Regis University
ePublications at Regis University

All Regis University Theses

Fall 2005

Integration of Interactive Voice Response Unit and Outage Management System

Chaomei Jiang
Regis University

MSC 696B

Professional Project

Integration of Interactive Voice Response Unit and Outage Management System

Version 1.5

August 2005

Chaomei Jiang

Regis University

School for Professional Studies

MSCIT Program

Certification of Authorship of Professional Project Work

Submitted to: Tim McKenzie

Student's Name: Chaomei Jiang

Date of Submission: August 8, 2005

Title of Submission: Integration of Interactive Voice Response Unit and Outage Management System

Certification of Authorship: I hereby certify that I am the author of this document and that any assistance I received in its preparation is fully acknowledged and disclosed in the document. I have also cited all sources from which I obtained data, ideas, or words that are copied directly or paraphrased in the document. Sources are properly credited according to accepted standards for professional publications. I also certify that this paper was prepared by me for the purpose of partial fulfillment of requirements for the MSC 696 or the MSC 696B course.

Student's Signature:

Regis University

School for Professional Studies

MSCIT Program

Advisor/MSC 696 and 696B Faculty Approval Form

Student's Name: Chaomei Jiang

Professional Project Title: Integration of Interactive Voice Response Unit and Outage Management System

Advisor's Declaration: I have advised this student through the Professional Project Process and approve of the final document as acceptable to be submitted as fulfillment of partial completion of requirements for the MSC 696 or MSC 696B course. The student has received project approval from the Advisory Board or the 696A faculty and has followed due process in the completion of the project and subsequent documentation.

ADVISOR

Name	Signature	Date
Michael R. Smith	Michael R. Smith	Aug 09, 2005

MSC 696 or MSC 696B Faculty Approval

Name	Signature	Date
Tim McKenzie	Tim McKenzie	August 9, 2005

Revision History

Revision	Date	Comments
0	May 10, 2005	Initial draft of MSC696B based on the final proposal of MSC696A.
1	May 30, 2005	Modifications to chapters 1 and 2.
2	June 15, 2005	Completed chapter 3.
3	July 5, 2005	Completed chapter 4.
4	July 30, 2005	Completed chapter 5 and reorganized the format of the initial part of the paper.
5	August 8, 2005	Completed the whole project paper.

Abstract

This project provides a technical solution that integrates the Interactive Voice Response Unit (IVR) and the Outage Management System (OMS) for the Public Service Company.

The IVR system is introduced because its automatic call logging functionality can process a large volume of phone calls from the customers reporting electrical service problems. This is a major improvement over the current Trouble Call Center (TCC) infrastructure that has a limited number of agents handling a relatively small number of customer calls. Because OMS can receive more calls, it then can predict the outage locations and causes more accurately, dispatch more repair crews on time, and restore outages quicker. As a result, the company benefits greatly on its customer relationship and revenue generation.

The goal of this project is to provide a simpler and less costly approach to the integration of IVR and OMS than the conventional integration approach provided by the IVR vender. The proposed solution utilizes the existing interface application between TCC and OMS. An adapter will be developed to communicate between IVR and the TCC-OMS interface. The adapter consists of a set of database tables and stored procedures. Compared with the vender's full integration approach from IVR to OMS, this proposed simple solution provides significant savings on the cost and time of the interface development as well as the system maintenance and upgrade.

Table of Contents

Table of Figures

1 Overview

1.1 Introduction

The Public Service Company provides electrical services to over a million customers. Its current TCC to OMS interface is not capable of handling a large volume of trouble calls during a storm situation. As a result, OMS cannot accurately predict the outage locations, crews cannot be dispatched on time, and the repair works are delayed extensively. Ultimately, the company takes hits on its customer relationship and suffers significant revenue loss. To improve the current call center infrastructure, the company is planning to introduce a new IVR system. However, because of the budgetary and scheduling constraint, the company cannot afford to implement the conventional integration approach provided by the IVR vender. It is looking for an alternate integration approach. In this project, a simpler design and development solution is proposed. The proposed solution utilizes existing TCC to OMS interface by building an adapter on a set of database tables and stored procedures that communicate with the IVR and the TCC to the OMS interface.

1.2 Existing Project Situation

At the Public Service Company, maintaining the quality of the electrical services to the customers is the number one priority of the company. TCC and OMS together serve this purpose. The customer representatives at TCC handle all customer calls reporting electrical problems. The trouble calls are sent to OMS for analysis. OMS predicts and locates the outage locations and causes. It arranges and dispatches crews for the repair and restoration. However, current TCC infrastructure can only handle a limited number of calls and a large volume of calls are blocked during a storm situation. The lost trouble calls affect the quality of the OMS predictions, resulting in delayed crew dispatching, repair and restoration.

To improve the call-taking capability, the company decided to implement a new IVR system. The IVR system can process a large volume of calls because of its automatic call taking and logging capability. But because of the tight budgetary and scheduling constraints, the company is struggling to accept the conventional integration approach provided by the IVR vender, which requires additional hardware/software purchases and a lengthy timeframe for development and implementation. As a result, the company is looking for an alternative approach to the IVR to OMS integration.

1.3 Goal of the Project

In this project, a simpler approach to the IVR to OMS integration is proposed. This approach provides following benefits:

a. It utilizes the existing OMS Oracle database server, eliminating the need for a designated hardware server and software packages for IVR.

b. It utilizes the existing TCC-OMS interface, eliminating the development of the IVR to OMS interface.

c. Architecturally, it is a simpler design, providing the benefit in every phase of the development, implementation, and maintenance.

d. It reduces the potential failure points and enhances the data quality.

e. It significantly shortens the development and implementation time to meet the scheduling constraint.

1.4 Barriers and Issues

This project was initiated by the client with specific requirements and concerns. The design process has to pay special attention to the project constraints in order to satisfy the client's need.

1.4.1 Limitation of OMS

One of the major concerns of introducing a new IVR system is that it may force the client to modify the current OMS. This may involve changes in the data model, control flow, and the interfaces from OMS to other systems. Because OMS is a mission-critical system, the client made it clear that the new IVR system will require none or minimal changes to the way the current OMS operates. Normally, an integration of two systems involves changes on both sides to achieve optimal performance. But this specific requirement puts the emphasis solely on the side of the IVR system, which really limits the selection of the IVR system and the interface design alternatives. Beyond of meeting normal system hardware, software, and functional requirements, the selected IVR system has to be flexible enough to be modeled into the OMS operation environment. On top of that, the design and development team has to have sound knowledge of both IVR system and OMS in order to carryout this specific integration task.

1.4.2 Budgetary Constraint

The client has stated that it has a very limited budget to support the integration development of this project. A major potion of the budget is allocated to hardware purchasing, deployment, and testing. It leaves little room for a lengthy interface development. This limits the initial design alternatives. A design with the best available technology and the most complete functionality may drive the project exceeding its budgetary limitation. So the design emphasis is focused more on the simplicity than on the functionality.

1.4.3 Scheduling Constraint

OMS is a mission-critical system and much of its heavy operations are weather related. The main purpose of introducing an IVR system is to enhance the OMS in its ability to manage the high outage volume during the storm season. A storm season typically starts in April and ends in November. During a storm season, the resources in the Trouble Call Center and the Operations Control Center are fully committed to support the daily operation. The development and deployment of the new IVR system has to be carried out during a non-storm season in order to get adequate support from the client. This only leaves a short time window of three months from December to late March to deliver the new IVR system. And even in a winter season, a snow storm may occur which may interrupt all regularly scheduled daily routines. So because of the unpredictability of the weather, it is expected to be a struggle in planning and managing the project development for dealing with frequent conflicts of scheduling and resource allocation.

2 Project Research

This chapter provides an overview of principle technologies and products used in this project with emphasis on the integration technology.

2.1 Outage Management System

OMS is a real-time and integrated software system that helps utility companies reduce restoration time, improve operational efficiency, and safeguard workers and the public. It supports trouble call management, outage analysis, operations dispatch, crew management, switching order development, safety documentation, reporting, and other critical operations. In addition to application software, OMS integrates multiple business information systems, such as Geographic Information System (GIS), Supervisory Control and Data Acquisition (SCADA), Customer Information System (CIS), and Work Management System (WMS).

2.1.1 Functional Components

OMS consists of following major components:

2.1.1.1 <u>Network Management</u>

Network management module offers basic capabilities of viewing, navigation, tracing, and operation on electronic distribution maps:

- <u>Loading and updating maps</u> - A map is extracted from GIS and loaded or updated into OMS. It may be an electronic network map containing conductors and electronic devices. It may also be a non-electronic map containing streets, buildings, and other land-based features. A map can be a new map or an existing map with the modified objects.

- Viewing and navigating maps - Maps can be viewed in various ways. They can be viewed in a real geographical mode or in a schematic drawing mode. Maps can be viewed with a selection filter to only view the selected objects by layer, type, pattern, and other attributes. Navigation includes zooming, panning, and searching.
- Tracing - Various tracing options can be performed on electronic maps: tracing by connectivity; directional tracing (upstream or downstream); tracing to specific types of devices; tracing by phases; isolated tracing between two objects; and tracing for special features, such as loops and meshed conditions.
- Operation - This includes: open or close a device; cut a wire; grounding; cross phasing connection; tagging; and installing jumpers.

2.1.1.2 Trouble Management

Trouble management module manages outages, unplanned trouble orders, and associated crews:

- Analyzing calls and outages - This includes viewing call and outage info, re-predicting outages upstream of downstream.

- Manage calls and outages - This includes acknowledging calls and outages, update call and outage conditions, confirming outages, restoring outages, outage reports, and post restoration repair jobs.
- Manage crews - This includes crew database maintenance, assign crews to outages, dispatching crews to outage locations, and crew job completion.

2.1.1.3 Storm Management

Storm management module produces accurate estimations for restoration times and crew staffing requirements for storm restoration:

- Setting estimation parameters - Estimation can be calculated by type of devices, by trouble causes, by geographic locations, by crew qualifications, by weather conditions, and by historical statistics.
- Managing jobs and resources - This includes setting the job priorities, allocating resources, tracking and reporting the restoration progress.

2.1.1.4 Switching Management

Switching management module prepares switching plans, validates the plans prior to execution, and executes the plans in real time:

- Design a switching plan in Study Mode - A switching plan includes detailed steps such as device operation (open or close), tagging a device, and issuing safety documents. Each step can be analyzed for its impact such as the number of affected customers and total load lost or gained.

- Executing a switching plan in Real Time Mode - Each step is validated and executed based on time and the projected electrical conditions. The planned outages during a switching plan execution are managed and reported.

2.1.1.5 Power Management

Power management module simulates, in real-time and in study sessions, the power flow, losses, and voltage profiles of the distribution system:

- Various analyses can be performed, such as load-flow analysis, short-circuit calculation, and protection calculation.
- Calculations can be performed in a study session for what-if analysis or in a real time mode to investigate current network conditions.
- Analysis reports and profiles are the bases for managing outage restoration and future enhancement to the network.

