Strategic Mobility 21

Development of Joint Data Standards and Communication Protocols in An Integration Tracking System

Contractor Report 0009

Prepared for:

Office of Naval Research
875 North Randolph Street, Room 273
Arlington, VA 22203-1995

Dr. Paul Rispin, Program Manager, ONR Code 331
703.696.0339   rispinp@onr.navy.mil

In fulfillment of the requirements for:

FY 2005 Contract No. N00014-06-C-0060
Strategic Mobility 21 – CLIN 0009

Prepared and Submitted by:

Dr. Lawrence G. Mallon, Program Manager
California State University, Long Beach Foundation
6300 State University Drive, Suite 220, Long Beach, CA 90815 562.985.7392

August 31, 2007

Acknowledgement of Support and Disclaimer: This material is based upon work supported by the Office of Naval Research under Contract No. N00014-06-C-0060. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Office of Naval Research.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 31-08-2007	**2. REPORT TYPE** Technical	**3. DATES COVERED** *(From - To)* Feb 2006-Aug 2007

4. TITLE AND SUBTITLE

Strategic Mobility 21
Development of Joint Data Standards and Communication Protocols in An Integration Tracking System

5a. CONTRACT NUMBER
N00014-06-C-0060

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Mallon, Lawrence G.
Chen, Andrew

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

California State University, Long Beach Foundation
6300 State University Drive, Suite 220
Long Beach, CA 90815

8. PERFORMING ORGANIZATION REPORT NUMBER
Contractor Report 0009

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
875 North Randolph Street, Room 273
Arlington, VA 22203-1995

10. SPONSOR/MONITOR'S ACRONYM(S)
ONR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
CLIN 0009

12. DISTRIBUTION/AVAILABILITY STATEMENT

DTIC Distribution Statement A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Integrated Tracking System (ITS) supports the flow of freight to and from the Southern California Logistics Airport (SCLA) and will be integrated with the Inland Port-Multi-Modal Terminal Operating System. The ITS design integrates Contract Line Item Numbers (CLIN) 0009 Joint Data Standard and Communications Protocol, CLIN 0010 the Regional Wireless Network Design, and CLIN 0012 the Regional IT Data Network.

This project CLIN 0009 is to build a Data Center to support all Information Technology requirements from the ITS and eventually the Joint Deployment and Distribution Support Platform (JDDSP). The design work includes configuring secure data capture and integration networks, creating the information interfacing layer, and establishing the web interface. The algorithm platform is on the basis of Web 2.0 which is featuring Wireless, RFID, Ontology, Unified Modeling Language (UML), Metadata, and XML.

15. SUBJECT TERMS

Integrated Tracking System (ITS); Joint Deployment and Distribution Support Platform (JDDSP); Unified Modeling Language (UML); Southern California Logistics Airport (SCLA).

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. Lawrence G. Mallon
U	U	U		53	**19b. TELEPHONE NUMBER** *(Include area code)* 1-562-985-7392

Reset

CCDoTT

CENTER FOR THE COMMERCIAL DEPLOYMENT OF TRANSPORTATION TECHNOLOGIES (CCDoTT)
California State University, Long Beach

September 13, 2007

Dr. Paul Rispin, Program Manager
Office of Naval Research, Code 331
875 North Randolph Street, Room 273
Arlington, VA 22203-1995

Subject: Deliverable Number 0009, Joint Data Standard and Communications Protocol

Reference: Strategic Mobility 21 Contract N00014-06-C-0060

Dear Paul,

In accordance with the requirements of referenced contract, we are pleased to submit this Joint Data Standard and Communications Protocol Document for your review.

Your comments on this document are welcomed.

Regards,

Dr. Lawrence G. Mallon
Strategic Mobility 21 Program Manager

cc: Administrative Contracting Officer (Transmittal Letter only)
Director, Naval Research Lab (Hardcopy via U.S. Mail)
Defense Technical Information Center
Stan Wheatley

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The Integrated Tracking System (ITS) supports the flow of freight to and from the Southern California Logistics Airport (SCLA) and will be integrated with the Inland Port-Multi-Modal Terminal Operating System. The ITS design integrates Contract Line Item Numbers (CLIN) 0009 Joint Data Standard and Communications Protocol, CLIN 0010 the Regional Wireless Network Design, and CLIN 0012 the Regional IT Data Network.

This project CLIN 0009 is to build a Data Center to support all Information Technology requirements from the ITS and eventually the Joint Deployment and Distribution Support Platform (JDDSP). The design work includes configuring secure data capture and integration networks, creating the information interfacing layer, and establishing the web interface. The algorithm platform is on the basis of Web 2.0 which is featuring Wireless, RFID, Ontology, Unified Modeling Language (UML), Metadata, and XML.

1.0 INTRODUCTION

1.1 Background

The Joint Deployment and Distribution Support Platform (JDDSP) requires an Integrated Tracking System (ITS) based on integrated data standards and standard communication protocols. The ITS is designed to improve goods movement and provide effective decision support for the full spectrum of military surge deployments, sustainment support, redeployments, and reset operations. The JDDSP conceptual system architecture will be composed of three layers: (1) a data capture and integration layer; (2) a mediation layer; and (3) an information interfacing layer to perform Extraction, Transformation, and Loading (ETL) processes. The ETL processes allow the raw data to be converted to useful information for both commercial and military shipment management.

Accordingly, using data and data flows to represent the physical movement and key control elements in the shipment tracking system, we seek to define the Information Technology (IT) architecture required to capture and enable shipment and asset tracking. The tracking data required by the JDDSP include commercial and military Electronic Data Interchange (EDI) segments, Car Location Messages (CLM), Auto Equipment Identification (AEI) technology, Radio Frequency Identification (RFID) tags, and sensor readers to track containers and equipment moving to and through the JDDSP.

The initial development of the JDDSP-ITS started with building an integrated database, currently named the SM21 Tracking Experimentation Database (STED). Development of the system included configuring secure data capture and integration networks, creating the information interfacing layer, and establishing the web interface. Future development includes testing the STED within an integrated ITS architecture, which will support the JDDSP Inland Port – Multi-modal Terminal Operating System (IP-MTOPS) defined within the Multi-Modal Terminal Operating Software System Specification. The initial testing arrangement is depicted in Figure 1. Using this configuration and leveraging a cooperative agreement with Dole Foods, SM21 will integrate the container shipping data of Dole processed foods into and out of the Southern California ports and throughout the region.

1.2 Objectives

To develop joint data standards and communication protocols in the Integrated Tracking System, we have set the following objectives:

- A collaborative community of interest comprised of military and commercial logistics planning managers (i.e., shipper activity planners, rail planners, truck planners, intermediate node planners, marine terminal planners, vessel load planners, and end users);
- An automated and secure datacenter with multi-tier networks, server process, domain name controller, firewall routers, SQL/Oracle databases, backup/recovery plans, and various ETL programs;

- Strategic and operational logistics data standards and communication protocol which can be used to track vessel manifest, rail waybill, throughput velocity, equipment security, and shipping synchronicity of container and military equipment movement; and
- All shippers and regional Distribution Centers to have instant visibility of container and equipment shipments which can enable timely supply chain distribution and force deployment decisions using XML internet from everywhere in the world.

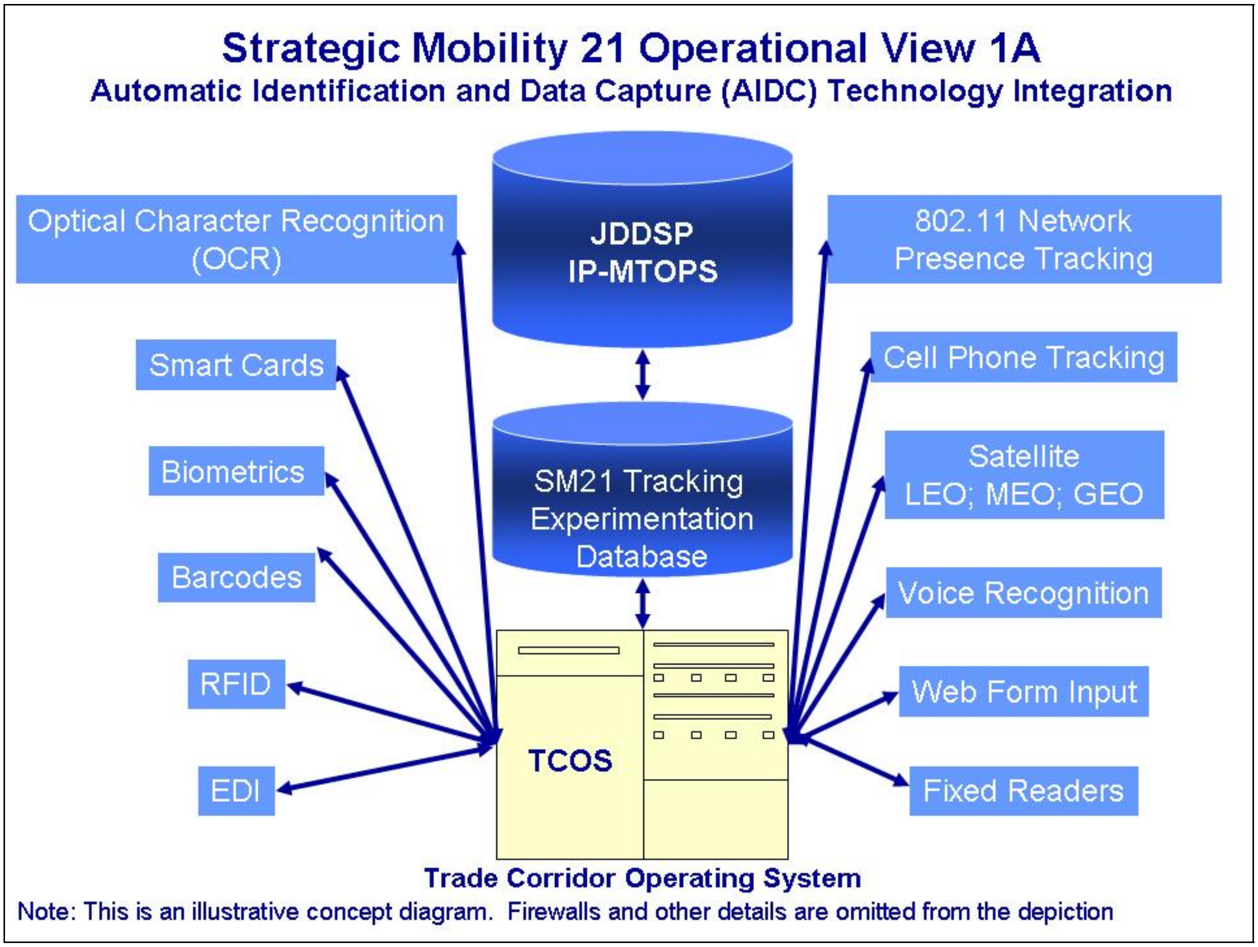

Strategic Mobility 21 Operational View 1A
Automatic Identification and Data Capture (AIDC) Technology Integration
JDDSP
IP-MTOPS
SM21 Tracking Experimentation Database
TCOS
Optical Character Recognition (OCR)
Smart Cards
Biometrics
Barcodes
RFID
EDI
802.11 Network Presence Tracking
Cell Phone Tracking
Satellite LEO; MEO; GEO
Voice Recognition
Web Form Input
Fixed Readers
Trade Corridor Operating System
Note: This is an illustrative concept diagram. Firewalls and other details are omitted from the depiction

Figure 1. Initial Testing Configuration

2.0 DATACENTER

2.1 Site and Hardware Allocation

The initial server site for the ITS is located at the Southern California Logistics Authority (SCLA), Victorville, California. The computer site and the respective hardware allocation map are shown in Figure 2. As outlined, the protocols of TCP/IP suite identify each device on the ITS network by its Internet Protocol (IP) address. So the router is a device with an IP address assigned to it. The Gateway router (IP: 172.16.1.1) is the entry point to communicate with external WAN and internet. The Virtual Private Network (VPN) router (IP: 66.162.38.10) is a device to provide firewall and security checking. From here, the networks split into two links to connect following two internal networks:

1. Local Area Networks (LAN) of Southern California Logistics Authority (SCLA), and
2. Local Area Networks of Victorville City Hall.

In the current configuration, the SCLA wireless firewall router (IP: 172.16.1.10) is the entry point to protect four servers as follows:

Table 1. Server Specifications

Server Name	IP Address	Functional Supports	Hardware Specifications
SM01	172.16.1.12	SQL Database Server	Dell PowerEdge 2650, Windows 2000 server, Hard disk 150 GBs
SM01	172.16.1.13	Web Portal Server	Dell PowerEdge 2950, Windows 2003 R2 Server, Hard Disk 300 GBs
SM03	172.16.1.14	Simulation and Oracle Server	Dell PowerEdge 2950, Windows 2003 R2 Server, Hard Disk 300 GBs
SM04	172.16.1.15	Geographic Information System Server	Dell PowerEdge 2950, Windows 2003 R2 Server, Hard Disk 1600 GBs

The four servers share a common wide screen monitor. All the workstations and laptops located at various sites are able to use Windows provided Remote Connections to sign into these four servers and run specific programs. The Victorville City Hall networks are not in the scope of this project and will not be included in this report.

2.2 Technical Support

The numerous ITS technical engineers are assigned to allocated positions. Together all job functions will be able to, locally or remotely, support all IT requirements at the Victorville Datacenter. Table 2 shows a summary of all technical supports by functions:

Table 2. Technical Supports by Functions

Positions	Number of Engineers	Job Functions
Chief Information Officer	1	Overall responsibility in IT planning, monitoring, budgeting, and Visioning.
Database Manager	1	SQL/Oracle implementation, data modeling, logical/physical design, backup/recovery, coding supports in data extraction/transformation/loading
Network Administrator	1	Implementation of WAN, LAN, routers, TCP/IP, Ethernet, DNS, and Active Directory
Server Room Operator	1	Power recovery, system reboot, application job maintenance, and documentation
Web Portal Manager and developers	6	SharePoint implementation, web planning & development, and web promotion
Application Developers	5	Development of stored procedures, triggers, Matlab programs, and stow plan software
GIS Developers	3	Implementation and development of ESRI IMS system and sizing on GIS server
Simulation Developers	4	Implementation and development of Rockwell Arena

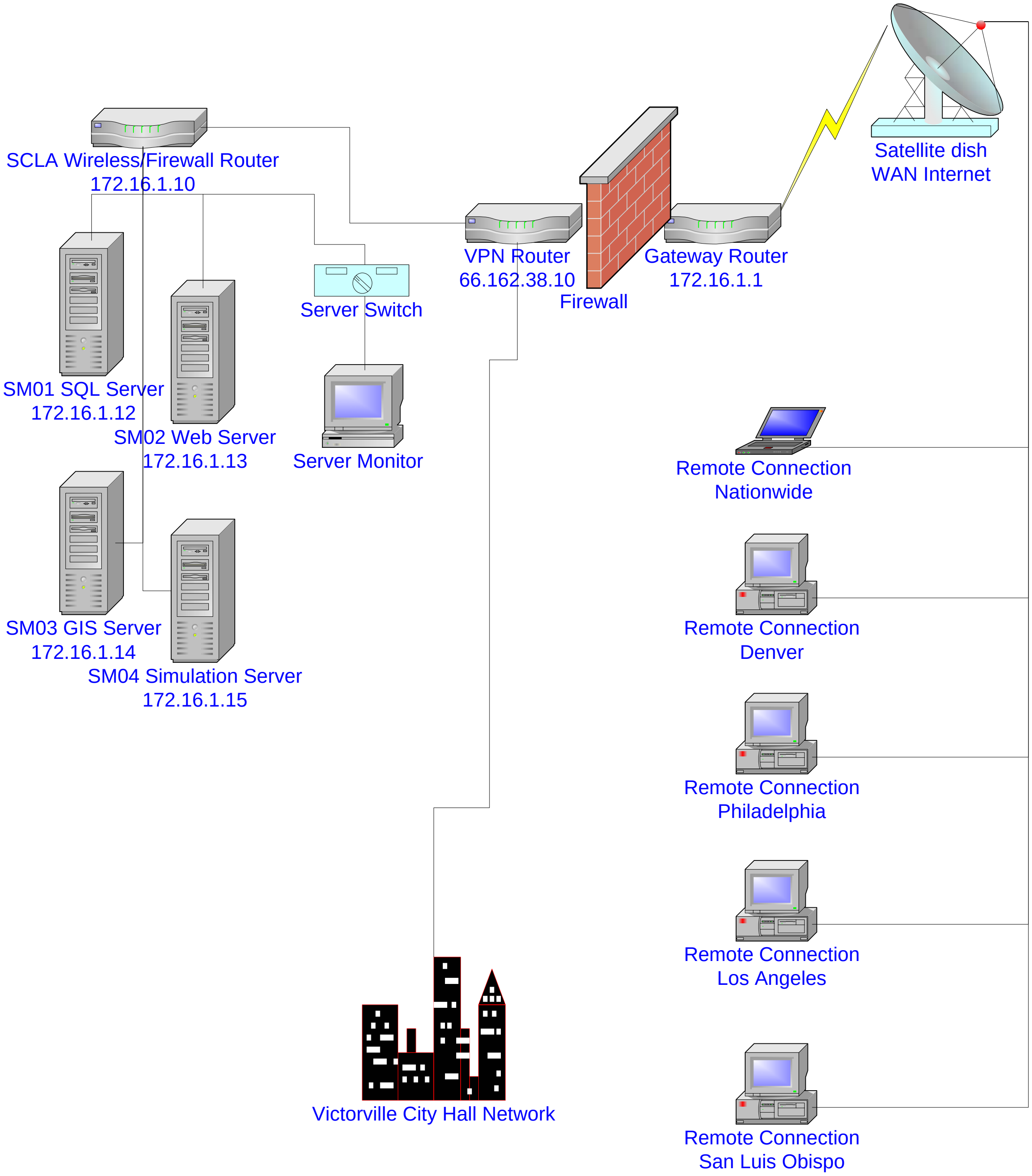


Figure 2. Computer Site and Hardware Allocation

3.0 CONFIGURATION OF SECURITY, ROUTERS, AND NETWORKS

The datacenter security consists of a series of hardware devices and supporting software applications and data communication. The devices include firewalls, routers, a Virtual Private Network (VPN), Encryptions, Domain Name Systems (DNS), Active Directory (AD), and SQL Authorization which will be described in the following sections.

3.1 Configurations of External Security Router, Firewall, Virtual Private Network

As shown in Figure 2, a Cisco PIX 501 router is implemented in the front-end WAN entry. This external router features first tier security protection of Gateway (IP: 172.16.1.1), Firewalls, and VPN (IP: 66.162.38.10). The purpose of this router is to permit, deny, and proxy connections configured by the overall security policy set by Victorville City Hall, SM 21, and Southern California Logistics Airport.

A Gateway is an entry and exit point from the datacenter networks to external WAN. A Firewall acts as a security guard, standing watch over data as it travels in and out of the Gateway. This PIX 501 router refers to a set of packet filtering rules (by ports and IP addresses) to determine what is allowed to pass as well as to keep unauthorized users out of the Gateway.

The VPN allows users to securely connect multiple computers over the Internet via IPSec, PPTP, and L2TP tunnels. The SM 21 designated users are authorized to use a name and a password to connect to this VPN (IP: 66.162.38.10). These Username/Password and authorization are currently managed by the top level Security Officer of the Network Administrator at Victorville City Hall.

