

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 01/26/2009 | 1.0 | Created | Jim Heary |
| 04/09/2014 | 2.0 | Updated | Jim Heary |
|  |  |  |  |

Table of Contents

Introduction 4

Overview 4

Scope 4

Risks 5

Business Activities 6

Create Schedule 6

System Analysis 7

Overview 7

Actors 7

Use Cases 8

View Schedule 8

Create Schedule 8

Update Schedule 8

Assign TLs 9

Assign Trailer 9

Assign Paperwork 9

Business Rules 9

Non-functional Requirements 9

Presentation 9

Performance 9

Security 9

Availability 9

Concurrency 9

Interoperability 9

Key Abstractions 10

States 10

System Design 11

Overview 11

Design Model 11

Structure 11

Behavior 12

User Interfaces 12

Persistence 13

Availability 13

Concurrency 13

Security 14

Deployment 14

Project Planning 16

Iteration 1 16

Appendix 16

Reports 16

# Introduction

## Overview

Argix Logistics needs to manage line hauling freight into and out of its’ National, local, and agent terminals. The Line Haul department in Jamesburg is responsible for coordinating the activities necessary to ensure carriers are scheduled to move trailers and agent are ready to receive sorted freight.

A Ship Schedule is used by the National terminal and each local terminal that sorts direct freight to manage the line haul of sorted freight out of each terminal to an agent terminal for local deliveries.

Outbound Ship Schedules

At the National terminal, sorted freight is line hauled to agent terminals using Argix and third party carriers. These line haul commitments are managed using the Jamesburg Ship Schedule. The Jamesburg Ship Schedule

At the local terminals,

Terminal Ship Schedule

Inbound Ship Schedules

The Shipper Ship Schedule specifies

Carriers

Email notifications

Agents

Email notifications

This document details known requirements and a candidate design. It does not address manpower requirements or project scheduling, although these can be determined from this architecture.

*How to read this document:*

Management- Introduction, Analysis (Overview, Business Activities, Actors), Project Planning

Operations- Introduction, Analysis, Project Planning

IT- Introduction, Analysis, Design, Project Planning

Context

The following diagram provides context for the system under discussion. This is a high level view of the structure of the system; it helps us to understand the people, hardware, and systems that interact with our system.



## Scope

Solution Boundary

The solution will provide...

Constraints

1. The solution shall conform to the principles of a Service Oriented Architectures (SOA).
2. The solution will use the current Microsoft-based platform and Microsoft-based development tools.

## Risks

1. There are no risks to address at the current time.

## Business Activities

The following diagrams show the business activities for the Ship Schedule system. Business activities are modeled using activity diagrams. Activity diagrams show a series of activities, the actors responsible for each activity, and the information consumed and created during the process. These diagrams use the language of the business. Activity diagrams drive the discovery of Actors, Use Cases, and Key Abstractions.

Create Schedule

Description.



# System Analysis

## Overview

The purpose of defining system behavior is to discover, capture, and analyze the requirements of the system under discussion. This is achieved by describing the requirements (i.e. the conditions or capabilities to which the system must conform) well enough so that an agreement can be reached between the business users and the system developers on what the system should and should not do. It begins by modeling the business processes with a series of activity diagrams. These diagrams drive discovery of the users of the system (i.e. Actors), the system functionality (i.e. Use Cases), and the vocabulary of the system (i.e. Key Abstractions). From these artifacts, an analysis model is created that drives system design and development.

## Actors

An actor specifies a role played by a user or any other system that interacts with the system under discussion. Actors influence UI design and security concerns. The list below provides definitions for the actors surrounding the Ship Schedule system.

* Ship Schedule- the system under discussion.
* LineHaul Supervisor
* LineHaul Clerk

## Use Cases

A Use Case is a list of steps, typically defining interactions between a role (i.e. Actor) and a system to achieve a goal. The actor can be a human or an external system. Use Cases describe the functional view of the system under discussion as a set of business transactions. Use Cases influence UI design, domain models, application service interfaces, and define business transactions. The following Use Case diagram shows some, if not all, of the actors and use cases involved in the system.



View Schedule

The Actor needs to…

Create Schedule

The Actor needs to…

Update Schedule

The Actor needs to...

***Pre-Conditions:***

**Flow of Events**:

Basic Path

Alternative Paths

The system does not have

Post-Conditions:

Assign TLs

The Actor needs to…

Assign Trailer

The Actor needs to…

Assign Paperwork

The Actor needs to…

## Business Rules

1. Rule 1.

## Non-functional Requirements

Presentation

Rich client user interface with spreadsheet capabilities. No tablet or mobile at this point.

Performance

Unspecified.

Security

Authentication

Users will be authenticated by the Argix network domain.

Authorization

Role-based authorization (i.e. Roadshow Specialist) using domain accounts.

Availability

Normal business hours.

Concurrency

1-3 simultaneous users.

Interoperability

Dispatch- Dispatch services are exposed as a web service.

## Key Abstractions

The class diagram below shows the key abstractions involved in the system. Key abstractions are the key concepts and abstractions that the system needs to handle. They are those things that, without which, you could not describe the system. Key abstractions drive design of the database schema and the domain model (if applicable).



