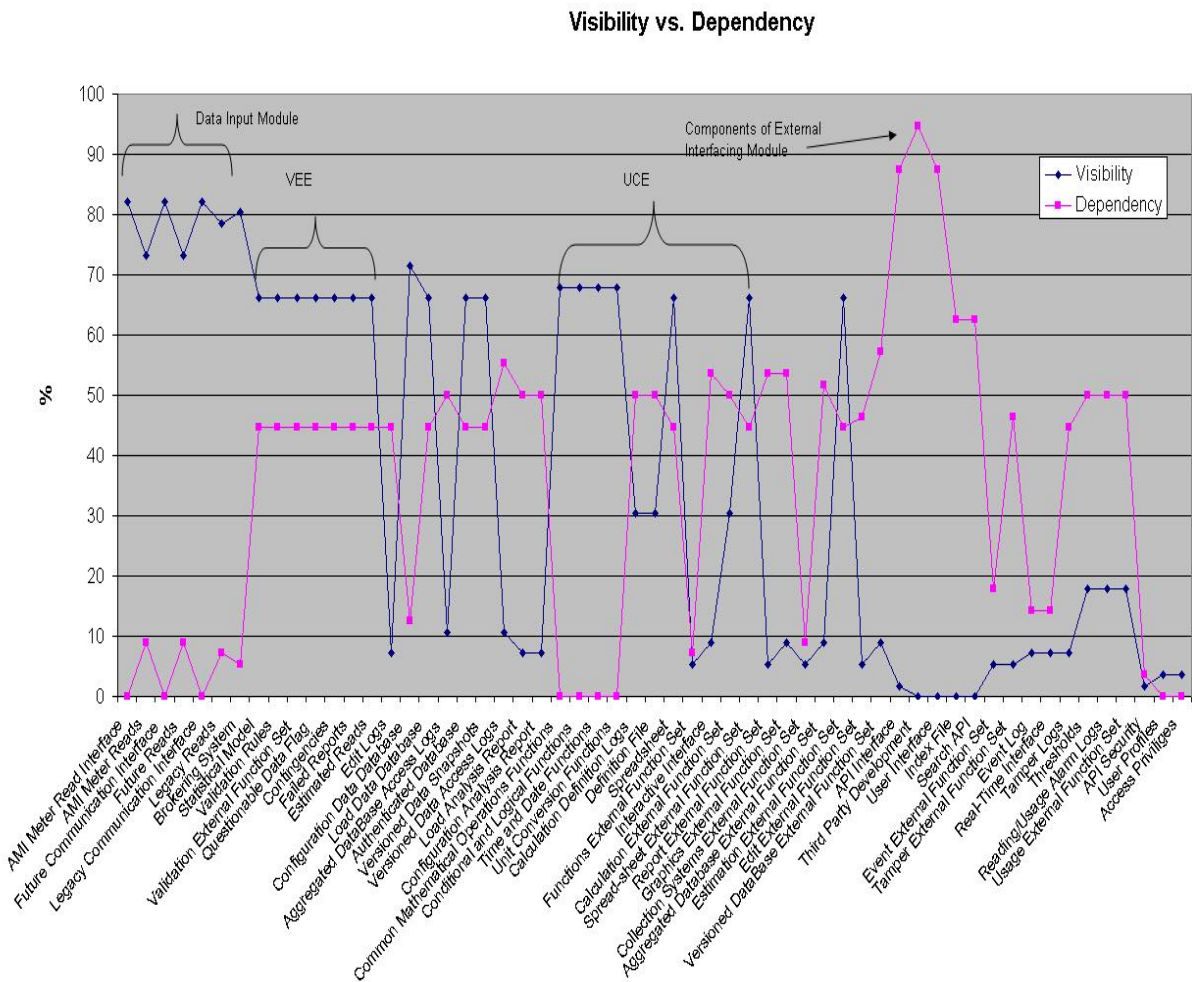
The clustered DSM combines the knowledge gained from OPM and Newman’s clustering analysis to identify modules such as the Universal Calculation engine and also to identify components that are shared by two modules such as the components shared by the VEE Engine and the Database. The Database module incorporates many components from the VEE Engine and Universal Calculation Engine because both the engines process the data in the database and hence are tightly connected.

In the OPM decomposition we had many different instances of the Database- Aggregated Database, Authenticated Database and the Versioned Database but after the DSM analysis we can conclude that all the different databases systems can be combined into a single large Database System. The large Database can have extra fields to differentiate between aggregated data,

authenticated data and versioned data. It should be noted however that the initial OPM Database decomposition was necessary to better understand the internal workings of the system.

The only suggestion based on the tacit knowledge of the P1 MDMS system will be to split the Configuration meter reads and Consumption meter reads into two separate databases since the database fields for these two reads are very different. The Consumption meter read database will have more activity than the configuration meter read database. This is because the Consumption Meter Read Database will be updated for every meter read whereas the Configuration Meter read Database will be populated once with the account information and accessed periodically for business applications. Each consumption meter read though will need a unique identifier to relate it to the specific entry in the Configuration Meter Read Database.

**Visibility Dependency Signature**



**Figure 47 Product 1: Visibility-Dependency Signature Analysis**

The Visibility-Dependency Signature for P1 is as shown in Figure 47. The Data Input Module has high Visibility because it feeds the database. The Data input module should have strict requirements for interfacing with any external collection system because any discrepancy will flow through the MDMS affecting the data and thereby the data analysis.

VEE Engine and UCE Engine also have high Visibility values because they handle the processing of the data before it is used by the other applications. Any changes in both should be carefully planned and special attention should be given to the dependencies with other modules in the system to prevent unplanned changes in the other dependant modules. At the same time, on the flip side, since these two modules are so centrally located they can be used to propagate change throughout the system.

Some components (API Interface, Third Party Development and User Interface) of the External Interfacing module have very high dependency. The Dependency values are at 90% so almost any change in the system will affect these three components in the External Interfacing Module. Since this modules interface with external systems, automatically communicating the changes to the external systems might be a good design decision to prevent costly delays and mistakes in the business ecosystem.

### 5.2.2 Product 1: Micro-Level Analysis

In order to analyze the micro-level characteristics of the system we need to first calculate the connectivity metrics.

#### *Connectivity Analysis*

The connectivity metrics of Indegree Centrality, Outdegree Centrality, Visibility and Dependency are calculated for P1 and the results are explained below.

#### Indegree Centrality

Component Name	Indegree
Aggregated DataBase Access Logs	14
API Interface	14
User Interface	14
Versioned Data Access Logs	10
Calculation Definition Logs	9
Interactive Interface	9
Failed Reports	8
Estimated Reads	8
Authenticated Database	8
Load Analysis Report	8
Configuration Analysis Report	8
Questionable Data Flag	7
Future Reads	5
Spreadsheet	5
Load Data Database	4
Functions External Function Set	4
AMI Meter Reads	3
Brokering System	3
Configuration Data Database	3
Aggregated Database External Function Set	3
Validation External Function Set	3
Estimation External Function Set	3
Edit Logs	3

**Table 3 Product 1: Indegree Centrality**

From Table 3 we can see that the nodes with the highest Indegree Centrality measures are the Aggregation Database Access Logs, API Interface and User Interface. Any access (read or write) to the Aggregated Database generates an input (write) to the Aggregation Database Access Logs that explains the high Indegree centrality measure. Almost all logs have a high measure of Indegree Centrality – Versioned Data Access Logs, Calculation Definition Logs. The API Interface and User Interface are the conduits of information from the MDM to the external operations and hence have a high number of inputs. Failed Report, Load Analysis Report and Configuration Analysis Report have a high degree of Indegree Centrality because of the number of nodes that are required to generate these reports.

### Outdegree Centrality

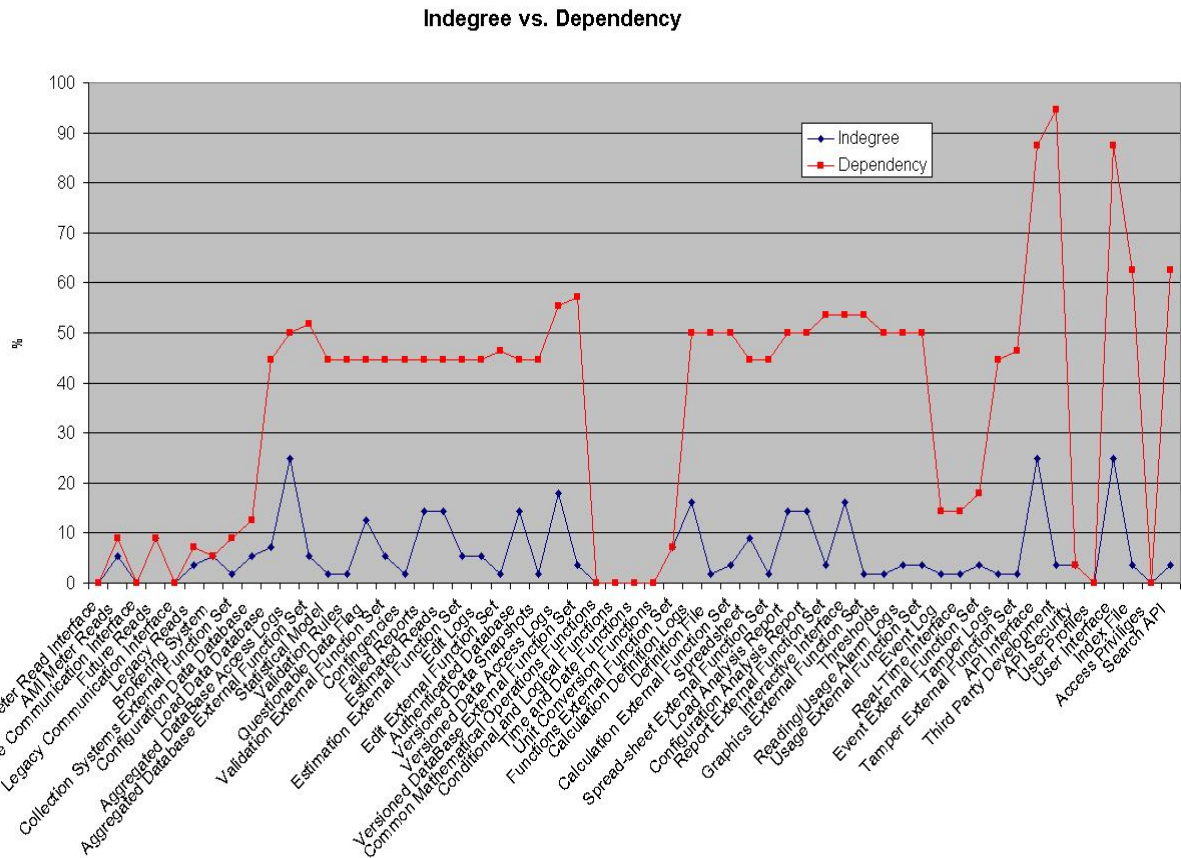
Component Name	Outdegree
Common Mathematical Operations Functions	11
Conditional and Logical Functions	11
Time and Date Functions	11
Unit Conversion Functions	11
Spreadsheet	11
Configuration Data Database	10
Versioned Data Snapshots	9
Load Data Database	8
Definition File	7
Legacy Reads	6
Questionable Data Flag	6
Aggregated Database External Function Set	4
Validation External Function Set	4
Estimated Reads	4
Versioned DataBase External Function Set	4
Calculation External Function Set	4
AMI Meter Read Interface	3
AMI Meter Reads	3
Future Reads	3
Legacy Communication Interface	3
Brokering System	3
Statistical Model	3
Contingencies	3
Failed Reports	3
Estimation External Function Set	3
Spread-sheet External Function Set	3
Graphics External Function Set	3
Thresholds	3
Usage External Function Set	3

**Table 4 Product 1: Outdegree Centrality**

From Table 4 we notice that the Components of the Universal Calculation Engine (UCE) have the highest measure of Outdegree Centrality. The UCE has high Outdegree because it is used extensively by both the internal systems (VEE Engine, Analytic Reports/Graphics) and the external Business Applications. The VEE Engines uses the UCE for estimation and validating purposes whereas the Analytic Reports/Graphics module uses the UCE for mathematical calculations. The external Business Applications uses the UCE for complex Load calculations.



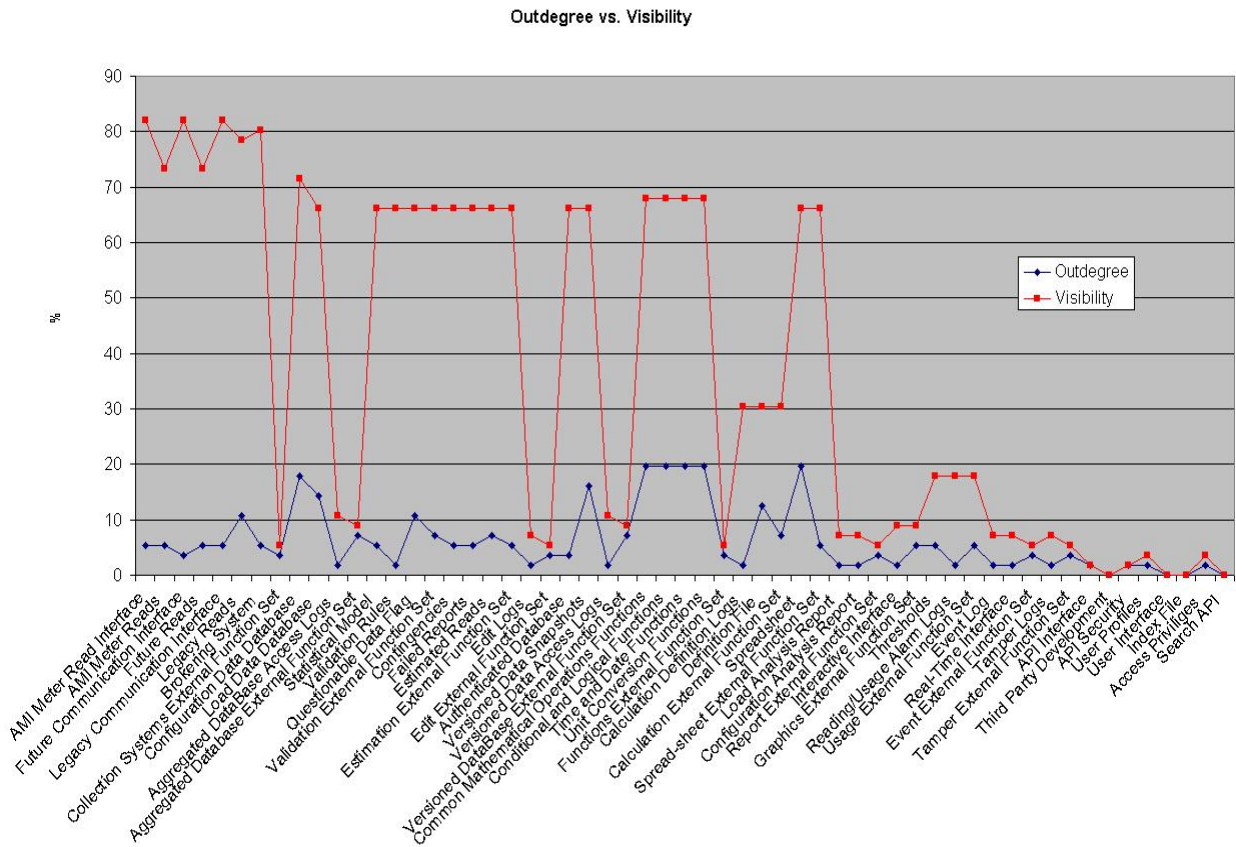
Dependency



**Figure 48 Product 1: Indegree vs. Dependency**

As demonstrated in figure 47, we notice that all the components of the VEE Engine have 50% dependency. Most of the Data Base components and the External Interface module components also have 50% dependency. The only modules with low dependency are the Data Input module and the Universal Calculation Engine. Interestingly the party development, index node and search API node although not identified in the Indegree centrality analysis as central nodes, they were captured here.