3.2 Implementation of Internal Security Router, Firewall, Wireless, and Encryption

As shown in Figure 2, another D-Link router DI-824VUP (IP: 172.16.1.10) is implemented in the entry point of the datacenter’s Local Area Networks (LAN). The purpose of this router is to further screen out the incoming messages to access each of the four servers at the datacenter.

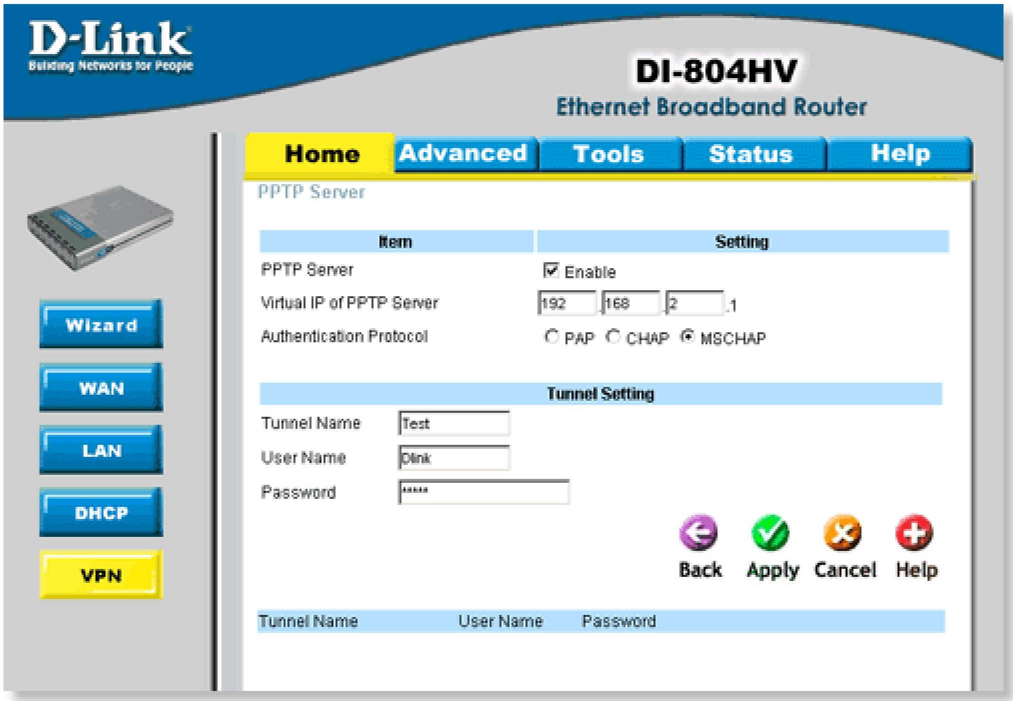
This D-Link device is a 4-Port Wireless Ethernet Broadband Router with Firewall, Wireless, and VPN capability. This router features the latest advanced wireless technology. It also includes robust Firewall security functions. Filters can be set based on MAC address, IP address, URL, and Domain Name. All these rules can be completely controlled by the SM 21 Network Administrator.

The procedures and specifications to configure this D-Link router are outlined as Table 3.

Table 3. Procedures and Specifications for D-Link Router Implementation

Procedures	**Specifications**
D-Link Router IP Address	192.168.0.1
Username/Password	XXXXX/YYYYY
VPN Channel	PPTP
Virtual IP Address	192.168.2.1
Authentication Protocol	MSCHAP
VPN Tunnel/Username/Password	Test/dlink/AAAA

This D-Link router provides a graphic menu for the configurations of WAN, LAN, DHCP, and VPN. This is a simple yet intelligent, web-based setup which can make the router secure for use by authorized SM 21 users to connect computers to share high-speed connections with the web portal, simulations, and other resources. To prevent unwanted Internet intruders from accessing the datacenter networks, this router serves as a solid Firewall through the use of filters. The major screen to set up the VPN is shown on Figure 3.



PPTP Server

Item	Setting
PPTP Server	☑ Enable
Virtual IP of PPTP Server	192 . 168 . 2 .1
Authentication Protocol	○ PAP ○ CHAP ● MSCHAP

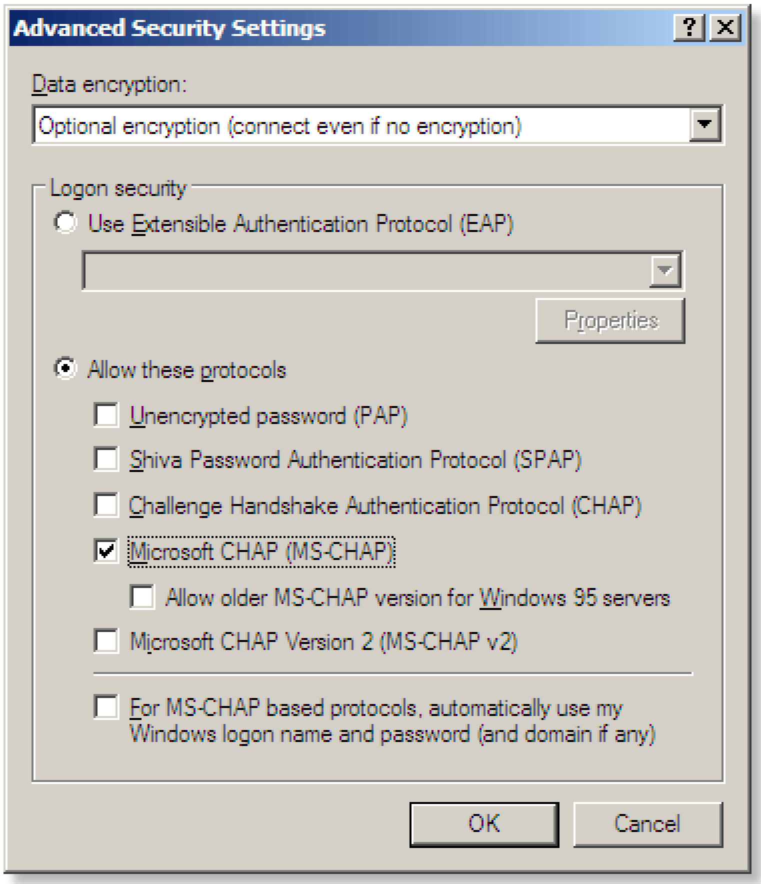
Tunnel Setting

Tunnel Name	Test
User Name	Dlink
Password	*****

Tunnel Name	User Name	Password

Figure 3. Configurations on D-Link Router VPN

As well, in the client PC workstation, a VPN connection must be configured appropriately to connect the internal router (192.168.0.1) with authorized Username and Password. The property security should be set as Figure 4.



Advanced Security Settings

Data encryption:

Optional encryption (connect even if no encryption)

Logon security

○ Use Extensible Authentication Protocol (EAP)

● Allow these protocols

- □ Unencrypted password (PAP)
- □ Shiva Password Authentication Protocol (SPAP)
- □ Challenge Handshake Authentication Protocol (CHAP)
- ☑ Microsoft CHAP (MS-CHAP)
  - □ Allow older MS-CHAP version for Windows 95 servers
- □ Microsoft CHAP Version 2 (MS-CHAP v2)

□ For MS-CHAP based protocols, automatically use my Windows logon name and password (and domain if any)

Figure 4. Configurations on Workstation VPN

Encryption is the process of obscuring information to make it unreadable without special knowledge, sometimes encryption is referred to as scrambling. Encryption can be used to ensure secrecy, but other techniques are still needed to make communications secure, particularly to verify the integrity and authenticity of a message. Therefore, the workstation VPN property must use Microsoft CHAP encryption to match the selection from the D-Link router as Figure 4.

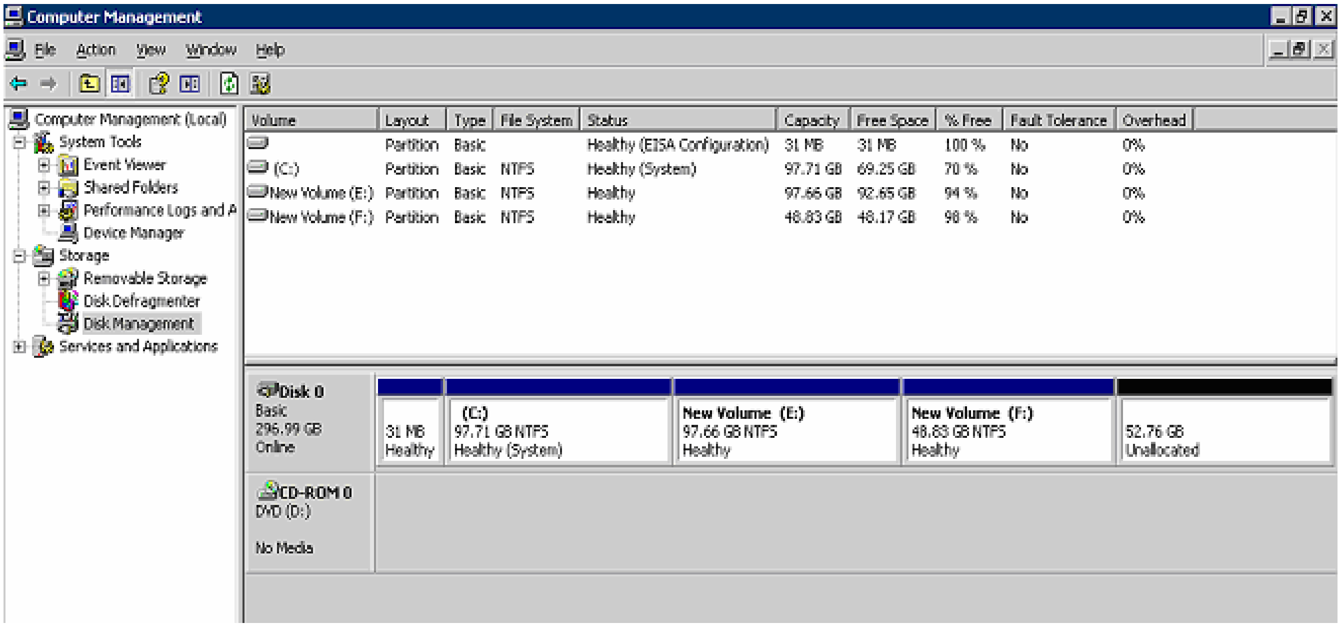
3.3 Connections of Servers In LAN

After passing the D-Link router, the user will be able to use another set of Username/Password to log into the servers located in the datacenter. All Usernames/Passwords and authorizations are managed by the SM21 Network Administrator.

4.0 CONFIGURATION OF OPERATING SYSTEMS, DOMAIN CONTROLLER, ACTIVE DIRECTORY, SQL SERVERS, AND VISUAL STUDIO

4.1 Reformat of Windows Operating System and Repartitions

The purchased Windows server had been partitioned with no more than 20 gigabytes in the system boot-up drive C:\. A SQL Server 2005 and a Visual Studio 2005 along will need more than 10 gigabytes to allow enough room for the future implementations of simulation software and Internet Information Services (IIS). As such, the new Windows 2003 server R2 (64 bits) required reformatting. The new disks are assigned 97 gigabytes to both C:\ drive and E:\ drive, and 48 gigabytes to F:\ drive. The remaining unused 52 gigabytes are reserved for future expansion. The D:\ drive is a default CD/DVD drive. Figure 5 outlines the disk management view after reformatting.



Volume	Layout	Type	File System	Status	Capacity	Free Space	% Free	Fault Tolerance	Overhead
	Partition	Basic		Healthy (EISA Configuration)	31 MB	31 MB	100 %	No	0%
(C:)	Partition	Basic	NTFS	Healthy (System)	97.71 GB	69.25 GB	70 %	No	0%
New Volume (E:)	Partition	Basic	NTFS	Healthy	97.66 GB	92.65 GB	94 %	No	0%
New Volume (F:)	Partition	Basic	NTFS	Healthy	48.83 GB	48.17 GB	98 %	No	0%

Disk 0 Basic 296.99 GB Online	31 MB Healthy	(C:) 97.71 GB NTFS Healthy (System)	New Volume (E:) 97.66 GB NTFS Healthy	New Volume (F:) 48.83 GB NTFS Healthy	52.76 GB Unallocated
CD-ROM 0 DVD (D:) No Media					

Figure 5. Disk Management on SM02 Server

4.2 Creation of Domain Name System and Domain Controller

A domain controller is a role that can be assigned to a server in a network of computers that use the Windows NT level operating systems. The Domain Name System (DNS) is implemented on this Domain Controller server. For security reasons, the datacenter networks use the a domain controller to manage all access to network resources (servers, databases, web portal, printers, and routers) for the SM 21 users, groups and other predetermined external users.

An Active Directory (AD) is created on the Domain Controller after the DNS and respective Service are appropriately configured. The Network Administrator uses the AD to create Username, Password, Roles, and Grant/Revoke authorization for a predetermined user. The user needs only to log in using the valid Username, Password, and Domain to gain access to the resources, which are located on the datacenter networks.

4.2.1 Domain Name System

The SM02 server is assigned to be the Domain Controller for the datacenter networks. The DNS is implemented on this server. Table 4 outlines the required procedures and respective specifications for the implementation.

Table 4. Procedures and Specifications for DNS Implementation

Procedures	**Specifications**
Windows Components of Control Panel	Networking Services
Networking Services	Domain Name System
Administrative Tools	DNS
Configuration Action	Forward Lookup Zone
Zone Name	Strategicmobility21.org
Zone File	Strategicmobility21.org.dns
Dynamic Update	No Dynamic Update
DNS IP Address	172.16.1.13

4.2.2 DNS Service

After the DNS is created, the DNS Service and Manager must be configured. The DNS Service is an internet service that translates domain names into IP addresses. Because domain names are alphabetic, they are easier to remember than their counterpart IP addresses. The Internet, however, is really based on IP addresses. Every time you use a domain name, a DNS service must translate the name into the corresponding IP address. A DNS Manager is used to monitor the operation of the DNS Service. Figure 6 shows an overview of the DNS Manager in the SM02 server.

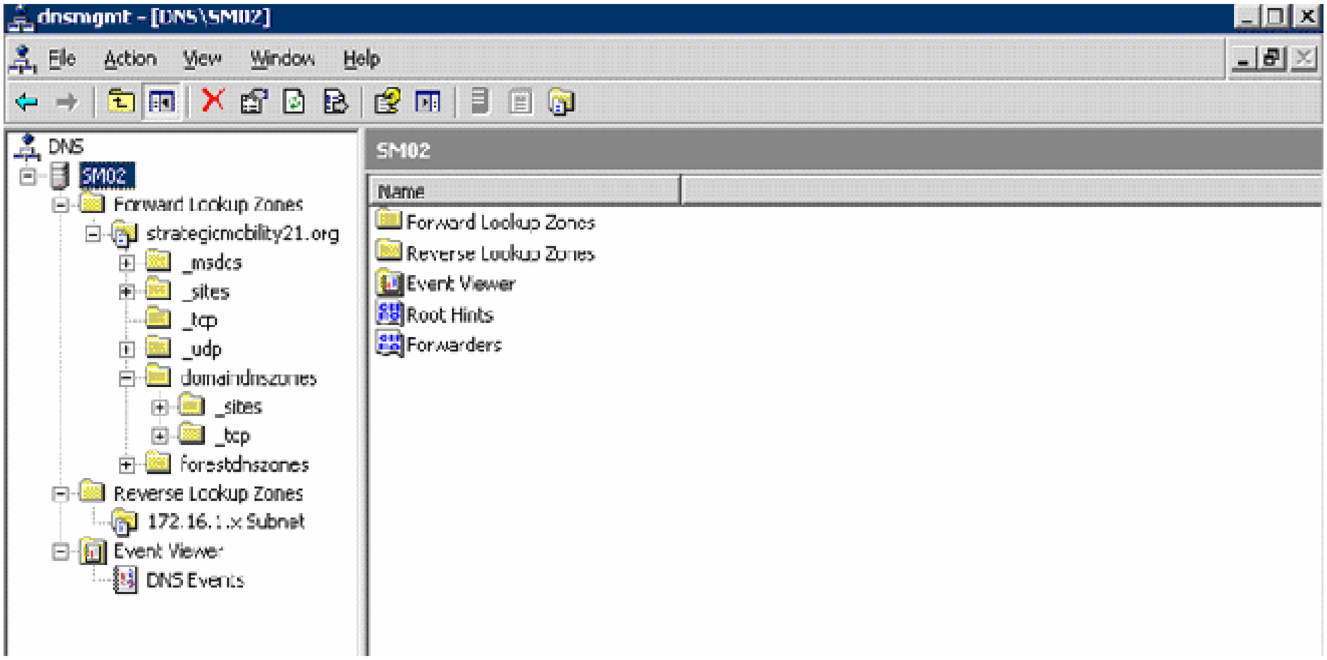


Figure 6. DNS Manager in SM02 Server

4.3 Creation of Active Directory

An Active Directory (AD) allows a Network Administrator to assign enterprise-wide Usernames, Passwords, Role authorizations, to deploy programs to many computers, and to apply critical updates to an entire organization. An AD is a directory service used to store information about the network resources across a domain. An AD structure is a hierarchical framework of objects. The objects fall into three broad categories: resources (e.g. printers), services (e.g. e-mail), and users (e.g. accounts, or users and groups). An AD provides information on the objects, organizes the objects, controls access, and sets security.

Each object represents a single entity whether a user, a computer, a printer, an application or a shared data source, and its attributes. Table 5 outlines all procedures and respective specifications for the AD implementation.

Table 5. Procedures and Specifications for AD Implementation

Procedures	Specifications
Run command	Dcpromo.exe
New domain	New forest
Default domain name	Strategicmobility21.local
Database and log files path	C:\winnt\ntds
Sysvol path	C:\winnt\sysvol
Password on AD	XXXXXXXXXXX

4.4 Installation of SQL Server 2005/2000 and Visual Studio 2005

A database is required to support a centralized datacenter and will be used by numerous users to read, write, update, and delete goods movement information. A Microsoft relational database system, SQL Server 2005/2000 Standard Editions, are implemented in the SM02/SM01 servers. This SQL Server includes following components:

- Server database engine
- Analysis services
- Reporting services
- Integration services
- Documentation and samples

A SQL Server 2005 Client Component is also implemented in numerous remote workstations. All designated users are able to connect to the SM02/SM01 SQL Servers using the SQL client in the workstations.

Another coding editor is required to help user programming process. A new Visual Studio 2005 is implemented and bundled with SQL Server 2005 in the SM02 server. This Visual Studio provides numerous editing functions to support the SM 21 project programming requirements in the following Languages:

- Transact SQL, Stored Procedures, Triggers (For SQL database loading)
- C# (For SQL data transformation)
- Visual Basic (For SQL data processing)
- HTML, XML, XSL (For web database query)
- ASP, JavaScript, VBScript (For web main menu)

5.0 OBJECT-ORIENTED UNIFIED MODELING LANGUAGE

The Object-Oriented Unified Modeling Language (UML) models completed in this section will be used by the Object-Oriented database design and Metadata export in Sections 6 and 7.

5.1 Background, Purpose, and Definitions

5.1.1 Object-Oriented

Traditionally, a program has been viewed as a logical procedure that takes input data, processing it, and producing output data. The programming challenge was seen as how to write the logic, not how to define the data. This method is still appropriate for use by a majority of applications developed today.

Object-Oriented (OO) database design and programming are new development practice organized around ‘objects’ rather than ‘actions’ and around ‘data’ rather than ‘logic’. The purpose of Object-Oriented database design and programming is to take the view that what we are interested in the objects to manipulate rather than the logic required to manipulate them.