2.1.1.6 Performance Data Mart

Performance data mart provides access and reporting to historical and statistical data.

- It operates on a separate database with its data extracted periodically from the OMS operations database.
- It contains a set of near real-time data (typically with 5 minutes delay) to support the reporting of current outage conditions.

- It also contains a set of historical data to support statistical reports.
- All reports are accessed through a web server. The reports include internal reports for executive review and general reports for the public.

2.1.2 System Architecture

OMS utilizes client/server technology. A typical OMS infrastructure consists following system components:

- <u>OMS Computer Room</u> – All OMS server equipments are installed in this room, including the DB server (and/or the secondary DB server for high-availability and fail-over), one or more application servers (scalable to the client specification), and other Real Time Interface (RTI) systems such as SCADA.
- <u>OMS Center</u> – This is the control room where the OMS operators/dispatchers work. It installs the OMS application workstations, the radio system, and the weather reporting system. It is located in the same building with the OMS Computer Room on a local area network (LAN) connection.
- <u>Network Hub</u> – This is the bridge between the OMS Computer Room and the external systems with a T1 connection.
- <u>Call Center</u> – This is the room where the customer representatives respond to customer calls and enter trouble calls into OMS. It installs the call center application workstations and the phone system.

- Field Office – This is the remote location for operators/dispatchers entering emergency calls and conducting limited OMS functions. It installs the monitors for the NT-based OMS applications running on a network server based on a thin client connection.

This typical OMS infrastructure is illustrated in the figure below.

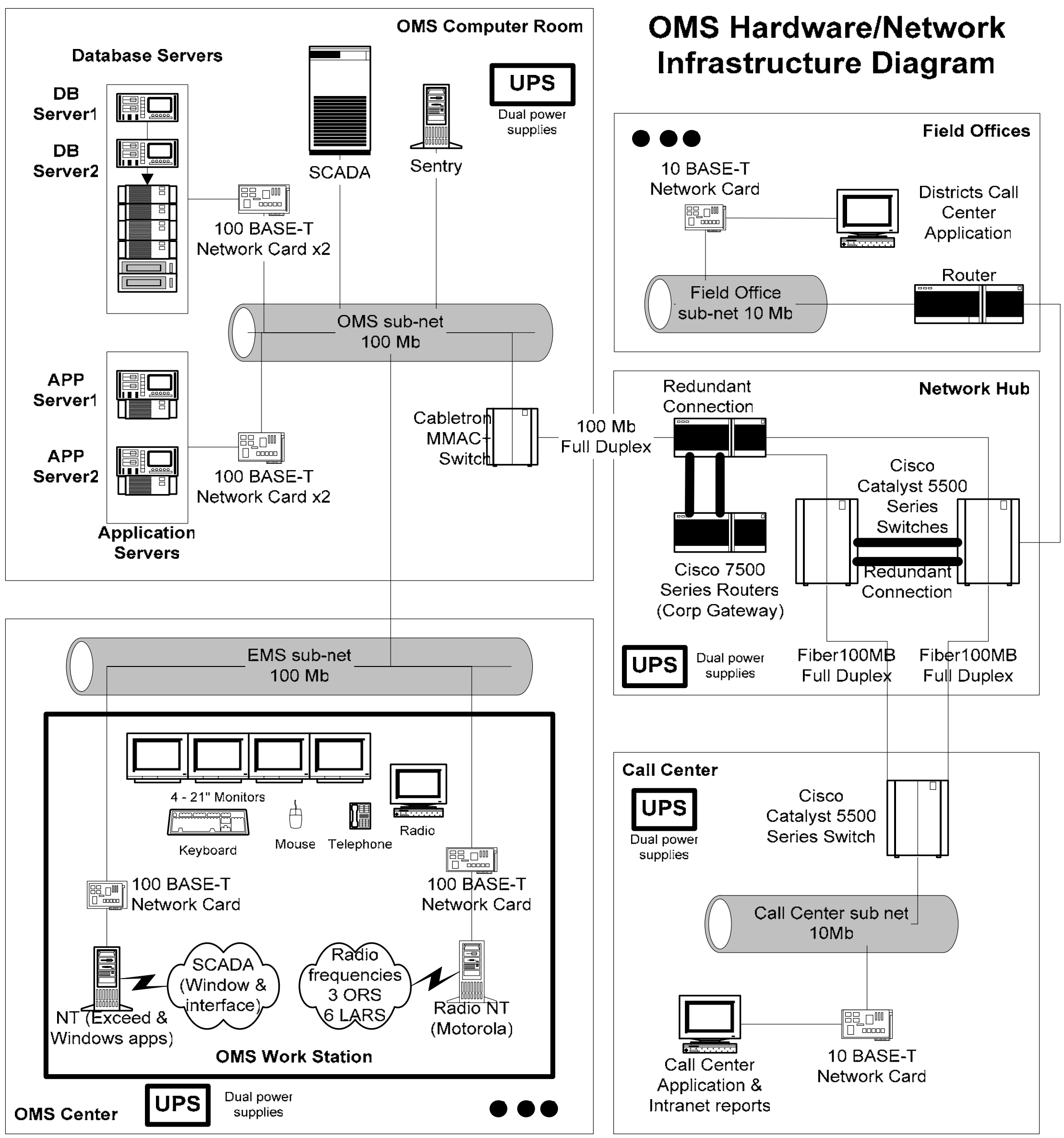

Figure 2-1 OMS Hardware/Network Infrastructure

2.1.2.1 Database Server

It provides data services to all client applications. An installation of Oracle 9i database on the server is required. It operates on all major UNIX platforms including:

- SUN Solaris
- HP/UX
- HP Tru64
- IBM AIX

2.1.2.2 Application Server

An application server hosts all client applications. It does not require any Oracle installation. It operates on all major UNIX platforms and Windows operating systems:

- SUN Solaris
- HP/UX
- HP Tru64
- IBM AIX
- Windows 2000/NT/XP (support thin clients)

2.1.2.3 Middleware Services

To support the information model in real time, OMS provides middleware services that cache key portions of the network model in memory and manage following transactions:

- Request to update or receipt of an event, such as a device status change from the SCADA system and a trouble call entry from the Call Center.
- Update the network and other data structures, such as updating a map from GIS and customer data from CIS.
- Validate and authorize the transactions, for example, if an operator is authorized to operate devices in a specific control zone, or if a device is tagged or issued with a safety document that prohibit any operating attempt.
- Update the appropriate information within the database instance(s), for example, changing the prediction rule set, or changing the operation mode from the normal mode to a storm mode.
- Broadcast an event notification to client process, such as dispatching a crew, confirming an outage, or restoring an outage.
- Reply with the acknowledgement or failure of the transaction ensuring the communication integrity of OMS with other systems.

2.1.2.4 Network Protocols

It is based on the TCP/IP protocol suite, including following:

- Internet Inter-ORB Protocol (IIOP) – Used for communication through Common Object Request Broker Architecture (CORBA) between Object Request Brokers (ORB) objects.
- Remote Method Invocation (RMI) – A Java architecture for remote communication that allows an object running in one Java Virtual Machine to invoke methods on an object running in another Java Virtual Machine.
- Hyper Text Transfer Protocol (HTTP) – A communication standard used to transfer data over the World Wide Web.
- SQL*NET – Oracle's client/server middleware product that offers transparent connection from client tools to the database or from one database to another.

Various OMS applications operate over LAN, wide area network (WAN), virtual private network (VPN), and Internet infrastructures.

2.2 OMS Integration Architecture

2.2.1 Integration Components

The figure below illustrates the software components involved in OMS integration:

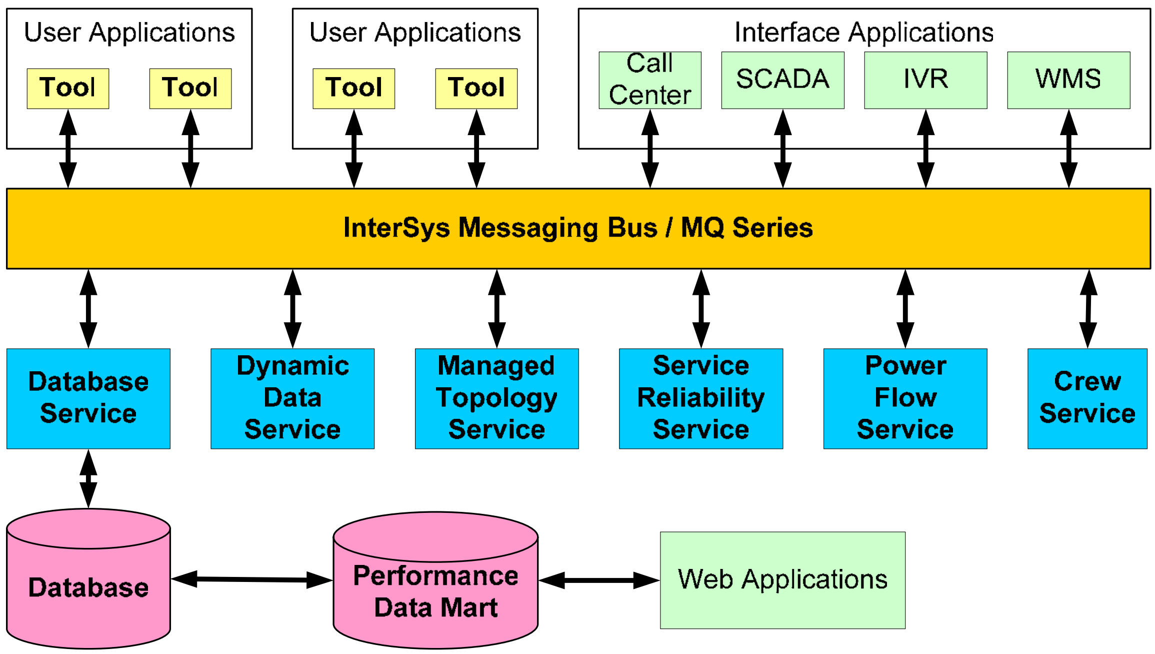

Figure 2-2 OMS Integration Components

2.2.2 InterSys Messaging Bus

InterSys Messaging Bus is the backbone of OMS. It provides the basis for the coordination and interoperation of processes within the system. It provides message transport using a reliable broadcast (multicast) protocol. The reliable messaging transport permits a sender to address a message to one or more receivers. This is accomplished by using a construct known as a process group as the destination address. Any process that is potentially interested in receiving messages of a particular type simply joins the appropriate process group. This reliable transport also ensures that all receivers of messages receive the messages in the proper order.

2.2.3 OMS Services

The server-side functionality of OMS is provided by a set of core services communicated through InterSys Messaging Bus.

