
Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By

Praveen Nagarajan, B.Tech.
Graduate Program in Computer Science
The Ohio State University
2010

Thesis Committee:
Dr. Rajiv Ramnath, Advisor
Dr. Srinivasan Parthasarathy
Dr. Jay Ramanathan
Copyright by
Praveen Nagarajan
2010
Abstract

To survive in today’s highly competitive, high demand business environment, management needs intelligence information to efficiently manage corporate operations and support decision-making, while support-level staff need better knowledge to provide better customer services for gaining satisfaction and retaining loyalty. Vast amounts of valuable operational data from which key insights may be drawn are scattered in various corporate databases and needs consolidating so that it may be accessed and shared among authorized users within a corporation and/or business partners. A system for integrated knowledge management and decision support is thus in great demand.

A synergy can be created by the integration of decision support and knowledge management, since these two processes involve activities that complement each other. The knowledge retrieval, storage, and dissemination activities for knowledge management enhance the dynamic creation and maintenance of decision support models, subsequently enhancing the decision support process. From the system design point of view, the challenge is that we need a new generation of knowledge-enabled systems that will provide an enterprise with an infrastructure to capture, cleanse, store, organize, leverage and disseminate, not only source data and information but also the knowledge or enriched information of the enterprise.
This thesis addresses this challenge through a real-life case study within a complex state government enterprise. In this thesis we develop and validate a conceptual framework for decision making based on the characterization of an enterprise as an ‘Adaptive Complex Enterprise’ (ACE). This framework addresses existing limitations by considering all the dimensions in an enterprise, viz. Business, Infrastructure, Operations, and Strategy. This framework introduces a Requirements-Execution-Delivery (RED) model to characterize the transactions in an ACE. The contributions of this thesis are as follows:

1) A new generation of knowledge-enabled operational management systems that provide enterprise with an infrastructure to capture, cleanse, store, organize, leverage, and disseminate not only source data and information but also the knowledge or enriched information of the enterprise.

2) An ACE ontology that provides the traceability among incoming Customer Service Request (CSR), the role sets processing them and performance.

3) ACE services ontology enhanced with Information Technology Infrastructure Language (ITIL) to show the complete set of collaborating objects in the processing of a CSR.

4) Leverage this ACE ontology by a tool set to analyze performance and make decision making easier. This ontology also forms the basis for a complete set of operational level performance questions for project management offices.
5) A visualization (dashboard prototype) to monitor the performance of an operational cyber infrastructure to enhance the basic balanced scorecard.
Acknowledgement

I would like to express my deep and sincere gratitude to my advisor Professor Rajiv Ramnath. His immense knowledge and critical thinking process have been of great value for me. I thank him for his patience and encouragement.

I also wish to express my warm and sincere thanks to Professor Jayashree Ramanathan for her continued support, insights and suggestions that helped to shape my research. I could not have imagined having a better mentor for my Master’s thesis.
Vita

2007..........................Bachelor of Technology, CEG, Anna University, India
2008..............................Program Analyst, Cognizant Technology Solutions

Field of study:

Major Field: Computer Science and Engineering
Table of contents

Abstract ........................................................................................................................................... ii
Acknowledgement............................................................................................................................ v
Vita .................................................................................................................................................. vi
Table of contents ........................................................................................................................... vii
List of figures ................................................................................................................................... x
Chapter 1: Introduction and Problem statement ........................................................................... 1
  1.1. Overview: ............................................................................................................................ 1
  1.2. Terminology: ........................................................................................................................ 1
  1.3. Introduction: .......................................................................................................................... 4
  1.4. Problem statement: .............................................................................................................. 5
  1.5. Approach: ............................................................................................................................. 6
Chapter 2: Background: Related Research and Best Practices .................................................... 7
  2.1. Overview: ............................................................................................................................. 7
  2.2. Enterprise architecture frameworks: .................................................................................. 8
  2.3. Best practices: Gartner, ITIL .......................................................................................... 13
  2.4. Existing frameworks and best practices: ............................................................................ 16
  2.5. Limitations of existing frameworks and best practices: ................................................... 19
  2.6. Traceability .......................................................................................................................... 20
    2.6.1. Need for BIOS Traceability for Alignment related Decision-Making ...................... 21
  2.7. Adaptive Complex Enterprise ............................................................................................ 23
    2.7.1. Introduction to ACE .................................................................................................... 23
    2.7.2. RED fractal view ......................................................................................................... 27
    2.7.3. RED contribution value to BIOS stakeholders: ...................................................... 28
  2.8. Existing decision support systems ...................................................................................... 33
    2.8.1. What is a decision support system? ............................................................................ 33
    2.8.2. Types of DSS: .............................................................................................................. 34
2.9. Limitations of the existing decision support systems ..................................................... 38
2.10. Project management tool – Clarity .............................................................................. 39
2.11. Limitations of the Clarity tool .................................................................................... 40
2.12. Data warehouse .......................................................................................................... 41
2.13. Schema integration ....................................................................................................... 42

Chapter 3: Enterprise Architecture Ontology case study for decision making .................... 44
3.1. Overview ....................................................................................................................... 44
3.2. CSR data ...................................................................................................................... 44
3.2.1. What can we interpret out of this data? ................................................................. 45
3.3. How is EA model going to help us better interpret the data ......................................... 46
3.4. Enterprise Ontology ..................................................................................................... 48
3.4.1. Business problem ................................................................................................... 48
3.4.2. Methodology for deriving Ontology ....................................................................... 51
3.4.3. Phase II ontology ................................................................................................... 57
3.5. Cyber infrastructure architecture: ................................................................................ 61

Chapter 4: Applications of the ACE EA ontology .................................................................. 66
4.1. Overview: ...................................................................................................................... 66
4.2. Process Consolidation .................................................................................................. 66
4.3. Performance Synthesis for decision making .............................................................. 67
4.4. Comparing across enterprises: ................................................................................... 72
4.5. Gaining a complete insight: ........................................................................................ 74

Chapter 5: Conclusions .......................................................................................................... 76

Chapter 6: Future research work ........................................................................................... 77

References .............................................................................................................................. 78

Appendix A: Performance views and dashboard implementation ........................................ 81
A.1. Performance questions and views: .............................................................................. 81
A.2. Mirror dashboard: ....................................................................................................... 83
A.3. ESB screenshot: ........................................................................................................... 84
A.4. File upload code snippet: ........................................................................................... 85
List of figures

Figure 1: Evaluation of existing best practices .................................................................18
Figure 2: BIOS dimensions ............................................................................................22
Figure 3: Sense and Respond .........................................................................................24
Figure 4: Generic RED transaction ...............................................................................26
Figure 5: RED transaction model ..................................................................................28
Figure 6: RED Service Interaction ...............................................................................31
Figure 7: ODJFS CSR data .............................................................................................45
Figure 8: Balanced score card (as in (22)) ....................................................................46
Figure 9: AS-IS ODJFS organization .............................................................................49
Figure 10: TO-BE ODJFS organization with four vertical pillars of roles .....................50
Figure 11: Phase I ontology ............................................................................................52
Figure 12: Interactions for a CSR ..................................................................................54
Figure 13: Validated ACE model ....................................................................................56
Figure 14: Phase II ontology ..........................................................................................57
Figure 15: Performance metrics ....................................................................................59
Figure 20: Dashboard prototype activity flow diagram ..................................................62
Figure 17: Operational cyber infrastructure database schema .......................................63
Figure 18: CSR with business system attribute added ....................................................67
Figure 19: Starlight visualization for CSR with business system attribute added ..........68
Figure 20: CSR with difficulty attribute added ..............................................................70
Figure 21: Starlight visualization for CSR with difficulty attribute added .......................... 71

Figure 22: Starlight visualization for comparison across ODJFS and City of Columbus .......... 73
Chapter 1: Introduction and Problem statement

1.1. Overview:

In this chapter, we start off in section 1.2 by describing the terminology that will be used throughout this thesis. In section 1.3 we give an introduction to the problem of enterprises not having a shared vocabulary, while section 1.4 summarizes the problem statement. Finally, we give an overview of the approach we take to solve the problem of not having a shared vocabulary, by using the enterprise architecture ontology in section 1.5.

1.2. Terminology:

Enterprise:

An enterprise is made up of one or more departments that come together for a business venture.

Enterprise Architecture:

Enterprise architecture (EA) describes the structure of an enterprise, its decomposition into subsystems, the relationships between the subsystems, the relationships with the external environment, the terminology to use, and the
guiding principles for the design and evolution of an enterprise. Enterprise architecture provides a holistic, systematic description of an enterprise. It encompasses business functions, business process, people, enterprise, business information, software applications and computer systems with their relationships to enterprise goals. The hope for enterprise architecture is that applying systematic rational methods to the design of an enterprise will produce one that more effectively and efficiently pursues its purposes (1).

Ontology:

Ontology is a term used to refer to the shared understanding, of some domain of interest, which is used to reduce or eliminate conceptual and terminological confusion and come to a shared understanding. Such an understanding can function as a unifying framework for the different viewpoints and serve as the basis for

1. Communication between people with different needs and viewpoints arising from their differing contexts,

2. Inter Operability among systems, and

Customer Service Request (CSR):

The CSRs are requirements that originate from multiple customer departments. The CSRs are the interface to the outside world. As the impact of the CSR is understood, individuals with specific skills are then assigned to the roles of the process to perform activities that research, use, create, review, and deploy the artifacts for the system enhancements.

The CSR typically starts a Life-Cycle Management (LCM) process to ensure that 1) functional and non-functional requirements of the change are identified, 2) implementation tasks are executed, and 3) deployment is assured with respect to the approved plans and monitored. Finally, the stakeholders must be satisfied that processes are effective and are providing value.

