Softstar Research, Inc.

Analysis/Design Artifacts

Revision History

Date	Version	Description	Author
6-Oct-00	<1.0>	Initial draft	David Rubin

Table of Contents

INTRODUCTION

This document provides an overview of the artifacts considered 'Analysis and Design' for the Softstar Research, Inc. Engineering Organization.

Like all other Softstar guidelines, this document is designed to help everyone involved in delivering IT solutions at Softstar, and is not intended to constrain any individual or project. If you have a strong reason to go against any recommendation specified here, then document your reason, and get on with it!

The purpose of Analysis and Design is to translate requirements into a specification that models and describes how to build and implement the system. Not all of these artifacts will be applicable for all projects.

Think of this list as a collection of artifacts that may or may not be used in any specific project. It is up to the Development Manager's and Architect for each requirement to determine which artifacts are appropriate, and to what level of detail those artifacts are applicable for any given project.

For more specifics on individual artifacts beyond the explanations in this document, refer to the Rational Unified Process (RUP).

Overview

Analysis Model

An object model describing the realization of use cases, and which serves as an abstraction of the Artifact: Design Model.

The Analysis Model contains the results of use case analysis and business rule analysis, and includes instances of the Artifact: Analysis Class, and other analysis.

Design Model

The design model is an object model describing the realization of use cases, and serves as an abstraction of the implementation model and its source code.

The design model is used as essential input to activities in implementation and test.

State Machine

A finite state machine specifies the states that an object can be in, the events and conditions that cause the object to reach those states, and the actions that take place when those states are reached.

Use-Case Realization

A use-case realization describes how a particular use case is realized within the design model, in terms of collaborating objects.

In general, a use-case realization will be represented using one or more sequence diagrams (a.k.a. Object Interaction Diagrams), along with a subset view of a class diagram containing the classes significant to the particular use-Case.

The realization may also contain a collaboration diagram, if it adds value to the design.

Analysis Class

Analysis classes represent an early conceptual model for 'things in the system that have responsibilities and behavior'.

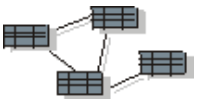
Design Class

A class is a description of a set of objects that share the same responsibilities, relationships, operations, attributes, and semantics.

Design Subsystem

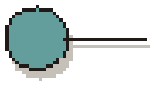
A model element which has the semantics of a package (it can contain other model elements) and a class (it has behavior). Classes or other subsystems it contains provide the behavior of the subsystem. A subsystem realizes one or more interfaces, which define the behavior it can perform.

Data Model



The data model is a subset of the implementation model, which describes the logical and physical representation of persistent data in the system. It also includes any behavior defined in the database, such as ~~stored procedures~~, triggers, constraints, etc.

Interface



A model element which defines a set of behaviors (a set of operations) offered by a classifier model element (specifically, a class, subsystem or component).

A classifier may realize one or more interfaces. Generally, interfaces are specified with regard to subsystems and/or packages (and/or services).

Event Diagram

An event diagram generally corresponds to a State machine, in that it represents different events and the possible actions that can result from those events. Event diagrams are specifically useful for documents events that cross object boundaries.

While it can be argued that any event that crosses object boundaries is really a use case, it is an event diagram that can help flush out the details to more accurately make that determination.

Artifacts

Analysis Model

Overview

An object model describing the realization of use cases, and which serves as an abstraction of the Artifact: Design Model.

The Analysis Model contains the results of use case analysis and business rule analysis, and includes instances of the Artifact: Analysis Class, and other analysis.

Other inputs into the analysis model will include the existing application artifacts and models, along with industry analysis and design patterns.

Note that the analysis model is a combination of graphic models and supporting text and documents.

Symbol

Format

The analysis model will be documented using the UML (Unified Modeling Language) notation.

Tool

Rational Rose

Template

N/A

Design Model

Overview

The design model is an object model describing the realization of use cases, and serves as an abstraction of the implementation model and its source code. The design model is used as essential input to activities in implementation and test.