Visibility



**Figure 49 Product 1: Outdegree vs. Visibility**

As can be seen in Figure 49 the components of the Data Input Module have high visibility because they feed meter data into the MDMS which is further processed and analyzed by different components within system. Most of the other nodes have already been captured in the Outdegree Centrality graph.

**Analysis Summary**

Aggregation Database Access Logs, Versioned Data Access Logs and Calculation Definition Logs have high Indegree centrality values. The logs writes are required to have great response times since the performance of the entire database is dependant on the log writes as can be seen from the high measure of Indegree centrality. It is industry wide best practice to put these log devices on faster performing storage devices. These logs are very essential in reproducing the states of the system at different points in time so redundancy of data should also be a big design consideration

Failed Report, Load Analysis Report and Configuration Analysis Report have a high degree of Indegree Centrality. P1 MDMS generates reports for successful transactions as well as failed transactions. Report generation will be a resource intensive task (Figure 42) and the underlying database design, server design and storage design should be able to meet these demands. By

scheduling the daily report generation at off peaks hours if and when possible, heavy demands on the MDMS can be avoided.

The UCE has High Outdegree Centrality and consists of set of functions, calculation definition files and a spreadsheet like interface. The UCE will be memory and processor intensive. Complex number crunching calculations will require lots of processing power and the manipulating the spreadsheet will consume a lot of memory. While designing the system hardware memory and CPU power are important things to factor. The universal calculation engine software should be designed such as to efficiently use all available CPU power. Decoupling the UCE hardware to another system might be something to be looked into since it is a critical component in the MDM architecture.

The design for redundancy/evolution discussion should identify important firms to partner with to enhance the P1 MDMS offering. For powerful processing capabilities the P1 MDMS vendor has to partner closely with Server vendors, Data storage vendors and System Integrators to get the best scalable performance.

### **5.3 PRODUCT 2 FORMULATION**

The following section formulizes the P2 MDMS architecture using the same framework used to analyze the P1 MDMS architecture.

#### **5.3.1 Product 2: Identifying the System Boundary**

Since both P1 and P2 have the same primary purpose the system boundary definition of P2 is identical to P1. Please refer to 4.1.1

#### **5.3.2 Product 2: Identifying the Object-Process System Levels**

Identifying the Object-Process System Levels involves breaking down the system into levels and then identifying the interconnections as proposed in section 4.1.2.

##### ***Object Decomposition***

The object decomposition of P2 MDMS is as shown in Figure 50 and each of the Level 1 Objects is explained below.

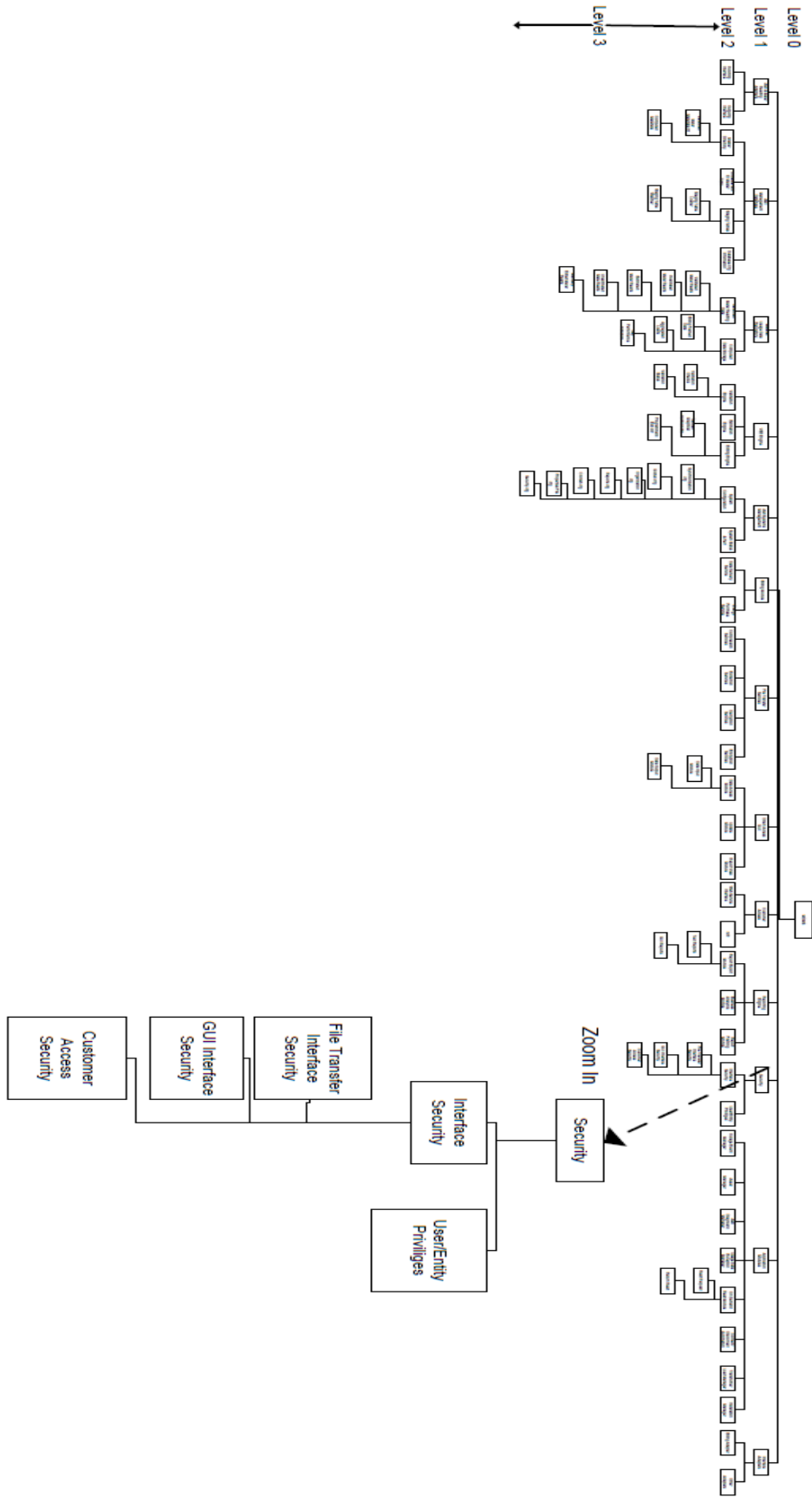


Figure 50 Product 2: Object Decomposition (Security Object Zoom In)

### AMI and Meter Reading Adapters

These adapters interface with the different external collection systems to populate the MDMS databases. The adapters support two way communications between the MDMS and the collection systems enabling on demand meter reads.

### AMI Management Database

AMI Management Database consists of the Universal Service Delivery Point Master Table, Master Directory, Staging Tables and Database Configuration Information. The Universal Service Delivery Point Master table has metadata information regarding all the meters individually or as a group. Only meters with information in the Universal Service Delivery Point Master table are allowed to communicate with the MDMS. Master Directory contains versioned metadata information and computed metadata information. The metadata contains all the meter configuration data information such as location and time zone. Staging Tables are used to pre-validate the data before loading into the database. The Database configuration file is used to configure the AMI Management Database

### Metered Usage Data Repository

The Metered Usage Data Repository stores all the consumption meter read information. It consists of versioned meter reading data and computed meter reading data.

### Validation, Estimation and Editing Engine (VEE)

The VEE Engine consists of the validation engine, estimation engine and editing engine. The validation engine validates the data before it is loaded into the database. The reads that do not pass validation are marked to be estimated or edited or re-read. The Estimation engine estimates reads based on pre-set guidelines. The meter reads that cannot be estimated by the estimation engine are marked to be edited. The editing engines handles all the reads marked to be edited. It generates a report to the System Administrator with all the reads to be edited and the system administrator can choose to manually edit the reads using historical data or the programmable edit interface

### MDM Systems Management

This module manages the MDMS. It is used to set parameters, modify configuration files and to manage security access. The MDM Systems Management module is also responsible for change management, release management, disaster recovery policy and backup policy.

### Billing Module

Primary use of the MDMS data is for billing purposes. This module make is easier by generating data in the right format for the billing system. The utilities employ different collection systems for gathering meter reads; the MDMS manages the different reads in different format and presents the billing utility with data in a format requested by the billing utility.

### File Transfer Services

File transfer services are used to transfer files between the MDMS and the other utility business units. It supports encryption/decryption and compression/extraction services.

### Direct Access GUI

Direct GUI access module is used for viewing the contents of the MDMS as well as an update module to manually change the MDMS values.

### Customer Access

Customer access module enables the utility consumers to check their utility bills and consumption trends via a web service interface or by using an Interactive Voice Response System.

### Reporting Engine

The MDMS system produces numerous reports and the reporting engine is the module responsible for generating the reports. The module generates daily batch reports as well as reports in response to specific business needs or queries. The reporting engine is made of the Business Analytics Module, Report framing module and the Report export module. The reports can be graphical or textual and can be transferred via the File transfer service, Direct Access GUI or email.

### Security

The Security module is responsible for all access to the system. It handles security for the File transfer services, GUI Interface and Customer Access Security.

### Application Module and Interface Adapters

This module is responsible for interfacing the external business modules with the MDMS. Some of the external applications it supports are Outage Event Manager, Asset Manager, Usage Data Exception Manager, Connect/Disconnect Automation, Transformer Manager, Billing adapter and Installation Manager. It also has the capability to add other adapters as required.

### ***Process Decomposition***

The process decomposition for P2 is as shown in Figure 51



### Read Meter Gathering

This process is the interface between the external collection systems and the MDMS. It consists of the incoming process and the outgoing process which handles the MDMS-AMI Data flow

### Loading

The Loading process is responsible for loading the meter reads into the databases. It consists of the Initial Loading process, Meta data loading and Meter Read Data loading. The Initial Loading process handles the functions to setup a new Meter instance. The Meta data loading loads the Configuration reads into the AMI Management Database and the Meter Read Data loading loads the consumption reads into the Metered Usage Data Repository.

### Processing

Processing consists of the Editing and Estimating process. Estimating is responsible for estimating consumptions reads that are marked to be estimated. The Editing process edits the consumption reads that are marked to be edited.

### Billing Processing

This process is responsible for generating and delivering meter reads as requested by the billing system.

### Configuring

Configuring process uses the AMI Systems Management module to configure the system parameters that define the behavior of the MDMS.

### Data Accessing

Data accessing is responsible for accessing the database by external applications and end consumers. It consists of the Querying process and the Customer Accessing process.

### Application Processing

Application Processing satisfies the request for data from the other Business Units.

### File transferring

File transferring is responsible for transferring files in and out of the MDMS. It uses the File Transfer Services to encrypt/decrypt and compress/extract the data and it uses the Security module to grant access to the MDMS.

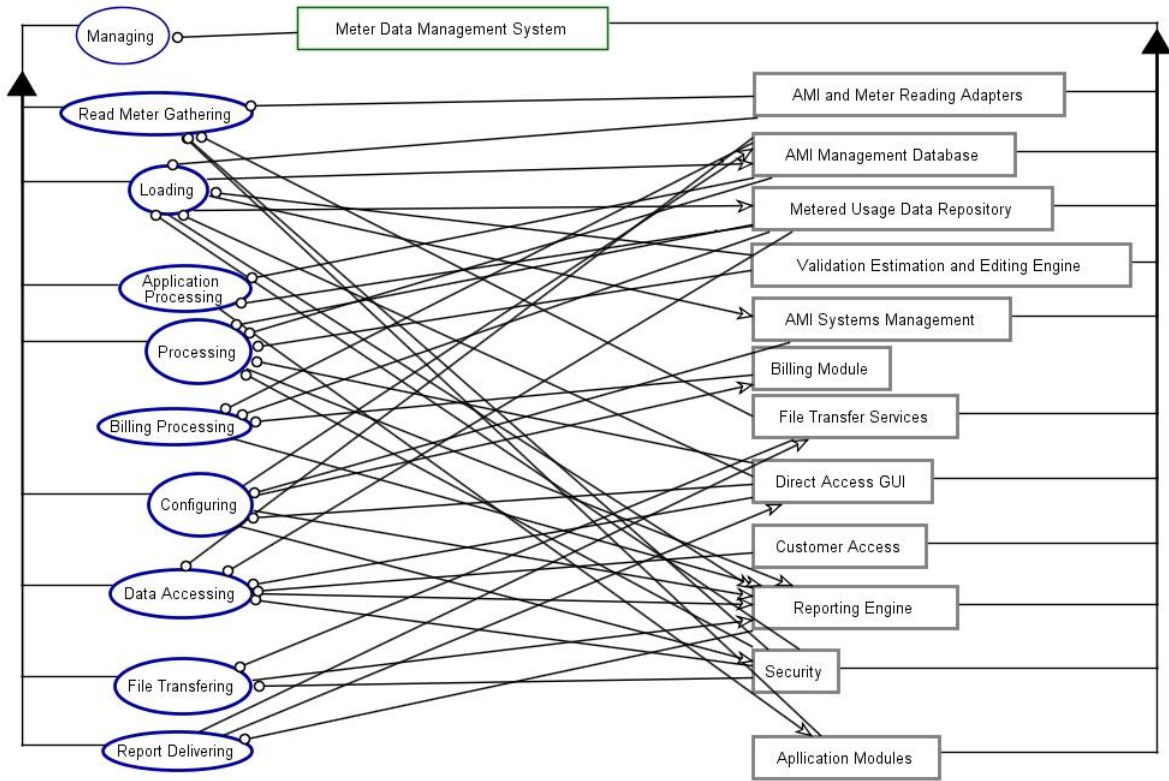
### Report Delivering

Report Delivering handles the displaying of report via the GUI interface.

### ***Combining the object and process decompositions and identifying the interconnections***

Combining the object and process decompositions starts at the top level. The input and output objects for each top level process are identified as shown in Figure 52.