Cyber Infrastructure:

The term "cyber infrastructure" describes the new research environments that support advanced data acquisition, data storage, data management, data integration, data mining, data visualization and other computing and information processing services over the internet. In scientific usage, cyber infrastructure is a technological solution to the problem of efficiently connecting data, computers, and people with the goal of enabling derivation of novel scientific theories and knowledge (3).
Adaptive complex enterprise (ACE):

The ACE framework is a conceptual view of the complex interactions between services provided by any collection of enterprises and systems. It permits improved decisions without requiring detailed modeling of the enterprise.

Business-Infrastructure-Operations-Strategy (BIOS) dimensions:

An enterprise is made up of 4 dimensions namely Business, Infrastructure, Operations and Strategy. Each dimension groups the stakeholder Roles, their desired goal states and required RED Interactions to achieve those states. More formally, a goal state is a prescriptive statement of intent whose satisfaction in general requires certain Interaction metrics to be achieved through Agents. The BIOS goals also collectively ensure the enabling agent services are available across Request types and at the right cost to meet the external customer requirements. The different dimensional perspectives ensure all aspects of each single Interaction are addressed.

1.3. Introduction:

An enterprise has various departments and each department follows its own terminology for defining its IT services. In a shared-service organization, this may lead to challenges – there will be redundancy in work and requirements will be unclear to roles. If they wish to identify the department that is underperforming,
they need to have a consistent view of the entire enterprise. This is not possible if the individual departments collect varying information.

Also, an enterprise has various departments and each department is lead by a manager. The managers gather operational data from the day to day operations and collect information which they feel is critical to the decision making process. Hence these managers end up having isolated and unrelated data sets for each department, most of which do not follow the same ontology.

Similarly, each enterprise collects information following its own ontology. Consider an enterprise which collects information about its projects based on priority, and another enterprise which collects information based on complexity. If we were to compare the throughput of the enterprises, we are measuring the number of projects delivered/ unit time on different scales i.e. ‘A’ finished 20 high priority projects ‘B’ finished 20 complex projects. Are the two enterprises performing equally well?

1.4. Problem statement:

1. How do we interpret multiple departmental processes and data to make decisions regarding service consolidation by a shared service enterprise?

   • This is a challenging problem, e.g. 1600 processes in an enterprise

2. How do we compare the performances across enterprises each following a disparate ontology for operational data collection?

   • E.g. no shared vocabulary
1.5. Approach:

In this thesis, we suggest an approach to effectively interpret multi-departmental operational data using enterprise architecture ontology, by reflecting the entities and their associations which collaborate to process the CSRs in the operational data collected. To interpret the processes, we use ITIL and enterprise architecture ontology to standardize the terminology. To have a shared vocabulary within and across enterprises that provides a common backdrop for comparison of the data, we suggest an approach similar to the ETL (Extract, Transform, Load) process. As in ETL, we take the disparate ontology from the operational data, transform it using policies we define and load the data in this new ontology.

In section 2 we will introduce the existing best practices for Enterprise architecture, decision support systems and their limitations.
Chapter 2: Background: Related Research and Best Practices

2.1. Overview:

This section focuses on the topics that provide the foundation for the work presented in the thesis. Based on the business problem, this section looks at the various best practices that exist for understanding, managing and trying to improve complex enterprises. Section 2.2 looks at the enterprise architecture frameworks like Zachman and The Open Group Architecture Framework (TOGAF) and why it is useful for an enterprise. Section 2.3 looks at the best practices like Information Technology Infrastructure Library (ITIL) and how it strives to improve IT management in enterprises. Section 2.4 points out the deficiencies in these best practices and the need to incorporate traceability. Section 2.5 introduces the concept of BIOS dimensions and how it helps us achieve this traceability. Section 2.6 gives a brief overview of the ACE framework and its concept of RED interaction which has been the basis for the developing the enterprise architecture ontology to implement BIOS, used in decision making. Section 2.7 gives a brief overview of the various decision support systems currently in the market and Section 2.8 points out the limitations of these existing decision support systems. Section 2.9 introduces a project
management tool ‘Clarity’ and section 2.10 discusses the limitations of the project management tools and the need for an improved decision making tool. Section 2.11 introduces data warehousing and the problems inherent in data warehousing and section 2.12 introduces schema integration and how it is relevant to our research

2.2. Enterprise architecture frameworks:

Enterprise architecture framework is a strategic information asset base, which defines the mission, the information necessary to perform the mission and the technologies necessary to perform the mission, and the transitional processes for implementing new technologies in response to the changing mission needs i.e. it captures not just all the important characteristics of the IT Architecture - but all the possible kinds of relationships between Business Services, Business Processes, Information, Applications and the Technology and Infrastructure viewpoints.

Enterprise architectures are ‘blueprints’ for systematically defining an enterprise’s current (baseline) and/or desired (target) environment. *Enterprise architecture is a coherent whole of principles, methods and models that are used in the design and realization of an enterprise’s organizational structure, business processes, information systems, and infrastructure* (4). It is used to describe how different elements in an enterprise – systems, processes, enterprises, and people – work together as a whole. They are essential for evolving information systems
and developing new systems that optimize their mission value. This is accomplished in logical or business terms (e.g., mission, business functions, information flows, and systems environments) and technical terms (e.g., software, hardware, communications), and includes a sequencing plan for transitioning from the baseline environment to the targeted e-government environment.

Some of the advantages of having an EA are,

• It acts as a way to pass from chaos and disagreement to order and structure
• It enables an integrated vision and a global perspective of informational resources
• It enables the discovery and elimination of redundancy in the business processes reducing information systems complexity
• It contributes to having information systems that reflect common goals and performance measures for all managers, to encourage cooperation rather than conflict, and competition within enterprises
• It becomes the bridge between the business and technical domains

The EA frameworks like Zachman, The Open Group Architecture Framework (TOGAF) and Federal Enterprise Architecture (FEA) framework are discussed in brief below and their relationship and influence on our research strategy is discussed in Section 3.
The Zachman Framework provides a structured way for any enterprise to acquire the necessary knowledge about itself with respect to the Enterprise Architecture. Zachman proposes a logical structure for classifying and organizing the descriptive representations of an enterprise, in different dimensions, and each dimension can be perceived in different perspectives. These perspectives are represented in a two-dimensional matrix that defines along the rows the type of stakeholders and with the columns the aspects of the architecture. Each row represents a total view of the solution from a particular perspective.

In Zachman Framework the rows are described as follows (5):

- **Planner's View (Scope)** - The first architectural sketch is a "bubble chart" or Venn diagram, which depicts in gross terms the size, shape, partial relationships, and basic purpose of the final structure. It corresponds to an executive summary for a planner or investor who wants an overview or estimate of the scope of the system, what it would cost, and how it would relate to the general environment in which it will operate.

- **Owner's View (Enterprise or Business Model)** - Next are the architect's drawings that depict the final building from the perspective of the owner, who will have to live with it in the daily routines of business. They correspond to the enterprise (business) models, which constitute
the designs of the business and show the business entities and processes and how they relate.

- **Designer's View** (Information Systems Model) - The architect's plans are the translation of the drawings into detail requirements representations from the designer's perspective. They correspond to the system model designed by a systems analyst who must determine the data elements, logical process flows, and functions that represent business entities and processes.

- **Builder's View** (Technology Model) - The contractor must redraw the architect's plans to represent the builder's perspective, with sufficient detail to understand the constraints of tools, technology, and materials. The builder's plans correspond to the technology models, which must adapt the information systems model to the details of the programming languages, input/output (I/O) devices, or other required supporting technology.

- **Subcontractor View** (Detailed Specifications) - Subcontractors work from shop plans that specify the details of parts or subsections. These correspond to the detailed specifications that are given to programmers who code individual modules without being concerned with the overall context or structure of the system. Alternatively, they could represent the detailed requirements for various commercial-off-the-shelf (COTS) or components of modular systems software being procured and implemented rather than built.
• *Actual System View* or The Functioning Enterprise

Each perspective focuses attention on the same fundamental questions, then answers those questions from that viewpoint, creating different descriptive representations (i.e., models), which translate from higher to lower perspectives.

The six categories of enterprise architecture components, and the underlying interrogatives that they answer, form the columns of the Zachman Framework and these are (6):

• The data description — What
• The function description — How
• The Network description — Where
• The people description — Who
• The time description — When
• The motivation description — Why

**The Open Group Architecture Framework** or TOGAF divides the enterprise architecture into four categories – Business, Application, Data and Technical Architecture. It is more of an architectural process than a framework. The most important part of TOGAF is the Architecture Development Method which is a process for creating architecture. Viewed as a process, TOGAF complements Zachman. Zachman tells you how to categorize your artifacts, TOGAF gives you a process for creating them. TOGAF views the world of enterprise architecture as a continuum of architectures, ranging from highly
generic to highly specific. It views the process of creating a specific enterprise architecture as moving from generic to the specific (7).

2.3. Best practices: Gartner, ITIL

Gartner is an enterprise architecture practice that describes architecture as a verb and not a noun. This means that architecture is an on-going process of creating, maintaining and especially leveraging, an enterprise architecture that gives enterprise architecture its vitality. An architecture that is just a bunch of stiff artifacts that site together in a corner gathering dust is useless, regardless of how sophisticated your taxonomy is for categorizing those artifacts or how brilliant your process is that guided their development.