Note that the design model is a combination of graphic models and supporting text and documents. Also, the design model classes are grouped and organized along classes that exhibit similar functionality and are represented as 'Design packages'.

Symbol

Format

The design model will be documented using the UML (Unified Modeling Language) notation.

Tool

Rational Rose

Template

N/A

State Machine

Overview

A finite state machine specifies the states that an object can be in, the events and conditions which cause the object to reach those states, and the actions which take place when those states are reached.

Not all objects will require an associated state machine diagram. Only those objects that represent significant complexity with regard to state and state transition.

Note: There may be a need to represent 'state machines' using more than the standard UML state notation (i.e. Harrel State Diagrams), for example, a state/event transition matrix.

Symbol

Format

The state machine will be documented using the UML (Unified Modeling Language) notation and/or Harrel State Diagram notation.

Tool

Rational Rose

Template

N/A

Use-Case Realization

Overview

A use-case realization describes how a particular use case is realized within the design model, in terms of collaborating objects.

In general, a use-case realization will be represented using one or more sequence diagrams, along with a subset view of a class diagram containing the classes significant to the particular use-Case. The realization may also contain a collaboration diagram, if it adds value to the design.

Symbol

Format

The use case realization will be documented using the UML (Unified Modeling Language) notation in the form of Object Interaction Diagrams (a.k.a. Sequence Diagrams) and/or Collaboration diagrams..

Tool

Rational Rose

Template

N/A

Analysis Class

Overview

Analysis classes represent an early conceptual model for 'things in the system, which have responsibilities and behavior'.

Symbol

Format

The analysis classes will be documented using the UML (Unified Modeling Language) notation. This documentation is generally in the context of an analysis model.

Tool

Rational Rose

Template

N/A

Design Class

Overview

A class is a description of a set of objects that share the same responsibilities, relationships, operations, attributes, and semantics.

Symbol

Format

The design classes will be documented using the UML (Unified Modeling Language) notation. This documentation is generally in the context of a design model.

Tool

Rational Rose

Template

N/A

Design Subsystem

Overview

A model element which has the semantics of a package (it can contain other model elements) and a class (it has behavior). Classes or other subsystems it contains provide the behavior of the subsystem. A subsystem realizes one or more interfaces, which define the behavior it can perform.

Symbol

Format

The design subsystem will be documented using the UML (Unified Modeling Language) notation. This documentation is generally in conjunction with Interface documentation.

Tool

Rational Rose

Template

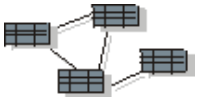
N/A

Data Model

Overview

The data model is a subset of the implementation model, which describes the logical and physical representation of persistent data in the system. It also includes any behavior defined in the database, such as ~~stored procedures~~, triggers, constraints, etc.

Symbol



Format

The data model will be documented using the IEFF modeling notation.

Tool

ErWin

Visual Modeler

Embarcadero

Template

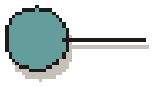
N/A

Interface

Overview

A model element which defines a set of behaviors (a set of operations) offered by a classifier model element (specifically, a class, subsystem or component). A classifier may realize one or more interfaces. An interface may be realized by one or more classifiers. Any classifiers which realize the same interfaces may be substituted for one another in the system. Each interface should provide an unique and well-defined set of operations.

Symbol



Format

Interfaces will be documented using the UML (Unified Modeling Language) notation. This documentation is generally in conjunction with subsystem and package documentation.

Tool

Rational Rose

Template

PROJECT DEPENDANT

Event Diagram

Overview

An event diagram generally corresponds to a State machine, in that it represents different events and the possible actions that can result from those events. Event diagrams are specifically useful for documents events that cross object boundaries.

While it can be argued that any event that crosses object boundaries is really a use case, an event diagram can help flush out the details to more accurately make that determination.

Symbol

Format

Event diagrams will be documented using the UML (Unified Modeling Language) notation. This documentation is generally in conjunction state machine documentation.

Tool

Rational Rose

Microsoft Word

Template

PROJECT DEPENDANT