An OO model of a database contains class, object, property, method, and association. Their major roles are to describe the data representation and object nature of relationships among entities. Table 6 illustrates OO surrogates and examples to be used by the Integrated Tracking System.

Table 6. Illustration of Surrogates Using Wireless Data Collections

SURROGATES	DEFINITIONS	EXAMPLES
Class	A certain related application entities, processing methods, and properties grouped together to represent a class.	Wireless tags, sites plus underneath programs, properties, associations, messages, and variables.
Instance	One instance data of a class representing one application data unit in a class.	One container.
Object	One physical representation in a class.	One container tag or a site location.
Property	A characteristic that describes an entity or an attribute.	All the data types and respective sizes of each object attribute in the central database.
Method	A function or a program that provides process for the class.	All subroutines, functions, stored procedures written to perform extraction, transformation, and loading to generate a central database for demonstration.
Association	A relationship that connect two entities or classes.	All foreign keys built in the central database.

5.1.2 Unified Modeling Language

The first step in OO is to identify all the objects you want to manipulate and how they relate to each other, which is often referred to as data modeling. In this Integration Tracking System, the UML will be used to analyze and design the models of object interface, class, activity, and deployment.

The UML is a standard modeling tool for specifying, visualizing, constructing, and documenting the artifacts of engineering systems, business systems, and other non-software systems. The UML represents a collection of best technical practices that have proven successful in the modeling of large and complex systems. The UML is also a major tool for developing an Object-Oriented system and its software development process. The UML uses mostly graphical notations to express the design of software projects.

The purposes of the UML used in this ITS are to:

1. Provide team members (of simulations, networks, wireless tracking, supply chains) with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models,
2. Provide extensibility and specialization mechanisms to extend the core wireless tracking concepts,
3. Be independent of particular programming languages (Stored Procedures, XML, and C#) and development processes,
4. Support higher-level development concepts such as collaborations, frameworks, patterns, and components, and
5. Integrate best practices for future system developments.

5.2 Objected-Oriented UML Analysis and Design

The UML diagrams are designed to let developers and stakeholders view an application system from different perspectives and in varying degrees of abstraction. Four UML diagrams which are commonly created in visual modeling tools are outlined in the following sections.

5.2.1 Use Case Diagram

A use case is a set of scenarios that describe an interaction between a user and a system to complete a task. A use case diagram displays the relationship among actors and use cases.

An Actor is the representation of a person or system which exists outside the system under study and who (or which) performs a sequence of activities in a dialogue with the system. A Use Case represents a single interaction between a primary actor (who initiates the interaction) and other (secondary) actors, and the system itself. The interaction is presented as a sequence of simple steps.

Use case development starts by listing a sequence of steps that a shipment and a stakeholder might take in order to complete an action. In the case of a container shipment, the following steps might apply:

1. Scan a container tag in a remote site.
2. Push records to central database server.
3. Extract, transform, and load records into a database.

Then a user stakeholder will continue with the following steps:

1. Select a truck route or a train route to browse container information.
2. Execute a stored procedure to select required records.
3. Generate XML and XSL document listing.
4. Display container information on a web.

These steps are represented in the following Use Case Diagram (Figure 7).

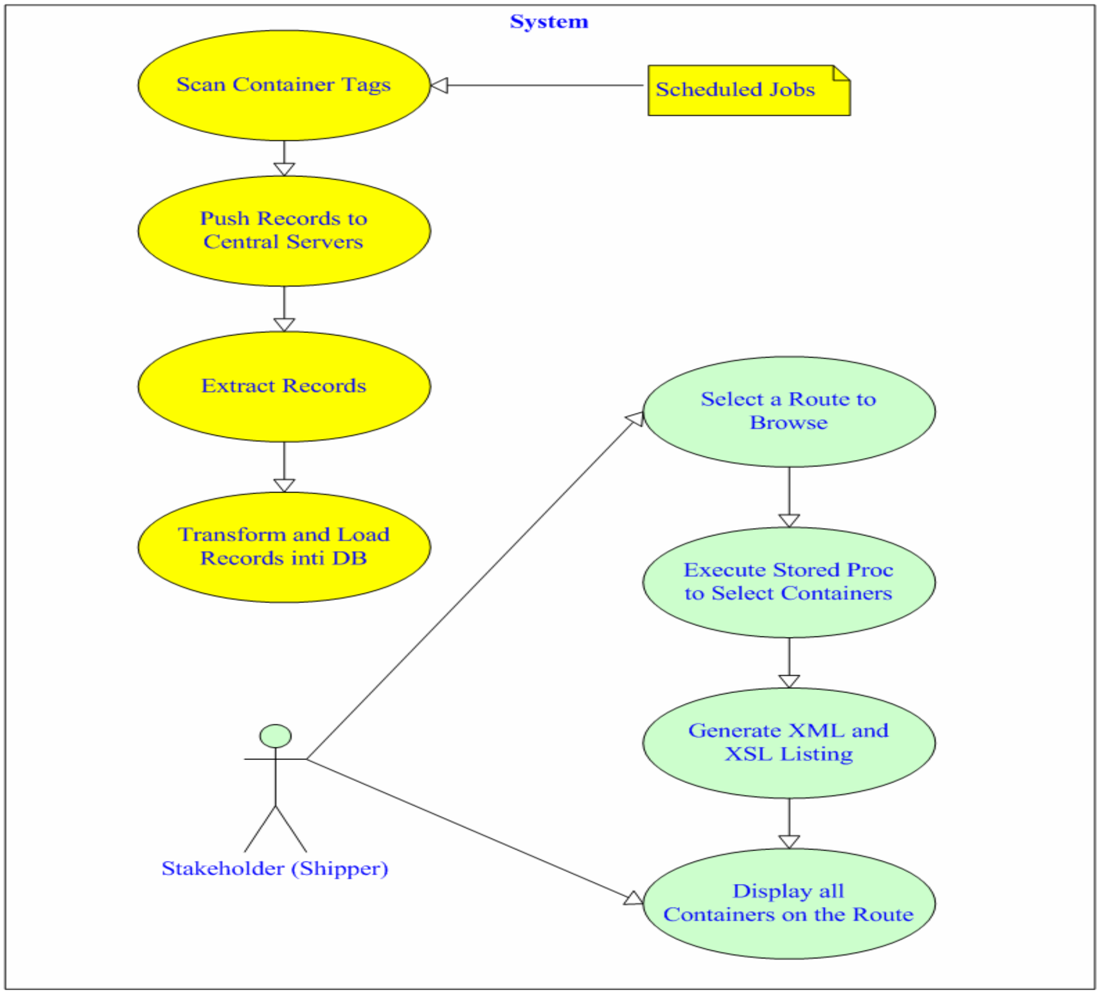

System
Scan Container Tags
Scheduled Jobs
Push Records to Central Servers
Extract Records
Transform and Load Records inti DB
Select a Route to Browse
Execute Stored Proc to Select Containers
Generate XML and XSL Listing
Display all Containers on the Route
Stakeholder (Shipper)

Figure 7. Use Case Diagram

5.2.2 Class Diagram

A class diagram is widely used to describe the types of objects in an OO system and their relationships and behavior (process). A class diagram configures class structure and behavior using design elements such as classes, objects, associations, and methods. Basically, a class diagram describes three different perspectives of:

1. Conceptual,
2. Specification, and
3. Implementation.

These perspectives can be transformed into five elements in a class:

1. Objects
2. Instances
3. Properties
4. Associations
5. Methods

These elements become evident as the diagram is created and help solidify the design. A class diagram can display an association (relationship) such as containment, inheritance, and association. An association shows the relationship between instances of classes. The multiplicity of the association denotes the number of objects that can participate in their relationship (1 to 1, 1 to N). The Integration Tracking System can be configured by using these elements to generate a Class Diagram as Figure 8.

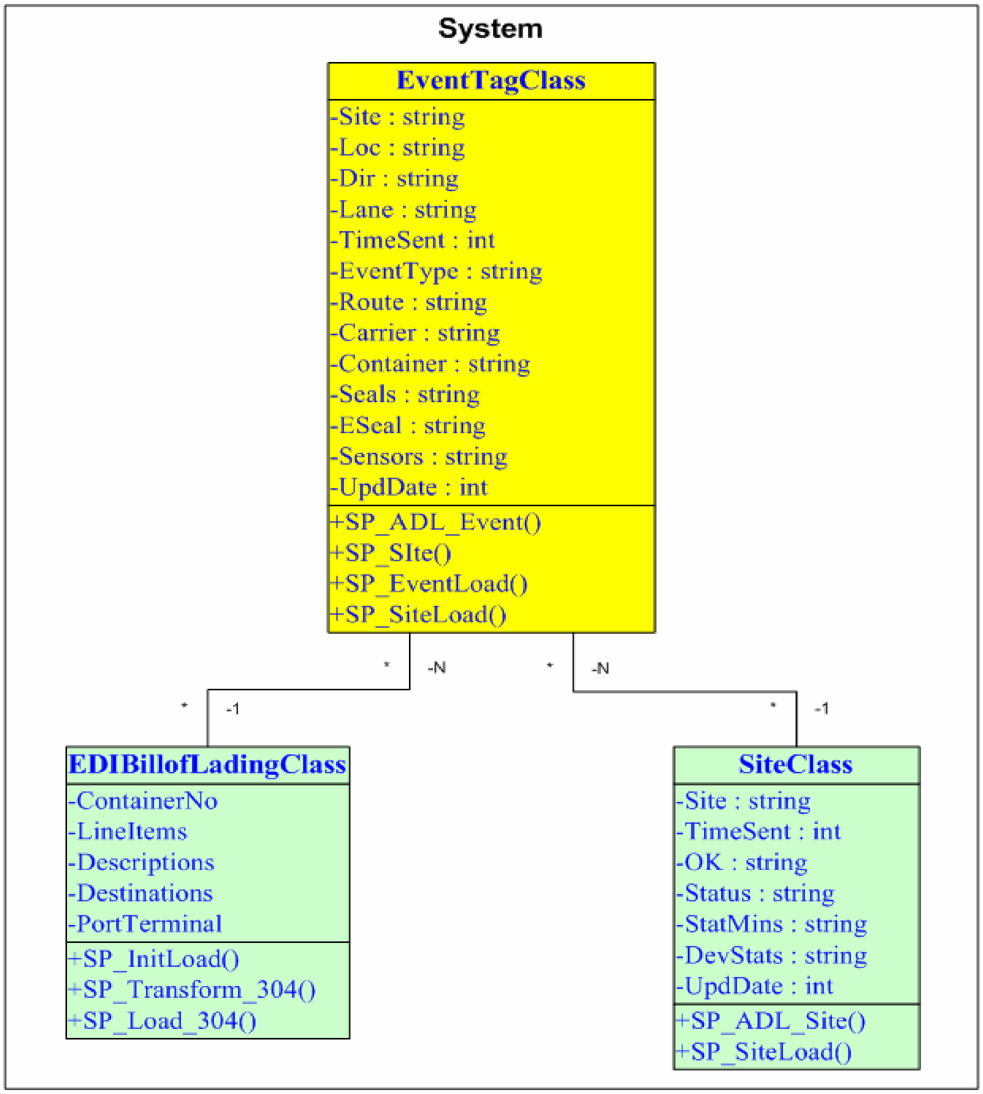


Figure 8. Class Diagram

5.2.3 Activity Diagram

An activity diagram describes the workflow behavior of a system. It can show activities that are conditional or parallel. An activity diagram is also useful for: (1) Analyzing a use case by describing what actions must take place and when they should occur; (2) Describing a complicated sequential algorithm; and (3) Modeling applications with parallel processes.

Activity diagrams are read from top to bottom and use branches and forks to describe conditions and parallel activities. A fork is used when multiple activities are occurring at the same time. Figure 9 shows how the activities of a wireless tag record are collected and where they are transferred.

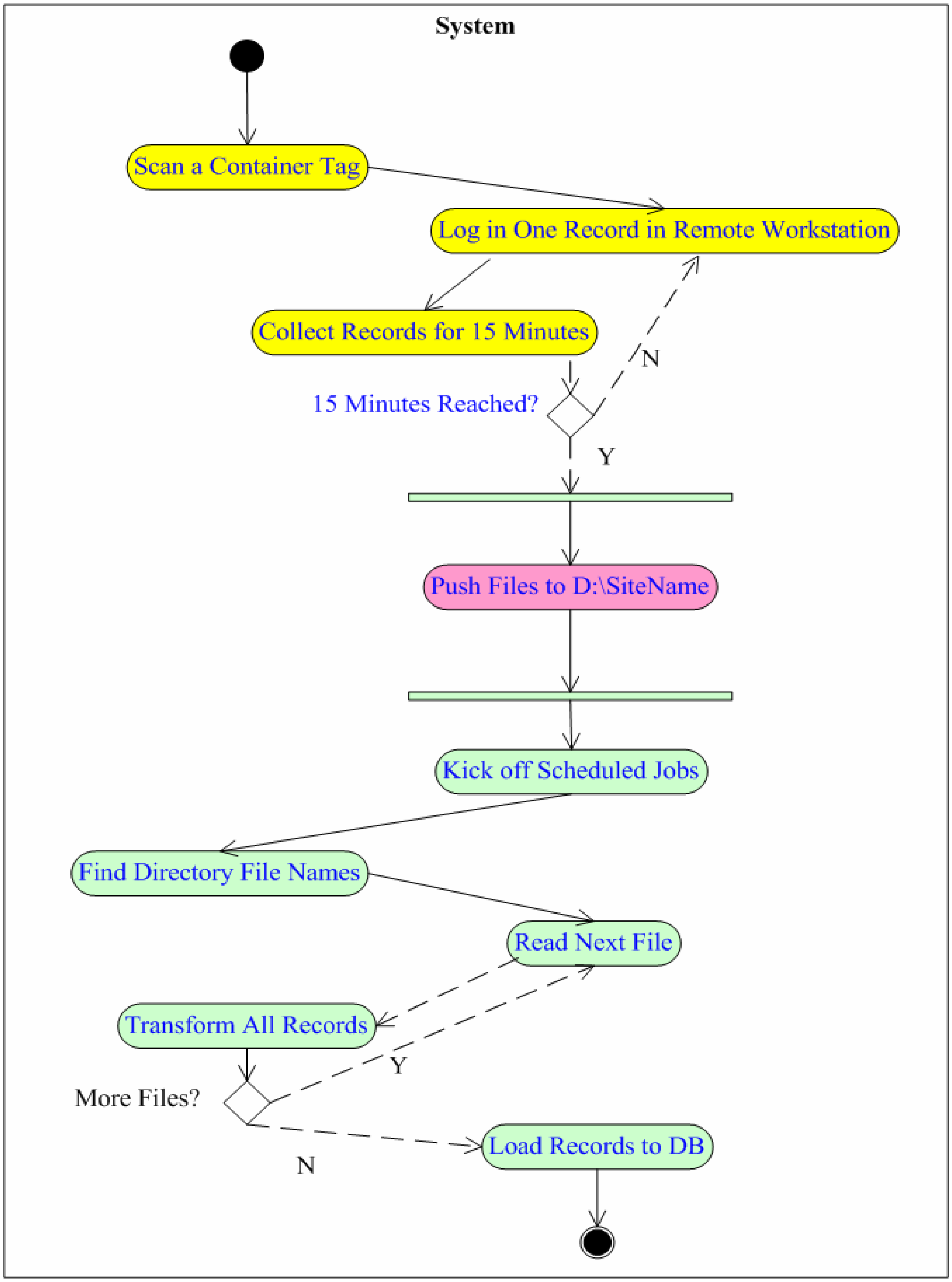


Figure 9. Activity Diagram

5.2.4 Physical -Component Diagram and Deployment Diagram

There are two types of physical diagrams: deployment and component. A deployment diagram shows the physical relationship between hardware and software in a system. A component diagram shows the software components of a system and how they are related to each other.

Many times the deployment and component diagrams are combined into one physical diagram. A combined deployment and component diagram consolidates the features (hardware and software) of both diagrams into one diagram.

The deployment diagram contains nodes and connections. A node usually represents a piece of hardware such as a server or a router in the ITS. A connection depicts the communication path used by the hardware to communicate and usually indicates a method such as TCP/IP and Ethernet.

The component diagram contains components and dependencies. Components represent the physical packaging of a module of codes (stored procedures or functions). The dependencies between the components show how changes made to one component may affect the other components in the system. Dependencies in a component diagram are represented by a dashed line between two or more components.

Figure 10 below shows the combined deployment and component diagrams providing a high level physical description of the completed system. In the Datacenter, there are four servers of SM01, SM02, SM03, and SM04. Each server has been loaded with critical application software and databases. Figure 10 provides the reader with a quick overall view of the entire system.

All four UML models will be used as a basis to develop OO databases and programs in the next sections.

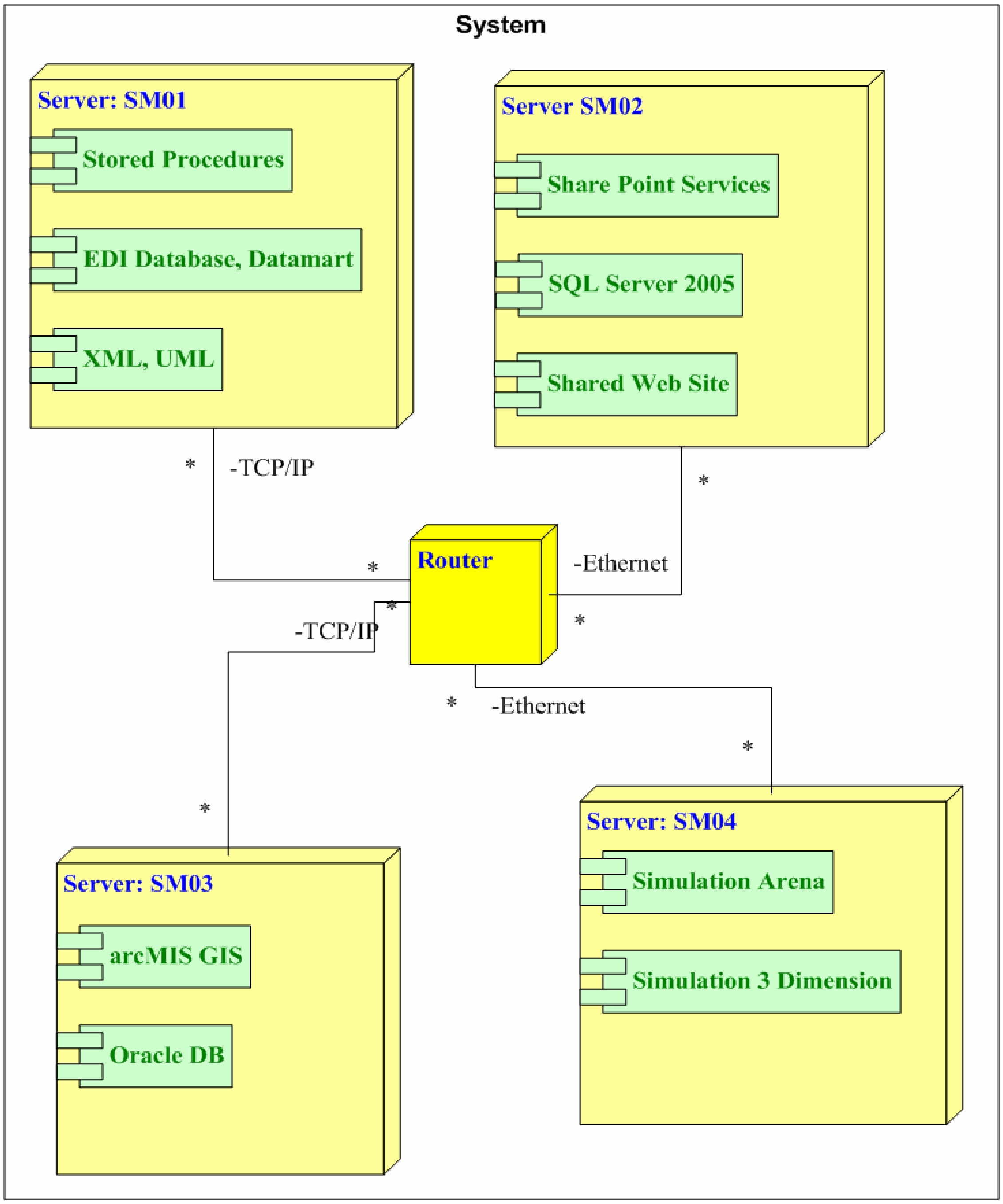


Figure 10. Combined Diagrams of Component and Deployment

6.0 INTEGRATION OF DATABASES AND DATACENTER

The tasks of Object-Oriented database design completed in this section were used for the Metadata creation found in Section 7.