Gartner believes that enterprise architecture is about bringing together three constituents: the business owners, the information specialists and the technology implementers. If you succeed in bringing together these three groups and unify them behind a common vision that drives business value, then you have succeeded. If not, you have failed. Success is measured in pragmatic terms, such as driving profitability, not by checking off items on a process matrix. The two things that are most important to Gartner are where an enterprise is going and how it will get there. Any architectural activity that is extraneous to these questions is irrelevant. Gartner does not have a firm step-by-step process and depends on the Gartner consultant’s expertise as well as the enterprise. The consultant does not see his role as creating an enterprise architecture but helping the enterprise
institute a process for allowing an enterprise architecture to emerge and evolve from the business strategy. (7)

The **Information Technology Infrastructure Library (ITIL)** is a set of concepts and policies for managing information technology (IT) infrastructure, development and operations. ITIL is published in a series of books, each of which covers an IT management topic.

ITIL v3, published in May 2007, comprises five key volumes:

- **Service Strategy**: Service strategy is shown at the core of the ITIL v3.1 lifecycle but cannot exist in isolation to the other parts of the IT structure. It encompasses a framework to build best practice in developing a long term service strategy. It covers many topics including: general strategy, competition and market space, service provider types, service management as a strategic asset, enterprise design and development, key process activities, financial management, service portfolio management, demand management, and key roles and responsibilities of staff engaging in service strategy.

- **Service Design**: The design of IT services conforming to best practice, and including design of architecture, processes, policies, documentation, and allowing for future business requirements. This also encompasses topics such as Service Design Package (SDP), service catalog management, service level management, designing for capacity management, IT service continuity, Information Security,
supplier management, and key roles and responsibilities for staff engaging in service design.

- **Service Transition**: Service transition relates to the delivery of services required by the business into live/operational use, and often encompasses the "project" side of IT rather than "BAU" (Business As Usual). This area also covers topics such as managing changes to the "BAU" environment. Topics include Service Asset and Configuration Management, Transition Planning and Support, Release and deployment management, Change Management, Knowledge Management, as well as the key roles of staff engaging in Service Transition.

- **Service Operation**: Best practice for achieving the delivery of agreed levels of services both to end-users and the customers (where "customers" refer to those individuals who pay for the service and negotiate the SLAs). Service Operations is the part of the lifecycle where the services and value is actually directly delivered. Also the monitoring of problems and balance between service reliability and cost etc are considered. Topics include balancing conflicting goals (e.g. reliability v cost etc), Event management, incident management, problem management, request fulfillment, asset management, service desk, technical and application management, as well as key roles and responsibilities for staff engaging in Service Operation.
• **Continual Service Improvement (CSI):** The goal of Continual Service Improvement is to align and realign IT Services to changing business needs by identifying and implementing improvements to the IT services that support the Business Processes. The perspective of CSI on improvement is the business perspective of service quality, even though CSI aims to improve process effectiveness, efficiency and cost effectiveness of the IT processes through the whole lifecycle. In order to manage improvement, CSI should clearly define what should be controlled and measured.

ITIL uses a configuration management database (CMDB) which represents the authorized configuration of the significant components of the IT environment. Although repositories similar to CMDBs have been used by IT departments for many years, the term CMDB stems from ITIL. The CMDB is a fundamental component of the ITIL framework's Configuration Management process. A key goal of a CMDB is to help an enterprise understand the relationships between these components and track their configuration.

### 2.4. Existing frameworks and best practices

The existing EA frameworks and methodologies are in some ways similar but in some ways different from each other and are complementary. Any one methodology does not address all the issues needed for business-IT alignment. Each one has its strengths and weaknesses. There are various criteria that can be
useful in evaluating and comparing the EA frameworks and architecture practice. Some of these criteria are discussed below and the evaluation on the basis of these criteria is in the following table.

- **Taxonomy completeness:** This refers to how well you can use the methodology to classify the various architectural artifacts. This is almost the entire focus of Zachman. None of the other methodologies focus as much on this area.

- **Process completeness:** This refers to how fully the methodology guides you through a step-by-step process for creating an enterprise architecture. This is almost the entire focus of TOGAF with its architectural design methodology.

- **Reference model guidance:** This refers to how useful the methodology is in helping you build a relevant set of reference models. This is almost the entire focus of FEA.

- **Practice guidance:** This refers to how much the methodology helps you assimilate the mindset of enterprise architecture into your enterprise and develop a culture in which it is valued and used.

- **Maturity model:** This refers to how much guidance the methodology gives you in assessing the effectiveness and maturity of different enterprises within your enterprise in using the enterprise architecture.
• **Business focus:** This refers to whether the methodology will focus on using technology to drive business value, in which value is specifically defined as either reduced expenses and/or increased income.

• **Specificity of ontology:** This refers to the degree of fine grained level of abstraction to which the methodology can give a fine grained level of abstraction in the ontology. None of the existing frameworks focus on this area (7)

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Zachman</th>
<th>TOGAF</th>
<th>FEA</th>
<th>Gartner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taxonomy completeness</td>
<td></td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process completeness</td>
<td></td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Ref model</td>
<td></td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Maturity model</td>
<td></td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Practice guidance</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Business focus</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Specificity of ontology</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 1: Evaluation of existing best practices**
ITIL has its own limitations. It provides best practice guidelines as it is focused on installed infrastructure alone (8).

2.5. Limitations of existing frameworks and best practices:

As the table above illustrates, there is inadequate focus on business in most of the frameworks, only Gartner brings the business focus in the picture.

- The frameworks do not address the need for a representation of the enterprise itself even before we can talk about improvement of the enterprise.
- While there is emphasis is on ‘what’ needs to be done in the form of best practices and guidelines, there are inadequate details on ‘how’ to achieve this and where exactly to start from.
- There is a lack of traceability from Business goals to Information use to Operations to Strategy Stakeholders (important for prioritizing and consolidating requirements).
- This limits continuous improvement and decision-making.
2.6. Traceability

Traditional definitions of traceability and the focus of research have primarily been on the management of software life-cycle artifacts and the use of various underlying associations between artifacts and their attributes (8) (9) (10). One limitation of all this, from a Business-IT complex system perspective, is that traditional traceability research has focused mainly on the creational phase of software development. Recently, goal-centric traceability has received attention because it helps in dealing with non-functional Business-IT requirements (11). Recognizing the importance of Business-IT alignment, this, goal-driven requirements engineering has been used to abstract the important relationships between the multiple business goals and their impact on associations and actions as in (11). Requirements traceability itself, however, has other issues which has also prompted research identifying tool gaps and lack of emphasis on pre- Requirement specification traceability that is linking the requirements back to the business goals that prompted those requirements (12).

Other emerging work related to traceability of complex systems is in the context of ISO20000 (ITIL (13)) where the traceability of installed configuration items within the CMDB is critical to the IT services such as incident resolution, problem and change management, service and operations management. A complementary body of knowledge in TOGAF stresses the importance of traceability between the conceptual business, logical system, and physical
architecture layers to facilitate their continuous improvement (14). That is, the emphasis is on inter-dimensional traceability as discussed in (13).

2.6.1. Need for BIOS Traceability for Alignment related Decision-Making

The BIOS dimensions and the acronym arises from the cause-and-effect cycle illustrated in Figure below: Business sets goals that determine the Infrastructure that enables the Operations which executes the services Strategy. Each of these four perspectives or dimensions is extremely important in shaping the enterprise and each of these is inter-related to the other dimensions as shown below. While there are methodologies and tools that try to comprehend the relationships between all the elements in a particular dimension, there are inadequacies in linking the elements of one dimension to another dimension. For example, site monitoring tools help us understand how our servers and other IT systems are performing, that is, it helps us understand the infrastructure dimension. However, it does not relate or provides partial information about things like which infrastructure elements are related to which applications running on them or which business services and these correspond to which the fulfillment of which business goals, hence the link with business or operations is not adequately defined. To achieve this, it requires a capability to trace requests for services across the dimensions.
If BIOS traceability is made available, enterprises can also be armed with predictive knowledge that will allow them to manage by objectives. Specifically, the tool requirements for this traceability can be divided into 1) knowledge mining, and 2) Decision support for continuous improvement. The example simulations and knowledge mining discussed later will illustrate the advantages of integrating these tool features and traceability methods as applied to the Ohio Department of Job and family services helpdesk. The issue of traceability in the context of our business problem is even more crucial and challenging since we are now looking at achieving end-to-end traceability not just in the context of software components but rather achieving planning-and-execution traceability across the Business, Infrastructure, Operations, and Strategy (BIOS) dimensions of a Business-IT system to provide an improvement method.
The following section on ACE discusses a framework that attempts to provide traceability across these dimensions through the concept of RED interaction and its fractal nature.

2.7. Adaptive Complex Enterprise

2.7.1. Introduction to ACE

The need for continuous adaptation of businesses and enterprises today has necessitated a shift from strategies that eliminate variation to those that embrace variation and changing conditions in businesses. Thus, enterprises have to move towards becoming ‘Sense and Respond’ (S-R) enterprises rather than ‘Make and Sell’ ones (15). The ACE framework is a conceptual view of the complex interactions between services provided by any collection of enterprises and systems.

- Why is ACE effective?

- Provides us with a lens that focuses on dynamic variation and how customer - provider agents provide value when they react.

- Allows us to relate the agent interactions to goals that must improve though prescriptive actions.
• Permits improved decisions without requiring detailed modeling of the enterprise.

• Spans Business, Systems Engineering and IT silos of knowledge.

The technical and enterprise ACE framework complements the business case method with measurement and predictive tools for proactive leverage using IT (16).