2.2.3.1 Database Service

Database service provides access to a relational database management system (RDBMS). It supports following database transactions:

- Query tables and views (PL/SQL SELECT)
- Add records to tables (PL/SQL INSERT)
- Modify records (PL/SQL UPDATE)
- Delete records (PL/SQL DELETE)
- Execute stored procedures

2.2.3.2 Dynamic Data Service

Dynamic data service manages a complex data structure of operation network model. It provides following data requests:

- Provide different ways to identify a class and an object

- Receive and update object status information
- Receive and issue control messages, such as open/close, tagging, and issuing of safety documents
- Receive and update dynamic measurement values obtained from a RTI, such a SCADA system
- Provide optimized access and presentation to the network data

2.2.3.3 Managed Topology Service

Managed topology service provides the ability to efficiently perform topological analysis and network tracing over a very large operations model. It maintains following electrical network attributes:

- Connectivity – If the devices are electrically connected to each other
- Flow direction – Electricity flow direction from upstream to downstream, non-determinate direction as in a loop, or no direction as a dead section with no electricity
- Phasing – Any combination of the single phases A, B, C, and neutral
- Energization – Energized (has electricity) or de-energized (no electricity)

2.2.3.4 Service Reliability Service

Service reliability service is responsible for analyzing trouble calls (incidents) and predicting the outages on probable outage devices. It provides following functions:

- Manage a set of complex analysis rules that can be configured to achieve certain operational objectives
- Analysis rules can be adjusted based on the data model and operational preference
- Identify the trouble location of the trouble calls even from incomplete call data
- Identify the trouble causes and prioritize the trouble calls
- Analyze the call pattern, predict and re-predict outages to the most likely locations
- Maintain outage info and distribute trouble events to client applications

2.2.3.5 Power Flow Service

Power flow service is a complete calculation tool that analyzes the network conditions to prevent overloads and voltage violations:

- It can operate in real-time to analyze current network conditions
- It can operate in a study case for design and planning purposes
- Its algorithm is a full, unbalanced solution with individual modeling of phases
- It utilizes customer type, load profiles, and other information to properly distribute the load to each individual distribution transformer
- It supports all common types of transformers and switch-able capacitors

- It can simulates an operation to look ahead for potential overloads before the user opens or closes a device
- It supports the validation of switching planning sequences to prevent overloads
- Its optimization algorithm can be used to generates a set of optimized switching sequences during a switching planning

2.2.3.6 Crew Service

Crew service manages crew information, assignment, dispatch, and scheduling:

- It maintains a crew database consisting of team makeup, individual skill level, equipment, and regional information
- It manages crew workload and scheduling
- It provides accurate outage restoration time estimates based on crew quality of skill, geographic location, and work type
- It coordinates crew activities during normal operation and storm operation
- It consolidates operations centers to ensure crew responsiveness, material supplies, and equipment readiness

2.2.4 Applications Programming Interfaces

OMS supports a wide range of application programming interfaces (API) for the development of layered applications, integration of existing applications, and integration of systems through the use of its adapter processes. The supported APIs include:

- InterSys C++ Library – It defines a set of wrapper classes that provide access to the core functions of OMS services. This is the primary API used to develop customized applications that directly communicate with OMS services.
- CORBA IDL – An Interface Definition Language (IDL) that provides standards-based interoperability and connectivity. It enables distributed Web-enabled applications to transparently invoke operations on remote network services.
- Java 2 Enterprise Edition (J2EE) – A platform for building distributed enterprise applications. Its services are performed in the middle tier between the user’s machine and the enterprise’s databases and legacy information systems. It consists of a specification, reference implementation and set of testing suites.
- Enterprise Java Beans (EJB) – The core component architecture for J2EE. It enables rapid and simplified development of distributed, transactional, secure and portable applications based on Java technology.
- Procedural Language/Structured Query Language (PL/SQL) – An Oracle programming language that is used to write triggers and stored procedures that are executed by the Oracle Database Management System.

- E-mail – The e-mail interface is primarily used only for sending notifications of major events from OMS. It supports Messaging Application Program Interface (MAPI), Simple Mail Transfer Protocol (SMTP), and Post Office Protocol 3 (POP3).

2.2.5 OMS to CIS MQ Series Gateway

OMS supports integration with a Message Queuing (MQ) Series infrastructure via the implementation of a gateway process. The gateway exchanges InterSys messages with an MQ Series infrastructure. Within the MQ Series infrastructure, messages are formatted using Extensible Markup Language (XML). The MQ Series Gateway supports following type of transactions:

- Customer data updates from CIS
- Creating customer data from CIS
- Updating customer data from CIS
- Getting current customer data from OMS
- Customer outage status inquiries
- Receiving customer trouble calls from CIS and IVR
- Device tagging, notes or other conditions issued through OMS
- Database queries to the OMS operations database
- Database transactions to the OMS operations database, such as INSERT, DELETE, and UPDATE

- Event status updates, such as confirming and restoration of outages
- Customer history inquires, such as the number of outage occurrences and duration
- Customer electrical disconnects and reconnections caused by switching events
- Crew statuses, such as idle, assigned, dispatched, on site, and job completed

2.3 Customer Information System

Customer Information System (CIS) maintains customer data and provides customer care services including technical operations, personalized communications and sophisticated billing statements.

2.3.1 CIS Features

- Managing customer billing, credit, and payment
- Web services providing reports internally for executive review as well as public access by customers
- Customer and service location information connecting the phone and address info to the geographic and network locations
- Work orders including routine maintenance, temporary repair, emergency repair, follow up work, and service upgrade
- Rate structures providing alternatives to the customers based on both economic and non-economic factors to promote energy conservation, maximize financial benefit, and minimize conflicts
- Asset tracking solution provides a foundation for lifecycle asset management and creates the framework for tracking the movement, location , ownership, and cost of assets
- Call center application supports both voice and data access for trouble call logging, outage status tracking and reporting

- Financial reporting provides tools for analyzing the asset management, energy management, and customer relationship management
- Historical data provides comprehensive customer information on billing and payment history, outage history, and facility maintenance history
- Meter reading especially Automatic Meter Reading (AMR) enhances accuracy in billing, energy consumption planning, and outage confirmation
- Interfacing with 3rd party software such as OMS and WMS
- Database independent, support multiple database platforms:
  - ✓ Oracle
  - ✓ SQL Server
  - ✓ DB2
  - ✓ Sybase
  - ✓ Informix
- Operating systems supported:
  - ✓ IBM OS/400
  - ✓ Sun Solaris UNIX
  - ✓ HP Tru64
  - ✓ Linux Advanced Server
  - ✓ IBM AIX
  - ✓ MS Windows 2000/NT/XP

- Multi-language capable and multi jurisdictional support for non-English speaking customers

2.3.2 Integration Components

CIS provides a structured integration framework. This framework consists of two levels of integrations: the integration between layers (group of components) and the integration between individual components. The figure below illustrates the CIS integration component framework:

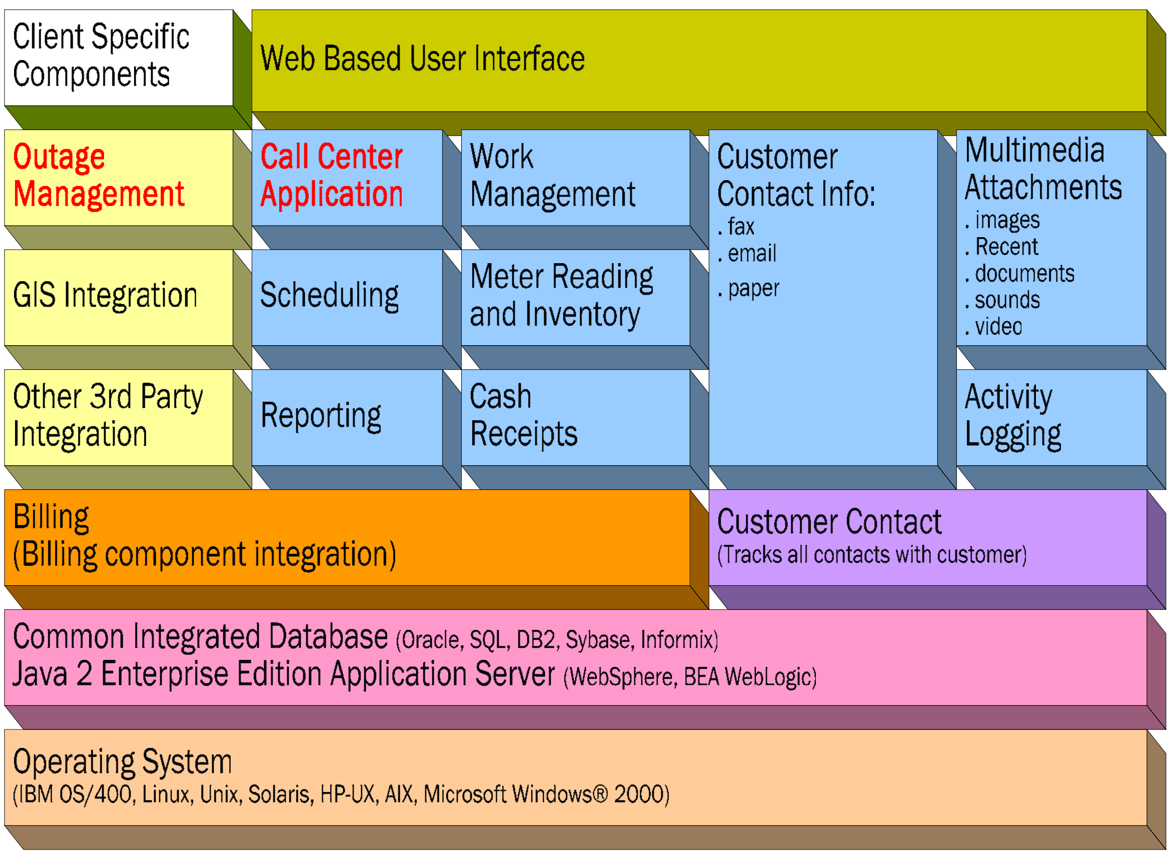

Figure 2-3 CIS Integration Components

The integration between Call Center Application and OMS is the focus of this project. The primary integration between the two systems is based on a MQ Series infrastructure. In this integration infrastructure, a gateway process is constructed as an integration messenger that handles the data exchange in XML format and support following communication models:

- One-to-one synchronous (request/reply). For example, Call Center Application sends a query to CIS database to identify a customer by phone number.
- One-to-one asynchronous (message queuing). For example, Call Center Application enters trouble calls into the OMS call queue.
- One-to-many asynchronous (publish and subscribe). For example, OMS broadcasts the outage status changes that can be subscribed by various CIS applications.

Another integration infrastructure between Call Center Application and OMS is based on the PL/SQL applications that support direct database transactions:

- Queries to the CIS database and the OMS database
- Stored procedures that updates the tables on both systems
- Incremental updates between the two systems through database triggers and replication processes
- Bulk updates by database import and export utilities

2.4 Interactive Voice Response Unit

Interactive Voice Response (IVR) unit is an automatic toll free phone answering system. It collects useful information from a caller reporting electricity interruptions or problems before the call is transferred to OMS for outage analysis or to a Call Center agent for further assistance.

