The Synergy Approach™

Raghav S. Nandyal
Chief Executive Officer

SITARA Technologies Pvt. Ltd.
'Bellevue', #559 • Road No. 3 • Banjara Hills • Hyderabad AP - 500 034 • INDIA
Email: raghav_nandyal@SITARATECH.com
URL: http://www.SITARATECH.com

A Near Zero-Defect Approach to Software Development

From The

Software Engineering Laboratory

March 1995

TO PROVIDE A FLAVOR FOR THE STEPS INVOLVED WITH THE USE OF OBJECT ORIENTED DESIGN, THE REAL TIME CLOCK EXAMPLE IS DISCUSSED HERE.

Real Time Clock using Object Modeling Technique

1.0 INTRODUCTION

This document presents Object Oriented Design of Time Display System. Object Modeling Technique (OMT) of James Rumbaugh is the design methodology followed in the design of the Time Display System.

2.0 TIME DISPLAY SYSTEM

- Analysis Phase

PROBLEM STATEMENT

Same as presented in the FRS in Lecture 2. **(SITARA SE/JUNE 2001)**

Analysis in OMT comprises of developing three models. The Object model, Dynamic model and the Functional model. The resulting three models are the inputs to the design phase. In Shlaer-Mellor Object Oriented Approach to Domain Analysis, the corresponding correlation to the three models is Information Model (Object Model), State Model (Dynamic Model) and Process Model (Functional Model).

2.1 OBJECT MODEL

Object model identifies the classes that could result from the problem definition, their associations and the attributes of each class. Object model only addresses the static behavior of the system.

The following steps will result in the Object Model:

1. Identify Objects and Classes
2. Identify Associations
3. Identify Attributes

OBJECT: An object is an abstraction of a set of real-world things such that all the real-world things in the set, "the instances", have the same characteristics; and all instances are subject to and conform to the same rules. That is, "object" in OOA represents a typical, but unspecified member of a class.

ATTRIBUTE: An attribute is the abstraction of a single characteristic possessed by all the instances that were themselves abstracted as an object.

RELATIONSHIP: A relationship is an abstraction of a systematic pattern of association that holds between real world things that were themselves abstracted as objects.

TYPES OF OBJECTS: Perhaps the easiest kinds of objects to develop are those based on abstraction of "tangible things". They may also be abstracted for "roles" of things, rather than for the thing itself. Specifications or quality criteria provide another basis for abstraction. Objects may be useful aggregations of equipment. Objects can be defined to reflect steps in the process of manufacturing.

STATES: Each state represents a condition of the object during which a defined set of rules, laws and policies apply.

EVENTS: Occurrences that signal when an instance of an object is moving from one stage of its lifecycle to another.

ACTIONS: The processes, queries and transactions necessary to draw an object through its lifecycle into actions in the object's state model.

Step 1: Identify Objects and Classes

Objects and classes can be recognized from the problem statement by identifying all the nouns and adjectives from it.

Following are the candidate classes that are identified from the knowledge of the problem statement.

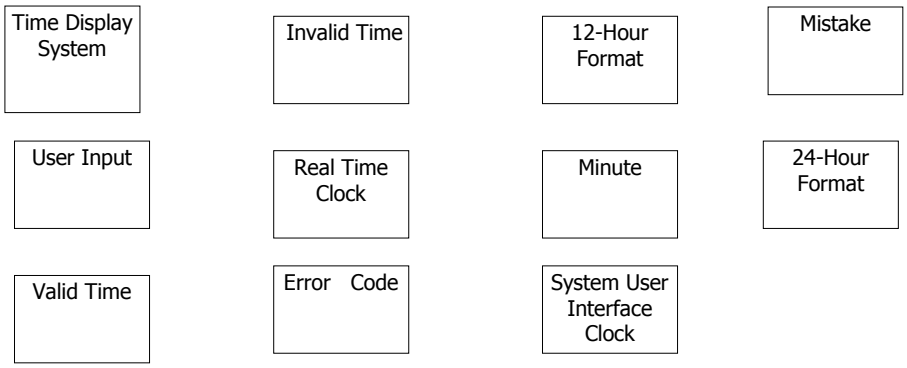


Figure 1. Time display system classes identified from the knowledge of problem domain.