The primary response or transaction loop of the S-R business unit begins by sensing the ecosystem as shown in figure below:

![Figure 3: Sense and Respond](image_url)
The sensing culminates in a request object (for example, an order, proposal request, or research requirements from the customer) that initiates the response processes within the business unit. The execution of the response commandeers the resources of the underlying infrastructure (shown in the lower half of Figure 1) to play roles that transform inputs and culminate in a deliverable object that satisfies the customer within the ecosystem. Within the business dimension, the transaction loop includes the prototypical Requirements and planning, Execution, and Delivery (RED) steps that incrementally add value to inputs.

A business unit handles different request types (for example, product change, process change, returns, request for quote, order, corrective actions, and new requirements) from external and internal customers. Each type of request has its own primary transaction loop and RED steps. In addition to the necessary inputs, the steps also identify roles such as customer, provider, sales, and logistics that must be assigned resources for execution. Active resources (such as humans or automated software) fill RED roles and become the agents that make decisions and execute these steps under some external policies and constraints to produce outputs. The final output is called a deliverable.

When each request arrives, needed resources from the underlying infrastructure are assigned to roles to execute the RED steps. The resulting agents complete steps by applying transformations to inputs and producing outputs (a deliverable being the final output).
The transformation tools and inputs are considered passive resources. Every incoming request causes interactions between the RED and business information objects (such as requests, other REDs, roles, resources, and deliverables) (17).

**Figure 4: Generic RED transaction**

During RED execution, the customer and provider roles make co-commitments and jointly agree to progress (illustrated by the red arrows in Figure 4, part A) to the next step. For example, filing the return request of a laptop purchase results in a RED execution. Here the ‘R’ step proposes a reimbursement plan based on warranty and defects. The proposal and the authorization by the customer form the co-commitment that causes the backoffice ‘E’ steps to replace the product. The next co-commitment is the ‘D’ or delivery by the provider, and the acceptance and payment by the customer. Note the roles for RED execution can be specialized. For example, the provider role for the first step could be the ‘product engineer’, but later the provider role can be ‘logistics’. In general, roles are careful characterizations of infrastructure capabilities provided by the business in order to execute and complete different types of REDs successfully. Roles can even be automated. Finally, each step has a
status—completed, executing, or to be started—as illustrated in Figure 5, part C.

2.7.2. RED fractal view

**RED** : RED is used to refer to the recursive (fractal) application of transaction loops and RED to refer to a single transaction loop. As illustrated by the fractal blueprint in Figure part B, the transaction loops are recursively applied by the control unit and within any of the R, E, or D steps. The resulting sub-RED enlists other resources in support of the primary deliverable, and so on. This provides the emergent (not predetermined) processing structure for coordinating and tracking resources needed for discovered conditions (17).
2.7.3. RED contribution value to BIOS stakeholders:

An enterprise is made up of dimensions. Each dimension groups the stakeholder Roles, their desired goal states and required RED Interactions to achieve those states. More formally, a goal state is a prescriptive statement of intent whose satisfaction in general requires certain Interaction metrics to be achieved through Agents. The BIOS goals also collectively ensure the enabling agent services are available across
Request types and at the right cost to meet the external customer requirements. The different dimensional perspectives ensure all aspects of each single Interaction are addressed. These BIOS perspectives are given next starting from business:

- **Business goals** refine external facing goals to ensure that business value is created by an optimal infrastructure capable of addressing different types of Requests and their Interaction requirements. This defines business value and priorities and used this to define budget allocations and investments.

- **Infrastructure-use goals** typically validate that all the workers that complete primary Interactions and sub-Interactions contribute value as efficiently as possible. Typical metrics that can be captured at the RED points include:
  
  - **Business-related:** As primary Interaction value delivered for the services used.
  
  - **Infrastructure-use-related:** As actual service and resource costs.
  
  - **Operations-related:** As throughput, service failure incidents, knowledge used for execution, and suggestions for improvement. These are often referred to as non-functional aspects of the system in software engineering.
• *Strategy–related:* As time to respond and requirements met in customers’ environment.

• *Operational goals* implement the specific provisioning for individual Interactions of a Request type to ensure that it meets business and strategy goals.

• *Strategy goals* are identified based on *customer* service needs and trends, new opportunities, and the competitive environment.

The following figure shows through a pit stop racing car example how the RED interaction relates to the BIOS goals and metrics for their achievement. The objective of the RED Interaction structure is to make explicit important transitions from all the perspectives and points of metrics collection. This is achieved as follows:

• *Customer perspective:* A primary Interaction begins with the *Request event* (for a particular type of Request) within the end-customer’s environment and ends with the related *service deliverable(s)* that adds value to the customer’s environment. During the Interaction, there are also many customer Roles (e.g. race car driver, sponsor, etc.) that review and participate as the service progresses.
Figure 6: RED Service Interaction

- Conceptual RED Interaction and performance perspective:

During the course of the Interaction explicit transitions between milestone events (represented as $\Rightarrow$) occur between the steps—Requirements, Execution, and Delivery. The Request is examined and the provider negotiates and understands the service requirements. Then both the customer and provider agree to proceed based on a formal/informal proposal or ‘agreement milestone’. For example, here the fuel provider would agree on the types of fuel combinations that could potentially be provided, based on weather conditions.
Next, transitioning from Requirements to Execution, the provider assigns Agents to the execution Roles to service the Request. The Interaction Roles (e.g. pit crew and fuel truck) together provide the capability to do the necessary set-up, execution, and completion of the service. At the same time the transformations create value from the customer’s perspective thus completing the next milestone. Simultaneously, feedback to the infrastructure results in future potential to improve the value. Finally the delivery in the customer’s environment in the third step results in provider’s compensation. In this case this includes compensating for the fueling and system check services.

In general, a service Interaction can be simple as in the case of delivering a product. Or it can be an Interaction like the order fulfillment, or the incident resolution, or even a ‘one-of-a-kind’ project. In these cases, Interactions often use sub-Interactions that are also initiated with a Request from an internal customer as discussed in the previous section.

- **Provider perspective:** The transitions between the milestone events are achieved using provider (and sometimes supplier or even customer) services. For each service Request type - say complaint type, or standard service type, or design type - handled by the enterprise, there is a RED service Interaction template with
required Roles that will need to be eventually filled by available
Agents with certain capabilities needed to execute the Interaction.

A RED Interaction further formalizes the meaning and semantics
of an Interaction. It takes a structured view of the Interaction between 1)
the customer-provider Role perspectives, 2) the Request and Deliverables
perspective, and 3) the infrastructure Roles and resources perspective. this
structure can be used to capture and monitor the events ‘black dots’ of the
Interaction in the above figure. For non-routine Requests typical of a
service enterprise, this allows us to look at an activity in a more nuanced
fashion and in particular make explicit its contributions to BIOS
stakeholder goals at these points. The metrics are also captured at the
milestones. (7)

Having discussed about enterprise architecture and the ACE
approach we are going to follow to develop the enterprise architecture for
ODJFS, we will now look at the existing decision support systems and
what kind of knowledge base they use to arrive at the decisions.

2.8. Existing decision support systems

2.8.1. What is a decision support system?

A Decision Support System (DSS) is a class of information
systems (including but not limited to computerized systems) that
support business and enterprise decision-making activities. A
A properly designed DSS is an interactive software-based system intended to help decision makers compile useful information from a combination of raw data, documents, personal knowledge, or business models to identify and solve problems and make decisions.

Typical information that a decision support application might gather and present are:

- an inventory of all of your current information assets (including legacy and relational data sources, cubes, data warehouses, and data marts),
- Comparative sales figures between one week and the next, projected revenue figures based on new product sales assumptions.

2.8.2. Types of DSS:

D.J. Power, enlisted the following as the different classes of Decision support systems (18)

- Model-driven DSS

  A model-driven DSS emphasizes access to and manipulation of financial, optimization and/or simulation models. Simple quantitative models provide the most elementary level of functionality. Model-driven DSS use limited data and parameters provided by decision
makers to aid decision makers in analyzing a situation, but in general large data bases are not needed for model-driven DSS.

Example: Dan Bricklin and Bob Frankston co-invented the software program VisiCalc (Visible Calculator). VisiCalc provided managers the opportunity for hands-on computer-based analysis and decision support at a reasonably low cost.

• Data-driven DSS

   In general, a data-driven DSS emphasizes access to and manipulation of a time-series of internal company data and sometimes external and real-time data. Simple file systems accessed by query and retrieval tools provide the most elementary level of functionality. Data warehouse systems that allow the manipulation of data by computerized tools tailored to a specific task and setting or by more general tools and operators provide additional functionality. Data-Driven DSS with On-line Analytical Processing (cf., Codd et al., 1993) provide the highest level of functionality and decision support that is linked to analysis of large collections of historical data.

   Example: John Rockart’s research stimulated the development of executive information systems (EIS) and executive support systems (ESS). These systems evolved from single user model-driven decision support systems and from the development of relational database products.
• Communications-driven DSS

Communications-driven DSS use network and communications technologies to facilitate decision-relevant collaboration and communication. In these systems, communication technologies are the dominant architectural component. Tools used include groupware, video conferencing, computer-based bulletin boards and web analytics.

Example: DeSanctis and Gallup (1987) defined two types of GDSS. Basic or level 1 GDSS are systems with tools to reduce communication barriers, such as large screens for display of ideas, voting mechanisms, and anonymous input of ideas and preferences. These are communications-driven DSS. Advanced or level 2 GDSS provide problem-structuring techniques, such as planning and modeling tools.