**Figure 52 Product 2: Top Level OPD**

The top level connections determine the linkages between the objects and processes at the levels below. This process is iteratively repeated for each level down that yields the complete system object-process diagram as shown in Figure 53 .In Figure 53 the black lines represent aggregation-participation links, the green lines represent instrument links and the red lines represent result/consumption links (2.2.1).

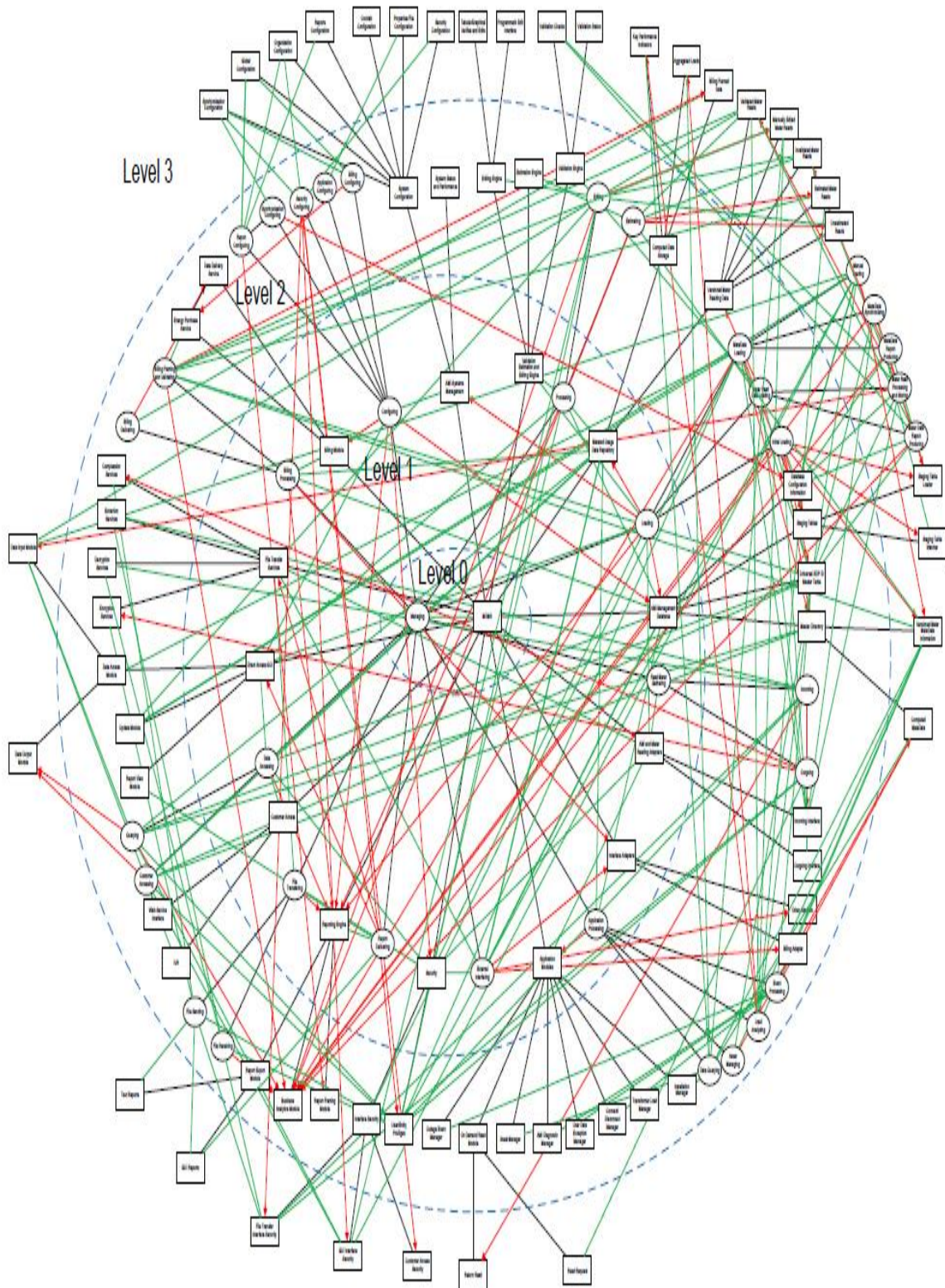
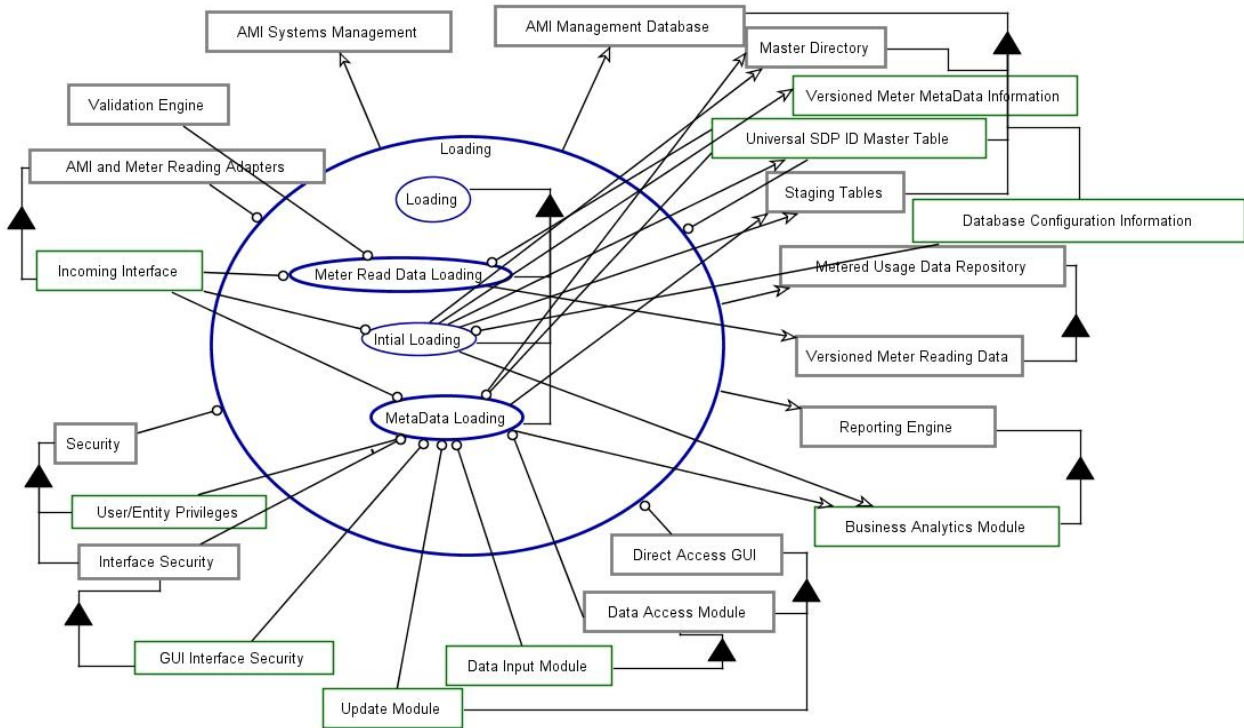


Figure 53 Product 2: System Level OPD

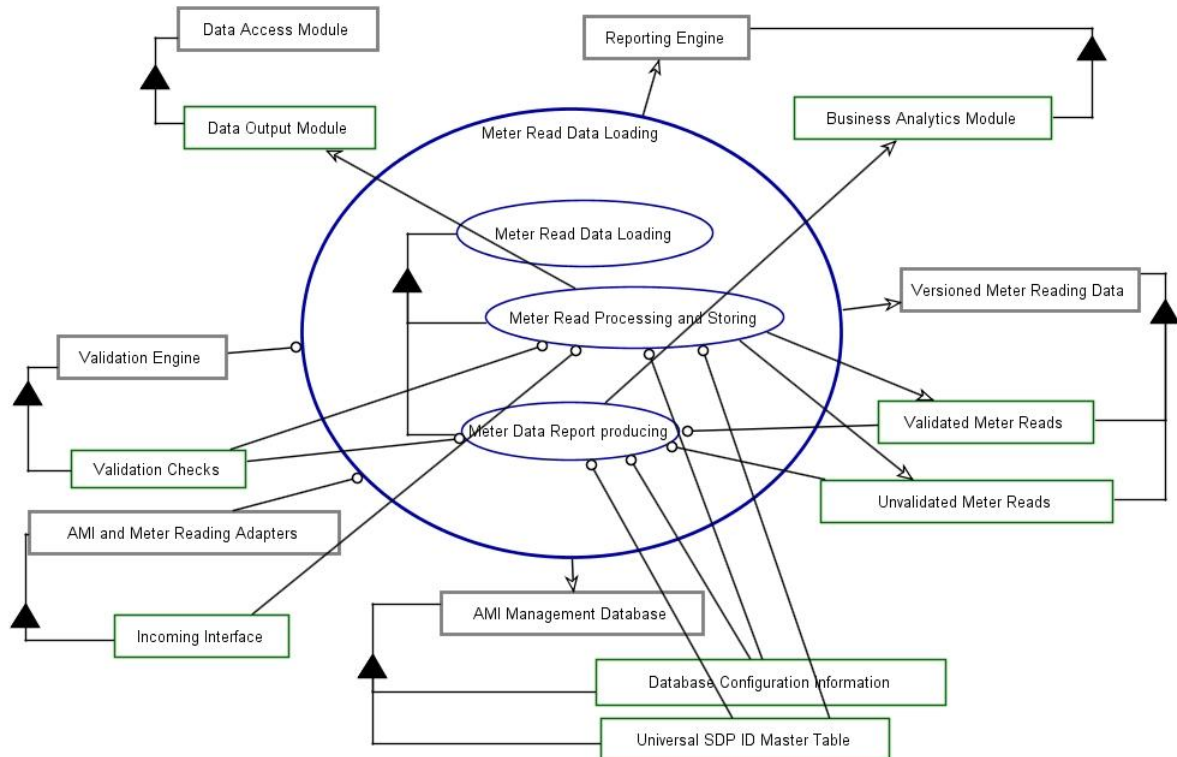
This section looks at the process centric views of some of the important processes in the P2 MDMS to identify the linkages between the objects.

*Interconnections in the “Loading” centric OPD*



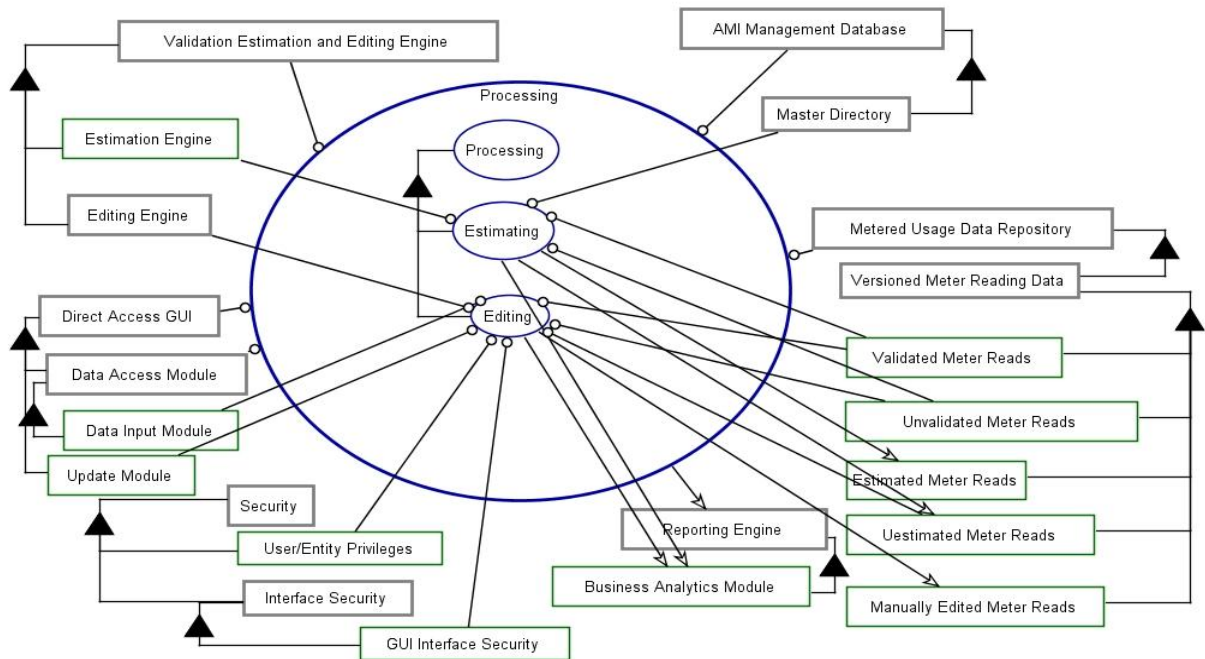
**Figure 54 Product 2: Level 1 Loading Centric OPD**

Loading process is responsible for loading the databases and it consists of three processes – Initial Loading process, Meter Read Data Loading process and MetaData Loading. The Initial Loading process is responsible for loading a new meter configuration instance. This step is important for setting up a link between the meter and the MDMS for all future communications. The MetaData Loading loads the configuration meter reads into the AMI Management Database. The Meter Read Data Loading is responsible for loading the consumption meter reads into the Metered Usage Data Repository and is shown in detail in Figure 55.