6.1 Data Collection and Transfer System for Radio Frequency Identification

A data collection and transfer system provided by RFID is proposed for this project. This system includes both site sensor controller hardware and software (remote sites), and Datacenter hardware and software. One remote site may support several locations (i.e. East Gate and West Gate) where the sensors are implemented separately. All the site controllers share a common set of software that is uniquely configured for each site. The site controller is responsible for collecting data from each location, as well as interfacing with sensor hardware, buffering sensor hardware status and RFID read events, and reporting to the Datacenter (via a secure Internet connection). The Datacenter receives reports from all of the site controllers in near real-time basis. The reported status and events are processed and stored in the JDDSP database. The XML Web application system provides near real-time access to this information for authorized users on the Internet via a secure VPN website (http://SM02/Regional_Planning/Defaultaspx). The JDDSP database also receives and distributes event reports from other sources such as the seaports and weigh stations via their respective information management systems (IMS). The data collection and transfer are displayed in Figure 11.

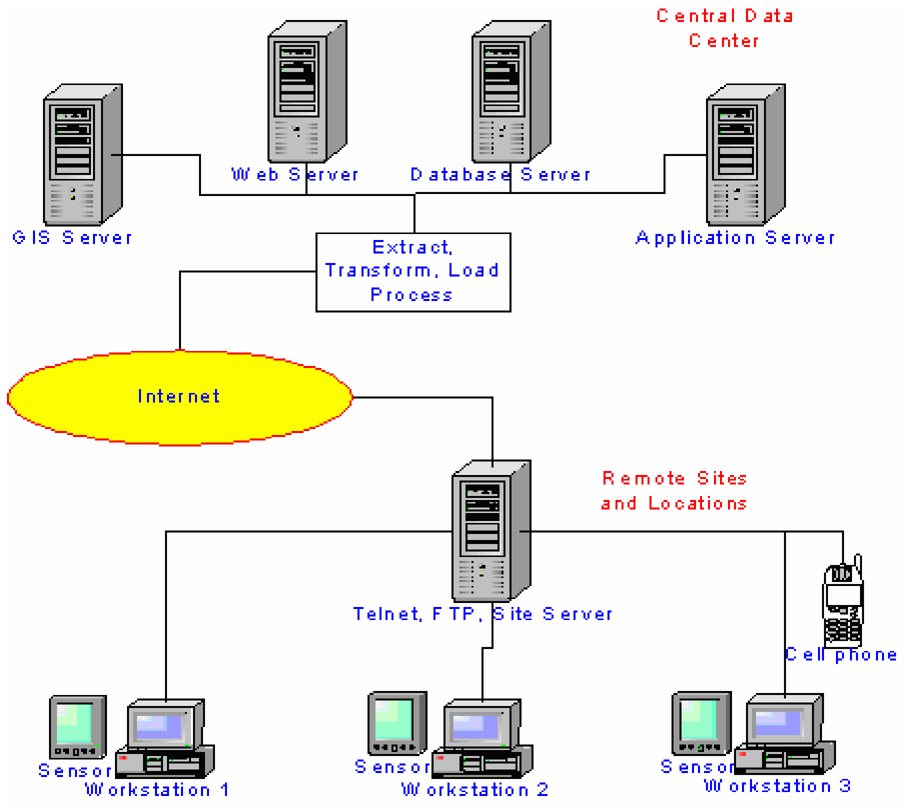


Figure 11. Data Collection and Transfer System of RFID

This data collection and transfer system consists of:

- Data Acquisition (by Sensors)
- Data Collection (by Workstations)
- Data Transfer (by internets)
- Data Loading (by SQL process)
- Data Demonstration (by XML Webs)

This software is developed in a flexible baseline collection system which can be easily adapted to other similar devices, i.e. CLM, AEI, Savi RFID technology, eSealing, etc. If necessary (for security or other reasons), the software can be cloned and modified to support other railways, warehouses, gate operation, or trade lanes at remote physical locations. This data collection and transfer system is not limited to commercial vehicle trucks, but also handles marine and rail transportation equipment. The software architecture of the datacenter and the remote Site Controller allows rapid revision and integration to support other device drivers, new sensors, and communication protocols as they become necessary.

Several data collection and record reformats are conveyed in the following sections.

6.2 Trade Corridor Operating System

The Trade Corridor Operating System (TCOS) is a system of hardware and software that was developed by TransCore Corporation originally for the Washington State Department of Transportation (WSDOT) to implement the Northwest International Trade Corridor and Border Crossing System (NWITC). The ports of Seattle (American President Lines (APL)) and Tacoma (Maersk Sealand) are integrated into the system along with the Washington State Commercial Vehicle Information Systems and Networks (CVISN) weigh stations and commercial vehicle database.

These sites are outfitted with roadside Automatic Vehicle Identification (AVI) sensors (to read CVISN Radio Frequency Identification vehicle tags) and other wireless sensors for reading electronic container seals (e-seals) and Free and Secure Trade (FAST) compatible vehicle sticker tags and driver/crew ID cards.

As a part of ITS, TCOS will capture sensor and other tracking data. As an example, TCOS will be used to pass e-seal tracking data to the JDDSP database. When a container sealed with an e-seal tag passes a gate sensor site the tag will be read and the record will be logged into the site workstation (a PC). Every 15 minutes, a file with all scanned records will be sent (PUSHed) to the JPPSP (SCLA) datacenter. A sample scanned record layout is provided in Table 7. This file in text format contains certain required data from a site scan reader.

Table 7. TCOS Event Record Layout

Field Name	Data Type	Length
Location	Varchar	32
Direction	varchar	20

Lane	Varchar	4
Time	Datetime	7
EventType	Varchar	20
Update	Varchar	1
Route	Varchar	32
Latitude	Numeric	5,2
Longitude	Numeric	5,2
VehType	Varchar	20
Veh	Varchar	32
Carrier	Varchar	32
Tag	Varchar	24
TagStatus	Varchar	20
Signal	Varchar	20
SignalReason	Varchar	20
TagData	Varchar	4000
Sticker	Varchar	16
StickerStatus	Varchar	20
VehGPS	Varchar	24
VehGPSStatus	Varchar	20
Crew	Varchar	100
CrewStatus	Varchar	100
Container	Varchar	12
ContainerChk	Varchar	2
ContainerTag	Varchar	24
ContainerStatus	Varchar	20
ContainerGPS	Varchar	24
ContainerGPSStatus	Varchar	20
ContainerDegreeC	Varchar	100
Agent	Varchar	24
Inbound	Varchar	17
Chassis	Varchar	24
ChassisTag	Varchar	24
ChassisStatus	Varchar	20
Seals	Varchar	100
eSeal	Varchar	24
eSealData	Varchar	4000
eSealStatus	Varchar	100
Tamperature	Varchar	1
Unit	Varchar	24
Sensors	Varchar	100
SensorStatus	Varchar	100
SensorCounts	Varchar	100
SensorSyncMins	Int	4
Reefer	Varchar	24

ReeferStatus	Varchar	24
ReeferDegreeC	Varchar	100
GeneratorSet	Varchar	24
GeneratorSetStatus	Varchar	24
GeneratorSetFuel	Numeric	5,2
Cargo	Varchar	100
CargoStatus	Varchar	100
Inspector	Varchar	24
GrossLbs	Numeric	10,2
HazMaterial	Varchar	1
OffRoute	Varchar	1
Timeout	Varchar	1
Evaluation	Varchar	24
EvaluationStatus	Varchar	20
VehClass	Int	4
VehGrossLbs	Numeric	8,2
VehLength	Numeric	5,2
VehHieght	Numeric	5,2
VehWidth	Numeric	5,2
VehSpeed	Numeric	5,2
VehAxles	Numeric	5,2
AxleSpeed	Varchar	100
AxleLbs	Varchar	100
LeftAxleLbs	Varchar	100

6.3 Car Location Messages

The information gathered by railcar AEI tag readers can be used by railroad traffic managers or customer service departments to locate a shipment for a customer at almost any point along its route. It also helps out in waybill generation by eliminating the information entry errors normally associated with manual reporting. These CLM message data sets are useful when tracking a train through certain areas to monitor train movement progress and determine delays and impediments to the train transit.

CLM are generated after a train passes an AEI tag reader site. CLM contain predetermined railcar location data like inventories of yards and train consists. The message records are generated by passive electronic tags affixed to railcars, one on each side of the railcar, passing electronic reader antennae located at specific points in the rail system. The collection of this information is critical for the management of an efficient rail schedule and for the asset management.

The CLM provides electronic connectivity among suppliers, shippers, and carriers with the exchange of vital transportation data sets. It serves as a single source of data collection from all reporting railroads to update rail shipments. CLMs are available from all Class 1 carriers and over 250 Shortline railroads.

An example of a CLM message record layout provided by Military Surface Deployment and Distribution Command (SDDC) is provided in Table 8.

Table 8. CLM Record Layout

Record Positions	Field Names
1	Equip Initial
5	Equip No
11	PO No
33	BOL
83	Ship Date: MMDDYYYYHHMI
95	Route
130	Origin City
160	Origin state/Prov
162	Destination City
192	Destination State/Prov
194	Shipper
254	Consignee
314	Original Estimate Time Arrived: MMDDYYYYHHMI
326	Dynamic Estimate Time Arrived: MMDDYYYYHHMI
338	AAR Code
339	CLM Code
348	CLM City
357	CLM State
359	CLM Destination
368	CLM Datetime: MMDDYYYYHHMI
380	Report Railroad
384	Note

6.4 Automatic Equipment Identification Data Collection

In the early 1990s something new came along that allowed railroads to retire their pencils and automate the way they keep tabs on a rolling stock. It is called Automatic Equipment Identification. AEI files are text files in a format designated by the Association of American Railroads (AAR). They contain key data elements such as the Railcar Number, the length of train and speed and direction at time of passing an AEI reader site, as well as the date and time of passing the reader site. The files are sent after passing a reader site via dialup service to the railroad which uses them to verify shipments and assets.

AEI messages provide information about railcars passing a certain point in the railcar tracking system of the railroad. The AEI tags have two record formats: the T94 Header and T94 Detail. The T94 Header format is shown in Table 9.

Table 9. T94 Header Layout

Record Positions	Field Names
4	Site ID
11	Event start date
19	Event start time
22	Event stop time
26	Time zone
29	Daylight savings indicator
30	Data format version
33	Sequence
37	Conversion status (locomotives)
38	Conversion status (cars)
39	Direction of travel
40	Switch direction
41	Unit of measure
42	Max train speed
45	Min train speed
48	Average train speed
51	Train movement type
52	Termination status
53	Transmission type
54	Adjacent track occupied
55	Train length
60	Equipment status
58	Total locomotives
60	Tagged locomotives
65	Total equipment
68	Tagged equipment
71	Total axles

The T94 Detail format is shown in Table 10:

Table 10. T94 Detail Layout

Record Positions	Field Names
4	Sequence number
7	EGC number
8	Equipment initial
12	Equipment number
22	Orientation
23	Reserved
24	Axle conversion code
26	Tag detail status

27	Hand shakes antenna 0
29	Hand shakes antenna 1
31	Speed
34	Axle count
36	Platform count

6.5 Electronic Data Interchange

Since 1983, the transportation industry in US has been using the American National Standards Institute (ANSI) accredited Standards Committee X12 Electronic Data Interchange (EDI) format (Muller, 1999), even though US rail and motor carriers EDI usage were non-compatible. The purpose of EDI data transmission is to allow transaction sets of different types to be transmitted from one party to another in the same structure. This hierarchical structure of headers and trailers allows the data to be segregated logically for easy interpretation by a receiver in the same industry. For the requirements to build the JDDSP database, we highlight the following data sets which are requested and collected in the related areas of ocean, highway, and rail shipments:

204 Motor Carrier Load Tender

This transaction set can be used to allow shippers or other interested parties to offer (tender) a shipment to a full load (truckload) motor carrier including detailed scheduling, equipment requirements, commodities, and shipping instructions pertinent to a load tender. It is not to be used to provide a motor carrier with data relative to a Less-than-Truckload bill of lading, pick-up notification, or manifest.

214 Transportation Carrier Shipment Status Message

This transaction set can be used by a transportation carrier to provide shippers, consignees, and their agents with the status of shipments in terms of dates, times, locations, route, identifying numbers, and conveyance.

304 Shipping Instructions

When this transaction set is transmitted to an ocean carrier, it provides all the information necessary to prepare and distribute a contract of carriage such as an ocean bill of lading, sea waybill, and other shipping documents. When this transaction set is transmitted to a freight forwarder or customs broker, it provides for the transmission of shipping and financial information required by the forwarder or customs broker to move cargo and provide the services requested.

309 Customs Manifest

This transaction set can be used by carriers, terminal operators, port authorities, or service centers to provide Customs with manifest data on cargo arriving in or departing from oceangoing vessels, railroad trains, or other types of conveyances. This transaction set can be also used by carriers to provide terminal operators, port authorities, or service centers with manifest data on cargo arriving at their facilities via the conveyances mentioned above.

315 Status Details (Ocean)

This transaction set can be used to provide all the information necessary to report status or event details for selected shipments or containers. It is intended to accommodate the details for one status or event associated with many shipments or containers, as well as more than one status or event for one shipment or container.

322 Terminal Operations and Intermodal Ramp Activity

This transaction set can be used to provide all the information necessary for a terminal operation, port authority or intermodal ramp to communicate terminal and intermodal ramp activities (e.g., "ingates" and "outgates") to authorized parties to a shipment.

404 Rail Carrier Shipment Information

This transaction set can be used to transmit rail-carrier-specific bill of lading information to a railroad. It is the initial tender of a shipment between a consignor and a rail carrier and can be used as notification of equipment release and/or a legal bill of lading.

418 Rail Advance Interchange Consist

This transaction set can be used to transmit advance information on equipment being interchanged to a connection rail carrier, from a consignor or to a consignee.

856 Ship Notice/Manifest (Generally used for commercial Shipments)

This transaction set can be used to list the contents of a shipment of goods as well as additional information relating to the shipment, such as order information, product description, physical characteristics, type of packaging, marking, carrier information, and configuration of goods within the transportation equipment. The transaction set enables the sender to describe the contents and configuration of a shipment in various levels of detail and provides an ordered flexibility to convey information. The sender of this transaction is the organization responsible for detailing and communicating the contents of a shipment, or shipments, to one or more receivers of the transaction set. The receiver of this transaction set can be any organization having an interest in the contents of a shipment or information about the contents of a shipment.

858 Shipment Information (Mainly used for Government Shipments)

This transaction set can be used to provide the sender with the capability to transmit detailed bill-of-lading, rating, and/or scheduling information pertinent to a shipment. This transaction set can also be used to exchange Government Bills of Lading (GBLs), Commercial Bills of Lading (CBLs), Transportation Control and Movement Documents (TCMDs), and Personal Property Government Bills of Lading (PPGBLs). It may be used by the U.S. Civilian Government, U.S. Department of Defense, and any of their trading partners to exchange information about the shipment of freight, household goods, and passengers. This transaction set fulfills information requirements established by U.S. Government transportation movement rules and regulations.

6.6 UMLER and Supporting Data

In order to provide certain functions in the database, specific supporting data are required. A Universal Machine Language Equipment Register (UMLER) is a data file that has all freight equipment characteristics and specifications recorded for proper use of rail industry equipment.

These data records are maintained by the Business Services Division Interline Service of the Association of American Railroads, RAILINC. These data are contained in the UMLER file. Specific excerpts from this very large UMLER file are normally used in the calculation of a train consist in our database ETL process. We also use this file to extract length, width, height, and weight of a container in this project. A sample file was provided free of charge here. This file in text format (Table 11) contains certain required data for all intermodal equipment in North America.

Table 11. UMLER File Layout

Field Name	Position	Length	Description
Mark	1-4	4	Alpha mark of car
Number	5-10	6	Numeric of car
Articulated Count	11-12	2	Number of platforms in set
Articulated Code	13-14	2	Articulated only
Equipment Type Code	15-18	4	For articulated, type is one position
Outside Length	19-23	5	Base only
Platform Length	24-27	4	
Platform Width	28-31	4	Base only
Bulkhead Height	32-35	4	Base only
Tare Weight	36-39	4	
Load Limit	40-43	4	
Trailer Capacity Code	44-45	2	Articulated only
Container Capacity Code	46-47	2	Articulated only
Axles	48-49	2	Base only

6.7 Object-Oriented Database Design

The integration of required RFID, CLM, AEI, and EDI data into an OO database is necessary for successful shipment and transportation asset tracking and management. The purposes of the JDDSP database design are to:

- Automate the ETL process and store the data without unnecessary redundancy.
- Allow users to query information in a net centric environment.
- Provide tracking services within a Service Oriented Architecture
- Secure all data to meet predetermined military and commercial policies.
- Configure the databases and the programs using objects and classes.

Therefore, a key step to the OO design is to establish relationships among objects. We define an entity as an object about which we store data, e.g., physical thing, person, place, event; establish relationships between entities, i.e., 1-to-1, 1-to-many, many-to-1; and form relational data model structure in only two-dimensional tables whereby columns represent attributes, and each row has a unique key.

The next major step in designing relations is to apply integrity rules involving five constraints:

- Primary key
- Foreign key
- Unique key
- Check function
- Null value

Maintaining a good relation and constraint integrity is helpful to avoiding table anomalies in the user database query such as insertion, deletion, and modification. Furthermore, the so-called normalization process must ensure functional dependencies, i.e., one-way relationship, with a determinant and another attribute dependent on it. We also decompose a relation into a group of smaller relations in accordance with the principles of a rationale for assigning attributes to entities and concepts of normal forms.

Of course, one significant goal of a database is for external application uses. A convenient query-response capability facilitating such applications is a structural query language called SQL with the following features:

- UML class model
- Tables and data
- Database and table structures – name, keys, indexes, columns, data types, null values
- Entity integrity
- Data entry and data management
- Queries and mathematical operations

We went through a number of UML data modeling activities based on each area of RFID, CLM, AEI, and EDI sources. The models contain classes, objects, entities, attributes, relationships, and entity-relationship diagrams (ERD). The ERD is used to map out the logical structure image of a database. The major role here is to describe the data representation and object nature of relationships among entities in the overall database. This process follows a development life cycle to yield an entity-relationship diagram that was used to develop the physical data modeling presented in the following section.