• Document-driven DSS

A document-driven DSS uses computer storage and processing technologies to provide document retrieval and analysis. Large document databases may include scanned documents, hypertext documents, images, sounds and video. Examples of documents that might be accessed by a document-driven DSS are policies and procedures, product specifications, catalogs, and corporate historical documents, including minutes of meetings and correspondence. A
search engine is a primary decision-aiding tool associated with a document-driven DSS

Example: Examples of documents that might be accessed by a document-driven DSS are policies and procedures, product specifications, catalogs, and corporate historical documents, including minutes of meetings and correspondence. A search engine is a primary decision-aiding tool associated with a document-driven DSS (Power, 2002). These systems have also been called text-oriented DSS

- Knowledge-driven DSS

Knowledge-driven DSS can suggest or recommend actions to managers. These DSS are person-computer systems with specialized problem-solving expertise. The "expertise" consists of knowledge about a particular domain, understanding of problems within that domain, and "skill" at solving some of these problems.

Example: Edward Feigenbaum created the DENDRAL expert system. DENDRAL led to the development of other rule-based reasoning programs including MYCIN, which helped physicians diagnose blood diseases based on sets of clinical symptoms.

- Web-based DSS

The World-wide Web and global Internet provided a technology platform for further extending the capabilities and deployment of
computerized decision support. The release of the HTML 2.0 specifications with form tags and tables was a turning point in the development of web-based DSS. This forms the basis of opinion mining, recommender systems.

Example: Power (1998) defined a Web-based decision support system as a computerized system that delivers decision support information or decision support tools to a manager or business analyst using a "thin-client" Web browser like Netscape Navigator or Internet Explorer. The computer server that is hosting the DSS application is linked to the user's computer by a network with the TCP/IP protocol.

2.9. Limitations of the existing decision support systems

As we see from the above decision support systems, most of them derive patterns and decisions from historical data. The modern day decision support systems are better equipped with an underlying ontology which is used as domain knowledge. But these DSS are constrained to domains of finance, crime and environment which use the ontology to perform time series analysis. After a thorough background research we understood the lack of any work in the field of enterprise ontology to support the decision making process of project managers.
But there have been some project management tools to aid the decision making. Let us consider an award winning project management tool Clarity (19).

2.10. Project management tool – Clarity

The CA Clarity Advanced Project Management (19) option helps reduce cost, improve efficiency and deliver greater value and project success. It helps you deliver more predictable project results and improve alignment of portfolios to your enterprise’s goals.

The benefits of using this tool are:

- Minimize impact of project change
  - Determine risks, issues and changes that might affect the project, document their characteristics and maintain a risk/issues/change register
  - Understand risks, issues and changes by analyzing trends and performing root cause analysis
- Improve project communication and collaboration
  - Distribute project related information to stakeholders to improve efficiency
  - Increase team collaboration to enable compressed decision timeframes
• **Improve project quality management**
  
  - Identify which quality standards are relevant to projects and determine how to satisfy them
  
  - Evaluate overall performance of projects regularly to provide confidence that projects will comply with relevant quality standards

Our approach concentrates on the ‘Improve project quality management’ feature and the relevance is pointed out in section 3.4.

### 2.11. Limitations of the Clarity tool

This tool relies heavily on user intervention. Users need to feed all the data in advance into the system which includes risks, costs and resources. The system applies algorithms like shortest path algorithm to find out the best way of allocating resources to different projects in order to maximize resource utilization.

The dashboard used in this tool gives us a status of all systems and how well they performing. It also identifies systems which are underperforming which are presented to the project managers for inspection. The project manager has to then assign Subject matter experts to find out the root cause of the problem. But it does not guide the process of root cause analysis and does not help answer the question of why those systems are underperforming.
2.12. Data warehouse

A **data warehouse** is a repository of an enterprise's electronically stored data. Data warehouses are designed to facilitate reporting and analysis. The essential components of a data warehouse are features to retrieve and analyze data, to extract, transform and load data, and to manage the data dictionary. An expanded definition for data warehousing includes business intelligence tools, tools to extract, transform, and load data into the repository, and tools to manage and retrieve metadata.

Data warehousing arises in an enterprise's need for reliable, consolidated, unique and integrated reporting and analysis of its data, at different levels of aggregation. The practical reality of most enterprises is that their data infrastructure is made up by a collection of heterogeneous systems. For example, an enterprise might have one system that handles customer-relationship, a system that handles employees, systems that handle sales data or production data, yet another system for finance and budgeting data etc. In practice, these systems are often poorly or not at all integrated and simple questions like: "How much time did sales person A spend on customer C, how much did we sell to Customer C, was customer C happy with the provided service, Did Customer C pay his bills" can be very hard to answer, even though the information is available "somewhere" in the different data systems. (20)

Our research is analogous to data warehousing – just as different ontology are merged into one in data warehousing, we are merging
different department-specific operational ontology using Enterprise Architecture ontology.

With this background, we can now move to the next section where we discuss our case study – Role Assignment for ODJFS, how we derived a solution for it and finally how it helped us develop a methodology for effective decision making.

2.13. Schema integration

Schema integration is the process of combining database schemas into a coherent global view. The Integration problems include:

- different data models
- incompatible concept representations
- different user or view perspectives
- structural conflicts within a model
- naming conflicts (homonym, synonym)

The idea behind schema integration techniques like RIM (Relational integration model) is that most (and probably all) schema conflicts can be resolved if we:

- eliminate all naming conflicts
- define a language capable of determining schema equivalence and performing transformations

The first task is eliminating naming conflicts:

- use a global thesaurus/dictionary like SSM
• map local schema names into global counterparts
• identical concepts can be identified by global name

The integration language must be defined:

• Capturing semantics of each LDBS in a machine-processable form
• Global names to identify concepts stored in LDBS

The concept of schema integration is closely related to our research, as we have used the enterprise architecture ontology combined with ITIL, as the global dictionary to map the local schemas into a global schema. (21)
Chapter 3: Enterprise Architecture Ontology case study for decision making

3.1. Overview

In this section, we introduce the operational data we received from Ohio Department of Job and Family Services (ODJFS) in 3.2. In 3.3, we look at how an EA model helps us better interpret this data. We then discuss how we came up with an ontology for ODJFS in 3.4, while section 3.5 describes the architecture of our operational Cyber Infrastructure.

3.2. CSR data

ODJFS is a public service organization which provides services like child care, unemployment compensation and many others. ODJFS has a MIS department which is responsible for creating and maintaining applications used to deliver these services.
Below is a snapshot of their CSR data:

<table>
<thead>
<tr>
<th>CLIENT_OFFICE</th>
<th>CREATE_DATE</th>
<th>PRIORITY</th>
<th>STATUS</th>
<th>ACTION_DATE</th>
<th>TITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHILD SUPPORT</td>
<td>1/22/2009</td>
<td>MEDIUM</td>
<td>IMPACT</td>
<td>1/6/2009</td>
<td>Female report for no new interventions in the last 12 months</td>
</tr>
<tr>
<td>CHILD SUPPORT</td>
<td>1/22/2009</td>
<td>MEDIUM</td>
<td>IMPACT</td>
<td>1/3/2009</td>
<td>Review Data Entry Screen (F0111) for CDHS 7715F</td>
</tr>
<tr>
<td>CHILD SUPPORT</td>
<td>1/16/2009</td>
<td>MEDIUM</td>
<td>CLOSE</td>
<td>2/3/2009</td>
<td>4047 to OSEB form P094 for Non IV-D caseload</td>
</tr>
<tr>
<td>CHILD SUPPORT</td>
<td>1/15/2009</td>
<td>HIGH</td>
<td>OPEN</td>
<td>1/16/2009</td>
<td>EMIS - CPA must have last names field populated</td>
</tr>
<tr>
<td>CHILD SUPPORT</td>
<td>1/15/2009</td>
<td>HIGH</td>
<td>CLOSE</td>
<td>2/4/2009</td>
<td>EMIS - populate Special support, use records</td>
</tr>
</tbody>
</table>

Figure 7: ODJFS CSR data

In the given dataset we have information about the client who initiated the request, the date on which the CSR was created, the priority of the request, the status, the date on which ODJFS MIS started acting on the CSR and a small description of the request.

3.2.1. What can we interpret out of this data?

We can see that some requests are closed, some are open and some high priority requests are also open. But why they are open? This can’t be answered out of this data because it uses a weak ontology which does not give the complete set of objects, and perhaps even more importantly the associations, used in the processing of a CSR, which inhibits us from
interpreting the operational data and identifying where things are falling through the cracks

3.3. How is EA model going to help us better interpret the data

Let us first consider a balanced scorecard as below:

Figure 8: Balanced score card (as in (22))

The balanced scorecard is a strategic planning and management system that is used extensively in business and industry, government, and nonprofit enterprises worldwide to align business activities to the vision and strategy of the enterprise, improve internal and external
communications, and monitor enterprise performance against strategic goals (22).

It was originated by Drs. Robert Kaplan (Harvard Business School) and David Norton as a performance measurement framework that added strategic non-financial performance measures to traditional financial metrics to give managers and executives a more 'balanced' view of enterprise performance. While the phrase balanced scorecard was coined in the early 1990s, the roots of this type of approach are deep, and include the pioneering work of General Electric on performance measurement reporting in the 1950’s and the work of French process engineers (who created the Tableau de Bord – literally, a "dashboard" of performance measures) in the early part of the 20th century.