Having identified the classes from the problem statement, we will now discard the classes that are redundant, irrelevant, vague, attributes of classes, operations, roles and are implementation constructs. Since this is purely judgmental, a good rule of thumb to decide this is to identify the classes that have adjectives qualifying them and if the class cannot stand by itself because of it being trivial, then it can be discarded. The following are the steps.

- The Class Time Display System represents the Module we are going to design, so its scope is too big to be considered as a class.
- Classes Error Code, Invalid Time, Mistake and Minute are vague, so are discarded.

- Classes Time Display System and Real Time Clock are redundant, so we discard one of them; say, Real Time Clock.

If adding a few classes based on the domain analysis of the problem helps to improve the overall structure of the system, we do so by adding a few classes. The following figure segregates the bad classes from the good classes.

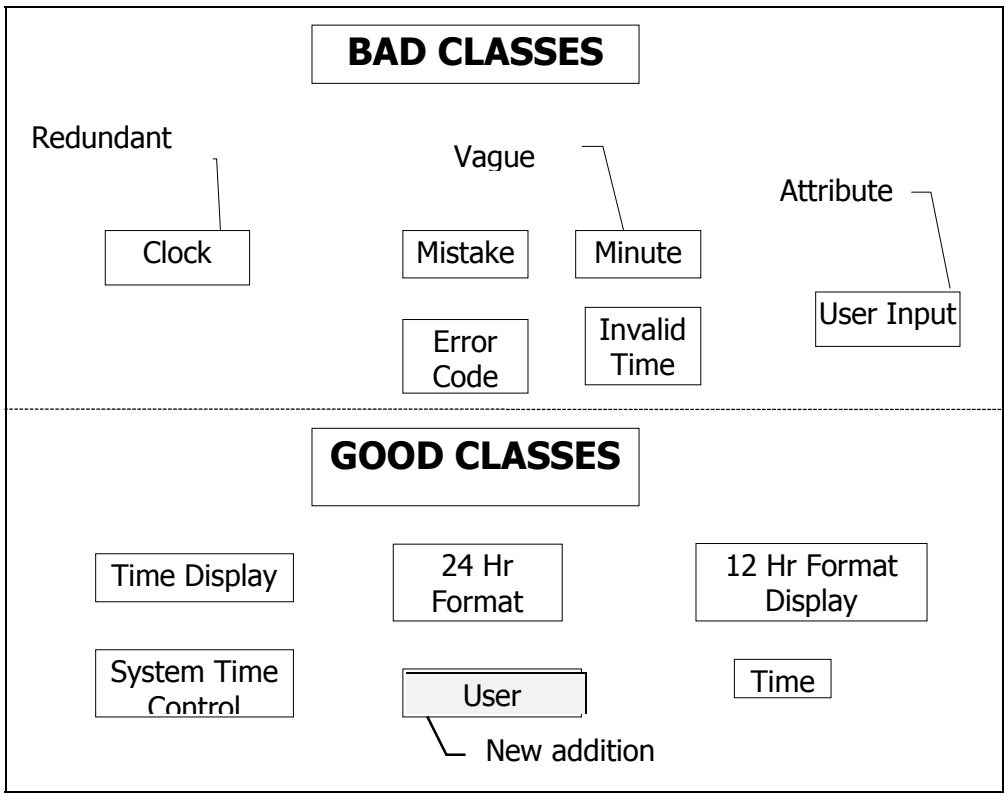


Figure 2. Eliminating unnecessary classes and adding new classes to the Time Display System problem

Step 2. Identify Associations

The following associations are identified for Time Display System problem:

Verb Phrases
Real time Clock shall ***display*** the time Time clock ***shall use*** the system clock Time can be ***set*** to any valid values ***Validation*** of user selected time must be done Invalid time must ***prompt*** the user Display shall ***update*** itself at the end of every minute

The following diagram shows the initial object model with the associations among the objects