2.4.1 IVR Software Features

- Automatic caller identification from phone number to customer record stored in database
- A scripting language for users to customize the process control flow
- Multiple IVR scripts to handle various caller inputs and responses
- Play message extracted from databases and the internet
- Database access (Oracle and SQL Server)
- Phone key input instead of voice input to ensure accuracy
- Text To Speech software (TTS) uses voice synthesis to convert text into spoken audio
- Multi-lingual support (English and Spanish)
- User touchtone navigation and selection of a set of input options
- Play music on hold
- Record caller message in database

- Route a call to a Call Center Agent when the call cannot be automatically accepted, for example, the phone number does not match with any customer record, or the caller reports an emergency
- IVR response logging and reports
- Operating systems supported: Windows, Unix, and Linux

2.4.2 IVR Hardware Features

- Intel/Dialogic telephony cards
- Client/server architecture
- Multiple digital T1 support (24 - 1000 lines)
- Analog solution (1 - 24 lines)
- Analog and digital phone line support
- 800 number systems and toll free systems
- Dialed Number Identification Service (DNIS) controlled

2.4.3 IVR Integration Architecture

The IVR system operates in a client/server environment. The IVR phone systems can reside on the same network or on a system that resides on the Internet. The IVR software program can run on the same server that contains the data and application programs. The figure below illustrates an IVR integration network infrastructure:

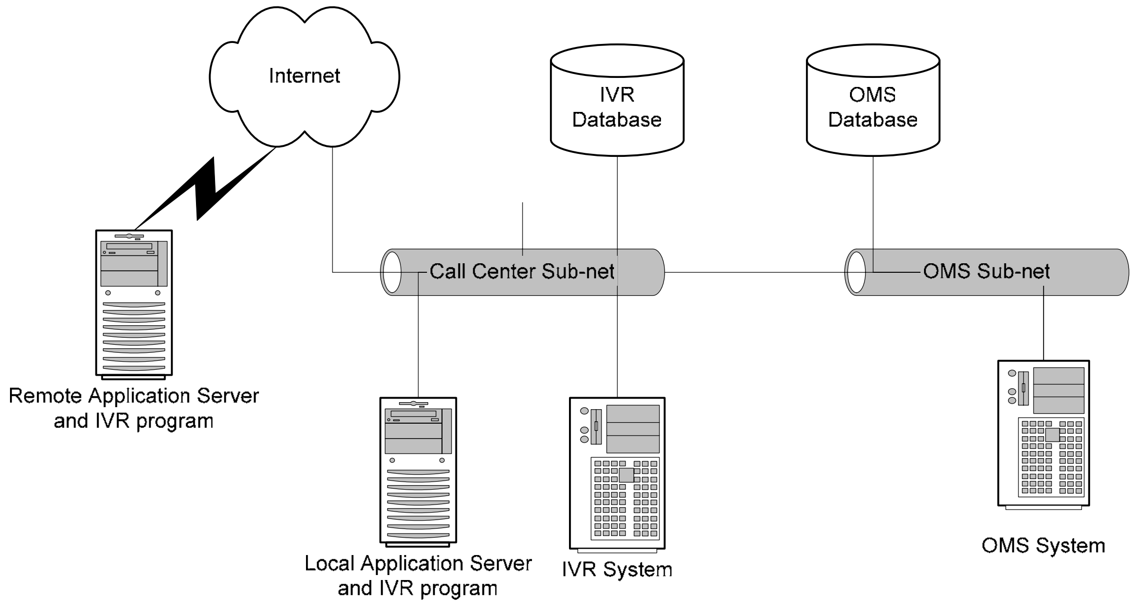

Figure 2-4 IVR Integration Architecture

The IVR software supports following programming APIs for developing IVR applications:

- A set of IVR C, C++ library – It defines a set of wrapper classes that provide access to the core IVR functions. This is the primary API used to develop customized applications.
- COBOL IVR API – A programming interface for Common Business Oriented Language (COBOL) for IVR implementation in IBM mainframes.
- Scripting Language - An internal simple scripting language for writing applications running on Windows, UNIX, or Linux. Scripting is primarily used to configure the IVR system and provide limited high level customization.

- Database applications – It provides a framework supporting user developed database applications and procedures that access its system tables.

3 Project Approach and Methodology

3.1 Systems Development Life Cycle

For this project, a variation of the Waterfall Model is adopted as the system development methodology. This is a systematic and orderly approach to the Systems Development Life Cycle (SDLC). This approach consists of five steps: Planning, Analysis, Design, Implementation, and Support. This is illustrated in the following diagram:

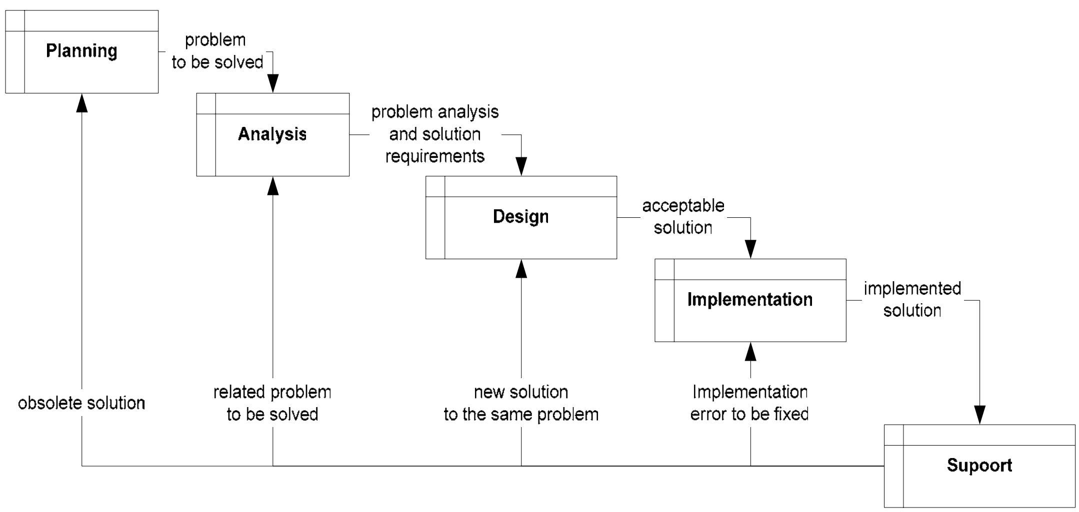

Figure 3-1 System Development Life Cycle

The key tasks and deliverables for each step of the SDLC are summarized below:

3.1.1 Planning

Goal: Identify the scope and boundary of the problem, and plan the development strategy and goals.

Activities: Conduct meetings and interviews with key users and managers to gather information. Identify and establish project direction and priorities.

Input: User's perspective on the corporate mission, vision, and goals.

Output: Feasibility analysis, management plan, and project plan.

3.1.2 Analysis

Goal: Define business requirements and functional requirements.

Activities: Conduct workshops with key users to gather business requirements. Consolidate and document the information. Analyze the problem domain and specify the requirements for a successful solution.

Input: Problem descriptions, causes, and effects.

Output: System integration requirements document.

3.1.3 Design

Goal: Establish and maintain a blueprint of the solution to be approved for implementation.

Activities: Identify data elements, data structures, data stores, and data flows. Establish data controls and processes. Identify and estimate the resource requirement. Specify the criteria for system acceptance testing.

Input: System integration requirements.

Output: System design document, project timeline, and resource estimate.

3.1.4 Implementation

Goal: Construct a successful integration system in time and under the budget.

Activities: Organize and manage the development team. Conduct coding and testing of the subsystems and integrated system. Document implementation issues and produce system reference manuals. Install the final system.

Input: System design details and project timeline.

Output: Installed integration system, acceptance testing document, and system reference manuals.

3.1.5 Support

Goal: Provide system maintenance.

Activities: Document and fix system defects. Maintain system performance. Analyze the implemented system and provide possible enhancement. Provide user training.

Input: Installed system.

Output: Bug fixes, system tuning, enhancement suggestions, user's manuals, and training courses.

3.2 Project Phases

The project is divided into five phases based on the selected SDLC model. The overall project takes six months to complete from the planning phase to the support phase. A maintenance contract extends the project for a year before its final closure.

3.2.1 Phase I - Planning

The goal of the planning phase is to identify the scope of the problem, plan the business strategy, and set the corporate goals. Following major activities are involved:

3.2.1.1 Collect Preliminary Information

Initial meetings are conducted with the client at the management level. The purpose is to gather general information regarding the client's perspective on the corporate mission, vision, and goals for this project.

3.2.1.2 Evaluate Existing Infrastructure

Review and evaluate existing system infrastructure of OMS, Call Center, and related systems. This evaluation process includes reviewing the current system documentation, conducting necessary meetings or interviews with managers and key technical personnel, and performing a preliminary feasibility study of the project. The purpose is to gain full technical background and to identify barriers and limitations.

3.2.1.3 Draft Project Management Plan

The management plan outlines the lines of communication and establishes project direction and priorities. It identifies the stakeholders and aligns the project goals with the business needs and corporate strategies. It prioritizes the requirements and creates a trace-ability matrix that identifies the major milestones and defines the criteria for the acceptance of the major deliverables.

3.2.1.4 Submit Project Plan

Submitting the project plan concludes the planning phase of the project. Once the project plan is approved by the client, it is served as the guideline through out the system development and implementation. The project plan defines the scope of the project. It lays out the development strategy with a project schedule. It also specifies the price structure and the project cost estimate including equipment cost, software cost, and labor cost. The outline of the project plan is listed below:

- Purpose – The abstract of the project objective
- Background – Problem description, history, project goal and objectives
- Scope – Description of project boundaries such as the stakeholders, data, processes, and locations
- Project Approach – Methodology, schedule and deliverables
- Managerial Approach – Team organization and experience, meeting schedules, management and reporting methods

- Assumptions and Constraints – Conditions and limitations in development time, budget, and technology
- Estimates – Schedule estimates and budgetary estimates
- Satisfaction and Risk – Criteria of success and measurement of failure or risk

3.2.2 Phase II - Analysis

The goal of the analysis phase is to define business requirements and functional requirements by analyzing the business environment and the related technical issues. Following major activities are involved:

3.2.2.1 Define Business Requirements

Conduct workshops with key users to gather the users' perspectives of how the system is supposed to work. The technical details are not included in the discussion of the business requirements. The focus is on the business or operation domain such as the problem descriptions, causes, and effects.