**Figure 55 Product 2: Level 2 Meter Read Data Loading Centric View**

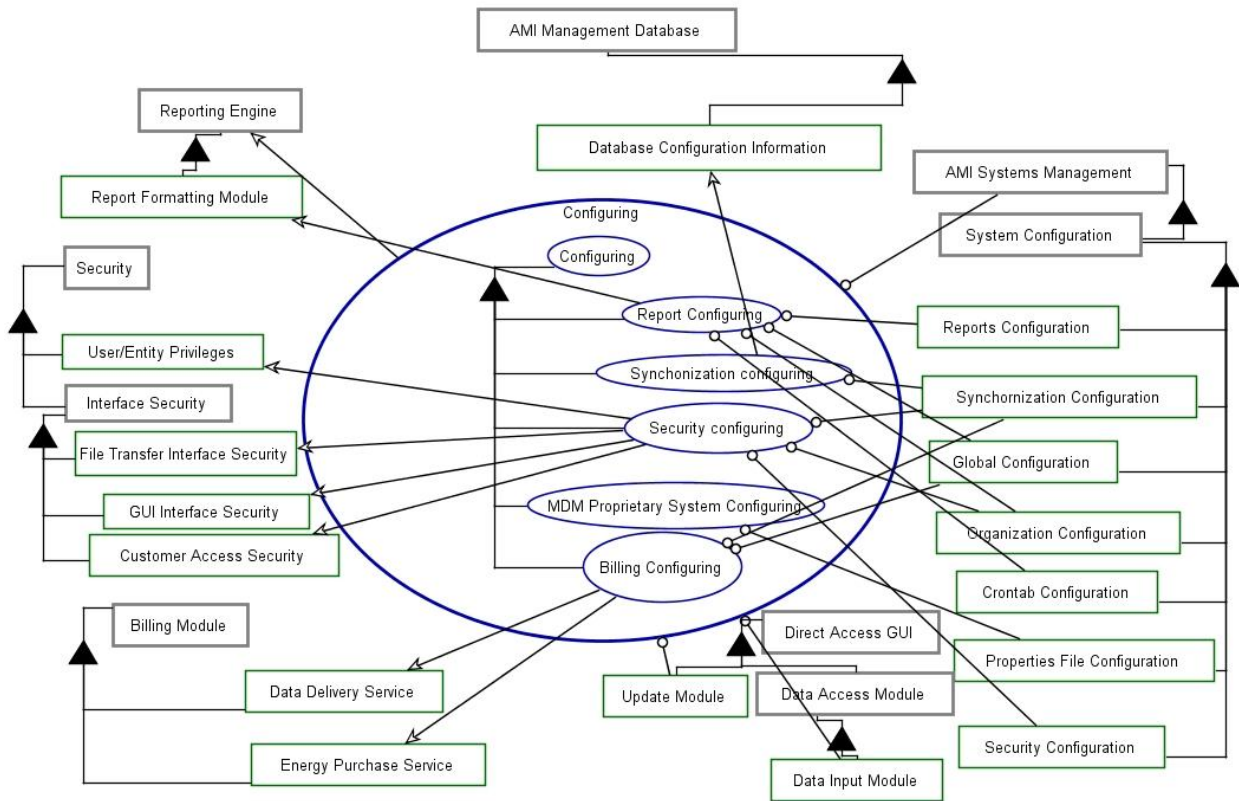
The Meter Read Data Loading process consists of Meter Read Processing and Storing, and Meter Data Report producing. The Meter Read processing and storing process validates the meter reads using the Validation engine and that validated data is stored as Validated Meter Reads whereas meter reads that cannot be validated are stored as Unvalidated meter reads. Both Validated and Unvalidated meter reads are stored in the Versioned Meter Reading Data System. The Meter Data Report producing process produces reports for successful transaction as well as a list of unsuccessful transactions. It pulls information from the AMI Management Database components as well as Versioned Meter Reading System and generates the reports using the Reporting Engine.

*Interconnections in the “Processing” Centric OPD*

**Figure 56 Product 2: Level 1 Processing Centric View**

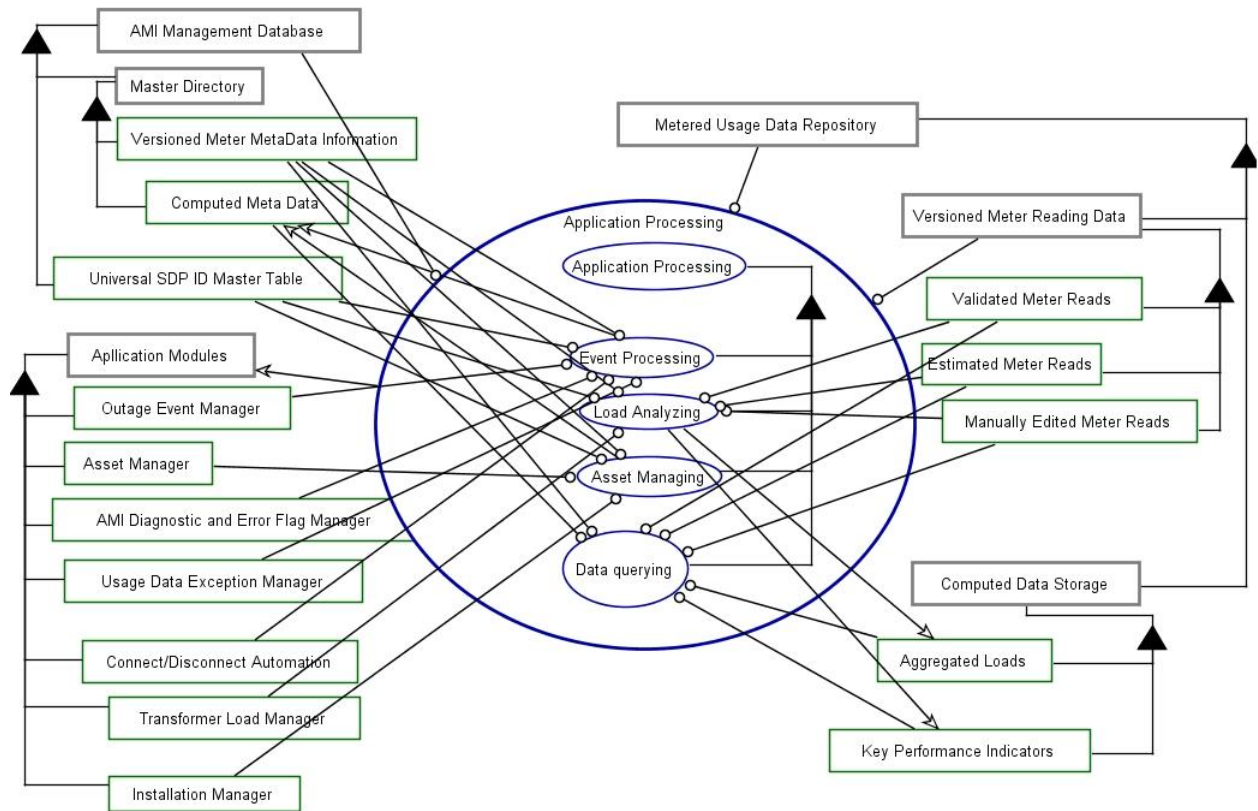
The processing process consists of the Estimating process and the Editing process. The Estimating process estimates the meter reads marked to be estimated by the Meter Read and Processing Process explained in the previous section. Reads that cannot be estimated are marked as Unestimated Meter Reads. The Unvalidated Meter Reads and the Unestimated Meter reads marked for editing are edited by the editing process. The editing and estimation process generate reports for successful and failed transactions.

*Interconnections in the “Configuring” Centric OPD*



**Figure 57 Product 2: Level 1Configuring Centric View**

Configuring process consists of all the processes needed to configure and manage the system. It shows the interconnections between the AMI Systems Management components and the configurable components in each module. It consists of the Report Configuring process, Synchronization Configuring process, Security Configuring process, MDM Proprietary System Configuring and Billing Configuring.

*Interconnections in the “Application Processing” Centric OPD*

**Figure 58 Product 2: Level 1 Application Processing centric view**

Application Processing is responsible for interfacing the MDMS with the external application adapters. It consists of Event processing, Load Analyzing, Asset Managing and Data querying. Event processing interfaces with the Application Module components such as Outage Event Manager and Usage Data Exception Manager to generate Computed Meta Data based on the events. Load Analyzing process uses information from the Transformer Load Manager, the AMI Management Database and the Versioned Meter Reading Data to produce the Aggregated Loads and Key Performance Indicators. Asset Managing is responsible for keep track of the utility assets and Data Querying provides the opportunity to query the databases.

### 5.3.3 Product 2: Creating the Dependency Matrix

All the interconnections from the OPD are transformed to a DSM as described in the framework section. The resultant DSM is as shown in Figure 59.

Level1		Level2		Level3	
AMI and Meter Reading Adapters	Incoming Interface Outgoing Interface				
AMI Management Database	Master Directory	Compiled Meter Data			
	Universal SDP ID Master Table	Versioned Meter Meter Data Information			
	Slicing Tables	Slicing Table Leader Slicing Table Watcher			
Metered Usage Data Repository	Dataverse Configuration Information	Validated Meter Reads Unvalidated Meter Reads Estimated Meter Reads			
	Versioned Meter Reading Data	Unvalidated Meter Reads Massively Edited Meter Reads			
	Compiled Data Storage	Billing Forward Data Aggregated Loads Key Performance Indicators			
	Validation Engine	Validation Checks			
	VEE Engine	Validation Engine Editing Engine			
AMI System Management	System Configuration	Tables/Graphical Verifies and Edits Synchronization Configuration Global Configuration Organizational Configuration Reports Configuration Credentials Configuration Preprocessor File Configuration Security Configuration			
	System Status and Performance				
	Billing Module	Data Delivery Service Energy Parameter Service Compression Service			
	File Transfer Services	Estimation Service Download Service Exception Service			
	Direct Access GUI	Data Input Module Data Output Module			
Customer Access	Update Module				
	Report View Module				
	Web Service Interface				
Reporting Engine	Business Analytics Module				
	Report Framing Module				
	Report Export Module	Text Reports GUI Reports			
Security	User/Entity Privileges				
	Interface Security	File Transfer Interface Security GUI Interface Security Customer Access Security			
	Outage Event Manager				
Application Module	Asset Manager				
	AMI Diagnostic and Error Flag Manager				
	Usage Data Exception Manager				
	General/Business Information				
	On Demand Meter Read Module	Read Request Release Read			
Interface Adapters	Transformer Load Manager				
	Installation Manager				
	Billing Adapter				
	Other Adapters				

Figure 59 Product 2: DSM

We transform the DSM into a Network Adjacency matrix as described in the framework section by replacing the marks by ‘1’ and the blanks by ‘0’.



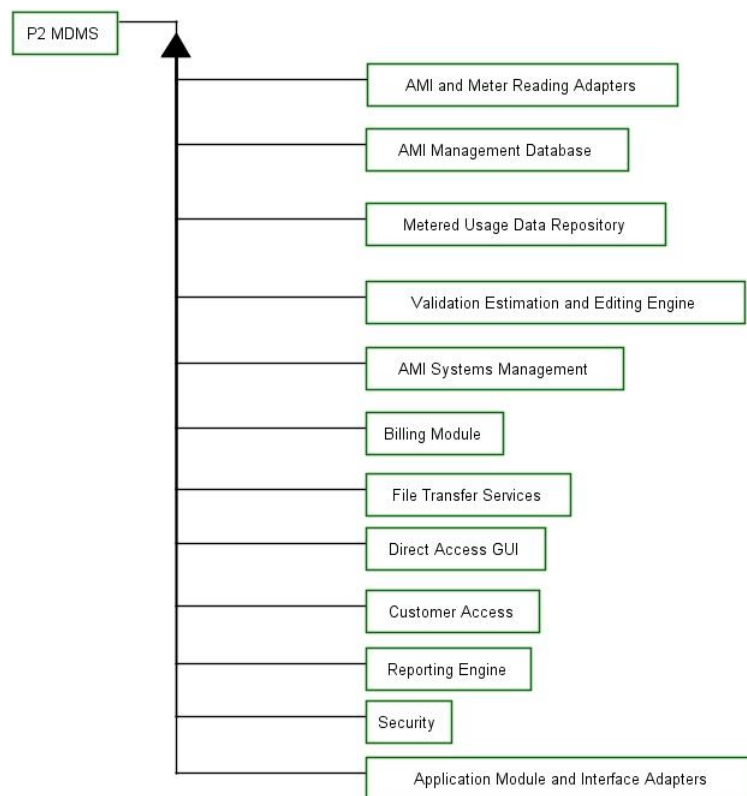
## 5.4 PRODUCT 2 ANALYSIS

The following sections explain the macro-level and micro-level analysis in detail

### 5.4.1 Product 2: Macro-Level Analysis

#### *Modularity from the OPM*

The initial modularity of the system is determined by the Level 1 OPM decomposition and consists of the following modules – AMI and Meter Reading Adapters, AMI Management Database, Metered Usage Data Repository, Validation Estimation and Editing Engine, AMI Systems Management, Billing Module, File transfer Services, Direct Access GUI, Customer Access, Reporting Engine, Security and, Application Module and Interface Adapters.



**Figure 60 Product 2 Modularity from OPM**

#### *Dependency Matrix Analysis*

Newman's clustering analysis on the adjacency matrix yields the grouping shown in Figure 61. The components with the same colors are grouped together. The clustering analysis revealed only 3 clusters which indicate that the P2 MDMS architecture is more integrated. From a top level functional point of view the P2 MDMS appears very modular but to achieve that modularity the lower level components appear to be tightly inter-connected.

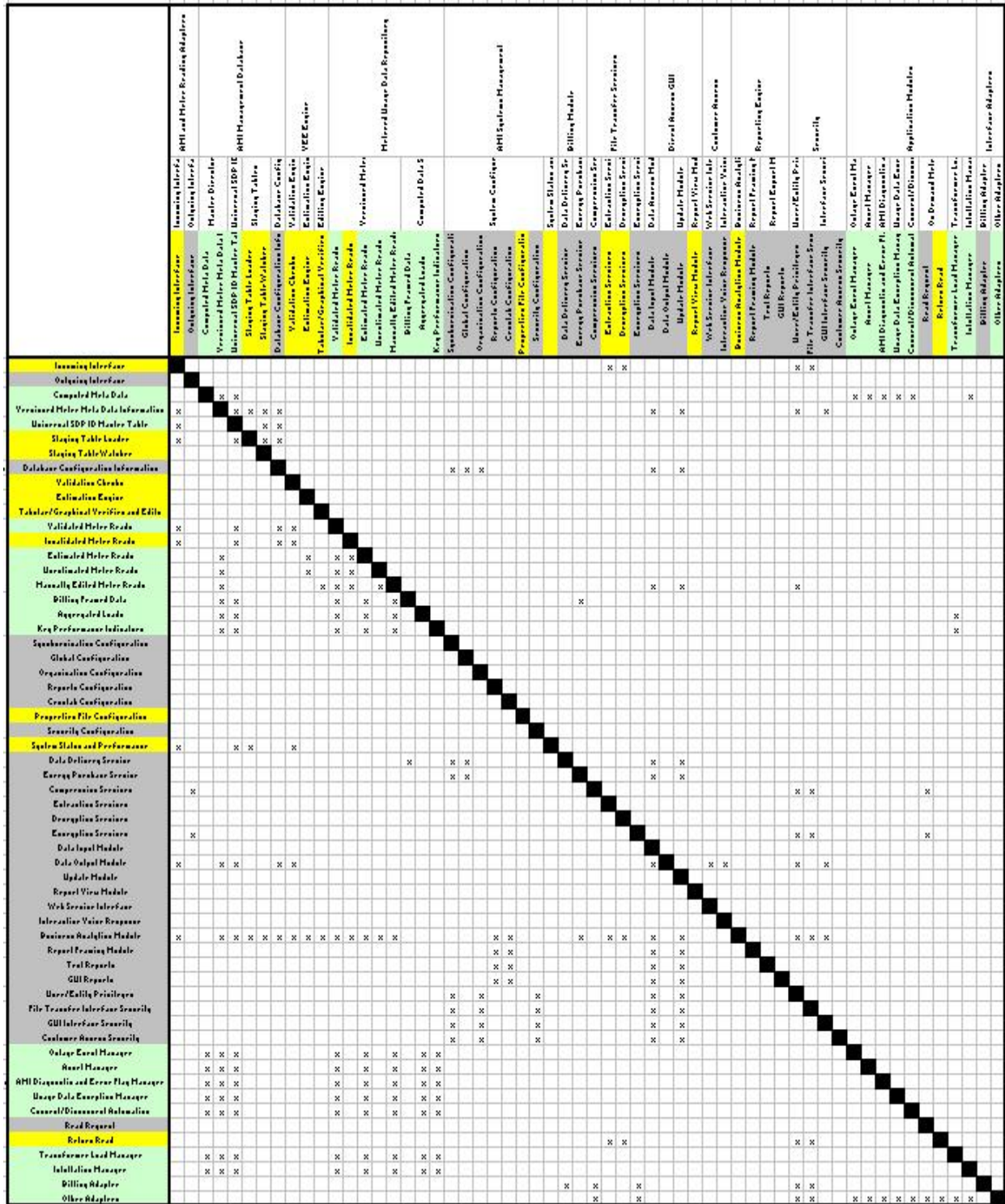


Figure 61 Product 2: Results of Newman Clustering Analysis

Based on the knowledge gained from the Newman’s clustering results, OPM understanding and using DSM clustering techniques a new clustering is proposed as shown in Figure 62. Four modules are identified – Application Module, Database Module, MDMS Management Module and Data Access Module. The Application Module interfaces the external application adapters with the Database components. The Database module consists of all the Database storing components such as AMI Management Database and the Metered Usage Data Repository. The

database processing modules -VEE Engine and the Business Analytics Module from the Reporting Engine are also contained within the Database cluster. The MDMS Management Module consists of all the components required to configure and manage the system. The Data Access Module includes all the different interface components and also the security components for these various interfaces.