The physical data modeling process is used to convert the ERD and normalized images into a physical model. A few objects are converted into equivalent counterparts in the physical model:

- Entity to Table
- Attribute to Column
- Tuple to Row
- Association to Relationship (Foreign Key)
- Candidate Key to Primary Key and Index
- English Object Name to Physical database name

An end user will be able to easily join several tables using foreign key relationships to query each specific transportation control number, shipment, container, railcar, shipping document, shipping date, or train station location. The EDI data loops can be assigned primary key columns to build relationships between outer levels and inner levels. An end user can also select data from tables between different transaction sets tables. The final database DDL scripts can be referenced by a request.

6.8 Database Management Tools

The SQL Server Enterprise Manager is a graphical interface tool as shown in Figure 12 to administer the server and the database of JointData (by this project). We use the Enterprise Manager to:

- Configure SQL Server options,
- Register databases,
- Create/edit/view database schema and data,
- Replicate stand-by databases,
- Perform maintenance and backups, and
- Monitor performance.

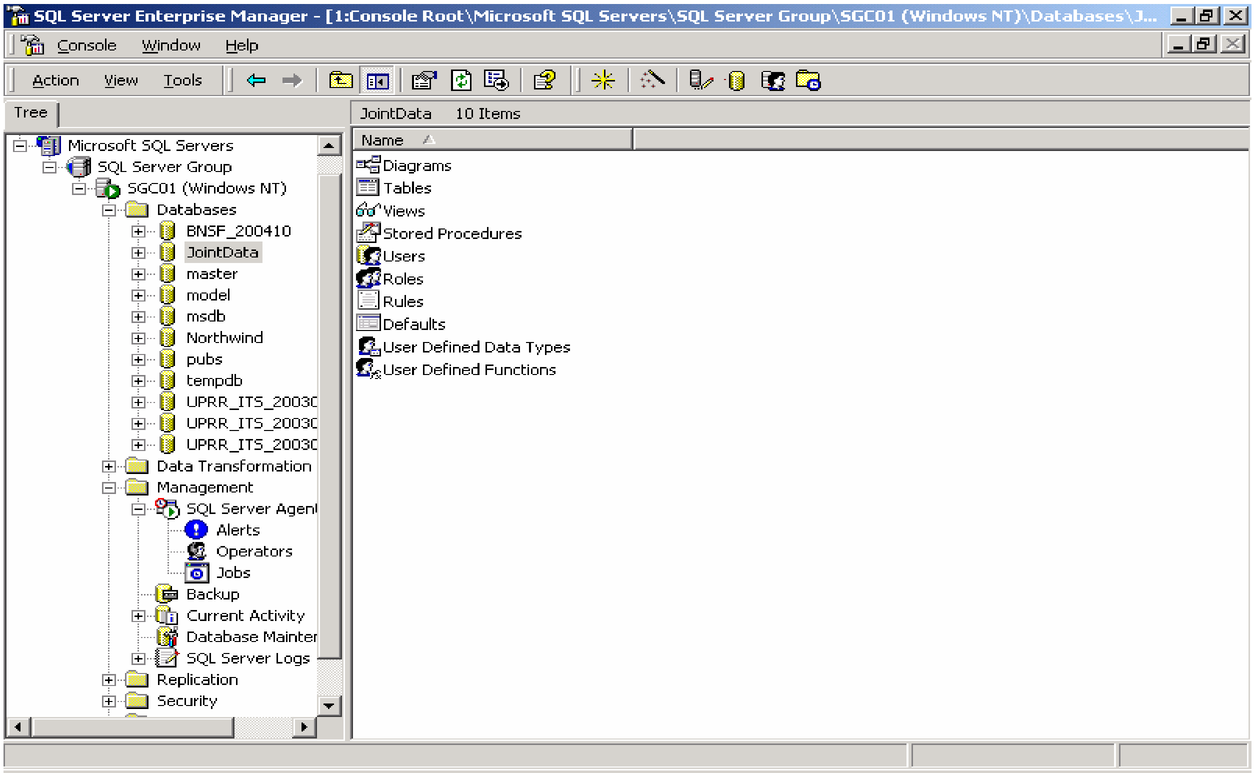


Figure 12. SQL Server Enterprise Manager

In the area of security, a backup and disaster recovery plan is created to provide 100% data restore protection. This backup plan is shown in Table 12:

Table 12. Backup Plan

Types	Cycle	Path and BAK File Names	Generations
Full Backup	1 am, Sunday	E:\Backup_Full\JointData_F_20070301013022.BAK	12
Diffeneral Backup	10 pm, daily	E:\Backup_Diff\JointData_D_20070301101512.BAK	14
Transaction Log Backup	Every 1 hour, daily	F:\Backup_Log\JointData_T_20070301080015.BAK	24

Another disaster recovery plan is created and must be able to restore all damaged transactions all the way to the crash time. As described in Table 13, if the crash time is at 10:30 am, Wednesday, the SQL Server Restore tool will be used. Three types of files must be set into the Restore Plan: (1) Sunday full backup file, (2) Tuesday night differential backup file, and (3) Wednesday 10 am transaction log backup file or 10:30 am transaction log file. Therefore, the Restore Plan tool will be able to restore all lost data to the crash time of 10:30 am Wednesday using timestamp and checkpoints.

Table 13. Disaster Recovery Plan

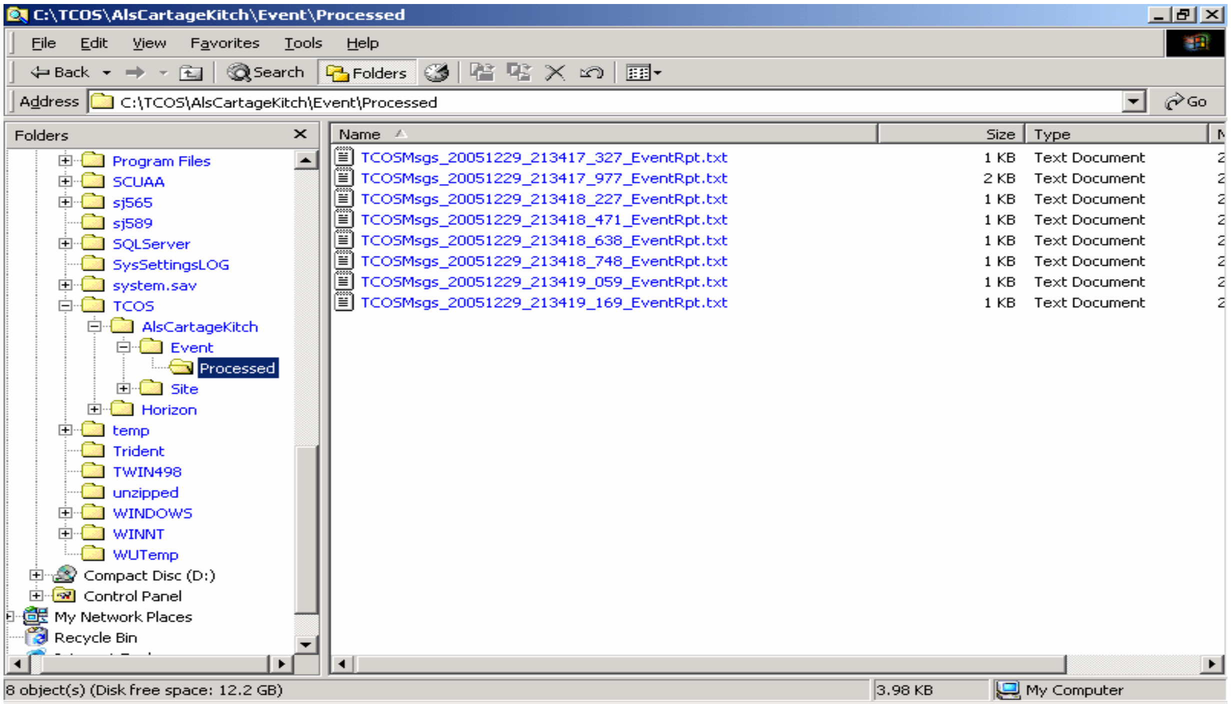
Crash Time	Restore Types	Time Generated	Path and BAK File Names
Database crashes at 10:30 am, Wednesday	Full Backup	1 am, Sunday	E:\Backup_Full\JointData_F_20070303013022.BAK
	Diffeneral Backup	10 pm, Tuesday	E:\Backup_Diff\JointData_D_20070306101512.BAK
	Transaction Log Backup	10 am, Wednesday	F:\Backup_Log\JointData_T_20070301080015.BAK
	Transaction Log	10:30 am, Wednesday	C:\SQLServer\Data\JointData_log.LDF

6.9 Extraction, Transformation, Loading Programs (OO Programming)

The Extraction, Transformation, and Loading (ETL) procedures are the critical programs (stored procedures, triggers, and C# DLL) we wrote to process collected RFID, CLM, AEI, and EDI data and load them into the database of JointData. The process has been automated from the remote scanner site locations to the JDDSP database without human being interruption as follows:

1. At remote scanner site, a record is logged into a file whenever a container passes through a designated lane. Every 15 minutes, a Windows scheduled job will PUSH this file to the datacenter via an internet or TELNET transfer protocol.

2. At the datacenter server, a path and folder will be assigned to each site to receive all files from all remote sites. For instance, the scanned files from TCOS's Site 'AlsCartageKtch' will be send to the C:/TCOS/AlsCartageKtch/Event of the Database Server (see Figure 13).
3. A SQL scheduled job will be kicked off every one minute to run the Auto Data Load program which will visit each folder specified from Step 2, read one file at a time until all files are completed, extract records and transform values, assign primary keys and foreign keys, and load them into a user-friendly relational database. The Figure 14 displays an example of this database table, TCOS_EventRpt_OSC3, which had been loaded with TCOS records.

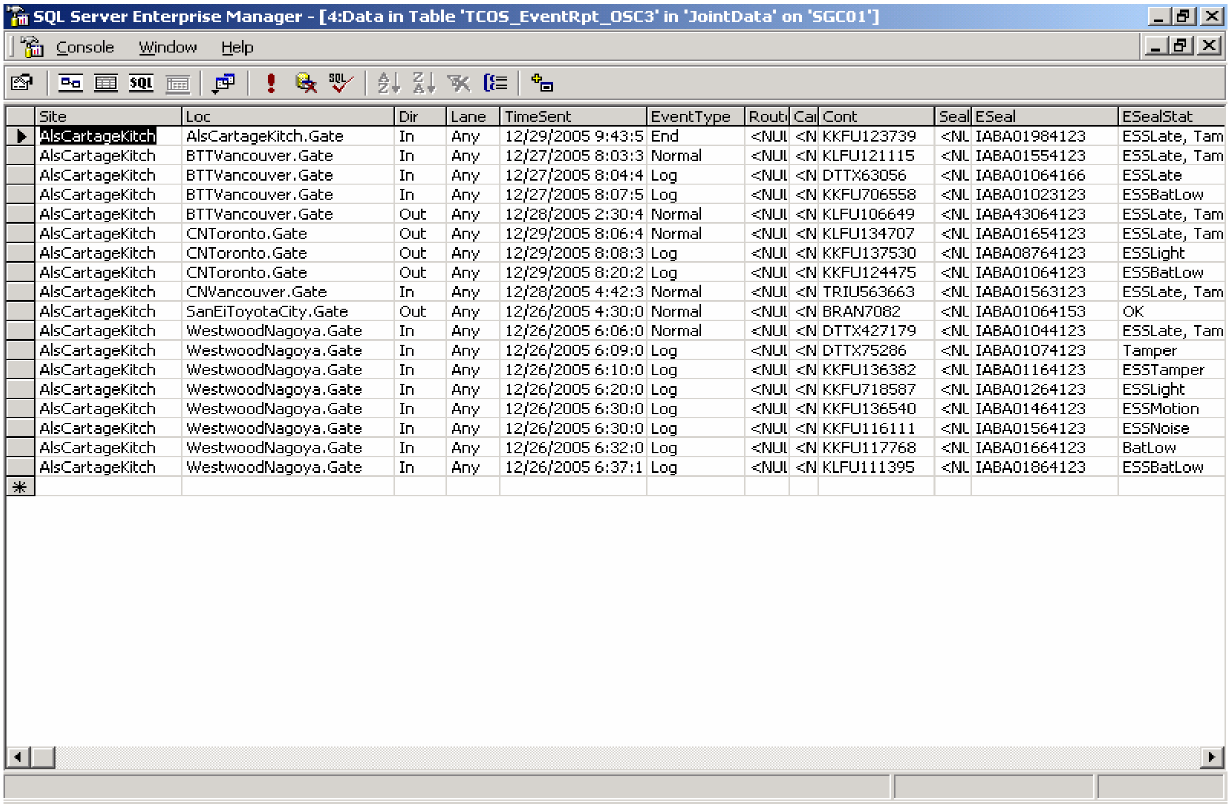


Address C:\TCOS\AlsCartageKitch\Event\Processed

Name	Size	Type
TCOSMsgs_20051229_213417_327_EventRpt.txt	1 KB	Text Document
TCOSMsgs_20051229_213417_977_EventRpt.txt	2 KB	Text Document
TCOSMsgs_20051229_213418_227_EventRpt.txt	1 KB	Text Document
TCOSMsgs_20051229_213418_471_EventRpt.txt	1 KB	Text Document
TCOSMsgs_20051229_213418_638_EventRpt.txt	1 KB	Text Document
TCOSMsgs_20051229_213418_748_EventRpt.txt	1 KB	Text Document
TCOSMsgs_20051229_213419_059_EventRpt.txt	1 KB	Text Document
TCOSMsgs_20051229_213419_169_EventRpt.txt	1 KB	Text Document

8 object(s) (Disk free space: 12.2 GB) 3.98 KB

Figure 13. Path and Folder to Receive TCOS's Site Files

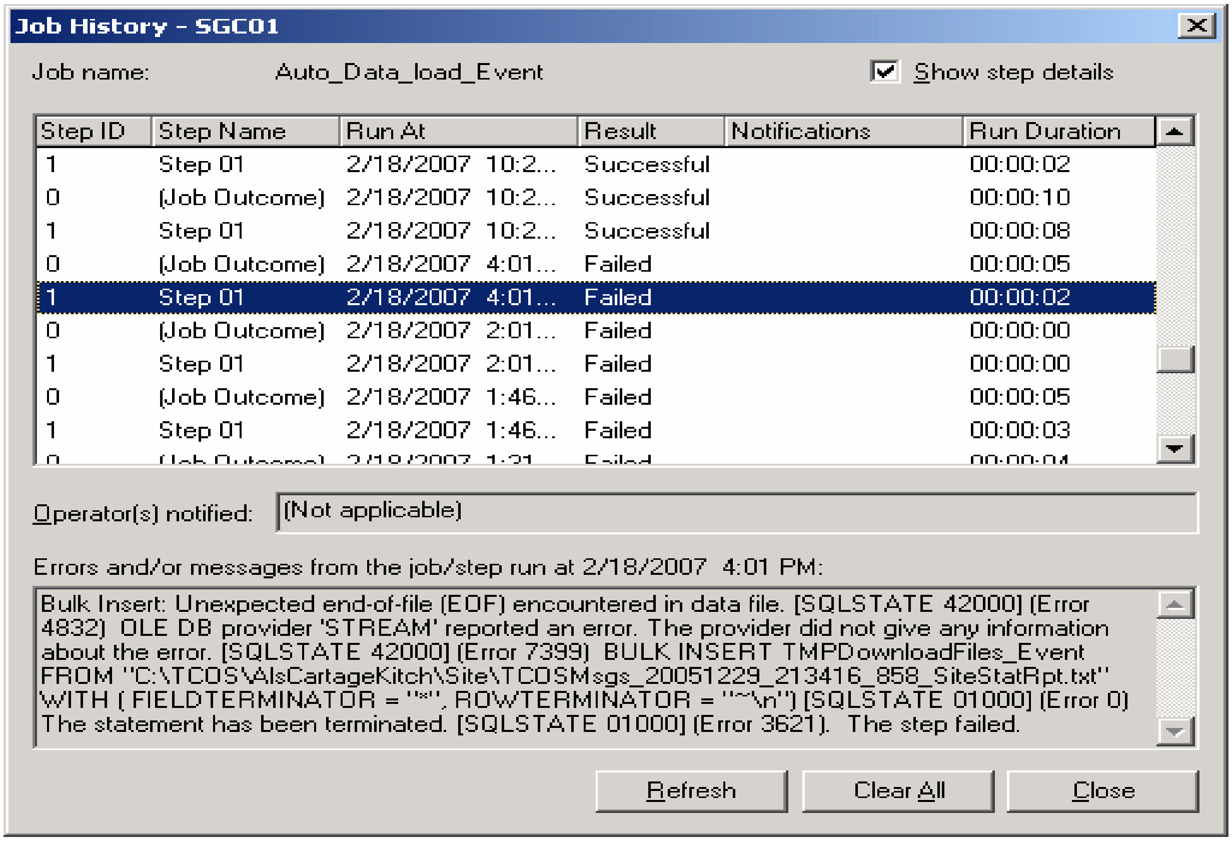

SQL Server Enterprise Manager - [4:Data in Table 'TCOS_EventRpt_OSC3' in 'JointData' on 'SGC01']
Console
Window
Help
SQL

Site	Loc	Dir	Lane	TimeSent	EventType	Routi	Cai	Cont	Seal	ESeal	ESealStat
AlsCartageKitch	AlsCartageKitch.Gate	In	Any	12/29/2005 9:43:5	End	<NUl	<N	KKFU123739	<NL	IABA01984123	ESSLate, Tam
AlsCartageKitch	BTTVancouver.Gate	In	Any	12/27/2005 8:03:3	Normal	<NUl	<N	KLFU121115	<NL	IABA01554123	ESSLate, Tam
AlsCartageKitch	BTTVancouver.Gate	In	Any	12/27/2005 8:04:4	Log	<NUl	<N	DTTX63056	<NL	IABA01064166	ESSLate
AlsCartageKitch	BTTVancouver.Gate	In	Any	12/27/2005 8:07:5	Log	<NUl	<N	KKFU706558	<NL	IABA01023123	ESSBatLow
AlsCartageKitch	BTTVancouver.Gate	Out	Any	12/28/2005 2:30:4	Normal	<NUl	<N	KLFU106649	<NL	IABA43064123	ESSLate, Tam
AlsCartageKitch	CNToronto.Gate	Out	Any	12/29/2005 8:06:4	Normal	<NUl	<N	KLFU134707	<NL	IABA01654123	ESSLate, Tam
AlsCartageKitch	CNToronto.Gate	Out	Any	12/29/2005 8:08:3	Log	<NUl	<N	KKFU137530	<NL	IABA08764123	ESSLight
AlsCartageKitch	CNToronto.Gate	Out	Any	12/29/2005 8:20:2	Log	<NUl	<N	KKFU124475	<NL	IABA01064123	ESSBatLow
AlsCartageKitch	CNVancouver.Gate	In	Any	12/28/2005 4:42:3	Normal	<NUl	<N	TRIU563663	<NL	IABA01563123	ESSLate, Tam
AlsCartageKitch	SanEiToyotaCity.Gate	Out	Any	12/26/2005 4:30:0	Normal	<NUl	<N	BRAN7082	<NL	IABA01064153	OK
AlsCartageKitch	WestwoodNagoya.Gate	In	Any	12/26/2005 6:06:0	Normal	<NUl	<N	DTTX427179	<NL	IABA01044123	ESSLate, Tam
AlsCartageKitch	WestwoodNagoya.Gate	In	Any	12/26/2005 6:09:0	Log	<NUl	<N	DTTX75286	<NL	IABA01074123	Tamper
AlsCartageKitch	WestwoodNagoya.Gate	In	Any	12/26/2005 6:10:0	Log	<NUl	<N	KKFU136382	<NL	IABA01164123	ESSTamper
AlsCartageKitch	WestwoodNagoya.Gate	In	Any	12/26/2005 6:20:0	Log	<NUl	<N	KKFU718587	<NL	IABA01264123	ESSLight
AlsCartageKitch	WestwoodNagoya.Gate	In	Any	12/26/2005 6:30:0	Log	<NUl	<N	KKFU136540	<NL	IABA01464123	ESSMotion
AlsCartageKitch	WestwoodNagoya.Gate	In	Any	12/26/2005 6:30:0	Log	<NUl	<N	KKFU116111	<NL	IABA01564123	ESSNoise
AlsCartageKitch	WestwoodNagoya.Gate	In	Any	12/26/2005 6:32:0	Log	<NUl	<N	KKFU117768	<NL	IABA01664123	BatLow
AlsCartageKitch	WestwoodNagoya.Gate	In	Any	12/26/2005 6:37:1	Log	<NUl	<N	KLFU111395	<NL	IABA01864123	ESSBatLow

Figure 14. TCOS Database Records

6.10 Database Applications and Maintenance

The Enterprise Manager also provides a Job History Screen to allow the Database Administrator to view the ETL status as Figure 15. If a source file is successfully Extracted, Transformed, and Loaded into the JointData database, the result will show a ‘Successful’ status and the file will be moved from C:/TCOS/AlsCartageKtch/Event to C:/TCOS/AlsCartageKtch/Event/Processed. If otherwise, the result will show a ‘Failed’ status and this file will stay in the same place C:/TCOS/AlsCartageKtch/Event. The DBA can view the error message in the bottom of the Job History screen and make required record changes in C:/TCOS/AlsCartageKtch/Event. So this file will be reprocessed and get moved to C:/TCOS/AlsCartageKtch/Event/Processed.