The balanced scorecard has evolved from its early use as a simple performance measurement framework to a full strategic planning and management system. The “new” balanced scorecard transforms an enterprise’s strategic plan from an attractive but passive document into the "marching orders" for the enterprise on a daily basis. It provides a framework that not only provides performance measurements, but helps planners identify what should be done and measured. It enables executives to truly execute their strategies.
How to Apply the Balanced Scorecard:

- Align activities to a focused value proposition
- Bring in “forward looking” measures rather than backward looking financial metrics
- Include “intangible” assets in the strategy

A balanced scorecard is used for decision making as it points us to the areas to be improved. Here we develop an EA model that identifies operational associations similar to the balanced scorecard and uses it as a ‘backbone’ to 1) populate an operational ‘data warehouse’ and 2) visualize answers to the questions related to the balanced scorecard. We will later show a dashboard prototype which uses a EA model as its backbone.

3.4. Enterprise Ontology

Let me first introduce how we came up with an ontology for ODJFS and then discuss about how it helps us enrich the CSR data to help us better make decisions.

3.4.1. Business problem

Within ODJFS the MIS department had been structured in the form of bureaus (as shown in figure 10) where each bureau was responsible from cradle-to grave-operations for each project i.e. from requirements
elicitation to final delivery of the project. But the problems with this as-is approach were as follows:

- A resource not allocated to any task in a particular bureau often had expertise required by another bureau. However, the resource could not be allocated to that bureau because of accountability structures.
- Many IT services were inherently based on shared resources such as servers, the Help Desk, and exploration of new technologies.

<table>
<thead>
<tr>
<th>ODJFS</th>
<th>Work Force Development</th>
<th>Requirement Elicitation</th>
<th>Analysis</th>
<th>Design</th>
<th>Delivery</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Sharing, Training, Jobs and Opportunities Information</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Children and Families</td>
<td>Requirement Elicitation</td>
<td>Analysis</td>
<td>Design</td>
<td>Delivery</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Child Welfare System, Family &amp; Children Service Info system</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ohio Health Plans</td>
<td>Requirement Elicitation</td>
<td>Analysis</td>
<td>Design</td>
<td>Delivery</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Medicated Management Info and Tech system</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Child Support</td>
<td>Requirement Elicitation</td>
<td>Analysis</td>
<td>Design</td>
<td>Delivery</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Support Enforcement Tracking system</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unemployment Compensation</td>
<td>Requirement Elicitation</td>
<td>Analysis</td>
<td>Design</td>
<td>Delivery</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tax, Insurance, Employer Information</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Family Stability</td>
<td>Requirement Elicitation</td>
<td>Analysis</td>
<td>Design</td>
<td>Delivery</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Registry Information, Integrated Client Management System</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 9: AS-IS ODJFS organization**

In order to address these problems, ODJFS decided to move to a shared resource IT enterprise (as shown in figure 11). The IT sub-enterprises were restructured based on 4 pillars of capabilities. The pillars were 1) Architecture and Engineering (A&E), 2) Application
However this step towards a shared IT enterprise brought about the following problems:

- **Problem 1:** The understanding of the Requirements is uncertain, thus correct resources can only be identified after the impact is understood.
- **Problem 2:** Often combination skills such as business analysis and technology are required. A single resource does not have the expertise needed for all the different roles he/she is assigned to. This situation also weakens the definition and
sense of roles and people become unclear about their responsibilities and their boundaries.

- Problem 3: During the reorganization they collected a list of processes from the project managers. The project managers collected a total of 1600 processes they were handling. But as per Gartner an enterprise should only have a maximum of 250 processes to be successful. So they wanted to identify the actual processes out of the 1600.

- Problem 4: Lack of any automated system to monitor performance and to identify the underperforming systems

ODJFS engaged the CETI team

1. To apply the ACE architecture to develop the processes to address proper assignment of resources to roles based on subject matter expertise and skills.

2. To analyze the 1600 processes and identify the actual processes among them for correct role assignment to those processes.

**3.4.2. Methodology for deriving Ontology**

In order to address the role assignment issue and the 1600 processes classification we decided to use the ACE transaction model to achieve the traceability between the incoming CSRs and the final resources used to
perform it. Based on a series of interviews with the project managers, section leads, pillar leads we understood the process flow of a CSR within ODJFS MIS and as a result we arrived at the following ontology

**Figure 11: Phase I ontology**

The framework above was used to model how a CSR is serviced within the enterprise.

- ODJFS had department goals such as optimum cost, low response time and high value, which are specific refinements of business goals for each department.
- These are the guidelines that drive the business and the activities.
- The flow of events begins with a request entering the enterprise from the external customers or the internal customers.
These requests were available in the CSR documents.

We studied these requests and analyzed which systems will be impacted and what roles were needed to accomplish this. Depending on the specific request type we defined a set of interactions needed to accomplish the task. These interactions are called **Request Execution Delivery** (RED) interactions.

These interactions were provided to us in a workflow matrix. The document also specified the role set needed to perform an interaction. From these role sets we mapped the interactions to the 4 pillars. The enterprise chart further helped us map these pillars to specific pillar roles and further down to individual resources. This framework resulted in traceability so as to identify which requests are being serviced by which roles. The possible set of interactions enacted when a CSR is serviced by ODJFS MIS is shown below:
Each box in the workflow interaction diagram above represents an interaction resulting from an input and delivers an output enabled by a role set and constrained by a set of constraints.

The figure above depicts the various interactions which are possible when a CSR is resolved by the ODJFS MIS. First the CSR is logged and classified by the MIS. If the request is a simple fix which requires just the expertise from production and operations pillar it is
considered as a tier 1 processing. But if the request cannot be resolved by
the production and operations pillar it is processed by tier 2 which is either
architecture & engineering or application development pillar depending on
the type of request. If the request can’t be solved by the tier 2 expertise it
is passed on tier 3 processing which is done using the subject matter
experts. Finally if the request can’t be solved by tier 3 the vendor is
contacted which is the tier 4 processing. Any CSR reaching the tier 2 or
tier 3 level first passes through the impact analysis phase which is
performed by the project manager and the customer together and the type
of CSR constrains the way impact analysis is done. The output from this
interaction is the scope of the project which is input to the business case
interaction and it also constrains the design interaction. The business case
is used to decide the priorities and it is done once again by the project
manager and customer and is constrained by the cost, Return on
investment and effort. The output from this interaction is the priority
document which is input to the project charter interaction. The project
charter includes a mission statement, including background, purpose, and
benefits, a goal, objectives, scope, assumptions and constraints. This is
done together by all 4 pillars. The output from this is the SRS document
which is input to design interaction. The design interaction is facilitated by
the pillars expect P&O. After these the interactions implementation,
testing and production follow each requiring a role set to perform the
interaction.
But all these interactions are not enacted for every CSR. For example if the CSR is a code change, impact analysis is done and we directly jump to implementation, testing and production. So the subset of interactions executed for each CSR depends on the type of the CSR and exact requirements of the CSR. Hence we need a dynamic triage model which will classify requests based on their type and also look at the needs in the CSR, to identify the interactions needed. Once we know the interactions we can refer to our model above to determine which role sets are required for each interaction.

The initial framework formed Phase I of our approach and gave us the transactional ontology. In Phase II, we chose to use ITIL to standardize our terminology. We fine-grained our ontology to a more specific level of abstraction by defining an ontology of its own for each object in the Phase I framework, in order to encompass all possible objects. This is illustrated in the figure below:

Figure 13: Validated ACE model
3.4.3. Phase II ontology

The combination of ACE with ITIL provided two new results, one of which was a standard unified vocabulary based on ITIL. While working with ODJFS, we observed that a lot of the terms we used to define certain things differed in meaning to them and vice versa. Once we had this unified vocabulary, communication and documenting became much more effortless.
The other outcome was the following framework which depicts objects and their linkages to each other. This framework helps us trace an issue to an object and analyze the root cause of the issue. Furthermore, this depiction finally placed us in a position where we had enough detail so as to be able to define and execute performance management.

The need to incorporate ITIL and the need to categorize the customer facing CSR's in a fine grained fashion required us to further expand the ontology as follows:

- The customer initially places a Customer service request (CSR). The CSR is a specific instance of an item in the Customer Facing Catalog, which has an associated Service Level Agreement (SLA).
- Once the CSR is launched a set of RED interactions are triggered with Deliverables to the customer. The interactions require services from the Internal Service Catalog (for e.g. project management services).
- These services are monitored by the metrics of Service use. For the fulfillment of these services we need services from both human roles and Business system. The Business system is enabled by the Production Support Systems, Business Systems and Production Support Infrastructure. Each interaction is fulfilled by a Role set which is from different Pillars.
- This assignment is done as per the 'triage' Assignment Policy. Thus this framework depicts the traceability from the request to the
services used. We see from the framework above that these categories fall into the objects depicted.

- Each entity in the ontology has an associated set of performance measures. They are listed in the table below

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>Entity</th>
<th>Measure</th>
</tr>
</thead>
</table>
| Business     | Cost and Returns | 1. Cost  
2. ROI                                      |
| Strategy     | Customer    | 1. Customer satisfaction  
2. Priority proposed / Priority delivered  
3. Span time |
| Operations   | Interactions | 1. Complexity  
2. Quality (number of bugs)  
3. Throughput (Number of units/Time) |
| Infrastructure | Resources  | 1. Skill  
2. Availability  
3. Work = Effort/Services Delivered  
4. Personnel turnover |

**Figure 15: Performance metrics**

Now these performance measures allow us to use the ontology as a resource to formulate the complete set of performance questions. They are complete because we exhaust all possible combinations of the performance measures and related key questions. They are as below:

1. How does the resource availability affect the throughput?
2. How does the resource skill set affect the throughput?
3. How does the resource availability affect the span time?
4. How does personnel turnover affect the throughput?
5. How does the resource availability affect the quality?
6. How does the resource skill set affect the quality?
7. How does the personnel turnover affect the quality?
8. How does the resource availability affect the work?
9. How does the resource skill set affect the work?
10. How does the personnel turnover affect the work?
11. How does quality affect the span time?
12. How does throughput affect the span time?
13. How does complexity affect the throughput?
14. How does resource skill affect the customer satisfaction?
15. How does resource availability affect the span time?
16. How does resource skill affect the span time?
17. How does resource availability affect the customer satisfaction?
18. How does quality affect cost?
19. How does quality affect ROI?
20. How does resource availability affect cost?
21. How does resource availability affect cost?
22. How does customer satisfaction affect ROI?
23. How does span time affect cost?
24. How does span time affect ROI?
25. How does throughput affect cost?
26. How does throughput affect ROI?
27. How does resource skill affect cost?
28. How does resource skill affect ROI?
29. How does complexity affect cost?
30. How does complexity affect ROI?
31. How does personnel turnover affect cost?
32. How does personnel turnover affect ROI?
33. How does work done affect cost?
34. How does work done affect ROI?