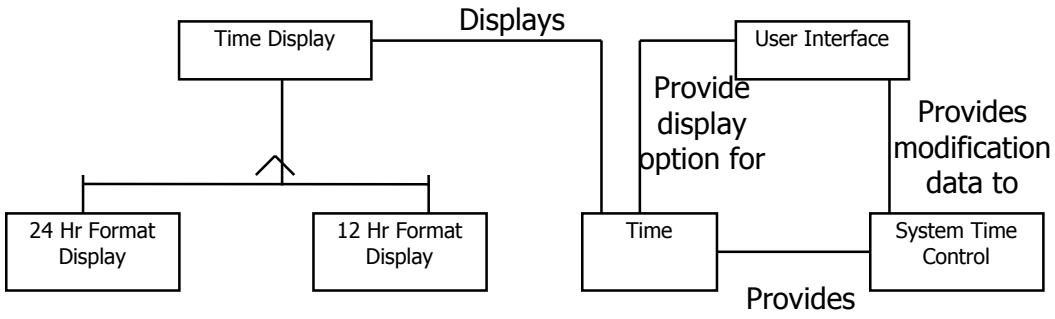


Figure 3. Initial object model with inheritance

Step 3. Identifying Attributes

Attributes are identified for each of the class required for the design of Time Display System. Following table gives the class names with their relative attributes:

Class	Attributes
Time	Display Option System Time
Time Display	
24Hr Format Display	24Hr Format Structure
12Hr Format Display	12Hr Format Structure
System Time Control	Change Information System time
User Interface	New Time Value Display Option

Until now we have identified the classes, their associations and attributes. Now we can combine this information to form our object model for the Time Display System with attributes and inheritance. Following figure gives this model:

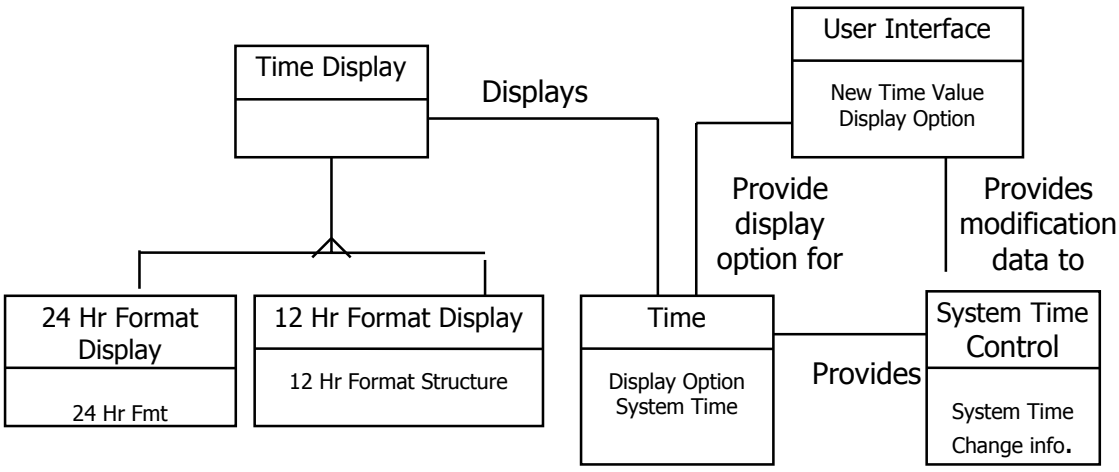


Figure 4. Object Model with inheritance and attributes

2.2 DYNAMIC MODEL

Dynamic model identifies the dynamic behavior of the system. It tries to emulate the behavior of the system under normal and abnormal conditions. This model is used to formalize the "lifecycles" or "life histories" of objects and relationships. In order to make the system of communication clear, state diagrams and transition tables communicate by means of events, and are organized in layers.

The following steps will result in the Dynamic Model

- Preparing Scenarios
- Identifying Event Trace for each Scenario
- Building state diagrams

Step 1. Preparing Scenarios

Scenarios of typical interactions are prepared.