3.2.2.2 Technology Assessment

Based on the business requirements, an analysis of the available technologies, software systems, and hardware systems is conducted. The purpose is to select the technical solution that best suite the business requirements of the project. The final selection is made based on the combination of following architecture principles:

- Functionality – available features and options
- Configurability and Flexibility – the capability of being setup to meet specific requirements without customized code change
- Extensibility and Interoperability – the integration mechanisms and relative ease for augmentation or aggregation
- Maintainability and Manageability – the characteristics that support relative ease of maintenance and management of operation and evolution
- Performance – the system operation measurement within defined time and cost constraints
- Portability – the capability of being instantiated on different development, test, and deployment platforms with little effort
- Reliability and Stability – the integrity measurement in terms of the fulfillment of the functional requirements with few defects
- Scalability – the capability to setup the system resources proportionally to support the increase/decrease in load and performance
- Security – the level of control of user access to the systems
- Usability – the capability of being used by business users, technical developers, system administrators, and other stakeholders
- Testability – the mechanisms to support testing of functionality, implementation, performance, and reliability

3.2.2.3 Functional Requirements

Based on the business requirements and the selected technologies/solutions, a detailed system integration requirements document is produced. This document defines the detailed functional descriptions of the system to be developed. It is the basis for system design, development, implementation, and acceptance. The document includes following components:

- General Description – Descriptions for business process, system perspective, product functions, user characteristics, general constraints, assumptions, and dependencies
- Process Requirements – Specific requirements describing input, processing, and output by operation mode and user class
- Data Requirements – Specific requirements describing attributes, functions, and messages by data class and object
- Functional Hierarchy – Specific requirements describing information flow, process relationship, data construct, and data dictionary
- External Systems – specific requirements for external interfaces, such as user interface, software/hardware interfaces, and communication interfaces
- Constraints – Descriptions of specific constraints on performance, design, system attributes, and other aspects of the project

3.2.3 Phase III - Design

The goal of the design phase is to establish and maintain a technical blueprint of the selected solution. A system design document is produced and is to be approved by the client for implementation. The system requirements document produced in the analysis phase defines “what” are to be accomplished. And the system design document provides details on “how” these requirements are to be accomplished. The design document contains following components:

3.2.3.1 System Architecture

This section details the components and major architectural view of the overall system. It is represented in following model elements:

- Solution Architecture – It defines the overriding system context and scenarios, and illustrates how the architecture supports their functionality. These scenarios are depicted in a series of diagrams, with references to other sections and/or documents where more detail of transactions can be found.
- Logical Architecture – It shows how each system platform maps to the Solution Architecture. It demonstrates the high-level approach to integration across all platforms including data transfer, location of data transformation, process flow reference, and overriding integration strategies not related to specific data. Finally, the logical architecture describes the logical building blocks for each sub-system.

- Physical Architecture – It describes the physical makeup of the hardware and the location of various components that supports the architectures.

3.2.3.2 Dependency Details

This section details the compatibilities and dependencies among the system components. A group of compatible system components (hardware units, software systems, and software modules) are carefully selected to support the expected operation environment.

3.2.3.3 Interface Details

This section describes the detailed design of the interface:

- Module Decomposition – Describes each interface components and their relationship such as databases, programs, and libraries.
- Data Decomposition – Describes the interface data elements, data structures, and data stores.
- Controls – Describes the data flow controls and processes.

3.2.3.4 Module Details

This section describes the internal design details for each interface module:

- Data Component – Table schema, class/object structure, attributes, and message formats.
- Graphical User Interface – Interface windows and their components.
- Process and Functions – Triggers, inputs, operations, and outputs.

3.2.3.5 Testing Plan

This section outlines the testing procedure and steps for system testing and acceptance testing. The testing procedure covers every functional requirement identified in the analysis phase. The detailed test scripts may be revised and finalized during the next phase (implementation phase).

3.2.3.6 Project Timeline and Resource Estimate

With a detailed design, the initial project schedule can be filled in with more accurate timeline and resource estimates.

3.2.4 Phase IV - Implementation

The expected system is constructed during the implementation phase. The development is managed closely to the blueprint in the detailed design produced in the design phase. Following major activities are involved:

3.2.4.1 Building the Team

For each major interface component or module, a small group of developers is assembled to build the module. Each group, including the testing group, has a domain expert as the technical lead of the group. A system architect is assigned as the overall team leader who manages the technical issues and coordinates the project management tasks with the project manager.

3.2.4.2 Coding

Coding is conducted individually or coordinated with a group of developers. Code review by the development team is conducted regularly to ensure that the development follows closely to the design specifications.

3.2.4.3 Testing

Extensive testing is required throughout the implementation phase for the purpose of quality assurance (QA) and quality control (QC). Generally there are four levels of testing involved in a system development:

- Unit Testing – The lowest level of testing normally conducted by individual programmers to verify the functional requirement of one program unit or module.
- Integration Testing – Testing a group of modules as an integrated component after each module is unit tested.
- System Testing – Testing the integrated system to ensure that the system works module by module and also as a whole to verify the functional, performance and reliability requirements. This testing involves with the development team and QA team.
- Acceptance Testing – The system testing conducted by the client, and based on the results, either granting or refusing the acceptance. The test environment is designed to be identical or close to the anticipated installation environment.

3.2.4.4 Managing the Development

Regular weekly meetings and special meetings are organized with the development team. It keeps track of the progress, identify and resolve any development issue. The goal is to construct the system in time and under the budget.

3.2.4.5 Documentation

All implementation issues are documented and maintained through out the implementation phase. The system reference manuals are drafted and the acceptance testing scripts are finalized at the completion of the system development.

3.2.4.6 Deployment

The integrated system is installed at the client site. The acceptance testing is conducted and signed off by the client. This concludes the system delivery.

3.2.5 Phase V - Support

The support phase provides system maintenance and support after delivering the integrated system. Following major activities are involved:

3.2.5.1 Customer Support

Provide adequate customer support. Report and document any system defect. Coordinate the efforts of bug fixing, applying code changes, conducting the test, and accepting the fixes.

3.2.5.2 Maintaining Performance

Monitor system operation behavior for any performance problem, especially during the high volume (storm) operation mode. Perform the system tuning to identify the optimal system parameters. A performance testing or stress testing may be conducted to verify the system stability against the performance specifications under the extreme conditions.

3.2.5.3 Enhancement

Analyze the implemented system and provide suggestions for possible future enhancement.

3.2.5.4 Training

Produce the user's manuals and conduct training courses.

3.3 Project Metrics and Measurements

The measurements for a successful project management are based on three major constraints:

- Time – Is the system delivered “on time”?
- Cost – Is the system delivered “within budget”?
- Quality – Is the system performance “acceptable” to the client?

Based on these three principles, the progress of the projects are constantly analyzed and measured to ensure that the project deliverables meet the projected goals or milestones.

3.3.1 Identifying Tasks

This project uses a Work Breakdown Structure (WBS) to hierarchically decompose the project into phases, tasks, and subtasks until the level of work packages are reached. Part of this decomposition is to identify certain subtasks as the milestones which signify key delivery points as the measurements of project success. The WBS for this project is illustrated in the diagram below:

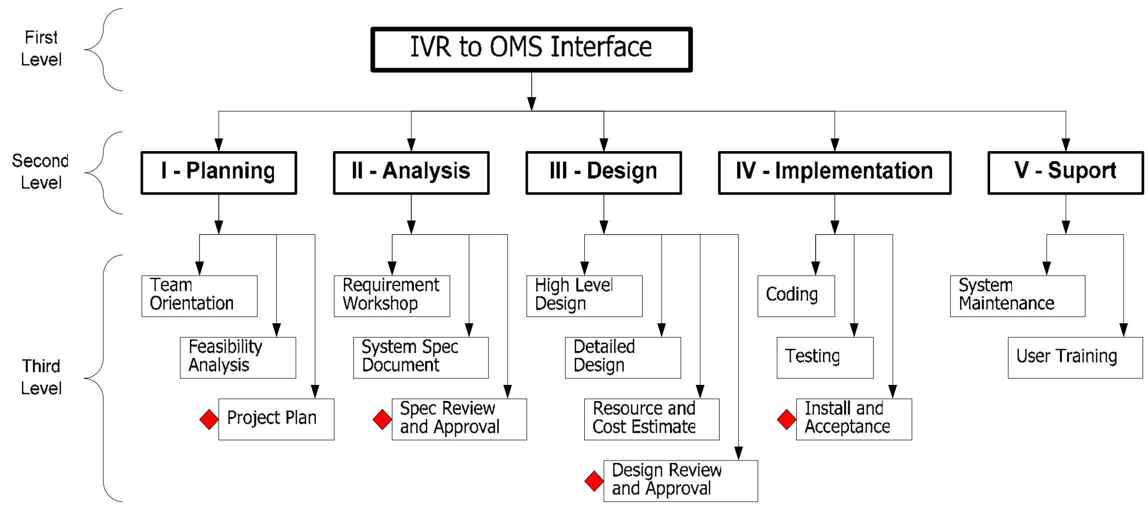

Figure 3-2 Work Breakdown Structure

3.3.2 Scheduling

Building the project work schedule involves estimating task durations, identifying inter-task dependencies, and setting scheduling strategy.

3.3.2.1 Estimate Task Duration

Estimating the task duration from start to finish is the first step in building a good schedule. Following are some of the commonly used techniques to estimate task duration:

- Expert Judgment - This type of estimation depends highly on individual's knowledge, past experience, and preference.
- Historical Data – Make predictions based on the data from the previous projects. The historical data may not be reliable for the current project if there are differences in project environment, resources, and constraints.

- Analogous Estimating – A top-down estimating technique, using the duration of previous activities that are similar to the current one.
- Simulation – This method uses well-engineered research data about the specifics of the program. One such method is Monte Carlo Analysis which calculates multiple durations with different sets of assumptions.

3.3.2.2 Specify Dependency

Inter-task dependencies define the activity sequencing that impacts the overall project duration. There are four types of dependencies between two related tasks:

- Finish/Start – The finish of one task triggers the start of another task
- Start/Start – The start of one task triggers the start of another task
- Finish/Finish – The two tasks must finish at the same time
- Start/Finish – The start of one task indicates the finish of another task

3.3.2.3 Scheduling Strategy

There are two ways to establish the project start and end time:

- Forward Scheduling – Establish the project start date, then calculate the start and end dates of each task based on the planned task durations, the inter-task dependencies, and the allocation of resources.
- Reverse Scheduling – Establish the project deadline, then calculate backward for the start and end dates of each task based on the planned

task durations, the inter-task dependencies, and the allocation of resources.