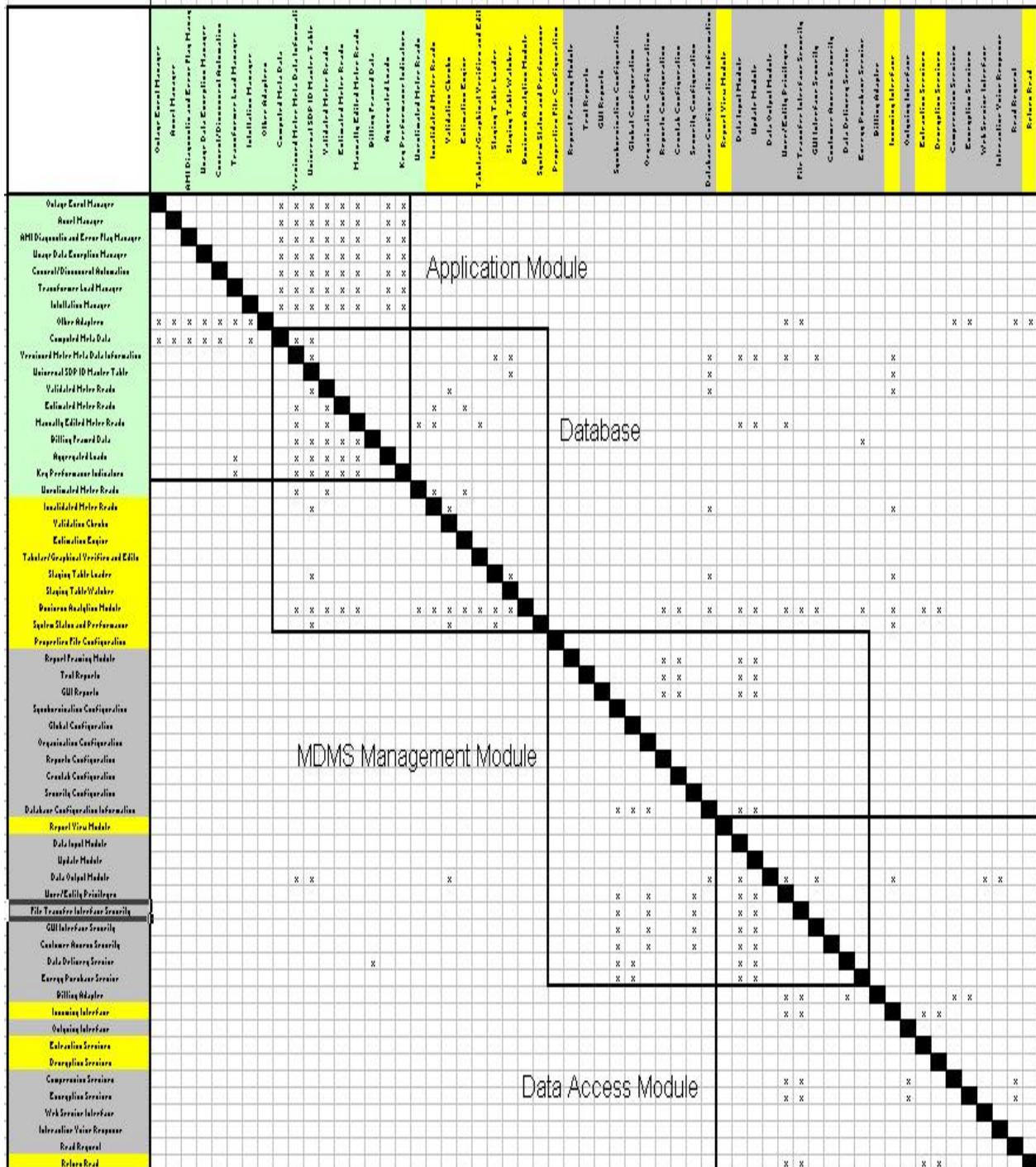
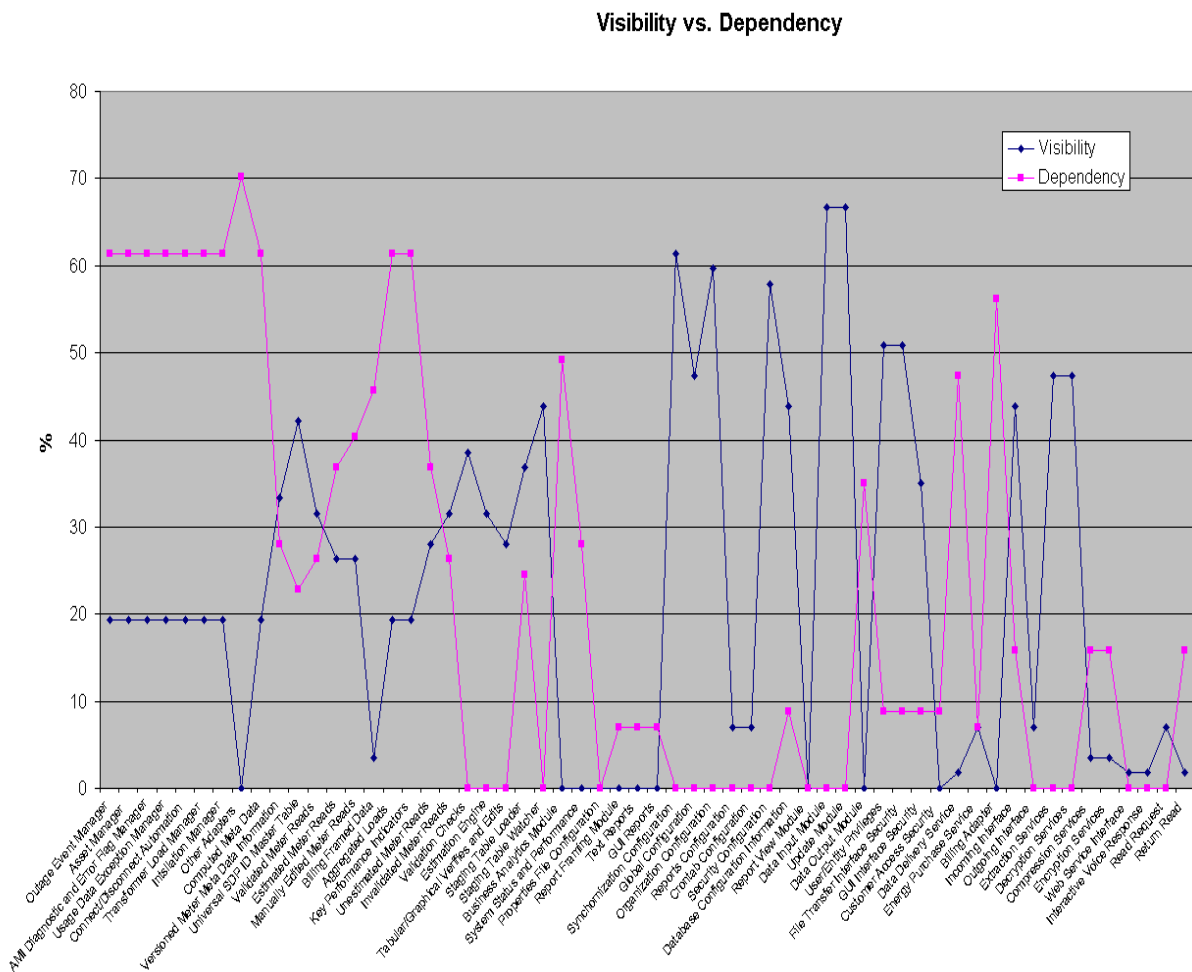


Figure 62 Product 2: Clustered DSM

**Visibility Dependency Signature**

As seen in Figure 63, the Application Module and the Database Cluster have high dependency. Also, Figure 62 shows that the Application Module is tightly interconnected with the Database module. The application module has adapters that interface with other business applications. These interfaces have to be well defined for efficient and correct functioning of the entire ecosystem. The high dependency of the Application module leaves it vulnerable to unplanned design changes because of changes to other components in the MDMS system. The adapters in the Application module should be well designed to somehow capture all these changes and give the same consistent data as requested by the external applications. This makes the design of the application modules very challenging.

The configuration module has high visibility as expected. This would signify that the P2 MDMS is highly user configurable which can be a big advantage for utility companies because they can change parameters according to their needs.



**Figure 63 Product 2: Visibility vs. Dependency Signature**

## 5.4.2 Product 2: Micro-Level Analysis

Micro-level Analysis studies the first order and higher order connectivity metrics.

### *Connectivity Analysis*

The connectivity metrics of Indegree Centrality, Outdegree Centrality, Visibility and Dependency are calculated for P2 and the results are explained below.

### Indegree Centrality

<b>Component Name</b>	<b>Indegree</b>
Business Analytics Module	24
Other Adapters	13
Data Output Module	10
Versioned Meter Meta Data Information	9
Computed Meta Data	8
Manually Edited Meter Reads	8
Outage Event Manager	8
Asset Manager	8
AMI Diagnostic and Error Flag Manager	8
Usage Data Exception Manager	8
Connect/Disconnect Automation	8
Transformer Load Manager	8
Installation Manager	8
Billing Framed Data	6
Aggregated Loads	6
Key Performance Indicators	6
Database Configuration Information	5
Data Delivery Service	5
User/Entity Privileges	5
File Transfer Interface Security	5
GUI Interface Security	5
Customer Access Security	5
Billing Adapter	5
Incoming Interface	4
Staging Table Loader	4
Validated Meter Reads	4
Unvalidated Meter Reads	4
Estimated Meter Reads	4
Unestimated Meter Reads	4
System Status and Performance	4
Energy Purchase Service	4
Compression Services	4
Encryption Services	4
Report Framing Module	4
Text Reports	4
GUI Reports	4
Return Read	4
Universal SDP ID Master Table	3

**Table 5 Product 2: Indegree Centrality**

Business Analytic Module is a part of the Reporting Engine. The MDMS generates multiple reports and that explains the high Indegree Centrality (Table 5). Other Adapters and the Application Modules (e.g. Asset Manager, Installation Manager) require a lot of information from many other components in the MDMS and hence have a very high Indegree Centrality value. Versioned Meter Meta Data Information, Computed Meta Data and Manually Edited Meter Reads are database components which are updated by many other components in the MDMS and therefore they have a very high Indegree Centrality Value.

### Outdegree Centrality

<b>Component Name</b>	<b>Outdegree</b>
Universal SDP ID Master Table	18
Versioned Meter Meta Data Information	16
Validated Meter Reads	14
Data Input Module	14
Update Module	13
Estimated Meter Reads	11
Manually Edited Meter Reads	11
User/Entity Privileges	10
Incoming Interface	8
Computed Meta Data	7
Database Configuration Information	7
Aggregated Loads	7
Key Performance Indicators	7
Synchronization Configuration	7
File Transfer Interface Security	7
Validation Checks	5
Organization Configuration	5
Staging Table Watcher	4
Unvalidated Meter Reads	4
Reports Configuration	4
Crontab Configuration	4
Security Configuration	4
Staging Table Loader	3
Estimation Engine	3
Global Configuration	3
Extraction Services	3
Decryption Services	3
GUI Interface Security	3
Read Request	3
Transformer Load Manager	3
Outgoing Interface	2
Unestimated Meter Reads	2
Tabular/Graphical Verifies and Edits	2
Energy Purchase Service	2
Compression Services	2
Encryption Services	2
Outage Event Manager	2
Asset Manager	2
AMI Diagnostic and Error Flag Manager	2
Usage Data Exception Manager	2
Connect/Disconnect Automation	2
Installation Manager	2

**Table 6 Product 2: Outdegree Centrality**

Universal SDP ID Master Table is used by many components in the MDMS to get the correct relationship between the SDP and the Meter reads. Versioned Meter Meta Data Information is used for in the VEE process and also for the application module which explains the high Outdegree Centrality. As expected, Validated Meter reads, Estimated Meter reads and Manually Edited Meter Reads have high Outdegree Centrality because they are used by all the downstream business applications. The Data Input Module, the Update module and the User/Entity Privileges module control the flow and access of information into the MDMS and hence have a lot of outputs to the components within the MDMS.