Job name: Auto_Data_load_Event

Step ID	Step Name	Run At	Result	Notifications	Run Duration
1	Step 01	2/18/2007 10:2...	Successful		00:00:02
0	(Job Outcome)	2/18/2007 10:2...	Successful		00:00:10
1	Step 01	2/18/2007 10:2...	Successful		00:00:08
0	(Job Outcome)	2/18/2007 4:01...	Failed		00:00:05
1	Step 01	2/18/2007 4:01...	Failed		00:00:02
0	(Job Outcome)	2/18/2007 2:01...	Failed		00:00:00
1	Step 01	2/18/2007 2:01...	Failed		00:00:00
0	(Job Outcome)	2/18/2007 1:46...	Failed		00:00:05
1	Step 01	2/18/2007 1:46...	Failed		00:00:03
0	(Job Outcome)	2/18/2007 1:31...	Failed		00:00:04

Operator(s) notified: (Not applicable)

Errors and/or messages from the job/step run at 2/18/2007 4:01 PM:

```
Bulk Insert: Unexpected end-of-file (EOF) encountered in data file. [SQLSTATE 42000] (Error
4832)  OLE DB provider 'STREAM' reported an error. The provider did not give any information
about the error. [SQLSTATE 42000] (Error 7399)  BULK INSERT TMPDownloadFiles_Event
FROM "C:\TCOS\AlsCartageKitch\Site\TCOSMsgs_20051229_213416_858_SiteStatRpt.txt"
WITH ( FIELDTERMINATOR = "*", ROWTERMINATOR = "~\n") [SQLSTATE 01000] (Error 0)
The statement has been terminated. [SQLSTATE 01000] (Error 3621).  The step failed.
```

Figure 15.   Job History Screen

7.0 METADATA

The creation tasks of the Metadata engine and repository database are based on the Object-Oriented UML models and database design of the preceding two sections.

7.1 Metadata are Abstract

The simple act of describing real-world phenomena generates abstract information that qualifies as Metadata. In data design, real-world phenomena are also described in abstract terms. For example, people are designated as employee (Metadata). In software design, the database structures that store data can be abstracted into Metadata classification schemes that make sense to developers and designers. In the OO development, a table (Metadata) is derived from an object, which, in turn, an object (Metadata) can be derived from a class.

There are multiple levels of abstraction in Metadata. You can describe a data instance, then describe that description, and continue to describe subsequent descriptions until you reach some practical limit. We have described class, object, instance, table, column, index, primary key, foreign key, and five constraints in the preceding sections, we will continue to describe the next level of abstraction as another Metadata for this central tracking database.

7.2 Definitions and Background

The Metadata are data used to describe other data. Metadata describe the structure and meaning of data, as well as the structure and meaning of applications and processes. It is important to remember that Metadata are abstract, have a context, and can be used for multiple purposes in the further development of systems.

Within the DOD, several types of Metadata can be found including:

1. Descriptive Metadata (table names, object names, etc.),
2. Structural Metadata (primary keys, foreign keys, Indexes, etc.),
3. Semantic Metadata (program documents, XML documents, etc.), and
4. Usage Metadata (application system documents, packages, etc.).

This section will generate all four Metadata. As well, we will rely on SQL Server Metadata Services in order to standardize retrieval of information from the JointData database created from the preceding sections.

7.3 Metadata Have Multiple Purposes

All project members can work with Metadata information just as one would with any kind of data design elements. Expressing standard design information as Metadata opens up new possibilities for reuse, sharing, and multiple tool support. For example, defining data objects as Metadata enables project members to see how they are constructed and versioned. Versioning support provides a way to view and retrieve any historical version of a particular DTS package or table layout definitions. When we develop codes based on Metadata, we can define a structure once and then reuse it to create multiple instances that can be versioned for specific tools and applications.

In summary, the major purposes of Metadata in the Integrated Tracking System are to:

1. Categorize the items of JDDSP database objects into groups.
2. Enable project members and stakeholders to retrieve data based on categories and versions.
3. Serves as a clearinghouse for official standards and documents across SM 21 projects.
4. Identify highly rated data for container search.
5. Enable the JDDSP database to interact with other systems such as simulations, wireless devices, marine terminals, and supply chains.
6. Promote data visibility and reusability for the future SM 21 projects.

7.4 Designing Metadata Using SQL Server Meta Data Services

The Meta Data Services of SQL Server Enterprise Manager will be used to design Metadata. The major Metadata architecture consists of (1) Repository Engine, (2) Metadata repository database, (3) Information Model, and (4) Exporting Metadata to XML for browse. All four components will be described as follows.

7.4.1 Repository Engine

The repository engine is a service in the SQL Server Enterprise Manager (called Meta Data Services) that provides basic functions for storing and retrieving objects and maintaining the relationships among them. The engine performs these functions within the context of an Information Model (see Section 7.4.3). The engine processes user-defined model information to determine how to store and support objects, relationships, and actions. When you click the Meta Data Services, the repository engine will manipulate instances and elements of Information Models, the engine does so only to the extent that the model structure allows. Figure 16 shows the Meta Data Services (in the middle caption of the figure) in the SQL Server Enterprise Manager.

7.4.2 Repository Database

SQL Server Meta Data Services is a software component providing not only a repository engine but also an object-oriented repository database that stores and manages metadata for SQL Server and its components. The repository database is a storage room to store all details of Metadata. Again Figure 16 shows the repository database with the caption – dbo on the left-hand side.

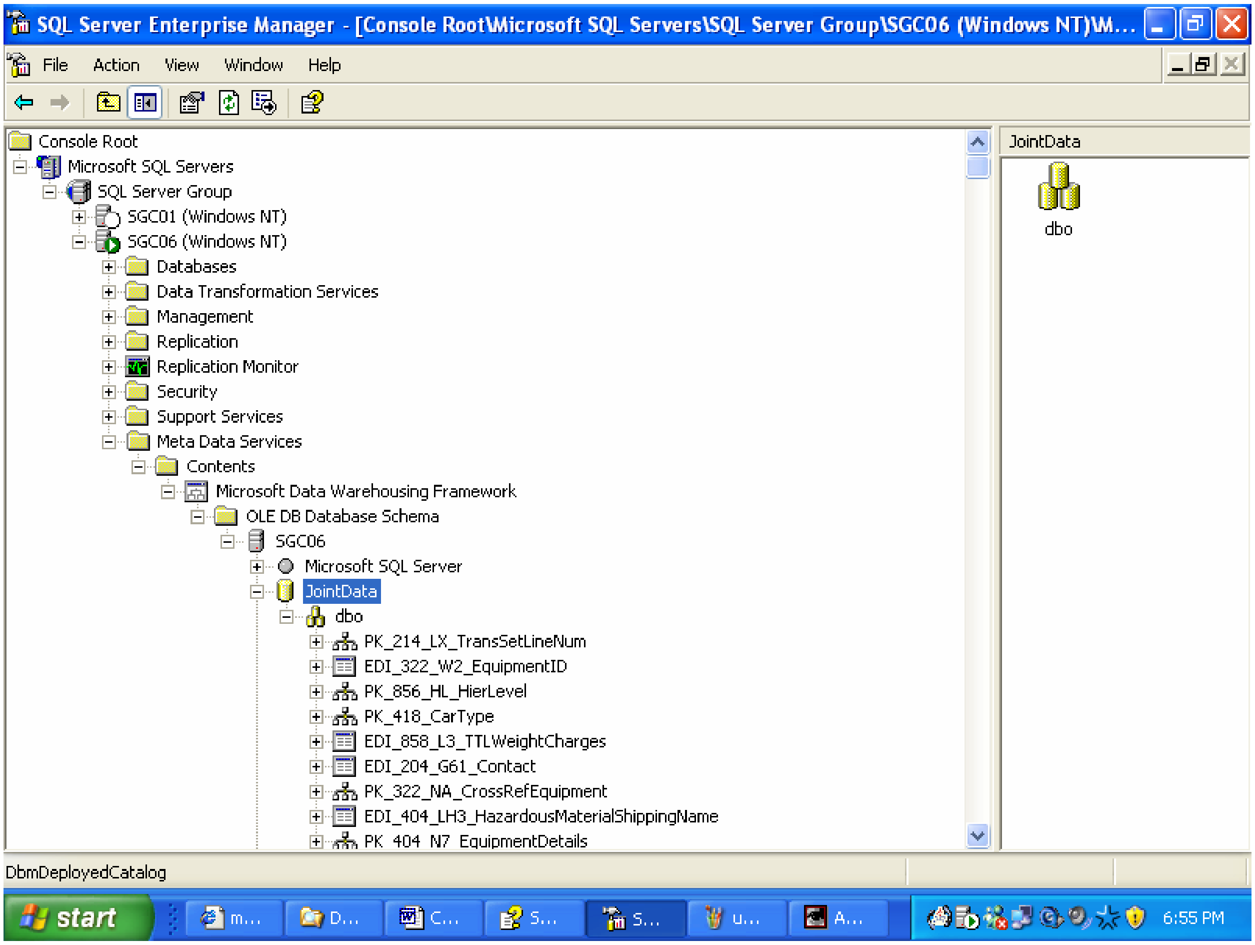


Figure 16. Meta Data Services – Repository Engine

7.4.3 Information Model of UML

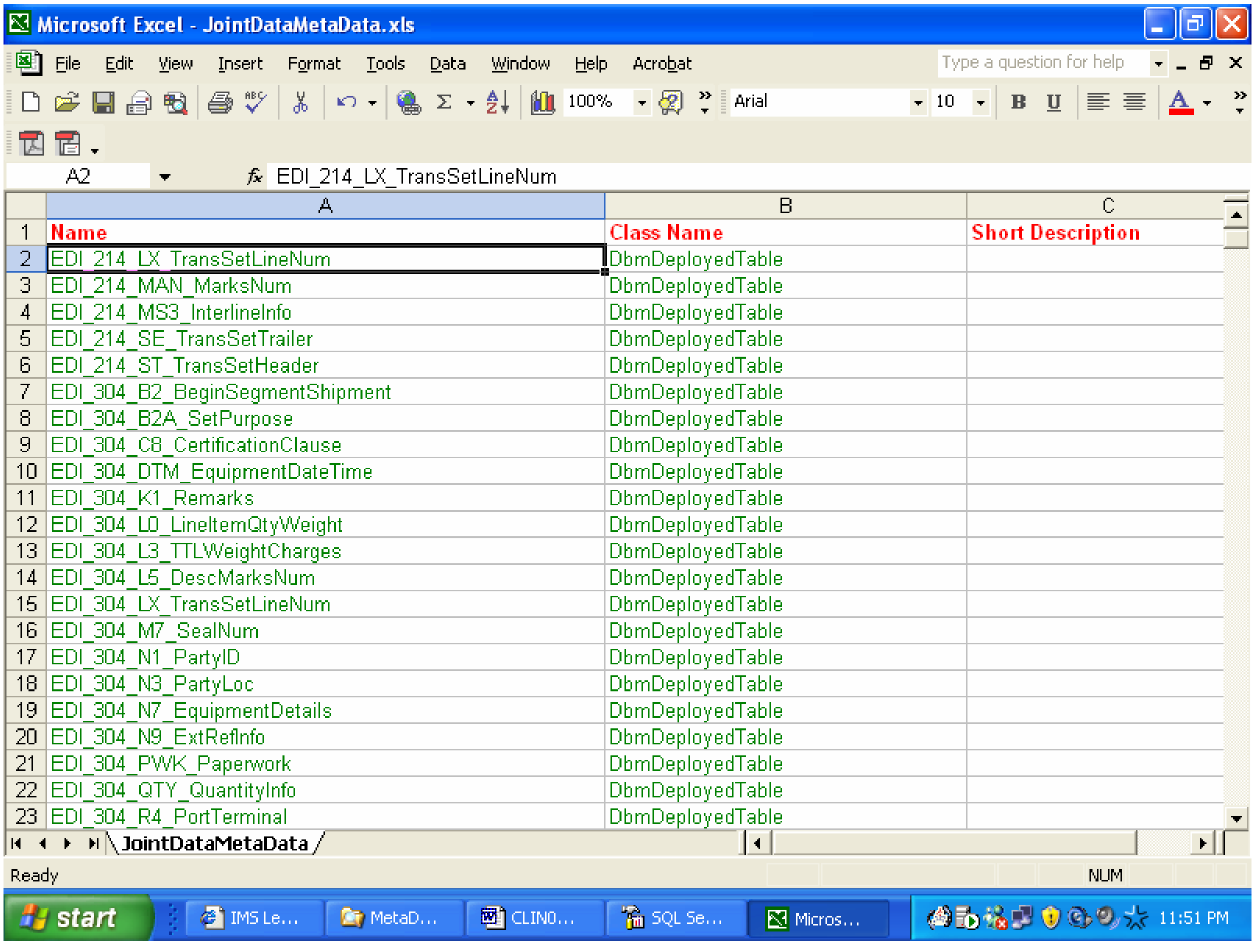
Deploying Microsoft SQL Server Meta Data Services technology begins with this Information Model. An Information Model defines Metadata types that are stored in a repository database and used by tools and applications. In Section 5, we developed four data models and they were used to create the Information Model of JointData database in Section 6. Meta Data Services is intended to be used in this section with information models that provide type information about the Metadata.

7.4.4 Metadata

We ran Meta Data Services to import metadata from the JointData database. This process took a few minutes to generate all descriptions (Metadata) from the JointData objects. Figure 16 gives a listing of Metadata in the lowest level starting with PK_214_LX_TransSetLineNum (a primary key for the 214_LX_TransSetLineNum table). If we continue to expand the next lower level, all underneath columns and data types will be displayed. These are all part of the Metadata.

7.4.5 Exporting Metadata

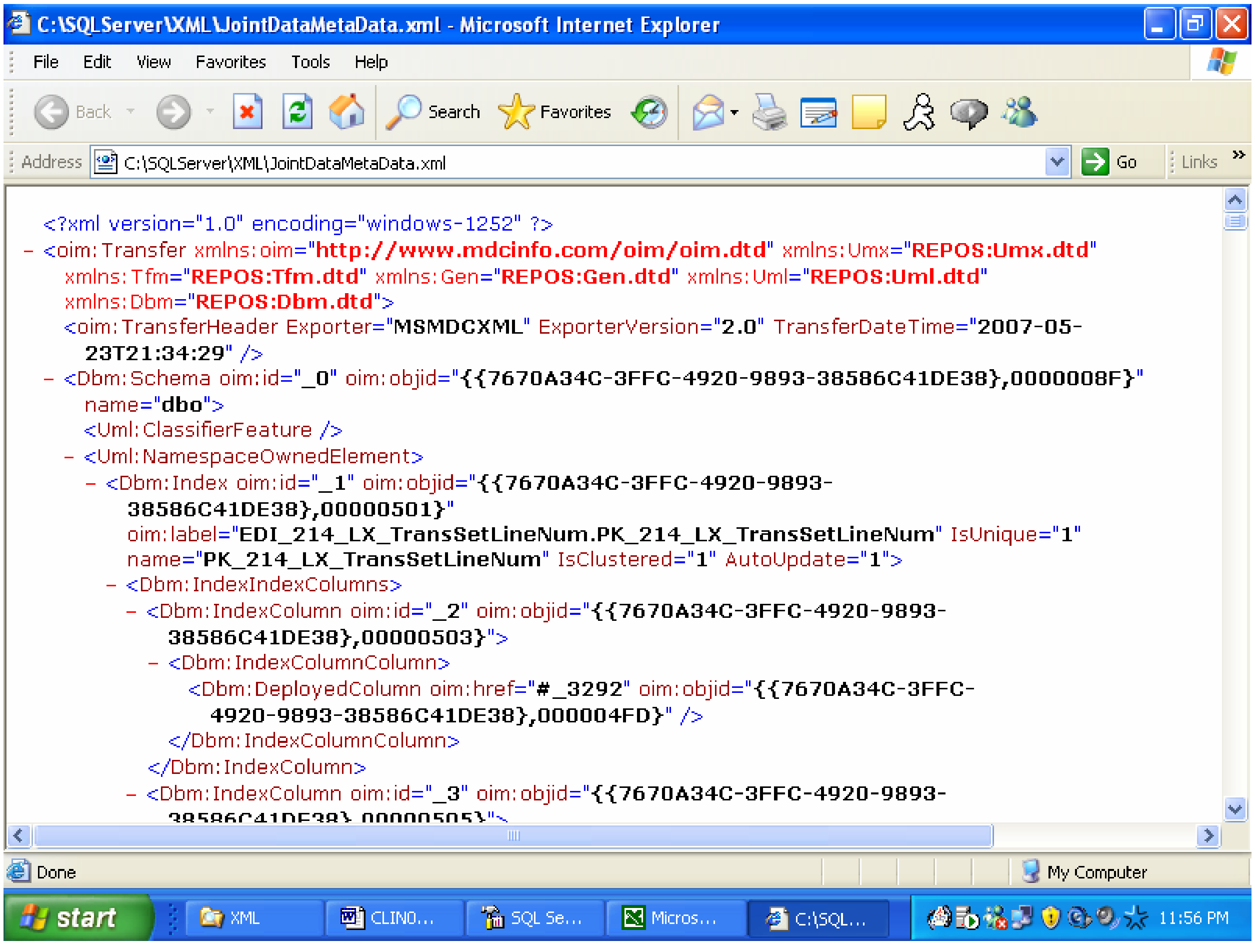
Since the Metadata listing is hard to read on the Meta Data Services, an easy way is to export the Metadata to Excel or XML, then we can sort out the Metadata to search descriptions and class names. Figure 17 displays an Excel export listing after sorting and Figure 18 displays an XML export listing. The very first entry is PK_214_LX_TransSetLineNum (which is a primary key for the 214_LX_TransSetLineNum table) in both export listings.