A Gantt chart for the baseline work schedule of this project is illustrated in the figure below:

Task	Task Name	Duration	Start	Finish
0	**Schedule for IVR to OMS Interface**	**150 days**	**Mon 10/3/05**	**Fri 4/28/0**
1	**Planning**	**10 days**	**Mon 10/3/05**	**Fri 10/14/0**
1.1	Team Orientation	5 days	Mon 10/3/05	Fri 10/7/0
1.2	Feasibility Analysis	2 days	Mon 10/10/05	Tue 10/11/0
1.3	Project Plan	3 days	Wed 10/12/05	Fri 10/14/0
2	**Analysis**	**20 days**	**Mon 10/17/05**	**Fri 11/11/0**
2.1	Requirement Workshop	5 days	Mon 10/17/05	Fri 10/21/0
2.2	System Specification Document	10 days	Mon 10/24/05	Fri 11/4/0
2.3	Spec Review and Approval	5 days	Mon 11/7/05	Fri 11/11/0
3	**Design**	**25 days**	**Mon 11/14/05**	**Fri 12/16/0**
3.1	High Level Design	8 days	Mon 11/14/05	Wed 11/23/0
3.2	Detailed Design	10 days	Thu 11/24/05	Wed 12/7/0
3.3	Resource and Cost Estimate	2 days	Thu 12/8/05	Fri 12/9/0
3.4	Design Review and Approval	5 days	Mon 12/12/05	Fri 12/16/0
4	**Implementation**	**65 days**	**Mon 12/19/05**	**Fri 3/17/0**
4.1	Coding	50 days	Mon 12/19/05	Fri 2/24/0
4.2	Testing	50 days	Mon 1/2/06	Fri 3/10/0
4.3	Installation and Acceptance	5 days	Mon 3/13/06	Fri 3/17/0
5	**Support**	**30 days**	**Mon 3/20/06**	**Fri 4/28/0**
5.1	System Maintenance	30 days	Mon 3/20/06	Fri 4/28/0
5.2	User Training	5 days	Mon 3/27/06	Fri 3/31/0

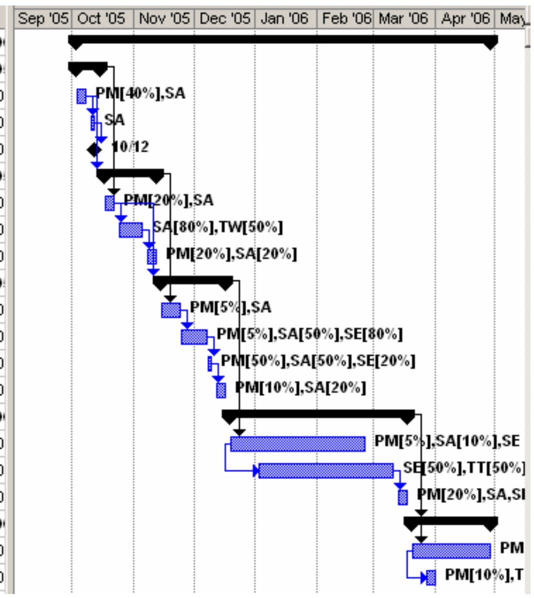

Figure 3-3 Baseline Work Schedule

3.3.3 Resource Planning

Resource planning involves team building, assigning resources to tasks, and cost estimating.

3.3.3.1 <u>Resource Structure</u>

The project resource structure contains the information on labor, services, facilities/equipments, supplies/materials, and other direct costs such as travel and training expenses. Resources are assigned to each task based on their roles, skills, and availability. The following chart shows a sample resource pool for the project:

		Resource Name	Type	Initials	Max. Units	Std. Rate	Cost/Use	Accrue At
1		Project Manager	Work	PM	100%	$150.00/hr	$0.00	Prorated
2		System Architect	Work	SA	100%	$150.00/hr	$0.00	Prorated
3		**Software Engineer**	**Work**	**SE**	**100%**	**$100.00/hr**	**$0.00**	**Prorated**
4		**Tester/Technician**	**Work**	**TT**	**100%**	**$80.00/hr**	**$0.00**	**Prorated**
5		Technical Writer	Work	TW	100%	$60.00/hr	$0.00	Prorated
6		Development Server	Material	DVS		$5,000.00	$0.00	Prorated
7		IVR-OMS Database S	Material	DBS		$10,000.00	$0.00	Prorated

Figure 3-4 Resource Pool

A sample resource usage chart of the allocated resources for the project is shown below:

	Resource Name	Work	Details	Qtr 4, 2005 Oct	Nov	Dec	Qtr 1, 2006 Jan	Feb	Mar	Qtr 2, 2006 Apr
	⊟ Project Manager	111.2 hrs	Work	28h	13.2h	18h	8.8h	7.2h	20h	16h
	Team Orientati	*8 hrs*	Work	8h						
	Project Plan	*12 hrs*	Work	12h						
	Requirement W	*8 hrs*	Work	8h						
	Spec Review ai	*8 hrs*	Work		8h					
	High Level Des	*3.2 hrs*	Work		3.2h					
	Detailed Desig.	*4 hrs*	Work		2h	2h				
	Resource and	*8 hrs*	Work			8h				
	Design Review	*4 hrs*	Work			4h				
	Coding	*20 hrs*	Work			4h	8.8h	7.2h		
	Installation and	*8 hrs*	Work						8h	
	System Mainte	*24 hrs*	Work						8h	16h
	User Training	*4 hrs*	Work						4h	
	⊟ System Architect	404 hrs	Work	146.4h	117.6h	44h	17.6h	14.4h	48h	16h
	Team Orientati	*40 hrs*	Work	40h						
	Feasibility Ana	*16 hrs*	Work	16h						
	Project Plan	*12 hrs*	Work	12h						
	Requirement W	*40 hrs*	Work	40h						
	System Specifi	*64 hrs*	Work	38.4h	25.6h					
	Spec Review ai	*8 hrs*	Work		8h					
	High Level Des	*64 hrs*	Work		64h					
	Detailed Desig.	*40 hrs*	Work		20h	20h				
	Resource and	*8 hrs*	Work			8h				
	Design Review	*8 hrs*	Work			8h				
	Coding	*40 hrs*	Work			8h	17.6h	14.4h		
	Installation and	*40 hrs*	Work						40h	
	System Mainte	*24 hrs*	Work						8h	16h
	⊟ **Software Engineer**	**807.2 hrs**	Work		32h	115.2h	264h	224h	92h	80h
	Detailed Desig.	*64 hrs*	Work		32h	32h				
	Resource and	*3.2 hrs*	Work			3.2h				
	Coding	*400 hrs*	Work			80h	176h	144h		

Figure 3-5 Resource Usage

3.3.3.2 Estimate Costs

Accurate cost estimating is one of the keys to measure project success. Following are some of the commonly used techniques to estimate costs:

- Analogous Estimating – A top-down estimating, using the actual cost of a previous project with similar characteristics as a base for estimating the cost of the current project.
- Parametric Modeling – It uses a mathematical model to predict project costs based on the project characteristics such as the historical data.
- Bottom-up – It takes the costs of individual work tasks and rolls up to get the project total.
- Computerized Tools – Various project management software tools provides the cost estimating function.

Following Gantt chart shows a sample cost estimate for the project based on the allocated resources:

	Task Name	Fixed Cost	Fixed Cost Accrual	Total Cost	Baseline
1	**Schedule for IVR to OM:**	**$0.00**	**Prorated**	**$190,000.00**	**$0.00**
2	**Planning**	**$0.00**	**Prorated**	**$13,200.00**	**$0.00**
3	Team Orientation	$0.00	Prorated	$7,200.00	$0.00
4	Feasibility Analysis	$0.00	Prorated	$2,400.00	$0.00
5	Project Plan	$0.00	Prorated	$3,600.00	$0.00
6	**Analysis**	**$0.00**	**Prorated**	**$20,800.00**	**$0.00**
7	Requirement Work:	$0.00	Prorated	$7,200.00	$0.00
8	System Specificati	$0.00	Prorated	$11,200.00	$0.00
9	Spec Review and	$0.00	Prorated	$2,400.00	$0.00
10	**Design**	**$0.00**	**Prorated**	**$27,600.00**	**$0.00**
11	High Level Design	$0.00	Prorated	$10,080.00	$0.00
12	Detailed Design	$0.00	Prorated	$13,000.00	$0.00
13	Resource and Cos	$0.00	Prorated	$2,720.00	$0.00
14	Design Review an	$0.00	Prorated	$1,800.00	$0.00
15	**Implementation**	**$0.00**	**Prorated**	**$95,800.00**	**$0.00**
16	Coding	$0.00	Prorated	$49,000.00	$0.00
17	Testing	$0.00	Prorated	$36,000.00	$0.00
18	Installation and Ac	$0.00	Prorated	$10,800.00	$0.00
19	**Support**	**$0.00**	**Prorated**	**$32,600.00**	**$0.00**
20	System Maintenan	$0.00	Prorated	$28,800.00	$0.00
21	User Training	$0.00	Prorated	$3,800.00	$0.00

Figure 3-6 Cost Estimates

3.3.3.3 Schedule Adjustment

Because of the dynamics of the project, the project schedule has to be analyzed and adjusted to deal with conflicts in resource allocations. Resource leveling is a strategy used to correct resource over-allocation, that is, too much work assigned to resources than what are available. Resource leveling involves with following activities:

- Critical Path Analysis – This is the process to identify a sequence of dependent tasks (critical path) that must be completed on time so that the overall project will not be delayed.

- Task Delaying – Add delay to the task until it is no longer over-allocated. This is typically done to a non-critical task having available slack time that its start time can be delayed without causing a delay in the project completion time.
- Task Splitting – A task is split into subtasks that do not require continuous work. Some of the subtasks are rescheduled to start in a delayed time when the resources become available. This is typically done to a critical path that cannot be delayed without delaying the entire project schedule.

3.3.4 Quality Management

Project quality is another important measurement of project success. Quality management must address both the management of the project and the progress of the project. It includes following processes required to ensure that the project will satisfy the quality policy, objectives, and responsibilities.

3.3.4.1 Quality Planning

This process identifies the required project quality standards and establishes the ways to satisfy these standards. The quality standards are derived from the organizational quality policy, the project scope statement, the product description, area-specific standards or regulations, and other process outputs. The quality planning process produces following outputs:

- Quality Management Plan – Description of how the project management team will implement its quality policy

- Operational Definition – Detailed description of what and how each project activity is measured by the quality control process
- Checklists – A checklist for a specific activity is a set of required steps that must be performed for quality assurance purpose

3.3.4.2 Quality Assurance

Quality Assurance (QA) is the process that systematically evaluates overall project performance on a regular basis to provide confidence or assurance that the project will satisfy the project quality standards. QA is planned in the beginning phase of the project and is verified with appropriate measures taken at every stage of the project.

3.3.4.3 Quality Control

Quality Control (QC) is the process performed at the end of each project stage to monitor specific project results and verify if they comply with the project quality standards. Unlike QA, QC by itself does not provide quality, but its methods may be used by QA as the verification tool to identify the ways to eliminate causes of unsatisfactory performance.

3.3.5 Progress and Control

To ensure that the project meets its objectives within budget and scheduled timescales at every stage, the progress is monitored and reported, the deliverables are tracked, the appropriate actions are taken to manage the changing project parameters such as cost, schedule, scope, and quality.