Dependency

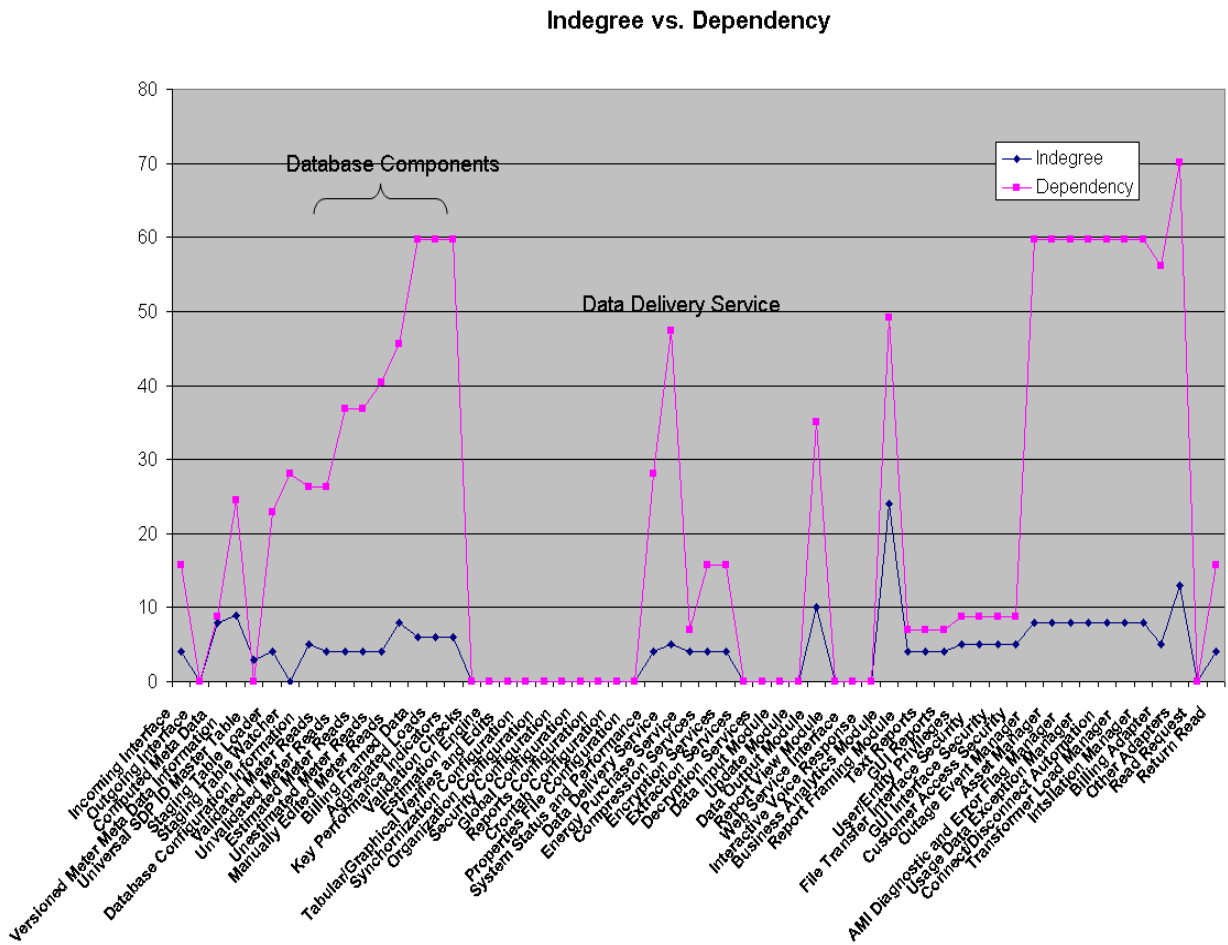


Figure 64 Product 2: Indegree vs. Dependency

The Database components have very high dependency because of the many MDMS components that are writing to the database indirectly to provide the desired functionality. Data Delivery Service which is a part of the Billing Module has very high dependency because it gets the information from the Database Components that have high Indegree Centrality and Dependency.

Visibility

System Configuration Components have high Visibility because they can change the configuration files that define the behavior of each module in the MDMS and hence can have far reaching effects. The Extraction and Decryption Services are part of the File transfer services and

they are responsible for feeding data into the MDMS and hence have high visibility. The File transfer Interface Security is used when the files are transferred into the MDMS and hence has high visibility.

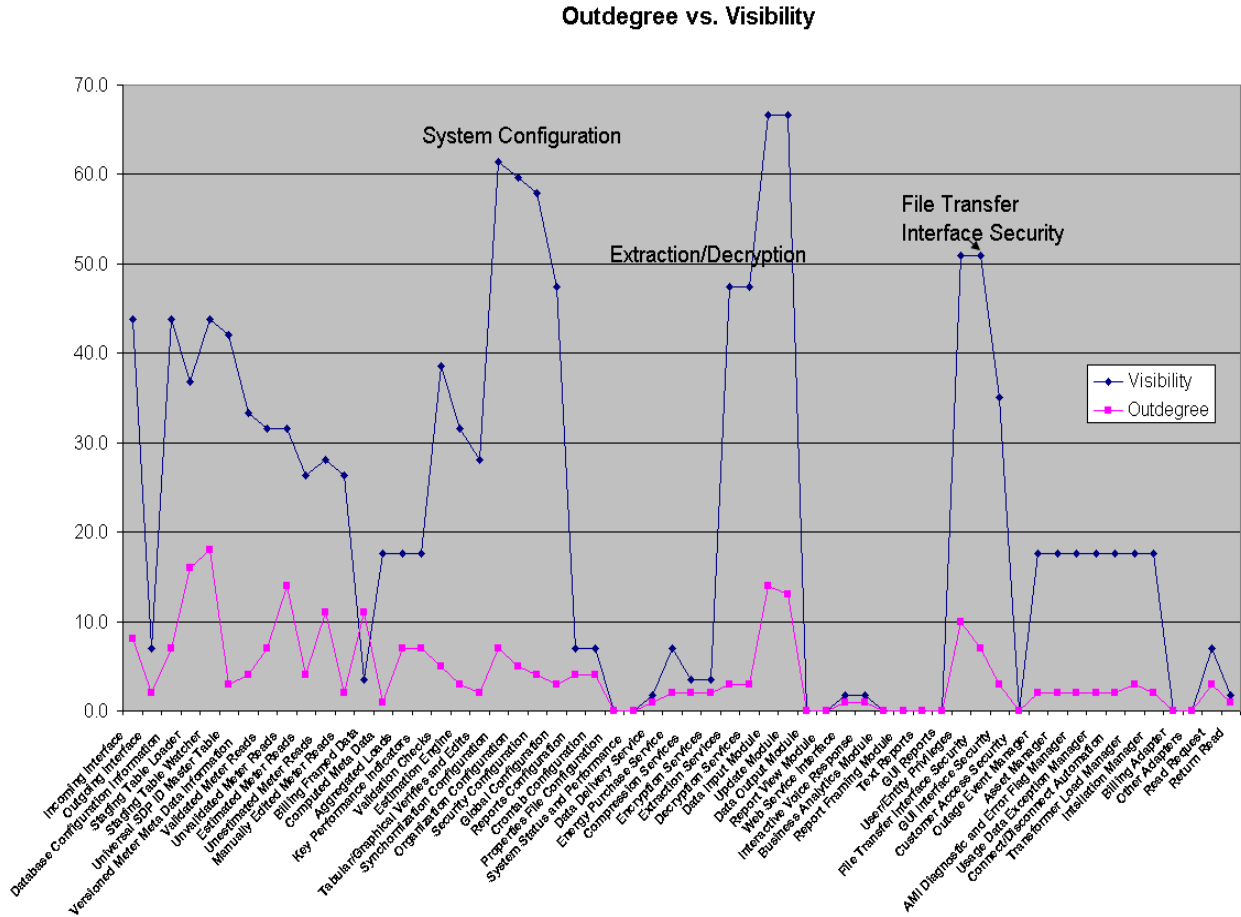


Figure 65 Product 2: Outdegree vs. Visibility

**Analysis Summary**

The Database components, Application Module components and the Business Analytics Module have high Indegree Centrality and Dependency which leaves them vulnerable to potential unplanned design issues. All these modules should have very strict and specific input interface guidelines which will prevent unplanned design issues and if the input modules to these components do not follow the guidelines an error report should be generated. The input modules to these components can also alternatively request a change in the interface if required thus making the design change a planned process.

Since MDMS is a primarily a database the Database components also have very high Outdegree centrality and visibility. Other components such as the Data Input module, Update module, User Privileges, Extraction/Decryption, System Configuration and the File Transfer Interface Security also have the same characteristics. Understanding the inter-connections is very important when



making design changes to such components because it will affect many components in the system.

The Database components are vital to the functioning of the MDMS. It is industry practice to have database redundancy in terms of backup, remote mirroring and disaster recovery strategies.

The Business Analytics Module is a central node in the MDMS. The Business Analytics Module is vital for generating reports and the MDMS generates a lot of reports. Multiple report requests arriving at once to the Business Analytic Module can lead to high response time thus affecting MDMS performance because this can slow the loading of data into the database. Since this is a software system slow system response can lead to bottlenecks affecting MDMS performance and hence high response time can lead to the system (MDMS) failure. The design, therefore, should allocate to the Business Analytic Module enough resources to process the incoming requests in a timely manner. The other option will be to stagger the report generation if possible to avoid bursty behavior.

A company that excels in graphical and analytical report generation can use the inputs of the Business Analytics Module to show the MDMS vendor the benefits of collaborating on that module. Similarly a firm which specializes in security can interface with the central security modules of the system to showcase their strengths and value add.

The MDMS vendor can also identify weakness in the system and take steps to address those issues. In this example the vendor might decide that based on the low connectivity metrics of the VEE module that its VEE capability is not up to par with the other vendor's VEE capabilities and therefore might need to offer more validating, estimation and editing routines to address those concerns.

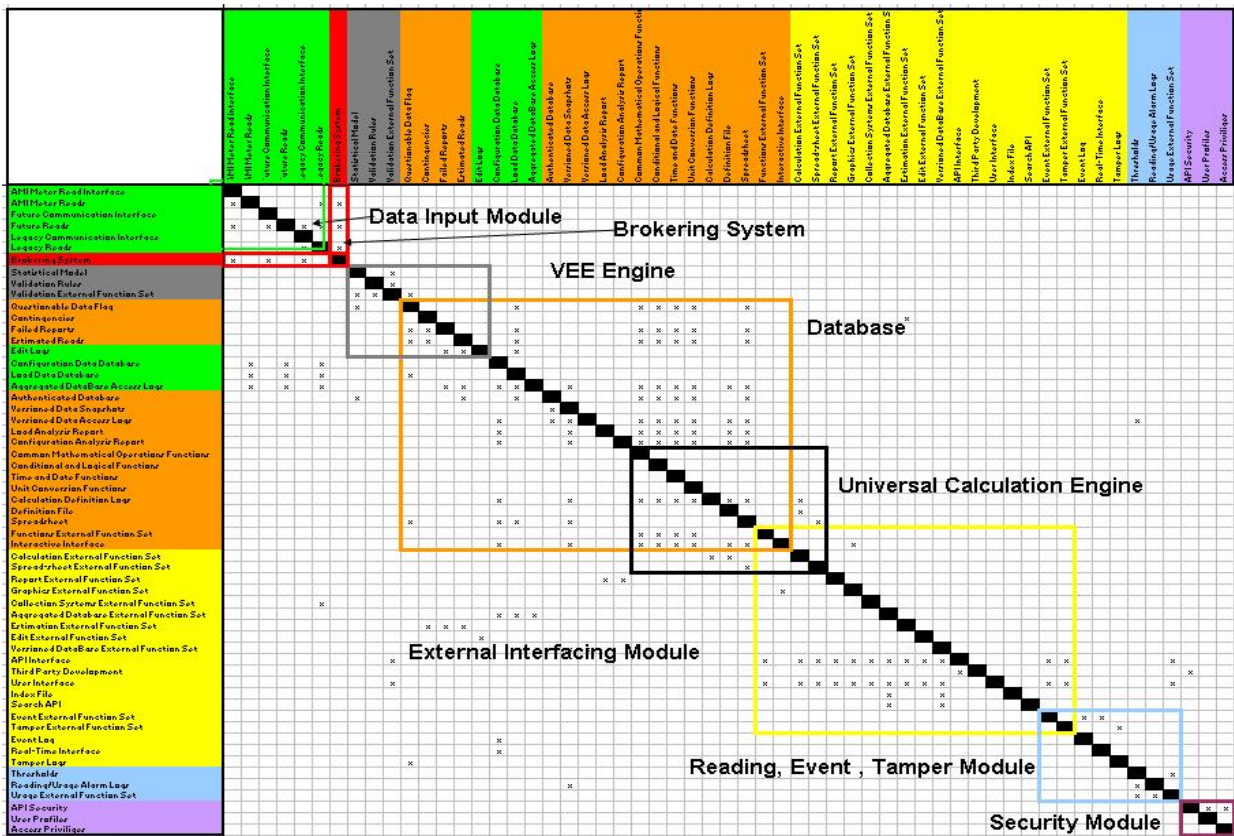
# CHAPTER 6. INFERENCES

This chapter simultaneously looks at the two MDMS architectures to identify strengths and weakness of each. Once this analysis is complete a combination of the best parts from both the architectures is synthesized into a proposed generic MDMS architecture. The last section of the chapter looks at the areas where companies can collaborate with the MDMS vendors to deliver value added services.

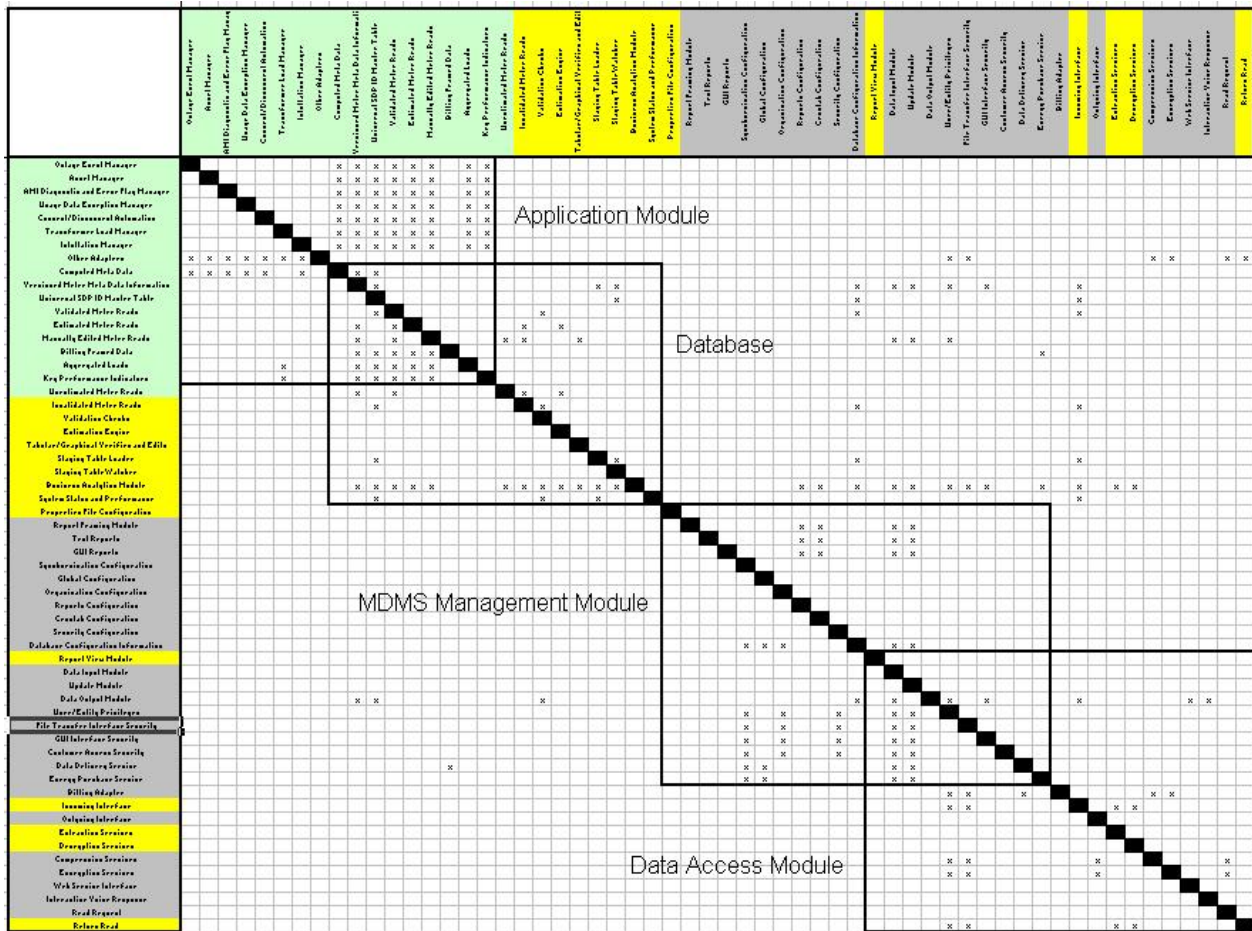
## 6.1 COMPARING THE ARCHITECTURES

Two systems that serve the same function may have some identical core modules. These core modules could provide the system’s basic functionality, but how well they provide such functionality depends on the implementation of supporting modules. These supporting modules can differentiate the strengths and weakness of each system’s architecture and can identify the designing firm’s distinct value proposition

For our two MDMS we propose first using a Macro-Level Dependency Matrix analysis to compare the two architectures. The Dependency Matrix clearly identifies the modules that make-up each architecture. By identifying modules which are present in a particular architecture but missing in the other we can weigh the relative strengths and weakness of the architectures. The Micro-Level analysis improves the understanding of the components within the modules. This can help us identify parts of a module that are more evolved in one architecture as compared to the other.