A2 EDI_214_LX_TransSetLineNum

	A	B	C
1	Name	Class Name	Short Description
2	EDI_214_LX_TransSetLineNum	DbmDeployedTable	
3	EDI_214_MAN_MarksNum	DbmDeployedTable	
4	EDI_214_MS3_InterlineInfo	DbmDeployedTable	
5	EDI_214_SE_TransSetTrailer	DbmDeployedTable	
6	EDI_214_ST_TransSetHeader	DbmDeployedTable	
7	EDI_304_B2_BeginSegmentShipment	DbmDeployedTable	
8	EDI_304_B2A_SetPurpose	DbmDeployedTable	
9	EDI_304_C8_CertificationClause	DbmDeployedTable	
10	EDI_304_DTM_EquipmentDateTime	DbmDeployedTable	
11	EDI_304_K1_Remarks	DbmDeployedTable	
12	EDI_304_L0_LineItemQtyWeight	DbmDeployedTable	
13	EDI_304_L3_TTLWeightCharges	DbmDeployedTable	
14	EDI_304_L5_DescMarksNum	DbmDeployedTable	
15	EDI_304_LX_TransSetLineNum	DbmDeployedTable	
16	EDI_304_M7_SealNum	DbmDeployedTable	
17	EDI_304_N1_PartyID	DbmDeployedTable	
18	EDI_304_N3_PartyLoc	DbmDeployedTable	
19	EDI_304_N7_EquipmentDetails	DbmDeployedTable	
20	EDI_304_N9_ExtRefInfo	DbmDeployedTable	
21	EDI_304_PWK_Paperwork	DbmDeployedTable	
22	EDI_304_QTY_QuantityInfo	DbmDeployedTable	
23	EDI_304_R4_PortTerminal	DbmDeployedTable	

Figure 17. Metadata Exports – Excel Listing


C:\SQLServer\XML\JointDataMetaData.xml - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address C:\SQLServer\XML\JointDataMetaData.xml
Go
Links
Done
My Computer
start
XML
CLINO...
SQL Se...
Micros...
C:\SQL...
11:56 PM

```
  <?xml version="1.0" encoding="windows-1252" ?>
- <oim:Transfer xmlns:oim="http://www.mdcinfo.com/oim/oim.dtd" xmlns:Umx="REPOS:Umx.dtd"
    xmlns:Tfm="REPOS:Tfm.dtd" xmlns:Gen="REPOS:Gen.dtd" xmlns:Uml="REPOS:Uml.dtd"
    xmlns:Dbm="REPOS:Dbm.dtd">
    <oim:TransferHeader Exporter="MSMDCXML" ExporterVersion="2.0" TransferDateTime="2007-05-
      23T21:34:29" />
  - <Dbm:Schema oim:id="_0" oim:objid="{{7670A34C-3FFC-4920-9893-38586C41DE38},0000008F}"
      name="dbo">
      <Uml:ClassifierFeature />
    - <Uml:NamespaceOwnedElement>
      - <Dbm:Index oim:id="_1" oim:objid="{{7670A34C-3FFC-4920-9893-
          38586C41DE38},00000501}"
          oim:label="EDI_214_LX_TransSetLineNum.PK_214_LX_TransSetLineNum" IsUnique="1"
          name="PK_214_LX_TransSetLineNum" IsClustered="1" AutoUpdate="1">
        - <Dbm:IndexIndexColumns>
          - <Dbm:IndexColumn oim:id="_2" oim:objid="{{7670A34C-3FFC-4920-9893-
              38586C41DE38},00000503}">
            - <Dbm:IndexColumnColumn>
                <Dbm:DeployedColumn oim:href="#_3292" oim:objid="{{7670A34C-3FFC-
                  4920-9893-38586C41DE38},000004FD}" />
              </Dbm:IndexColumnColumn>
            </Dbm:IndexColumn>
          - <Dbm:IndexColumn oim:id="_3" oim:objid="{{7670A34C-3FFC-4920-9893-
              38586C41DE38},00000505}">
```

Figure 18. Metadata Exports – XML Listing

8.0 WEB AND XML DEMONSTRATIONS

8.1 Background and Goals

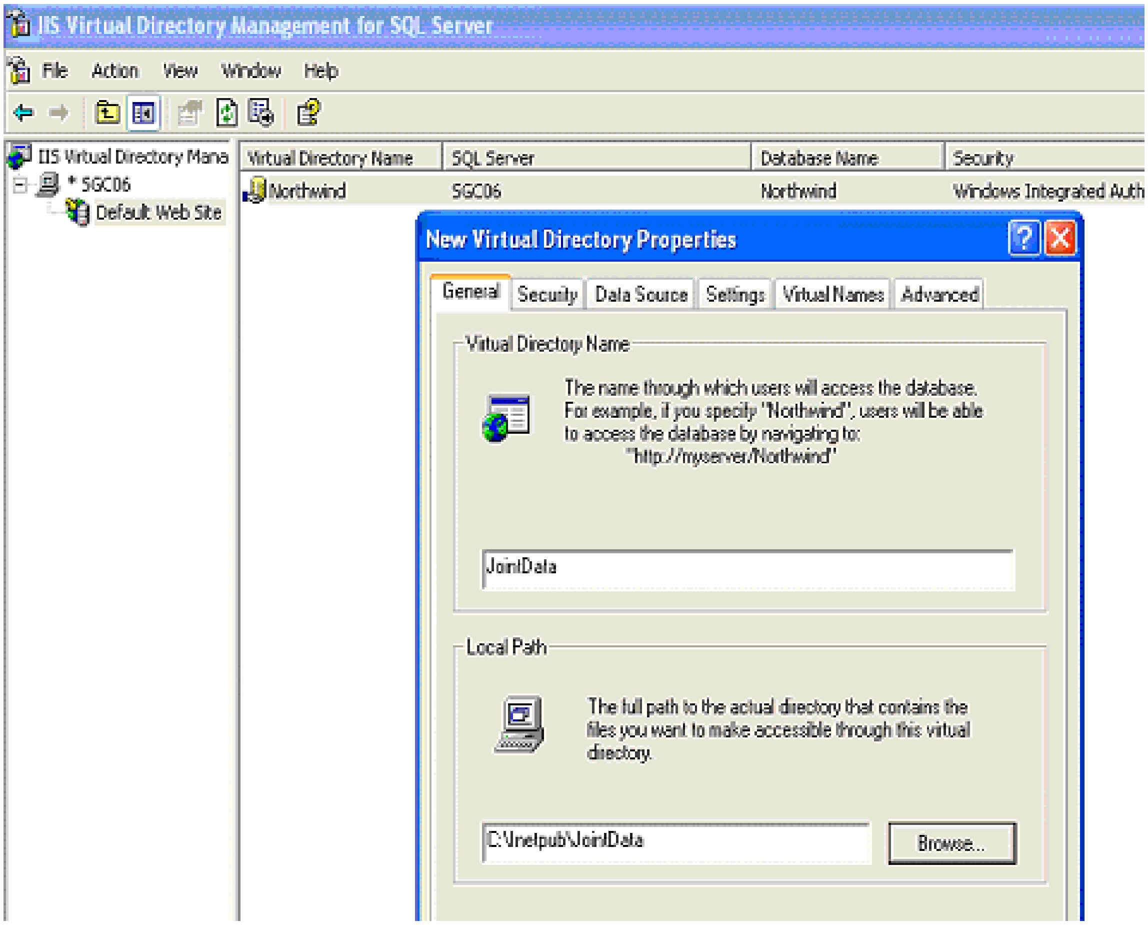
After a user-friendly relational database is created, we need a front-end screen for an end-user or a stakeholder to read, write, update, and delete the data and reports. An XML Web system is extremely appropriate in this application in terms of its remote accessibility and affordability.

The purpose for applying an XML Web application on the relational database system is:

- To allow a stakeholder to use a remote web page to easily interoperate with a database and make decisions on a near real-time basis.

8.2 Configuration of Internet Information Services

An XML Web application requires writing client-side scripts and server-side scripts. The client-side scripts can be executed using the Internet Explorer from a Personal Computer. The server-side scripts must be executed on a server. So we need an Internet Information Server (IIS) service implemented in a PC workstation. The IIS is a service to be installed in a Personal Computer Windows. This IIS will be functioning as a group of Internet servers (including a Web or Hypertext Transfer Protocol server and a File Transfer Protocol server) with similar capabilities to Windows Server operating systems. Therefore, the IIS allows us to create and test the server-side scripts of an XML Web application in a Personal Computer without using a real Windows 2003 server. In the XML development, we must also use Windows provided folder structure which resides at C:\inetpub\JointData\ and C:\inetpub\wwwroot to conduct testing. After the XML Web application is completed, all programs will be moved to a real Windows web server to be published. The SQL Server provides a tool to configure an IIS Virtual Directory as shown in Figure 19.



Virtual Directory Name	SQL Server	Database Name	Security
Northwind	SGC06	Northwind	Windows Integrated Auth

The name through which users will access the database. For example, if you specify "Northwind", users will be able to access the database by navigating to: "http://myserver/Northwind"

JointData

The full path to the actual directory that contains the files you want to make accessible through this virtual directory.

C:\Inetpub\JointData

Figure 19. Configuration of IIS Using SM02 Server

Table 14 outlines the required procedures and respective specifications for the IIS configuration.

Table 14. Procedures and Specifications for IIS Configuration

Procedures	Specifications
General IIS Name	JointData
Local Path	C:\inetpub\JointData
Logon	sa/password
SQL Server/Database	SM02/JointData
Settings	Allow URL, Template, XPath, Post
Virtual Name	Template

8.3 Coding of XML, XSL, and Stored Procedures

The SQL Server database is an XML-enabled DBMS, which can read and write XML data, return data from databases in XML format, and read and update data stored in XML documents. The XML documents can be embedded in a Web Portal and SQL Server has a number of different ways to support XML.

Our XML logic starts with a request from a user. This request executes a stored procedure which will query the database using SQL. Then the database returns a number of records and generates a XML/Tag document listing to embed the records. This XML document needs an XSL style sheet to format and demonstrate the results on an XML web page. Figure 20 describes this logic. The examples of stored procedure, SQL query, XML/Tag codes, and XSL can also referenced by a request.

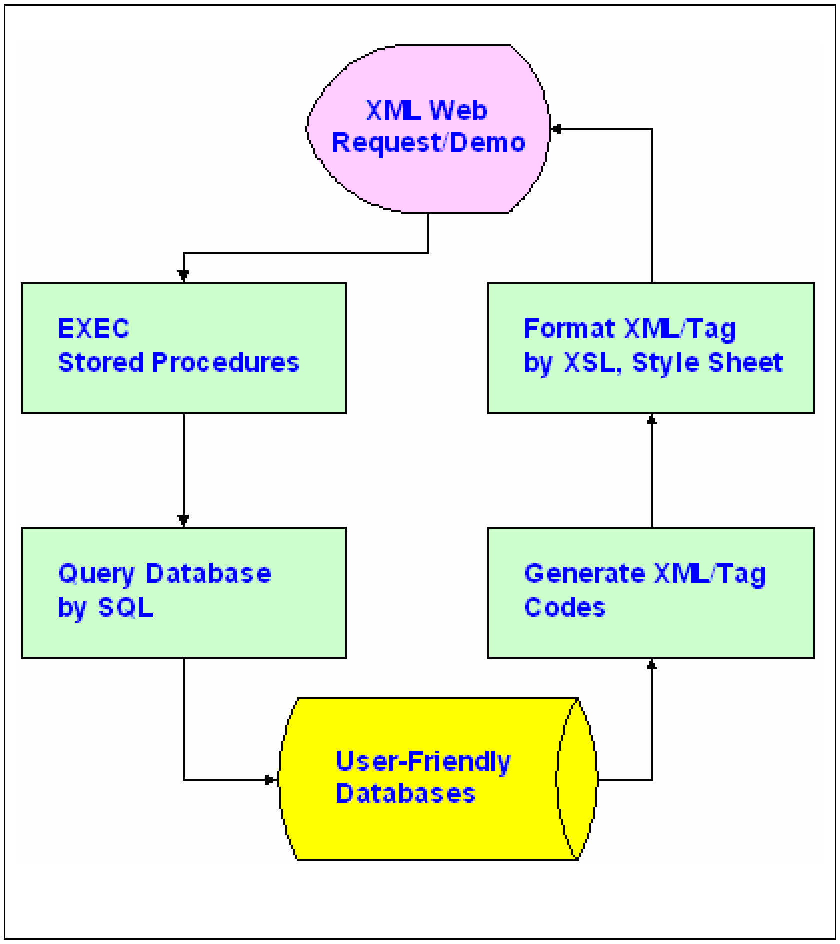
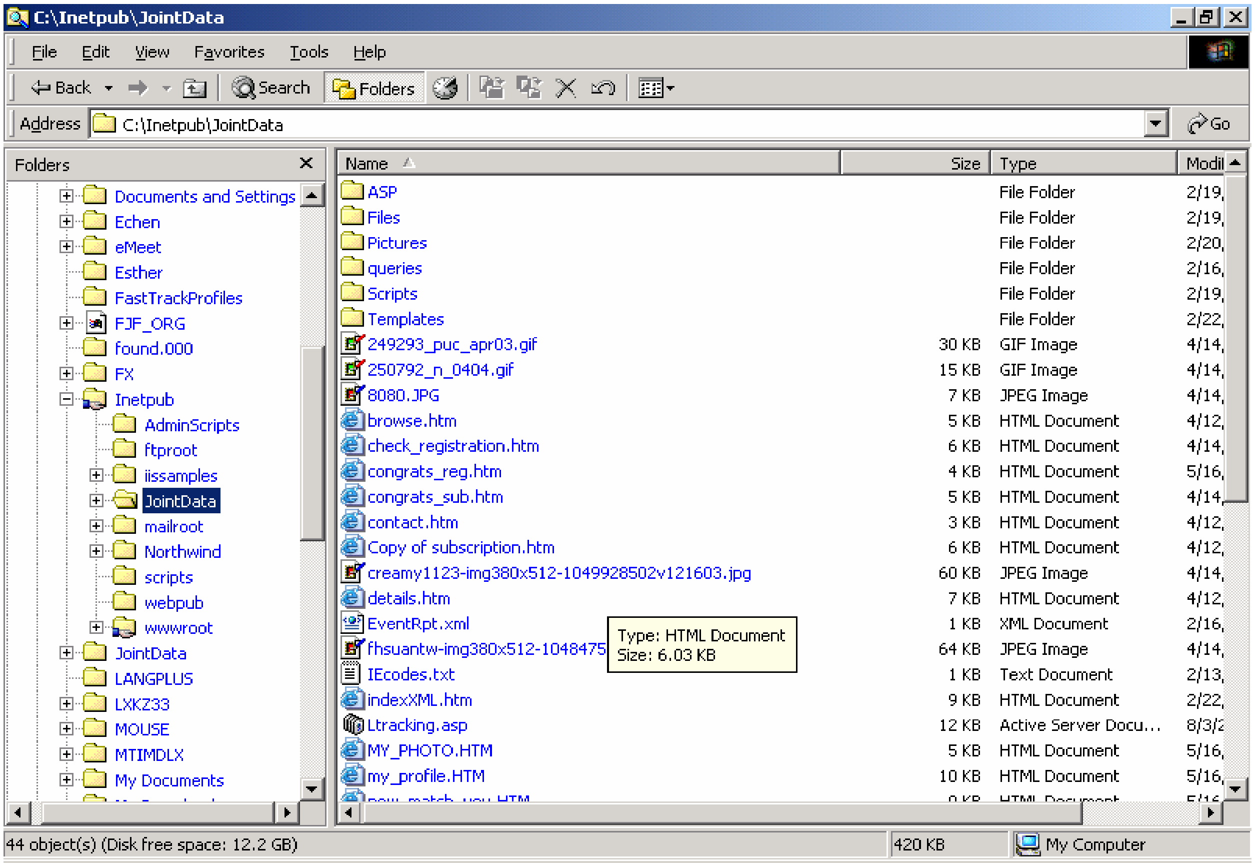


Figure 20. Logic of XML Request/Demo Procedure

8.4 Web Demonstrations

The XML Web application resides in one section of the SM 21 Web Portal in the SharePoint internet area. The XML scripts, database programs, and pictures are hosted in C:\Inetpub\JointDara\ in the SM02 server as shown in Figure 21.


C:\Inetpub\JointData
File Edit View Favorites Tools Help
Back Search Folders
Address C:\Inetpub\JointData
Go
Folders

Documents and Settings
Echen
eMeet
Esther
FastTrackProfiles
FJF_ORG
found.000
FX
Inetpub
AdminScripts
ftproot
iissamples
JointData
mailroot
Northwind
scripts
webpub
wwwroot
JointData
LANGPLUS
LXKZ33
MOUSE
MTIMDLX
My Documents

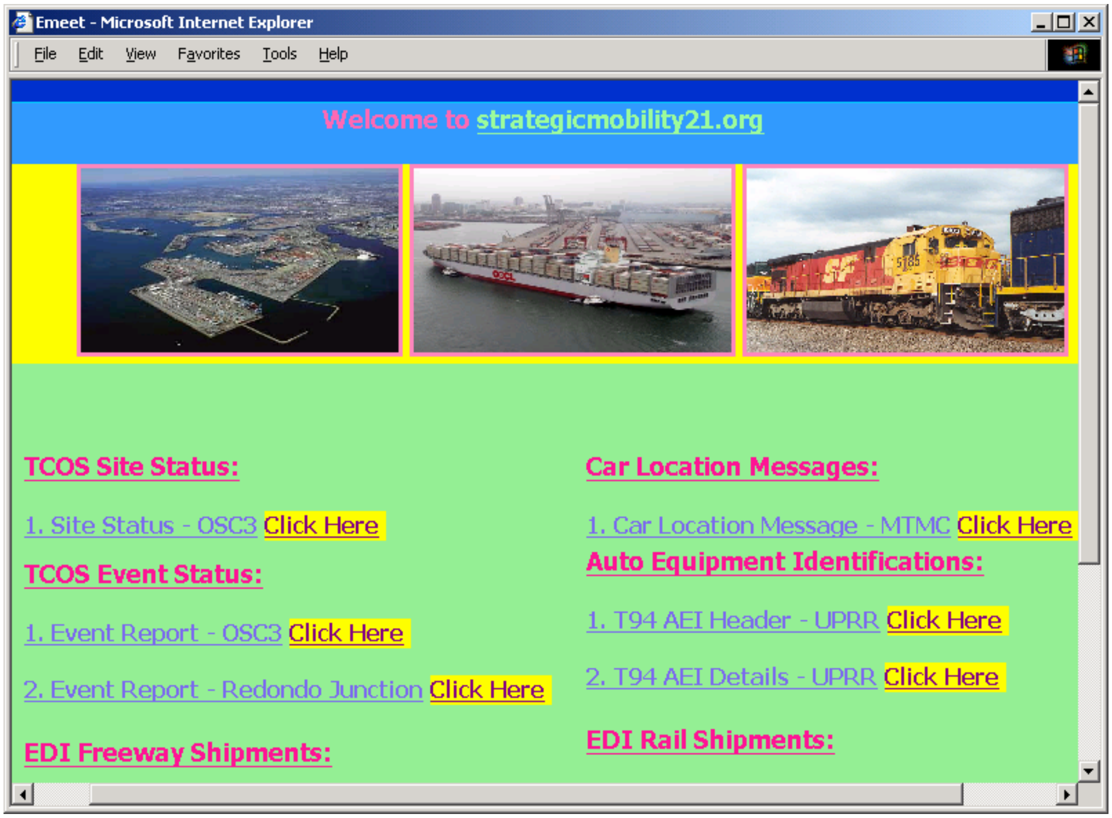
Name	Size	Type	Modi
ASP		File Folder	2/19,
Files		File Folder	2/19,
Pictures		File Folder	2/20,
queries		File Folder	2/16,
Scripts		File Folder	2/19,
Templates		File Folder	2/22,
249293_puc_apr03.gif	30 KB	GIF Image	4/14,
250792_n_0404.gif	15 KB	GIF Image	4/14,
8080.JPG	7 KB	JPEG Image	4/14,
browse.htm	5 KB	HTML Document	4/12,
check_registration.htm	6 KB	HTML Document	4/14,
congrats_reg.htm	4 KB	HTML Document	5/16,
congrats_sub.htm	5 KB	HTML Document	4/14,
contact.htm	3 KB	HTML Document	4/12,
Copy of subscription.htm	6 KB	HTML Document	4/12,
creamy1123-img380x512-1049928502v121603.jpg	60 KB	JPEG Image	4/14,
details.htm	7 KB	HTML Document	4/12,
EventRpt.xml	1 KB	XML Document	2/16,
fhsuantw-img380x512-1048475	64 KB	JPEG Image	4/14,
IEcodes.txt	1 KB	Text Document	2/13,
indexXML.htm	9 KB	HTML Document	2/22,
Ltracking.asp	12 KB	Active Server Docu...	8/3/2
MY_PHOTO.HTM	5 KB	HTML Document	5/16,
my_profile.HTM	10 KB	HTML Document	5/16,

Type: HTML Document
Size: 6.03 KB

44 object(s) (Disk free space: 12.2 GB) | 420 KB | My Computer

Figure 21. All XML Web Programs and Pictures in C:\Inetpub

Users can visit these XML web pages by: (1) Connecting to VPN (66.162.38.10) and (2) Entering http://SM02/Regional_Planning/Defaultaspx in the Address line of an Internet Explorer. Then the Home page of the SharePoint Web Portal will be displayed. The user can continue to click the Icon (JointData) under Site Hierarchy to hyperlink to the Main Menu of the JointData database as shown in Figure 22.