3.3.5.1 Progress Reporting

Periodical project status report is produced to measure the project success and identify the problems in project progression. Following measurement attributes may be included in the status report:

- Cost Comparison – shows the actual hour and cost data for period and to-date total, estimates to complete, estimates at completion, and budget
- Rate Analysis – shows the rate and multiplier data for period, to-date total, to complete, estimates at completion, and budget so that plans and forecasts can be verified
- Progress – shows period and to-date total percent completion data
- Productivity – shows how the work is accomplished compared with the cost of the work
- Performance – compares budget, earned value, and actual data of the schedule and cost variances to verify if the project is ahead/behind schedule and under/over budget

3.3.5.2 Change Management

Change management is a process that defines management procedures undertaken to identify, approve, and control changes to the scope, deliverables, timescales or resources of the project. The change management process consists of following activities:

- Submit Change Request – A change request is submitted by any member of the project team. It identifies a requirement for change in scope, deliverables, timescales, and organization. It describes the reasons for the change, as well as the benefits, costs, and impacts of the change.
- Review Change Request – The change request is reviewed by the project manager to determine if a feasibility study is required.
- Identify Change Feasibility – This study is conducted to investigate all change requirements, options/recommendations, costs/benefits, risks/issues, and impact.
- Approve Change Request – This is the formal review of the change request. Based on the risk and impact of implementing the change request, one of the decisions may be made:
  - ✓ Reject the change
  - ✓ Request more information
  - ✓ Approve the change
  - ✓ Approve the change with specified conditions
- Implement Change Request – Execution of the approved change request. It includes following steps:
  - ✓ Identify the change schedule
  - ✓ Perform pre-implementation testing if necessary
  - ✓ Implement the change

- ✓ Review the success of the implementation
- ✓ Document and close the change

4 Project History

4.1 Project Initiation

The project was initiated with meetings between the Public Service Company and the Utility Business Consulting Company. The participants from the client included the VP of Energy Services, the VP of Information Technology, and the managers from the Information Technology (IT) Department, the Customer Services Department, and the Operations Department. The participants from the Utility Business Consulting Company included the executive consultant and the project manager of System Integration. The purpose of the meetings was to identify the business need for improving the business processes of Trouble Call Center and Operations Control Center through the application of information technology.

4.1.1 Identify the Need to Improve Business Process

The initial discussions between the senior managers and the executive consultant were focused on the history, issues, and opportunities in the business processes of the customer services and operations. It was concluded that the there were significant problems and deficiencies in the current business processes and the customer services area. It was identified in particular that there were needs, opportunities, and benefits to improve the processes in Trouble Call Center and Operations Control Center to better handle high customer call volume during a storm situation.

4.1.2 Identify Project Sponsor and Project Organization

Based on the type of business process, it was determined that the activities involved were primarily within the customer services area. The Customer Services Department would take the role of the project sponsor or the principle authority of the project on issues regarding the business needs, priority, scope, and resources.

The Operations Department, the IT Department, and a system integration team from the Utility Business Consulting Company would be the other key stakeholders providing the support to the project.

A manager from the Customer Services Department was appointed as the project manager responsible for the project planning and execution.

4.1.3 Project Concept Review and Approval

The results of the meetings were documented in the project concept proposal by the project manager. The concept proposal outlined the need or opportunity to improve the operations in Trouble Call Center and Operations Control Center, identified the deficiencies in the current infrastructure, highlighted the benefits and strategic goals, and provided the basic funding estimate. The project concept proposal was reviewed and approved by the company's Chief Information Officer and Executive Review Board. This concluded the project initiation phase.

4.2 Project Conceptualization and Definition

This was the first stage to launch the project. The preliminary preparation and study were conducted to validate the approved project concept proposal and to formally define the project objectives in terms of specifications and requirements. The outcome of the project conceptualization and definition provided the basis for making a decision of go or no-go for further project development.

4.2.1 Define Project Boundary and Scope

A formal agreement was established among the stakeholders on the project boundaries and scope. An organizational strategic plan was hammered out outlining the long-term goals, high level business requirements, benefits, assumptions, costs, and schedule:

- <u>Mission</u> – It identified the deficiencies in the current business processes of Trouble Call Center and Operations Control Center, the needs and opportunities to address those deficiencies, and how it would benefit the company to reach its long-term goal of improving customer service quality and corporate profitability. It also identified the critical success factors and the performance measurement to achieve the business objectives.
- <u>Requirement</u> – The high level functional requirements were specified for existing system/methods/procedures, system users, network infrastructure, and architectural framework.
- <u>Assumptions and Constraints</u> – It identified the assumptions and constraints that would affect the project planning and execution including

the options for employing new technology or system upgrade, organization support level, budgetary and scheduling constraints, and equipment and resource limitation.

- Cost, Schedule, and Performance – It established the commitment to funding the project cost, identified major scheduling milestones, and defined the basic measurable performance metrics.

4.2.2 Feasibility Study

This task was conducted by an expert consultant with the support of the technical experts from various departments. The purpose of this study was to identify and validate feasible solutions that would satisfy the project objectives and requirements. Based on the project requirements, the evaluation was conducted on the current systems, methods, procedures, and the alternative solutions. The evaluation criteria were outlined based on the project objectives. A recommendation was made in the conclusion of the study. The recommendation was to improve the call-taking process at Trouble Call Center by expending current IVR system or introducing a new IVR system that directly communicate with the OMS.

4.2.3 Risk and Benefit Analysis

The cost-benefit study was conducted to evaluate and compare the alternative solutions. The cost analysis profile was produced covering each phase of the project including design, development, implementation, operation, and maintenance. The benefit analysis included the tangible benefits such as the revenue increase, saving in time/money, and reduction in labor. It also included the intangible benefits such as the improved system performance, increased process volume, and reduction in system down time. Based on the project evaluation criteria, the costs and benefits were summarized and the comparisons were made. The results became the basis for the selection of the recommended solution.

The risk analysis was conducted to produce a risk management plan identifying the possible risks on each project phase, assessing the probability and the impact of each risk, and devising the steps or plans to reduce the risks.

4.2.4 Resources Allocation

Based on the proposed or recommended solution, a preliminary project estimate was made and a request to obtain the project funding, staff, and resources was submitted for approval by the project sponsor.

4.3 Project Planning

After the project definition in the previous stage was approved, the full project development work was officially started with the project planning phase. In this phase, the project concepts were further developed with detailed descriptions in various project development domains such as tasks, processes, schedule, budget, and resources.

4.3.1 Development Strategy

Following project development strategy was first laid out to set the stage for further constructing the detailed project tasks:

- As the integrator of the current TCC application and OMS, the Utility Business Consulting Company was contracted for the design and implementation of the new IVR to OMS interface.
- The Customer Services Department would manage the overall project with the support from the Operations Department and the IT Department.
- It was determined that the existing TCC to OMS infrastructure and processes already implemented in the Call Center Application could be extended to the IVR to OMS application. The current Call Center IVR vendor was contracted to provide the hardware and software supports for the new IVR to OMS interface.
- The new IVR to OMS interface would be an extension of the existing TCC to OMS interface and processes. The OMS vendor was not needed for the development work but was contracted to provide technical support for handling the added trouble call volume.

4.3.2 Communication Plan

The communication plan was developed to coordinate project activities among team members and external entities. It specified frequencies, target audiences, formats, locations, and types of information delivered.

4.3.2.1 Lines of Communication

Four communication entities were identified: project manager, development team, technical support team, and vendors. The lines of communication among them were outlined in the figure below:

Entity 1	Entity 2	Lines of Communication
Project Manager (PM)	Development Team	All project related issues, status reports, and deliverables were submitted to the PM. The PM provided the project directions, resolutions, and approvals.
Project Manager	Support Team	The PM sent the queries and requests to the support team. The support team provided the technical advises and reviews to the PM.
Project Manager	Vendor	Any request for vendor action was normally conveyed only between the PM and the vendor.
Development Team	Support Team	Only the requests for information were exchanged directly between the development team and the support team.

Figure 4-1 Lines of Communication

4.3.2.2 Form of Communication

Scheduled and unscheduled communications were specified in the communication plan as outlined below:

- Reoccurring Events – The weekly status meetings with all team leaders were normally scheduled and conducted through conference calls. The progresses, problems, and issues were documented and updated in the project schedule.
- Pre-scheduled Events – The normally scheduled communication events that occurred at the important junctions of the project, involving key personnel of each team. Such events included the project plan review meeting, the functional requirement workshop, the requirement review meeting, the design review meeting, and the review meeting for the acceptance test plan.
- Unscheduled Events – Special meetings and processes managed by the project manager involving key personnel of each team to address and resolve major project issues, for example, meetings to resolve the delay of the project or to process a change order for adding a new functionality.

4.3.3 Project Management Plan

The project management plan was carefully developed to outline the lines of communication and established project direction and priorities. It documented the details on the functional units involved, required job tasks, cost and schedule estimates, allocated resources, task interrelationships, performance measurements, milestones and review scheduling. The management plan was reviewed and revised at the end of each project phase address the management oversight activities of the project. The management plan included following components:

- Tasks and Subtasks – A detailed work breakdown structure defining system functional units and the required work elements.
- Schedule Estimates – Task durations, resource availability, task interrelationships and dependencies.
- Resource Estimates – Costs in time and currency for labor, equipment, facilities, and other expenses.
- Milestones – Key delivery points of the project schedule, such as acceptance of the project plan, approval of the system integration requirements document, approval of the system design document, completing the system acceptance testing.
- Performance Measurement – Project and task completion percentage, comparison of budget, earned value, and cost variances.

4.3.4 Quality Management Plan

The quality management plan for this project was broadly framed to specify the quality standards and implementation procedures based on the requirements of the project. This included following:

- QA Roles – The organizational and functional alignment of the QA staff, their roles and responsibilities at the management, team leader, and specialist levels.
- Operational Definition – Peer and process reviews used to ensure that processes and products associated with hardware, software, and documentation are monitored and tested to ensure compliance with methodology, contracts, and standards.
- Reporting and Action – The procedures and formats for the preparation, tracking, and management involvement in identifying problems and recommending corrective actions.

4.4 Analysis and Design

The analysis and design activities was conducted by the system architect from the Utility Business Consulting Company through the workshop, meetings, interviews and other means of communication with key users and technical personnel from the client.

4.4.1 Functional Requirement

A workshop with key system users and technical personnel was conducted to develop detailed functional requirements. The business requirements defined in the planning phase was used as the basis for analyzing the intended use of the system and specifying the functional requirements, data requirements, and the fundamental processes to support the desired business functionality. The outcome of the workshop was the initial draft of the System Integration Requirements Document.

The System Integration Requirements Document was submitted to and reviewed by the project manager and the technical support team. Revisions were made and the final draft was approved by the project manager.

4.4.2 Test Criteria and Plan

Based on the functional requirements, the test criteria were established and a test plan was developed to ensure that the implementation of the requirements could be verified. The test criteria and plan specified the test locations, testing environment, responsible parties for the testing, system units or modules to be tested, and test procedures.