Product 1



Product 2

Figure 66 Comparing DSMs

Figure 66 compares the DSMs of the two architectures. P1 has the Universal Calculation Engine (UCE) which the P2 architecture is lacking. The UCE in P1 gives it an advantage over P2. It allows the external applications to perform complex calculations within the MDMS without having to worry about unit conversions and data integrity issues. The UCE prevents replication of data across multiple systems. In addition, it reduces chances of errors by having all the applications work off the same platform. The UCE also enhances MDMS security. The raw data is always stored in the MDMS and there is a log of all the calculations and database accesses. Finally, the UCE is also used by internal modules such as the Validation Estimation and Editing (VEE) engine making it more competent than the P2 VEE Engine offering.

Validating, editing and estimation processing is one of the main MDMS value propositions. This function is performed by the VEE Engine in P1 and P2. A well implemented VEE engine provides a significant advantage. The VEE Engine in P1 is better defined and has more functionality than the one in P2. This combined with the fact that the P1 VEE Engine can directly interface with the UCE makes the P1 VEE Engine more powerful than the P2 offering.

The data input module loads the MDMS. In the P1 architecture it is well-defined and has more functionality (migration) as compared to the P2 architecture.

The VEE and UCE components in P1 add more functionality to its MDMS offering but at the same time can hamper scalability of the MDMS. The Micro analysis of connectivity metrics reveals that the VEE and UCE components are high interconnected with the other nodes in the MDMS. As the system grows the demands on the UCE and VEE modules will increase. If not properly designed these central modules will become system bottlenecks and affect the performance of the entire MDMS.

The application module is responsible for interfacing the external business modules with the MDMS. The P2 application module has numerous adapters to leverage the MDMS. P2 also has a well defined System Management Module to configure and manage the MDMS. These two modules are lacking in the P1 architecture. The P2 Data access module which also includes the security components for each interface (GUI, IVR, and File Transfer) is more advanced than the P1 External Interfacing Module. Data flow through the P2 architecture is much better managed as compared to the P1 architecture.

Micro-analysis reveals that P2 System Configuration Components have high Visibility because they can change the configuration files that define the behavior of each module in the MDMS and hence can have system wide effects. Each utility might prefer to configure their MDMS in a different fashion and therefore, the more user-configurable the system the better, but that also increases the potential failure points in the system.

## **6.2 PROPOSING A GENERIC MDMS ARCHITECTURE**

A comparative analysis of both architectures reveals the strengths and weakness of the competing products. A generic architecture can then be proposed which would combine the best, most well-defined modules from each product. The generic architecture is important to competitors designing similar products but also to firms collaborating with these multiple products.

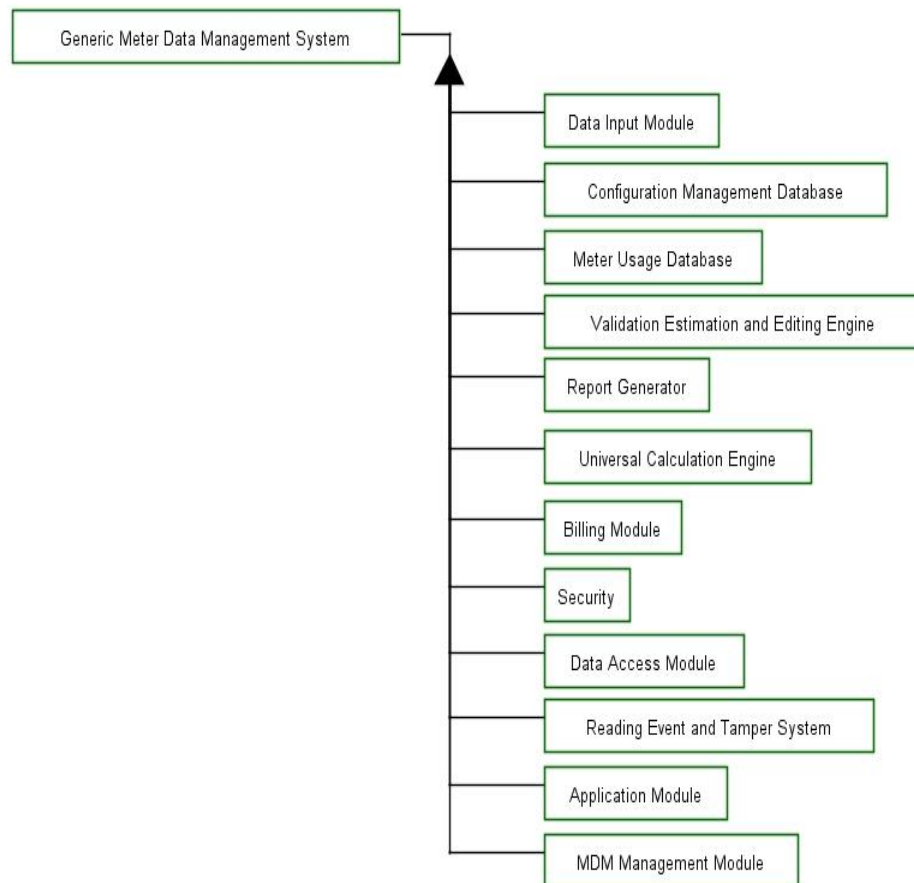
Section 6.1, describes the strengths and weakness of the individual architectures. By combining the strengths of the two architectures and using the knowledge gained from OPM analysis a generic architecture is proposed and the top level OPD is as shown in Figure 67.

The modules proposed for a generic architecture are as follows:

- Data Input Module – Feeds data from the various collection systems in the MDMS. Should have capability to enable seamless migration of Legacy meter read Collection Systems to AMI Collection Systems.
- Configuration Management Database – Stores the configuration information (e.g. time-zone, contract id, power characteristics) for each Service Delivery Point. Service Delivery point can be a single meter or bunch of meters combined together.
- Meter Usage Database – Stores meter consumption information. Life cycle Management of the consumption data – Initial Read, Validated Reads and Edited Reads. Stores

computed data from other Business application. Version control of Data for auditing and reuse purposes.

- Validation, Estimation and Editing Engine – Validates the reads based on business guidelines and flags reads which need to be estimated or edited. Estimates the flagged reads using pre-defined contingencies. Edits the flagged reads manually or via a programmable interface. Track all estimates and edits for auditing purposes.



**Figure 67 Generic MDMS Architecture top level OPD**

- Report Generator – Generates MDMS data loading reports, VEE reports, Business-Unit reports and User-Defined reports.
- Universal Calculation Engine – Mathematical engine to handle all complex calculations in one platform thereby reducing errors, preventing data duplication, reducing data transfers and increasing data security.
- Billing Module – Generate billing ready data as requested by the billing system.
- Security – Controls all access to the system via file transfer, web or APIs. Keeps detailed logs of all access to the system.

- Data Access Module – Enable other applications to access the data via different methods (web, file transfer, thin clients).
- The Reading, Event and Tamper System – Synchronous system to communicate real-time information to other business units e.g. Outage Management System.
- Application Module – Adapters and Interfaces to enable leveraging of the MDMS Data.
- MDM Management Module - Handles the management of the system. Setting up the configuration files and security access. Responsible for change management, release management, disaster recovery policies and backup policies.

The Data Input Module has high Visibility because it feeds the Database and hence should have strict requirements for interfacing with external collection systems because any discrepancy in the incoming data will flow through the MDMS affecting the data and the data analysis. The functionality to interface with multiple collection systems can be inbuilt into the MDMS or can be a separate Collection Interface System.

Based on our understanding of the MDMS from the Macro OPM analysis, the Database module in the DSM has been split into two separate modules – Configuration Management Database and Meter Usage Database. The database fields for the two databases are different since the Configuration Management Database will store information such as contract id, location and time zone whereas the Meter Usage Database will store electricity load information. The Meter Usage Database will have more activity (throughput) than the Configuration Management Database. This is because the Consumption Meter Read Database will be updated for every meter read whereas the Configuration Meter read Database will be populated with the account information only once and will be accessed periodically for business applications.

From the micro analysis we expect that the Database Log components of the Database Module will have very High Indegree Centrality metrics because any access (read or write) to the Database generates an input (write) to the log devices. This is done for database restore and/or auditing purposes. The Log device writes are required to have great response times since the performance of the entire database is dependant on the log writes and it is industry wide best practice to put these log devices on faster performing storage devices.

Furthermore, The VEE Engine and the UCE module have components that work very closely with the Database module components and hence they have many interdependencies. The design decisions involving any of these three modules should take into consideration the effects on the other two modules. This will prevent unforeseen bottlenecks in the system thus improving system scalability.

VEE Engine and UCE Engine also have high Visibility values because they handle the processing of the data before it is used by the other applications. Any changes in both should be carefully planned and special attention should be given to the dependencies with other modules in the system to prevent unplanned changes in the other dependant modules. On the flip side, since these two modules are so centrally located they can be used to propagate design changes throughout the system.

The Billing module was stated as a separate module instead of just including it in the Application module because seamlessly generating bills from meter reads collected via multiple Collection Systems is one of the main value propositions of the MDMS.

The components of the Report Generator module have high Indegree centrality. The Report Generator was explicitly stated because of the volume of reports the MDMS generates and hence attention must be paid to the module because it can be extremely resource intensive. The underlying database design, server design and storage design should be able to meet these demands. The reports storage database should be separate than the other Database modules since its access pattern will be predominantly sequential.

The Application Module components have high dependency metrics and it interfaces with external systems. To prevent costly delays and mistakes in the business ecosystem, internal MDMS design changes that affect the application module should be automatically communicated to the external systems

### 6.3 LEVERAGING THE MDMS ARCHITECTURE

The knowledge gained from the MDMS macro and micro analysis is used to identify six potentially areas to collaborate with the MDMS vendor.

Generic MDMS Architecture Modules	Areas to Collaborate
Data Input Module	Data Loading
Configuration Management Database	Data Storing
Meter Usage Database	
Validation, Estimation and Editing Engine	Data Manipulation
Universal Calculation Engine	
Security	Data Security
Report Generator	Data Leveraging
Billing Module	
Data Access Module	
Reading, Event and Tamper System	
Application Module	
MDM Management Module	System Management

Table 7 Areas to Collaborate

Table 7 summarizes the areas of collaboration. In the figure the Modules derived from the Macro-Level Analysis are grouped based on their functionality; the areas of collaboration are based on these groupings. Once a firm decides to collaborate with a particular MDMS vendor then the Micro-Level understanding of that particular architecture is required to deliver the required functionality.

#### ***Data Loading***

The Data input module of the MDMS has high visibility since its feeds data into the MDMS which is then processed and analyzed by the other modules in the MDMS. The amount of data collected by the AMI systems to be fed into the MDMS is expected to grow exponentially (Figure

13). This will put a lot of strain on the Data Input module of the MDMS architecture. The potential value-added tools would include:

- Intelligent ways to stagger the load to prevent huge bursts of data coming down at once to the MDMS.
- Data input module design with large buffer allocation to capture bursts when they do occur to maintain system performance.

### ***Data Storing***

Data storing include all the traditional storage related systems such as – Primary storage, Backup and Recovery, Disaster recovery, and Data archiving and restoration.

The biggest value-added here is the ability to differentiate the data load flowing through the system based on characteristics such as IO size, throughput and access pattern. This allows the Storage Administrator to make design trade-offs while allocating resources to different modules to achieve maximum system performance. From the analysis we can identify the central components that process the data and the central components that access this data within the MDMS. The load can be differentiated by understanding the flow of load through the MDMS and then analyzing whether the data load characteristics change when it is processed by a particular component.

For example, the data load characteristics for the Configuration Database is very different when compared to the Meter Usage Database. The components of the Configuration Database have lower connectivity metrics as compared to the components of the Meter Usage Database. From the understanding gained during the OPM analysis we also know that the Meter Usage Database is updated more frequently than the Configuration database and hence it should be designed to handle a heavier load than the Configuration Database.

The data flowing through the other modules of the system can also be analyzed to design a storage system that will prevent bottlenecks and thus enhance system scalability. For example the components of the UCE and Report Generator have high connectivity metric values. The characteristics of the data (IO size, access pattern) flowing into the UCE might be very different from the ones flowing out of the UCE. Similarly the data flowing into the Report Generator will be randomly accessed (read) from multiple tables in the database whereas the output (write) will be a report file which will be sequential in nature. Potentially the System Administrator can then allocate a different pool of storage resources for the data flowing in as compared to the data flowing out.

For backup and archiving purposes data deduplication will help in making efficient use of the storage.

### ***Data Manipulation***

Data Manipulation includes services such as VEE and Complex load calculations. From the discussion in 6.1 we know that the P2 architecture does include a dedicated calculation engine. P2 can collaborate with an external vendor to incorporate this technology into their architecture.



MDMS vendors can also collaborate with other vendors to enhance the VEE engine or UCE capabilities. Data Manipulation functions are resource intensive so any technique to increase their efficiency will improve the overall MDMS performance. The high resource requirements will also call for partnering with hardware firms that deliver the performance required.

The components of the UCE and the VEE have high visibility values because they handle the processing of the data before it is used by the other applications. For powerful processing capabilities the P1 MDMS vendor has to partner closely with Server vendors, Data storage vendors and System Integrators to get the best scalable performance.