Welcome to strategicmobility21.org

TCOS Site Status:

1. Site Status - OSC3 Click Here

TCOS Event Status:

1. Event Report - OSC3 Click Here

2. Event Report - Redondo Junction Click Here

EDI Freeway Shipments:

Car Location Messages:

1. Car Location Message - MTMC Click Here

Auto Equipment Identifications:

1. T94 AEI Header - UPRR Click Here

2. T94 AEI Details - UPRR Click Here

EDI Rail Shipments:

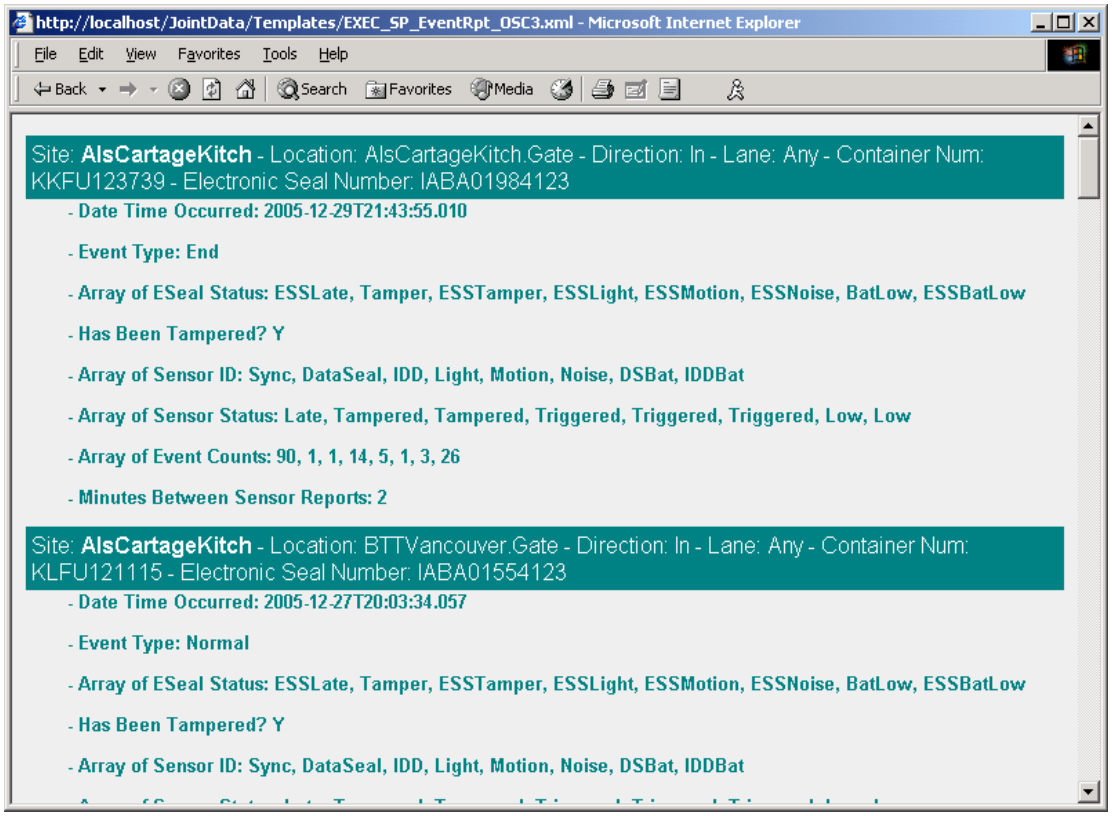
Figure 22. Main Menu of JointData Database

The Main Menu outlines a number of categories which allow a user to click and select a specific XML document listing. The categories collecting numerous example records from different systems are shown in Table 15.

After the user clicks and selects one of the categories, a quest will query the required records from the relational database, generate a XML document listing, match a style sheet for the display format, and then populate the results on a web page. Figure 23 shows an XML Document page on TCOS event status and a Rail Consist is shown in Figure 24.

Table 15. Categories, Systems, Record Formats of Main Menu

Categories	Systems	Record Format
Radio Frequency Identifications	Trade Corridor Operation System (Seattle, Tocoma, Alemada Corridor)	1. Sites 2. Events 3. Login
Car Location Message	Military Surface Deployment and Distribution Command	1. SDDC Heads 2. SDDC Details
Auto Equipment Identifications	T94 – Common Data Format	1. T94 Heads 2. T94 Details
Electronic Data Interchange	Ocean booking, vessel manifest, Bill of Lading, Terminal Operation Activity, Load Tender, Transportation Status, Rail Waybill, Rail Consist	1. EDI 204, 214 2. EDI 301, 304, 309, 322, 324 3. EDI 404, 417, 418 4. EDI 856, 858
Joint Data	Joint Data fro various databases	1. EDI and RFID 2. EDI and CLM 3. EDI and AEI


http://localhost/JointData/Templates/EXEC_SP_EventRpt_OSC3.xml - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Media

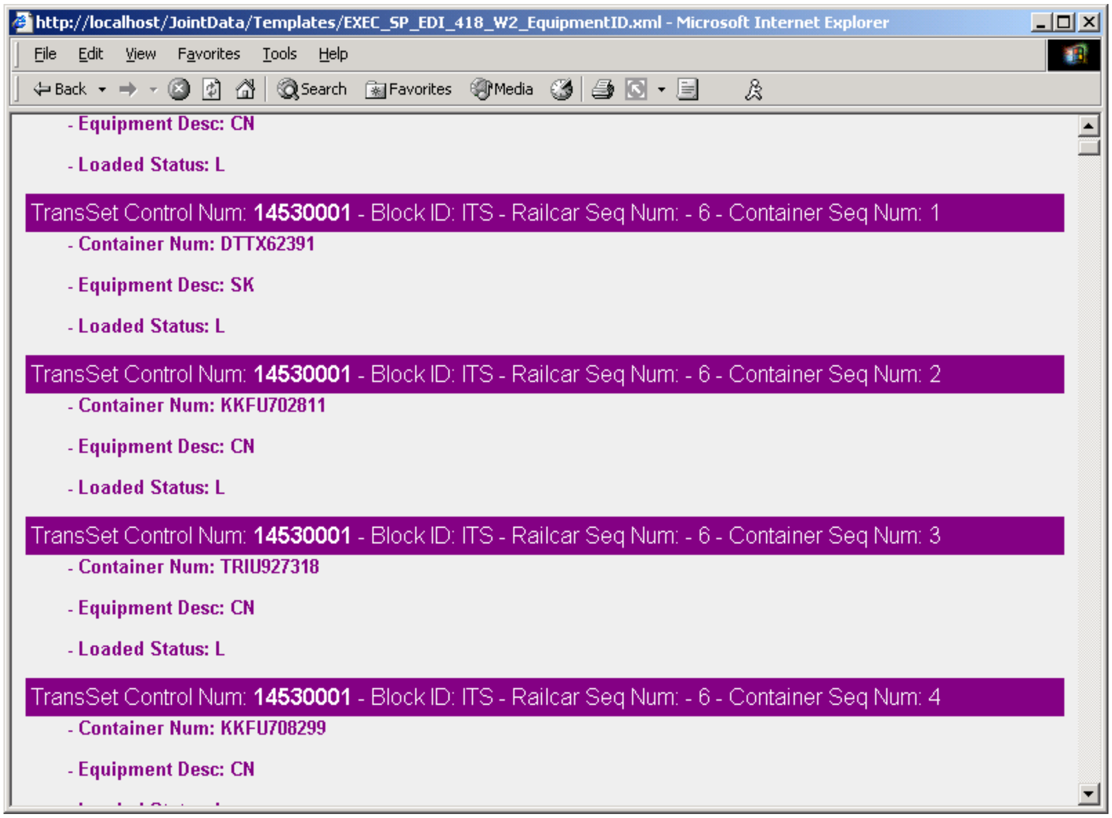
Site: **AlsCartageKitch** - Location: AlsCartageKitch.Gate - Direction: In - Lane: Any - Container Num: KKFU123739 - Electronic Seal Number: IABA01984123

- Date Time Occurred: 2005-12-29T21:43:55.010
- Event Type: End
- Array of ESeal Status: ESSLate, Tamper, ESSTamper, ESSLight, ESSMotion, ESSNoise, BatLow, ESSBatLow
- Has Been Tampered? Y
- Array of Sensor ID: Sync, DataSeal, IDD, Light, Motion, Noise, DSBat, IDDBat
- Array of Sensor Status: Late, Tampered, Tampered, Triggered, Triggered, Triggered, Low, Low
- Array of Event Counts: 90, 1, 1, 14, 5, 1, 3, 26
- Minutes Between Sensor Reports: 2

Site: **AlsCartageKitch** - Location: BTTVancouver.Gate - Direction: In - Lane: Any - Container Num: KLFU121115 - Electronic Seal Number: IABA01554123

- Date Time Occurred: 2005-12-27T20:03:34.057
- Event Type: Normal
- Array of ESeal Status: ESSLate, Tamper, ESSTamper, ESSLight, ESSMotion, ESSNoise, BatLow, ESSBatLow
- Has Been Tampered? Y
- Array of Sensor ID: Sync, DataSeal, IDD, Light, Motion, Noise, DSBat, IDDBat

Figure 23. XML Document Page for a TCOS Event Status


http://localhost/JointData/Templates/EXEC_SP_EDI_418_W2_EquipmentID.xml - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media

- Equipment Desc: CN

- Loaded Status: L

TransSet Control Num: **14530001** - Block ID: ITS - Railcar Seq Num: - 6 - Container Seq Num: 1

- Container Num: DTTX62391

- Equipment Desc: SK

- Loaded Status: L

TransSet Control Num: **14530001** - Block ID: ITS - Railcar Seq Num: - 6 - Container Seq Num: 2

- Container Num: KKFU702811

- Equipment Desc: CN

- Loaded Status: L

TransSet Control Num: **14530001** - Block ID: ITS - Railcar Seq Num: - 6 - Container Seq Num: 3

- Container Num: TRIU927318

- Equipment Desc: CN

- Loaded Status: L

TransSet Control Num: **14530001** - Block ID: ITS - Railcar Seq Num: - 6 - Container Seq Num: 4

- Container Num: KKFU708299

- Equipment Desc: CN

Figure 24. XML Document Page for a Rail Consist (EDI 418)

9.0 CONCLUSIONS

9.1 Summary

The proposed Joint Data Standards and Communication Protocols are vital for achieving efficiency for the Integrated Tracking System and to enhance the global supply chain. We have shown that the network security, database integration, and web demonstration are facilitated by using standard information flows as surrogates to characterize movement and interactions.

The overall objective of the Integrated Tracking System is to provide the JDDSP Multimodal Terminal Operating Systems the required data for tracking containers (empty and full), chassis, rail cars, and trailers. Tracking can be enabled by the EDI or wireless identification equipment, which is described as one combination of the following: optical character readers, active and passive radio frequency identification tags and readers, differential GPS, and CVISN.

This project involves the development of an overall Information Technology conceptual design based upon:

- Data layer collection (RFID, EDI, etc),
- Relational database design (Central Database),
- XML Web demonstration (Web Portal), and
- Datacenter management.

The Integration Tracking System is supported by the Joint Data Standards and Communication Protocols for:

- Automated ETL data transfer (no manual intervention) and
- Instant query using XML in a net centric environment.

The ITS supported by the Joint Data Standards and Communication Protocols can support the improvement of military surge force deployments, military sustainment shipments, and commercial shipments. The JDDSP database will provide the tracking data necessary for operating the JDDSP multimodal terminal operating system and will able exception management of shipments into and out of the facility.

9.2 Future Research

Additional research on the integrating RFID tag messages and EDI/AEI databases to populate a comprehensive listing of all container contents, bill of lading, and shipping details can significantly benefit the industry. Therefore, further research to extend completed work includes:

- Given a container number or military equipment identification, how can this shipment position be located and displayed on a geographic information system map?

- Given a shipping destination for a container number or military equipment identification, how can a shipping destination change be directed from a routing instruction?
- Given rapid volume growth rate, how can the integration database system be converted to a data warehousing system?

As conceived and executed, this project is intended to enhance goods movement by providing an integrated means for describing the dynamics of shipment tracking modeling for public and private stakeholders and to conduct data collection by standardizing the transforming process. The JDDSP Database provides a means of obtaining answers to “What if?” questions. The database is intended to answer questions such as those associated with the impact of increasing shipment volume on throughput capacity, the impact of changes in marine terminal loading processes, and alternative future infrastructure capacity (e.g., Victorville inland cargo operation and freeway, railway movement visibility).

REFERENCES

AAR AEI file format T-94, SAIC publication of T-94 file formats.

ASC X12 Standards for EDI (2001), Version release 004050, Data Interchange Standards Association, Inc., October, 2001.

Braun, David, Jeff Sivils, Alex Shapiro, Jerry Versteegh. Unified Modeling Language Tutorial, Atlanta: Addison-Wesley, 2007.

CLM (Car Location Message), AAR Shipment Status file.

DISA (Data Standards Interchange Association), Release Version 004030.

Fleet, Dina, and Warren, Matt (1997), Active Web Database Programming, Indianapolis: Sams.net, 1997.

Inland Empire Main Line Rail Study Final Report, prepared for Southern California Association of Governments, Contract 04-010, June 30, 2005. The report may be downloaded from the SCAG web site, http://www.scag.ca.gov/goodsmove/pdf/FinalElasticityReport0905rev1105.pdf .

Lu, Quan, Maged Dessouky, and Robert C. Leachman, "Modeling Train Movements Through Complex Rail Networks," *ACM Transactions on Modeling and Computer Simulation*, Vol. 14, No. 1, January, 2004, p. 48-75. See also Dessouky, Maged, Robert C. Leachman and Quan Lu, "Using Simulation Modeling to Assess Rail Track Infrastructure in Densely Trafficked Metropolitan Areas," *Proceedings of the2002 Winter Simulation Conference*, E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds., 2002.

Mallon, Larry G., J. D. Hwang and R. C. Leachman, *Optimization of Military and Commercial Goods Movement Through Southern California Using Information Technology*, prepared for US Navy Space and Naval Warfare Systems Center, Center for Commercial Deployment of Transportation Technologies, Cal State University at Long Beach, Sept., 2003.

Morneau, Keith, and Batistick, Jill (2001), Active Server Pages, Cambridge: Thomson Learning, 2001.

Pritsker, A. B. and J. J. O'Reilly. 1999. *Simulation with Visual SLAM and Awesim.* John Wiley & Sons, Inc., New York.

Rob, Peter, and Coronel, Carlos (2002), Database Systems: Design, Implementation, and Management, 5th Edition, Boston: Course Technology, 2002.

T94 Record Format, AEI Systems Operation Manual (2000), AEI Systems, Inc., November, 2000.

Wynkoop, Stephen (2003), Using Microsoft SQL Server, 2nd edition, Que Corporation, Hollis, NH, August, 2003.

LIST OF ABBREVIATIONS

AAR	Association of American Railroads
AD	Active Directory
AEI	Automated Equipment Identification
AI	Artificial Intelligence
ANSI	American National Standards Institute
APL	American President Lines
APS	Agile Port System
ASC	American Standards Committee
ASP	Active Server Pages
AVI	Automatic Vehicle Identification
BNSF	Burlington Northern Santa Fe
CBL	Commercial Bill of Lading
CLM	Car Location Messages
CSULB	California State University, Long Beach
CVISN	Commercial Vehicle Information System and Networks
DB	Database
DBA	Database Administrator
DBMS	Database Management System
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DPO	Distribution Process Owner
DSS	Decision Support System
DSTU	Draft Standards for Trial Use
EDI	Electronic Data Interchange
EMT	Efficient Marine Terminal
ERD	Entity-relationship Diagram
ES	Expert System
ETL	Extraction, Transformation, Loading
FAST	Free and Secure Trade
FK	Foreign Key
FTP	File Transfer Protocol
GBL	Government Bill of Lading
GIS	Geographic Information System
GUI	Graphical User Interface
ICTF	Intermodal Rail Transfer Facility
ID	Identification
IIS	Internet Information Services
IMS	Information Management System
IP	Internet Protocol
ISP	Internet Service Provider
IT	Information Technology
ITS	Integrated Tracking System
JFCOM	Joint Forces Command
LA/LB	Los Angeles/Long Beach

LAN	Local Area Network
NF	Normal Form
NWITC	Northwest International Trade Corridor
OCD	Operational Concept Document
ODBC	Open Database Connectivity
PC	Personal Computer
PK	Primary Key
POD	Point of Debarkation
POE	Port of Embarkation
PPGBL	Personal Property Government Bill of Lading
RFID	Radio Frequency Identification
RSCS	Regional Supply Chain Simulation
SCLA	Southern California Logistics Authority
SDDC	Surface Deployment and Distribution Commands
SM	Strategic Mobility
SQL	Structured Query Language
SW	Software
TCMD	Transportation Control and Movement Document
TCOS	Trade Corridor Operating System
TCP	Transmission Control Protocol
TPL	Third-party Logistics
UML	Unified Modeling Language
UMLER	Universal Machine Language Equipment Register
UP	Union Pacific
URL	Uniform Resource Locator
USTRANSCOM	US Transportation Command
VPN	Virtual Private Networks
WAN	Wide Area Networks
WSDOT	Washington State Department of Transportation