* Ship Schedule- .
* Ship Schedule Trip-
* Ship Schedule Stop-

## States

The state machine view describes the dynamic behavior of objects over time by modeling the lifecycles of objects of each class. Each object is treated as an isolated entity that communicates with the rest of the world by detecting events and responding to them. Events represent the kinds of changes that an object can detect. Anything that can affect an object can be characterized as an event.

Ship Schedule

A Ship Schedule is in a new state...



# System Design

## Overview

The system design includes a rich Windows client that provides management of Ship Schedule. It accesses back-end databases through web services arranged in a Service Oriented Architecture (SOA). Security is provided by an ASP.Net membership database hosting domain accounts. Program services are secured using pre-defined user roles.

## Design Model

Structure

The design is an n-tier physical and logical design composed of a .NET 4.0 Win Forms client (DeliveryPoints.exe), several .NET 4.0 WCF Services (Argix10.Terminal.Services), and a SQL Server 2008 R2 database. The design as a whole is built to conform to a Service Oriented Architecture. Service Oriented Architectures describe ways to build loosely-coupled systems composed from individual services. All software components are built with Microsoft Visual Studio 2010 and compiled to operate in.Net 4.0 framework environments. The structures of the design are defined by drawings of the components and interfaces. The behaviors of the design are defined by sequence diagrams involving the interactions between the components.

Enterprise Patterns

In [software engineering](http://en.wikipedia.org/wiki/Software_engineering), a [design pattern](http://en.wikipedia.org/wiki/Design_pattern) is a general reusable solution to a commonly occurring problem within a given context in [software design](http://en.wikipedia.org/wiki/Software_design). It is a description or template for how to solve a problem that can be used in many different situations. Patterns are formalized [best practices](http://en.wikipedia.org/wiki/Best_practice) that the programmer must implement in the application. [Object-oriented](http://en.wikipedia.org/wiki/Object-oriented) design patterns typically show relationships and [interactions](http://en.wikipedia.org/wiki/Interaction) between [classes](http://en.wikipedia.org/wiki/Class_(computer_science)) or [objects](http://en.wikipedia.org/wiki/Object_(computer_science)), without specifying the final application classes or objects that are involved. The following enterprise design patterns are used in throughout the design:

Transaction Script- a Transaction Script organizes business logic by procedures where each procedure handles a single request from the presentation. They are an excellent choice when we don’t have a middle tier domain model. Transaction scripts are used in the web services to provide business transactions and security for one or more use cases.

Gateway- a Gateway is an object that encapsulates access to an external resource such as a web service or database server. Gateways are used throughout the design for user interface access to middle tier web services, for web service access to backend database services, and for web service access to external hardware including a label printer.

Record Set- a Record Set is an in-memory representation of tabular data; record sets (i.e. Dataset) work well with .NET user interface components.

Data Transfer Object- a Data Transfer Object (DTO) is an object that carries data between processes in order to reduce the number of method calls. Data Transfer Objects are serialized classes exposed by the interfaces of the web services for transactional calls to the middle tier (i.e. Service::Method(DTO dto)).

System Structure

The following drawing shows the components and interfaces (structure) of the Ship Schedule design, and the connections between the various components. Components are denoted with tabbed containers. The rectangles within the components are software units stereotyped with their functional purpose such as a gateway or transaction script.



Behavior

System Behavior

The following drawing shows the behavior of …

[Design Behavior Drawing]

User Interfaces

Windows Client

The Ship Schedule Windows application is built using .NET Win Forms for the .NET 4.0 Framework. It communicates with a middle-tier WCF service using the WcfHttpBinding binding. It is secured using an ASP.NET Membership services database through a WCF Role Service.

The application is built using Visual Studio 2010 and is deployed onto a LAN-based IIS 7.5 web server where it is installed onto client computers using the Click-Once technology.

Persistence

Database Schema

The database schema is initially derived from the Key Abstractions. The drawing below

The database is built and deployed with a SQL Server 2008 R2 database server. The schema is exposed through stored procedures.



## Availability

As dictated by the platform.

## Concurrency

TBD.

## Security

Authentication

Windows network login.

Authorization

The windows application uses ASP.NET role-based security to authorize access to application features and services. The RoleServiceGateway encapsulates the details to access the RoleService from the MembershipServices web service. The RoleServiceGateway also acts as a policy file where role-based access is queried by user interface objects. The following drawing shows the processes and components that realize the application authorization design.



## Deployment

The current deployment consists of a centralized application on the Jamesburg local area network (LAN) with client computers running on the Jamesburg LAN as well. Access to the application will be via the Argix Corporate Portal (i.e. SharePoint) using a Click-Once deployment that assumes the deployment server is always available; although the deployment is cached on the client, the server is always checked for a version update. Additional information concerning the click-once deployment can be found in the release notes accessible from the Help menu.

The deployment platform is as follows:

* Client- Windows7 operating system with .NET Framework 4.0 (full, not client)
* Web/Application Servers- Windows Server 2008 R2 and IIS 7.5 with .NET Framework 4.0
* Database Servers- Windows Server 2008 R2 and SQL Server 2008 R2
* Enterprise Service Bus- Windows Server 2008 R2 and BizTalk Server 2010
* Reporting Server- Windows Server 2008 R2 and SQL Server Reporting Services 2008 R2



# Project Planning

## Iteration 1

Implement the following Use Cases:

Use Case 1

Use Case 2

Address the following risks:

Risk 1

# Appendix

## Reports

Report 1

[Report 1 Image]