Note the relationship of these questions to the balanced scorecard in Figure 8, where we consider the same 4 dimensions and how the measures associated with each dimension affects one another. Also note that these questions identify the quality standards and the relations between them just like the ‘improved project quality management’ feature of the ‘Clarity’ tool.

3.5. **Cyber infrastructure architecture:**

Our second claim was that an enterprise architecture model will equip us with a shared vocabulary that will help us compare across enterprises. In order to prove this claim we decided to implement a prototype dashboard for our cyber infrastructure ‘Mirror’. In the figure below we show the activity flow in the dashboard which will help us prototype the visualizations we need to analyze performance
We have a flex front end through which any enterprise can upload their CSRs in the excel file format. This excel file is transported to the server, where a stateless bean reads it and populates our mirror schema.

We have an ESB configured on a remote machine which polls the database for every 10 seconds, converts the database contents into XML which can be used to create visualizations in Starlight.

Thus we have the following as part of the dashboard prototype:

- Translation service to upload and translate operational data based on ontology
- Transport operational data from multiple departments, enterprises
- Starlight to visualize performance and provide decision support
  - Answers critical identified project management questions

Below is the database schema for the operational cyber infrastructure:
Figure 17: Operational cyber infrastructure database schema
The database description is as below:

<table>
<thead>
<tr>
<th></th>
<th>TABLE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Scape</td>
<td>Scape details, references to presence, owner/creator User, ad has multiple Scape Rules</td>
</tr>
<tr>
<td>2</td>
<td>Presence</td>
<td>A geographical location and orientation data holder</td>
</tr>
<tr>
<td>3</td>
<td>Constraint</td>
<td>Specifies the constraints that a rule can apply. This is to be modified as we write the rules in a rules engine instead of here. Probably, the rules engine could be populated by such a table.</td>
</tr>
<tr>
<td>4</td>
<td>Event</td>
<td>Holds the information about ‘event’</td>
</tr>
<tr>
<td>5</td>
<td>Event Rule</td>
<td>Specifies what rules to trigger in case an event occurs, and what consistency constraints should be applicable in the context of that event (To be modified when using with rule engines)</td>
</tr>
<tr>
<td>6</td>
<td>Scape Rule</td>
<td>Specifies what consistency constraints are applicable in the context of a scape</td>
</tr>
<tr>
<td>7</td>
<td>Asset</td>
<td>Any object (not user) in the mirror, has geo information, relation to other assets</td>
</tr>
<tr>
<td>8</td>
<td>Value</td>
<td>Stores the value of an asset owned by some user or group, in the context of a scape. (Note: the way a scape interprets the value may be different; but the actual ‘value’ of an asset is the property of mirror than any one scape, so it is consistent.)</td>
</tr>
<tr>
<td>9</td>
<td>AssetState</td>
<td>Maintains the state of an asset in different scapes. E.g. a Stadium might be used in one scape as a shelter. It might be ‘half-filled’, or a house may be ‘occupied’, a theater may be ‘full’, etc.</td>
</tr>
<tr>
<td>10</td>
<td>GeoRelation</td>
<td>Inter-relation (orientation and geography-wise) between assets</td>
</tr>
<tr>
<td>11</td>
<td>OwnedBy</td>
<td>Stores what Asset is owned by which user/group</td>
</tr>
<tr>
<td>12</td>
<td>User</td>
<td>Stores basic user information</td>
</tr>
<tr>
<td>13</td>
<td>Group</td>
<td>Group information – owner/creator/administrator</td>
</tr>
<tr>
<td>14</td>
<td>Membership</td>
<td>Stores the user-group memberships</td>
</tr>
<tr>
<td>15</td>
<td>UserProfile</td>
<td>Stores all the profile information of user</td>
</tr>
<tr>
<td>16</td>
<td>UserRelation</td>
<td>Relation between users (friend, spouse, neighbor, enemy, etc)</td>
</tr>
<tr>
<td>17</td>
<td>Role</td>
<td>Description of a responsibility that a user carries out, in context of some scape interaction</td>
</tr>
<tr>
<td>18</td>
<td>Interaction</td>
<td>Any correspondence between two users or user and the system.</td>
</tr>
<tr>
<td>19</td>
<td>InteractionContains</td>
<td>The assets that might be involved in an interaction</td>
</tr>
<tr>
<td></td>
<td>InteractionType</td>
<td>Type of interaction: e.g. Communication, upload, enhancement request, feedback, etc</td>
</tr>
<tr>
<td>---</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>20.</td>
<td>InteractionInfo</td>
<td>Stores what users taken part as what roles in which interaction, and under what scape context</td>
</tr>
</tbody>
</table>

The event table is what we will use in this thesis because a CSR is an event and we will use the event table to store the CSR data of different organizations. In the next chapter we will see how this schema gives us a shared vocabulary to compare across organizations.

Having described about how we came up with an Enterprise Architecture ontology we will now describe the applications of this ontology to process consolidation and decision making in the next section.
Chapter 4: Applications of the ACE EA ontology

4.1. Overview:

In this section we start off by describing the results of process consolidation using the Enterprise Architecture ontology in section 4.2. In section 4.3 we show how to use the Enterprise Architecture ontology for root cause analysis within an enterprise and in section 4.3 we show how we can use the ontology to compare across enterprises.

4.2. Process Consolidation

Once we arrived at the ontology we categorized the 1600 processes into each one of these objects based on the ITIL terminology. Through this categorization we discovered that most of the 1600 processes were not processes and they fell into categories like production system support, roles etc. Once this categorization was done

- We were able to cut down the number of processes to 234.
• We were able to identify the interactions needed for each of these processes which let us decide which pillars owned the processes, thus enabling us to perform easier role assignment.

4.3. Performance Synthesis for decision making

As we saw explained earlier the existing CSR dataset has a weak ontology. Now that we created an EA model and a complete set of performance questions, we will add some details from the EA model to the CSR dataset and see how it helps us answer the performance questions and how a stronger ontology is created for decision making.

First we decided to add the dimension ‘Business system’ which the target system against which a CSR is filed. We derive this knowledge by using the ontology which helps us find the entities involved in the CSR processing.

<table>
<thead>
<tr>
<th>CONT OFFICE</th>
<th>Business system</th>
<th>CREATE DATE</th>
<th>PRIORITY INITIATOR</th>
<th>STATUS</th>
<th>ACTION DATE</th>
<th>TITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHLD SUPPORT</td>
<td>SETS</td>
<td>1/6/2009 11:17</td>
<td>MEDIUM</td>
<td>IMPACT ANALYSIS</td>
<td>1/6/2009 12:40</td>
<td>Scenario report for repair intervention in the last 12 months</td>
</tr>
<tr>
<td>CHLD SUPPORT</td>
<td>SETS</td>
<td>1/6/2009 12:23</td>
<td>MEDIUM</td>
<td>IMPACT ANALYSIS</td>
<td>1/6/2009 12:40</td>
<td>Review Data Entry Screen (PG111) for ODHS7710P</td>
</tr>
<tr>
<td>CHLD SUPPORT</td>
<td>MTS</td>
<td>1/18/2009 8:27</td>
<td>HIGH</td>
<td>OPEN</td>
<td>1/18/2009 8:27</td>
<td>EMIE - CTA must have first name field populated</td>
</tr>
<tr>
<td>CHLD SUPPORT</td>
<td>BTSC</td>
<td>1/18/2009 8:54</td>
<td>HIGH</td>
<td>CLOSE</td>
<td>2/4/2009 14:05</td>
<td>EMIE - populate Special support case records</td>
</tr>
</tbody>
</table>

Figure 18: CSR with business system attribute added
The Starlight visualization for CSR records which are open in the dataset above is shown below:

![Starlight visualization for CSR with business system attribute added](image)

**Figure 19: Starlight visualization for CSR with business system attribute added**

As we see from the visualization, 3 business systems are not performing well namely ‘SETS’, ‘CRISE’, ‘IA’. Each business system is handled by one team in the ODJFS MIS. Having included this additional information we are now able to identify exactly which set of resources is under performing. So we have reached a point where we get to know where to look out for problems.