### ***Data Security***

The MDMS interfaces with the external systems using multiple techniques such as file transfer, web, interactive voice response, automatic emails and thin clients. The security system should grant only authorized personnel and systems access to the MDMS. The security module should also be able to handle different access rights for each system and personnel. All the above requirements make the security of the system a very challenging job. If the MDMS vendor does not have expertise in security it should partner with a firm that can offer the high level of security for effective and secure communication with the MDMS.

The security module components that have the highest connectivity metrics are the ones handling the file transfer interface whereas the remaining components have very low connectivity metrics. This enables designing the security module as a separate entity and then interfacing it with the MDMS and thus making the Security module a prime candidate for collaboration.

### ***Data Leveraging***

Data leveraging focuses on the Business Intelligence applications that use the MDMS data for performing their analysis. The MDMS Report Generator module and Billing module are prime examples for delivering value-add services to the Business Intelligence Applications.

Future value-add applications can be designed after the external applications are analyzed in detail to identify their needs. Since MDMS is the shared resource between multiple Business Intelligence Units it can be used for inter Business Unit communications. For example the Outage Management System detects a power outage in real time and then it has to notify the Workforce Management System and Asset Management System based on the diagnosis. MDMS should be able to provide an interface to handle that communication thus enabling seamless system hand-off.

### ***Systems Management***

The MDMS is a large system to manage and configure. System Configuration Components have high Visibility metrics because they can change the configuration files that define the behavior of each module in the MDMS. Systems Management also deals with usability of the MDMS in terms of metrics such as responsiveness, presentation. As the MDMS grows larger the MDMS vendors will need expertise to scale the system management capabilities at the same rate. The Systems Management is the window into the MDMS and a MDMS with poor usability metrics will lose out to the competition.

## CHAPTER 7. CONCLUSION

The thesis presents a method of analyzing architecture at two levels macro and micro. A framework to understand the architecture at different levels by quantifying the findings at the macro-micro level and using that micro-macro analysis to draw conclusions is the crux of this thesis.

### 7.1 PROPOSED FRAMEWORK

The framework proposed consists of two phases: Formulation and Analysis. Formulation is the first phase and involves breaking a complex system down into smaller entities, identifying the relationship between those entities and representing the system in an easy to analyze format. OPM was the tool used to break down the system into smaller chunks and to identify the relationship between those chunks. The Dependency Matrix was used to represent the relationships in an easy to analyze matrix format. Formulation generates two forms of system representation the OPD and the Dependency Matrix, both of these are used for architectural analysis. The Formulation step is as important as the Analysis steps in drawing conclusions. The tacit knowledge gained while decomposing the system into smaller chunks and then connecting the chunks is vital to the holistic understanding of the system.

The Analysis consists of a Macro-Level Analysis and a Micro-Level Analysis. Macro-Level Analysis draws conclusion based on high-level analysis of the system. It specifically identifies the system modularity based on Dependency Matrix clustering techniques and the knowledge gained from the OPM. Macro-Level Analysis also identifies the influence of those modules on the system and the impact of the system on those modules with the help of connectivity metrics derived from the Dependency Matrix. Macro-Analysis is vital in terms of comprehending system scalability and functionality. The modules and their interactions influence the scalability of the system while the absence of certain modules within a system might indicate missing system functionality.

The Micro-Level Analysis studies the system at the lower component levels and primarily uses the Dependency Matrix representation. The connectivity metrics calculated from the Dependency Matrix quantify the relative centrality of each component in the system. Micro-Analysis reveals that within a system some components have many connections while others have very few connections and this information is very important in understanding their attributes and behavior. Connectivity metrics influence redesign/design efforts. Understanding how the redesign of a particular node will affect the entire system helps in planning and implementation. On the other hand design Modification/enhancement of nodes with low connectivity can be achieved without affecting the performance or architecture of the entire system. Identifying the highly central nodes also helps the architect understand whether the system has enough redundancy built in to withstand the failure of the central nodes. Potential system bottlenecks can also be identified by using the micro-level analysis.

## 7.2 MDMS ANALYSIS INSIGHTS

The framework was applied to the architectural analysis of two leading MDMSs. The approach identified the modularity of the individual architectures and determined the central nodes in each architecture.

The Macro-Micro Analysis was then used to compare the relative strengths and weakness of the two architectures based on their modules. The hypothesis was that each module performs a specific function. The absence of a module in a system signifies the absence of functionality or a limited implementation of that functionality in a particular architecture. Through these, the two MDMS architectures were compared by identifying the modules present in one of the architectures and absent or minimized in the other.

For example, the analysis revealed that Product 1 has the better data processing capabilities. In turn, Product 2 had better external interfacing capabilities. Knowing this, the firms can exploit each other weakness to make better sales pitches while at the same time investing resources in fixing their deficiencies vis-à-vis their competitor. The deficiencies can be overcome via internal R&D or external collaboration. On the other hand the analysis revealed that for both P1 and P2 the Report Generator module is an important module in the MDMS because of the high number of reports generated within the system. Hence the design of the Report Generator module should be able to handle this load so that it does not become a performance bottleneck.

Similarly by combining the strengths of the two architectures a generic MDMS architecture was proposed. This architecture delivers a more comprehensive functionality, because it combines the features of both the MDMS architectures. The generic MDMS architecture has the following modules:

- Data Input Module
- Configuration Management
- Meter Usage Database
- Validation, Estimation and Editing Engine
- Report Generator
- Universal Calculation Engine
- Billing Module, Security
- Data Access Module
- Alarm System
- Application Module
- MDMS Management Module

These modules deliver the core central data repository functions, while at the same time providing value-added services such as complex processing power, legal compliance and change management thus increasing the value proposition of the MDMS for the utilities.

Once the generic architecture was synthesized areas of collaboration were proposed based on the generic architecture and the Macro-Micro understanding of the system. The potential areas of collaboration are as follows:

- Data Loading
- Data Storing
- Data Manipulation
- Data Security
- Data Leveraging
- System Management

The firms with core competencies in the above areas should position their solutions to align with and take advantage of the MDMS architecture. The first step to collaboration is to understand the MDMS architecture which the framework proposed in this thesis can provide.

### **7.3 FUTURE RESEARCH**

This thesis focused on the object-object relationships of the OPM. Future research can also focus on object-process relationships and process-process relationships. This would solidify the conclusions of this thesis's research and possibly draw further undiscovered conclusions.

The second area of research could be to use a weighted Dependency Matrix; the weights in the matrix will signify the strength of the relationships between the nodes. For e.g. the relationships between the objects of the MDMS could be weighted based on the amount of data flowing between the nodes or the frequency of the data flow. A weighted Dependency Matrix has more information and the analysis will be richer.

Analysis is only as good as the data. This thesis relied on white papers and online documentation to understand the two MDMS architectures. Access to proprietary information of the MDMS would have allowed performing a more in-depth research and have higher confidence in the results. However, this thesis is a public document. It is understandable that companies would not want to divulge proprietary information and seek to protect their intellectual property. To a large extent by analyzing two separate systems we have greatly increased the confidence of the results in the framework proposed.

The System Level OPD was transferred to a Dependency Matrix for analysis purposes. This transformation results in loss of information that was captured in the OPD such as specific object-object relationships that deliver a particular functionality. Analysis methodologies that capture all the OPD information need to be investigated for generating results without any loss of information from the OPD representation.

## REFERENCES

- (1) Yassine AA, Wasserman S. The Implications of Product Architecture on the Firm. *Syst. Eng.* 2007;10(2):118-137.
- (2) Sosa ME, Eppinger SD, Rowles CM. Identifying Modular and Integrative Systems and Their Impact on Design Team Interactions. Available at:  
[http://web.mit.edu/eppinger/www/pdf/Sosa\\_JMD\\_June2003.pdf](http://web.mit.edu/eppinger/www/pdf/Sosa_JMD_June2003.pdf).
- (3) Ulrich K. The role of product architecture in the manufacturing firm. *Research Policy* 1995 5;24(3):419-440.
- (4) Gershenson JK, Prasad GJ, Zhang Y. Product modularity: definitions and benefits. *J.Eng.Des.* 2003;14(3):295.
- (5) Sosa ME, Eppinger SD, Rowles CM. A Network approach to define modularity of components in complex products. ; 2007. p. p. 1118-1129.
- (6) Sharman DM, Yassine AA. Characterizing Complex Products Architectures. *Systems Engineering* 2004;7(1):35-60.
- (7) Crawley E. ESD. 34 System Architecture. Lecture Notes. 2008.
- (8) The MIT ESD Architecture Committee. The Influence of Architecture in Engineering Systems. Available at:  
<http://esd.mit.edu.libproxy.mit.edu/symposium/pdfs/monograph/architecture-b.pdf>.
- (9) DORI D. Object-process Analysis: Maintaining the Balance Between System Structure and Behaviour. *Journal of Logic and Computation* 1995;5(2):227.
- (10) Dori D. Object-Process Methodology. : Springer; 2002.
- (11) Sharman DM, Yassine AA. Architectural Valuation using the Design Structure Matrix and Real Options Theory. *Concurrent Engineering* 2007 June 1;15(2):157-173.
- (12) Eppinger SD, Pimmler TU. Integration Analysis of Product Decompositions. Available at:  
[http://dandelion-patch.mit.edu/people/eppinger/pdf/Pimmler\\_DTM1994.pdf](http://dandelion-patch.mit.edu/people/eppinger/pdf/Pimmler_DTM1994.pdf).
- (13) C. I. G. Fernandez. Integration analysis of product architecture to support effective team co-location. Cambridge, MA: MIT; 1998.
- (14) A Genetic Algorithm for Developing Modular Product Architectures. Proceedings of the ASME 2003 International Design Engineering Technical Conferences, 15th International Conference on Design Theory & Methodology; 2003; Chicago, Illinois; 2003.
- (15) US Department of Energy. Available at:  
[http://www.oe.energy.gov/DocumentsandMedia/DOE\\_SG\\_Book\\_Single\\_Pages\(1\).pdf](http://www.oe.energy.gov/DocumentsandMedia/DOE_SG_Book_Single_Pages(1).pdf).
- (16) GTM Research. Available at: <http://www.gtmresearch.com/report/smart-grid-in-2010>.

- (17) EnergyICT: World leader in Energy Management solutions. Available at: <http://www.energyict.com/page.php?lang=EN&itemno=35>.
- (18) Viewpoint G. Electricity in NH - Bulk Electricity Markets Granite Viewpoint 2009 Wed, 11 Nov.
- (19) Meter Data Management, Advanced Pricing Programs - Utility Automation/Electric Light & Power. Available at: [http://www.elp.com/index/display/article-display.articles.electric-light-power.volume-87.issue-6.sections.meter-data\\_management.html](http://www.elp.com/index/display/article-display.articles.electric-light-power.volume-87.issue-6.sections.meter-data_management.html).
- (20) Newcomb PJ, Bras B, Rosen DW. Implications of Modularity on Product Design for the Life Cycle. ASME September 1998;120(3):483-490.
- (21) Lee HL. Effective Inventory and Service Management Through Product and Process Redesign. OPERATIONS RESEARCH 1996 January 1;44(1):151-159.
- (22) Lee HL, Tang CS. Modelling the Costs and Benefits of Delayed Product Differentiation. MANAGEMENT SCIENCE 1997 January 1;43(1):40-53.
- (23) Ulrich KT, Ellison DJ. Holistic Customer Requirements and the Design-Select Decision. MANAGEMENT SCIENCE 1999 May 1;45(5):641-658.
- (24) Yu JS, Gonzalez-Zugasti JP, Otto KN. Product Architecture Definition Based Upon Customer Demands  
. ASME September 1999;121(3):329-335.
- (25) Gonzalez-Zugasti JP, Otto KN, Baker JD. A Method for Architecting Product Platforms  
. Res. Eng. Des October 2000;12(2):61-72.
- (26) Planning for Product Platforms - The Magazine - MIT Sloan Management Review Available at: <http://sloanreview.mit.edu/the-magazine/articles/1998/summer/3942/planning-for-product-platforms/>. Accessed 4/22/2010, 2010.
- (27) Clarkson PJ, Simons C, Eckert C. Predicting Change Propagation in Complex Design. ASME September 2004;126(5):788-797.
- (28) Sosa ME, Eppinger SD, Rowles CM. The Misalignment of Product Architecture and Organizational Structure in Complex Product Development. MANAGEMENT SCIENCE 2004 December 1;50(12):1674-1689.
- (29) Saleh JH. Perspectives in Design: The Deacon's Masterpiece and the Hundred-Year Aircraft, Spacecraft, and Other Complex Engineering Systems  
. ASME September 2005;127(5):845-850.
- (30) Simon HA. The Architecture of Complexity. Proc.Am.Philos.Soc. 1962 Dec. 12;106(6):467-482.
- (31) Freeman LC. Centrality in social networks conceptual clarification. Social Networks 1979;1(3):215-239.

- (32) Linton C. Freeman. A Set of Measures of Centrality Based on Betweenness. *Sociometry* 1977 Mar.;40(1):35-41.
- (33) Sabidussi G. The centrality index of a graph. *Psychometrika* 1966 December;31(4):581-603.
- (34) Wasserman S, Faust K. *Social Network Analysis*. NY: Cambridge University Press; 1994.
- (35) Gebala DA, Eppinger SD, Sloan School of Management. *Methods for analyzing design procedures*. Cambridge, Mass.: Sloan School of Management, Massachusetts Institute of Technology; 1991.
- (36) Park H, Cutkosky MR. Framework for Modeling Dependencies in Collaborative Engineering Design Processes. *Res. Eng. Des* 1999;11:84-102.
- (37) Michelena N, Papalambros PY. A Network Reliability Approach to Optimal Decomposition of Design Problems. *ASME* 1995;117(3):433-440.
- (38) Chen L, Li S. Analysis of Decomposability and Complexity for Design Problems in the Context of Decomposition . *ASME* July 2005;127(4):545-557.
- (39) Ulrich K, Eppinger SD. *Product Design and Development*. Third ed. New York: McGraw Hill; 2004.
- (40) Baldwin CY, Clark KB. *Design Rules: Volume 1: The Power of Modularity*. Cambridge, MA: MIT Press; 2000.
- (41) Tradeoff Between Modularity and Performance for Engineered Systems and Products. *International Conference on Engineering Design*; 2005; Melbourne: ICED.
- (42) Peleg M, Dori D. Extending the Object- Process Methodology to Handle Real-Time Systems. *JOOP* 1999;11(8):53-58.
- (43) Reinhartz-Berger I, Dori D, Katz S. Object-Process Methodology for Developing Web Applications. *Ann. Softw. Eng.* 2002;13:141-161.
- (44) Newman MEJ. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* 2006 June 6;103(23):8577-8582.
- (45) Freeman LC. *The Development of Social Network Analysis: A Study in the Sociology of Science*. : Booksurge Publishing; 2004.