4.4.3 System Design

The System Design Document was drafted to transform the functional requirements into complete, detailed specifications. The design was conducted iteratively, producing first a high-level system design that emphasizes the functional features of the system, then a more detailed system design that expands the general design by providing all the technical detail.

The System Design Document was submitted to and reviewed by the project manager and the technical support team. Revisions were made and the final draft was approved by the project manager.

4.4.4 Implementation Plan

The implementation plan was developed based on the design document. The implementation plan was integrated into the overall project plan and schedule by adding new tasks and subtasks into the work breakdown structure. The project plan and schedule were adjusted with more accurate estimates for equipments, facilities, materials, and personnel.

5 Lessons Learned and Next Evolution of the Project

5.1 Lessons Learned

One of the principle responsibilities for every team member is to review and analyze the case histories through out the software development cycle. The objective is to identify and document key lessons learned so that the valuable knowledge and experience gained in the past projects become an essential part for continuous improvement in the future software development projects.

5.1.1 Get the Executive Leadership Commitment

As a consultant, one of the major mistakes we made early in the project was not being able to get the commitment from the executive leadership of the client. The project started without real vision and guideline. As a result, not knowing what was really needed, we ended up having to rework repeatedly to get the planning and requirements done.

The three major stakeholders on the client side were the Customer Services Department, the Operations Department, and the IT Department. Each of them had different agendas and priorities in this project. For the Customer Service Department, they were concerned primarily with the customer relationship and the revenue. The Operations Department focused on fixing the equipments and managing the crew workloads. Although the IT Department played more of a supporting role, they had the technical authority on issues that would impact the company wide IT strategy.

The initial project vision and scope was vague. It was not clear who was really running the project on the client side. So we came in working primarily with the Operations Department because we had great relationship working with them in the past when getting their OMS implemented. It did not go well. The effort of mapping the functional requirements to business requirements was repeatedly rejected by other departments, resulted in many reworks and rewrites.

Further more, as a large utility company; the service territory was divided as the East and the West regions each operated independently. We did not realize that each region had its own operation requirements. So our work with the West region was not received well with the East.

Having realized the cause of the problem, we eventually pressed the client to get their senior leadership involved in the decision-making process. As a result of internal negotiation among the stakeholders of the client, it was determined that the Customer Services Department was the principle authority of the project. A project manager from the Customer Services Department was appointed as the overall team leader.

The lessons are summarized as following:

- Identify the principle stakeholder early and to get their full commitment to the project.
- Define the right leadership roles to resolve issues regarding the business needs, priority, scope, and resources.

5.1.2 Establish a Framework in Requirement Process

The initial business requirements done by the client was extremely fragmented and poorly structured making it very difficult to substantiate functional requirements. The client was not familiar with the terminologies we used with most other utility companies making it difficult for us to facilitate effective communication. The requirements provided by the client were neither analyzed nor prioritized. They were simply a list of features that the client wanted without knowing exactly what they really needed.

The problem was primarily caused by the downsizing of the IT Department. The project team from the client side simply lacked the personnel with adequate knowledge and skills in software development process. They were clearly not prepared for the tasks at hand. Unfortunately, our planning was not conducted thoroughly to deal with this problem. As a result, the project progression was too fast, too confusing, and too demanding for the client. Frequent adjustments to the project plan and a lot of rework had to be done to finally bring the client up to the task.

The lessons learned from the requirement specification process are summarized below:

- Establish a framework of requirement process and present it to the client.
- Validate the framework for the business cases before jumping into details of building the business cases.

- Conduct necessary training for the requirement analysis for key client representatives.
- Establish a formal mechanism to control changes to requirements.
- Prioritize the real requirements to determine those that should be met in the first stage and those that can be addressed subsequently.

5.1.3 Analyze the Impact of the Constraints in Project Planning

We made a mistake during the project planning stage for not fully analyzing the impact of the major constraints. One of the major constraints was the resource availability during the storm season between April and August. In the initial planning stage, we underestimated the impact of this constraint. Our plan had the major system testing and delivery going well into the storm season. This was based on the initial commitment from the IT Department and the Customer Services Department that adequate resources would be available. However, the assumption was inaccurate because the main task of testing and validation had to be conducted by the operators in the Operations Department. There would be no availability of resources in the Operations Department during the storm season when they were fully committed to the daily operations.

Fortunately, we realized this problem early. By examining the alternatives and workarounds, we decided the best solution was to overhaul the original plan and reschedule it so that the project delivery would be completed before the storm season. Although it was a major struggle, we avoided delaying the project for additional six months.

The lessons learned in the planning stage are summarized below:

- Identify the major constraints and obstacles as early as possible and analyze the full impact to the project.
- Establish a well designed work breakdown structure with major dependencies and milestones clearly identified.
- Make the best possible initial resource estimates for the main tasks and confirm the allocation of the resources for those tasks.

5.1.4 Pay Close Attention to System Performance in Design

Establishing the system performance expectation early proved to be very important for producing an adequate system design and ultimately a system delivery and acceptance. When the Call Center operates at its full capacity, it activates 125 customer representatives; each can process between 60 to 80 trouble calls per hour. This translates to a maximum operation volume of 7500 to 10000 calls per hour for the OMS. With the new IVR system, the call volume is expected to reach 20000 to 30000 calls per hour and the client actually hoped to extend the system capacity to handle up to 40000 calls per hour.

The OMS vender claimed that the current OMS server was adequately configured to support processing 20000 calls per hour. And with additional tuning to the system, database, and operations, the system can handle up to 30000 calls per hour. Based on this assumption, the initial design was laid out based on a single OMS server directly taking calls from both Call Center and IVR.

Not feeling too confident about this architecture, we decided to conduct our own volume testing simulating the expected IVR call volume. The test failed miserably. Additional testing conducted together with the OMS vender further proved the inadequacy of the current OMS to handle such a high volume. It would have been a performance disaster had we stayed with the original design.

As a result, we had to redesign the IVR-OMS integration system. The new design called for a designated IVR server to preprocess the IVR calls. The process would group the calls on same location and with similar priority before sending then to the OMS server. A monitoring and control mechanism was also designed to control the flow of the IVR calls by delaying or blocking calls with low priority when the load on OMS was too high.

The lessons learned can be summarized as below:

- Establish an acceptable performance standard in the early design stage.
- Look for design alternatives that provide flexibility and leverage for potential performance enhancement.
- Conduct necessary performance testing early to validate the design concept.

5.2 Next Evolution of the Project

The detailed system design document and testing plan have been approved. The project is currently in its implementation phase.

5.2.1 Hardware Procurement

The hardware procurement mostly involves actions from the client and the IVR vender:

- Plain Old Telephone Service (POTS) and Integrated Services Digital Network (ISDN) communication lines have been installed for the West and the East regions.
- Installation of the frame relays and routers for the West region has been completed.
- Installation of the frame relays and routers for the East region is scheduled.
- The purchasing order of the IVR database server is being processed.
- The IT Department provided the development server by using one of its training servers.

5.2.2 Software Development

The software development activities involve the Utility Business Consulting Company, the client IT Department, and the IVR vender:

- A development team from the Utility Business Consulting Company has been formed for the coding and testing of the interface.
- A database administrator (DBA) from the IT Department has been appointed to support the development team in system administration and database management.
- A programmer from the IT Department has been appointed to develop IVR call entry scripts with the support from the IVR vender.

5.2.3 System and Performance Testing

Preparations for the testing have been scheduled based on the approved System Testing Plan. The activities include:

- The IT Department is responsible for testing IVR hardware.
- The IT Department and the Call Center is responsible for testing the IVR scripts and database replication processes between IVR server and OMS server.
- The Operations Department will build a testing environment and prepare for the integration test data.
- The Operations Department will also conduct high volume performance testing of the IVR-OMS interface.
- The final acceptance testing will be conducted jointly for the completion of the implementation phase.

6 Appendices

Appendix A Definitions, Acronyms, and Abbreviations

AMR	Automatic Meter Reading
API	Application Programming Interface
CIS	Customer Information System
COBOL	Common Business Oriented Language
CORBA	Common Object Request Broker Architecture
DBA	Database Administrator
DNIS	Dialed Number Identification Service
EJB	Enterprise Java Beans
GIS	Geographic Information System
HTTP	Hyper Text Transfer Protocol
IDL	Interface Definition Language
IIOP	Internet Inter-ORB Protocol
ISDN	Integrated Services Digital Network
IT	Information Technology
IVR	Interactive Voice Response Unit
J2EE	Java 2 Enterprise Edition
LAN	Local Area Network
MAPI	Messaging Application Program Interface
MQ	Message Queuing
OMS	Outage Management System
ORB	Object Request Brokers
PL/SQL	Procedural Language/Structured Query Language

PM	Project Manager
POP3	Post Office Protocol 3
POTS	Plain Old Telephone Service
QA	Quality Assurance
QC	Quality Control
RDBMS	Relational Database Management System
RMI	Remote Method Invocation
RTI	Real Time Interface
SCADA	Supervisory Control And Data Acquisition
SDLC	Systems Development Life Cycle
SMTP	Simple Mail Transfer Protocol
TCC	Trouble Call Center
TCP/IP	Transmission Control Protocol / Internet Protocol
TTS	Text To Speech Software
VPN	Virtual Private Network
WAN	Wide Area Network
WBS	Work Breakdown Structure
WMS	Work Management System
XML	Extensible Markup Language

Appendix B References

1. Effective Project Management, Robert K. Wysocki, July 2003
2. MSC 696A Classroom-Based Course Module, Regis University
3. Systems Analysis and Design Methods, Jeffrey L. Whitten, Lonnie D. Bentley, 1998
4. The Business Case, Project Charter & Project Plan, Rose Minton, January 2004, http://www.tmgconsulting.com/articles/0401_minton.htm
5. Mitigating CIS Selection Risks, Mario Bauer, April 2004, http://cisworld.com/articles/0310_bauer.htm
6. Core and Extended CIS Components, Greg Galluzzi, June 2004, http://www.tmgconsulting.com/articles/0406_galluzzi.htm
7. Smart IVR Solutions, Database Systems Corp, http://www.databasesystemscorp.com/psivr.htm
8. SPL Customer Care & Billing for Integrated Utilities, SPL, http://www.splwg.com/resources/brochures/Integrated-Utilities.pdf
9. SPL Outage Management Solutions, SPL, http://www.splwg.com/resources/brochures/datasheets_0305/OMS_ds_0305.pdf
10. Call Center IVR, VOXEO, http://www.voxeo.com/markets/call-center-ivr.jsp

11. Configurable IVR Platform, Simplified Telephony Solutions,

http://www.sim-phony.com/Downloads/simpivrtechoverview.pdf

12. Centricity an Overview, SPL,

http://www.splworldgroup.com/products/centricity.asp