But this information is still insufficient. We are still left with the question as to why these Business systems are underperforming. At first glance the
possible causes for this below par performance (throughput) appear to be related to infrastructure measures (resource skill, availability, and personnel turnover) and operation measures (complexity)

Out of the above measures we decided to include complexity as one more attribute called the difficulty. The policy employed to define difficulty is:

\[ X = \text{Number of interactions} + \text{Number of roles} + \text{Routine/ Non Routine} + \text{Internal service catalog needed} \]

Now if \( X \) is lesser than 13 it’s a easy request

If \( X \) is greater than 13 and lesser than 20 it’s a medium request

If \( X \) is greater than 20 it’s a hard request
Below is the CSR with the ‘difficulty’ attribute added:

<table>
<thead>
<tr>
<th>Business system</th>
<th>Creation date</th>
<th>Action date</th>
<th>Priority</th>
<th>Status</th>
<th>Difficulty</th>
<th>CSR description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETS</td>
<td>20/2/2009</td>
<td>22/2/2009</td>
<td>MEDIUM</td>
<td>Impact Analysis</td>
<td>Medium</td>
<td>create report for no new interventions in the last 12 months</td>
</tr>
<tr>
<td>CASE</td>
<td>10/2/2009</td>
<td>22/5/2009</td>
<td>MEDIUM</td>
<td>Closed</td>
<td>Easy</td>
<td>prevent generation of JFS 4047 for ORES from FGS for Non IV-D cases</td>
</tr>
<tr>
<td>MB</td>
<td>22/2/2009</td>
<td>22/2/2009</td>
<td>HIGH</td>
<td>Open</td>
<td>Medium</td>
<td>CMS/MSA-Medical Support Adjustment</td>
</tr>
</tbody>
</table>

Figure 20: CSR with difficulty attribute added
The Starlight visualization for the above dataset is as below:

![Starlight visualization for CSR with difficulty attribute added](image)

**Figure 21: Starlight visualization for CSR with difficulty attribute added**

The yellow dots are the closed requests, red dots are the requests in progress and the blue dots are the open requests. As we see above most of the requests which are open are the easy and medium difficulty ones which eliminates one high level questions i.e. the requests are open not because they are complex which increases the probability of the other cause which is resources being insufficient.
Thus we see that as we include more information from the ontology into our analysis we are able to reduce the set of the probable causes of underperformance. This kind of decision making makes the job of decision makers easier. When the decision makers are equipped with an EA model and the complete set of questions, they can reduce the number of questions to look at using the EA model and thus their task becomes easier.

4.4. Comparing across enterprises:

Consider the following 2 data formats of the CSR for 2 different enterprises

- ODJFS (HELPDESK)

  CSR_ID|STATUS|PRIORITY|ACTION DATES

- CITY OF COLUMBUS (HELPDESK)

  CSR_ID|STATUS|ACTION DATES

As we see from the datasets, there are no performance attributes common across the datasets. So we can’t compare the performance. Relational database schema can also serve as ontology by specifying relations that can exist in some shared database and the integrity constraints that must hold for them. So we decided to map both the datasets to the database schema of our operational cyber infrastructure
(Mirror) as shown in the figure 17. This database schema provides us a common backdrop for comparison of data across enterprises.

We then mapped the data from the 2 enterprises to this common database schema shown in figure 17 and visualized it using Starlight and the results are as below:

**Figure 22: Starlight visualization for comparison across ODJFS and City of Columbus**

In the figure above, blue dots denote the closed requests and black dots denote the open requests. As we see from the visualizations, both the datasets are now in the same schema and now we can compare them.
The observations are as below:

- As we see City of Columbus IT helpdesk has a greater throughput.
- In City of Columbus IT helpdesk most of the unsolved requests were of high complexity whereas in ODJFS most of unsolved requests are of low complexity. When we investigated further we found out that City of Columbus did not perform well with non-routine requests and when the IT department received a non routine CSR they stored them in a separate list and these requests constituted the maximum percentage of the open requests.

Thus we can now ask the same set of questions we explained earlier to compare the performances of both the enterprises

4.5. **Gaining a complete insight:**

Thus we can see that when we include the performance measures into the CSR data we are now able to eliminate the performance measures which are not the root cause for underperformance. Once we include all the performance measures, collect data for them and visualize the data we will gain a complete insight into the performance of the enterprise. In the
appendix we show how these performance questions can be answered by visualizing the data using Starlight.
Chapter 5: Conclusions

1. Enterprise architecture ontology provides us a 'standard' lens to view the enterprise as a set of collaborating objects and the associations between them. As we have shown, this view helps us in the following ways:
   - Perform better role assignment to increase throughput
   - Combined with ITIL, gives us a shared vocabulary across the enterprise to avoid confusion and replication of work
   - Provides a complete insight about the enterprise based on performance questions

2. Operational data collected helps perform more effective root cause analysis. More specifically this helped us develop a roadmap for future actions within the organization

3. The ontology can be used to create a business roadmap, to identify current and potential business challenges. This includes two activities:
   - improving operational efficiencies
   - enhancing customer responsiveness
Chapter 6: Future research work

- Deploy this approach to collect operational data for all performance measures associated with each object in the ontology - this will allow us to answer all the performance questions
- Prioritize the performance questions to explore the high impact questions with precedence
- Automate the dashboard by creating an API to Starlight
References


7. Sessions, Roger. Comparison of the Top Four Enterprise Architecture Methodologies, EA Comparisons, ObjectWatch, Inc.


34. Hilary Cheng a, Yi-Chuan. An ontology-based business intelligence application in a financial knowledge management system.
Appendix A: Performance views and dashboard implementation

A.1. Performance questions and views:

How does the resource availability affect the span time

The previous visualization shows how resource availability affects span time

- A particular business system ‘CRISE’ which has very less resource availability and the span time taken to complete the incoming high priority CSRs
The previous visualization also shows how span time affects throughput.

- The requests which have a span time of 0 are the open requests (start date and end date are the same) and the requests having span time greater than 0 are the closed requests.

How does complexity affect the throughput?

The previous visualization helps us answer how complexity affects the throughput

- Here we see that complexity is proportional to throughput. Lower the complexity lower the throughput
A.2. Mirror dashboard:

The snapshot below shows the file upload feature of the dashboard. The load data tab shows the format of the CSR excel file that can be uploaded. Once the CSR has been formatted, we can use the file upload button to upload the CSR data to the mirror database.
A.3. ESB screenshot:

The snapshot below shows how the CSR fields are mapped to the mirror database schema in the ESB:
A.4. File upload code snippet:

Flex:

    private const FILE_UPLOAD_URL:String = "http://mirror.cse.ohio-
    state.edu/mirror/FileUploadHandler2";

    private var docFilter:FileFilter = new FileFilter("Documents",
    "*.xls");

    private function init():void {

        fileRef = new FileReference();

        fileRef.addEventListener(Event.SELECT, fileRef_select);

        fileRef.addEventListener(ProgressEvent.PROGRESS,

            fileRef_progress);

        fileRef.addEventListener(Event.COMPLETE, fileRef_complete);

    }

    private function browseAndUpload():void {

        fileRef.browse([docFilter]);

        uploadMessage.label = "";

    }

EJB:
HSSFWorkbook wb = newHSSFWorkbook(input);

HSSFSheet sheet = wb.getSheetAt(0);

Iterator rows = sheet.rowIterator();

while( rows.hasNext() ) {
    HSSFRow row = (HSSFRow) rows.next();

    Iterator cells = row.cellIterator();

    while( cells.hasNext() ) {

        HSSFCell cell = (HSSFCell) cells.next();

        switch ( cell.getCellType() ) {

            case HSSFCell.CELL_TYPE_NUMERIC:

                if(HSSFDateUtil.isCellDateFormatted(cell))

                    d[cnt1++]=cell.getDateCellValue();

                break;

        }

        e.setStartDate(d[0]);

        e.setEndDate(d[1]);
    }
}
A.5. ESB code snippet:

```java
String sqlStatement = "select * from event";

LinkedList<MessageExchange> sendList = new LinkedList<MessageExchange>();

NormalizedMessage inMsg = exchange.getMessage("in");

NormalizedMessageHandler nmh = new NormalizedMessageHandler(inMsg);

String sqlRequest = "<jdbc_request xmlns="http://cbesb.bostechcorp.com/jdbc/1.0">" +
    "<execute>" +
    "<statement>" + sqlStatement + "</statement>" +
    "</execute>" +
    "</jdbc_request>";

StreamSource src = new StreamSource(new StringReader(sqlRequest));

nmh.addRecord(src);

nmh.generateMessageContent();

sendList.add(exchange);
```
return sendList;

}

A.6. ESB component flow diagram:

The snapshot below shows the component flow in the ESB. We have a poll script which polls the mirror database housed in the mirror VM. The contents of the mirror database are converted to XML using the JDBC component and the XML file is written to the local directory using the file writer.

![ESB component flow diagram]

A.7. Starlight:

Future Point's Starlight Visual Information System™ is a comprehensive visual analytics platform that transforms mountains of data into actionable intelligence. We provide software for visually managing, understanding, and
deriving new knowledge from massive quantities of heterogeneous and complexly related information.

The Starlight software solution tames information overload and enables users to extract new value from data of all types. We can automatically organize and summarize large collections of information and present the results in easily interpretable graphical formats. In essence, Starlight is a “force multiplier”, enabling analysts to do the work of many with greater efficiency, accuracy, and higher levels of productivity. Using Starlight, analysts rapidly attain high-value results leading to faster time-to-decision. Below is a snap shot of Starlight:
A.8. User manual:

In order to run the file import utility open http://mirror.cse.ohio-state.edu/mirror/mirror-flex/mirror.html and click on Load Data tab. The tab displays the format of the the CSR file that can be imported. Click on the upload file button and select the excel file to be imported. Once it is selected the CSR records are copied to the mirror database on the mirror VM. Once the upload is complete, an alert box saying file uploaded flashes.

In order to visualize it using Starlight, logon to the Starlight VM. The Starlight VM has a ESB configured which pulls the contents of the mirror database and creates a XML file in C:\inetpub\ftproot. Open starlight and use the import wizard to select the XML file just created. Once the XML records are imported we can select the different views on the data.