Scenario 1

Following is a scenario when user needs to initialize the display option.

User provides the display option Display option is written to the display storage from where it is to be read at the beginning of every session

Scenario 2

Following is a scenario when user changes the time successfully

User provides the time User provided time is checked for validity Format of user provided time is acceptable Time to display is modified according to the information provided by the user

Scenario 3

Following is a scenario when user fails to change the time successfully

User provides the time User provided time is checked for validity Format of user provided time is not acceptable User is prompted to provide correct format

Scenario 4

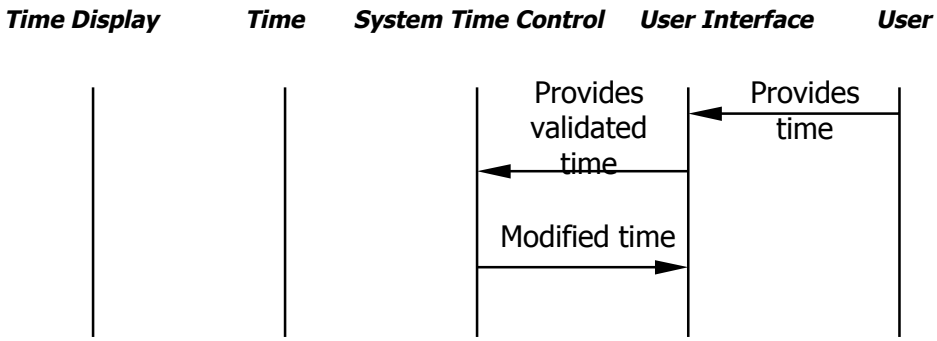
Following is the scenario when user need to display time

User requests for time display System time is obtained from OS Display option is obtained from display storage Time is displayed depending upon the display option Time is updated after every one minute showing latest system time Display continues until user discontinues the display

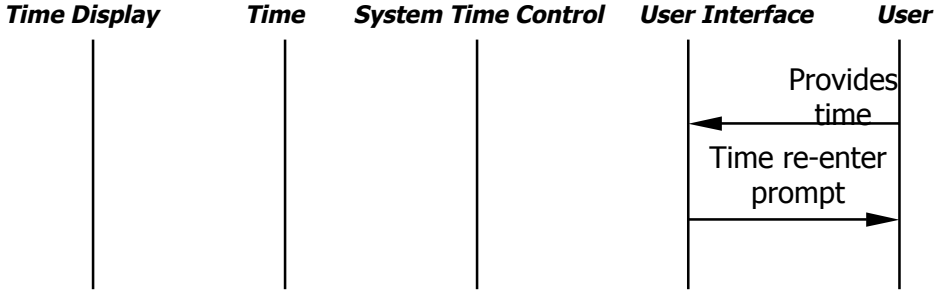
Step 2. Identifying Event-Trace for each Scenario

Event traces are identified for some of the scenarios. It shows how the events are traced between each of the object class.

Event trace for scenario 2.

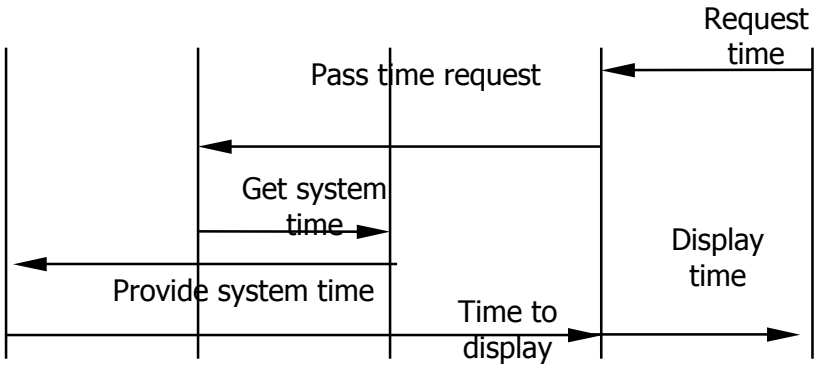


Event trace for scenario 3.



Event trace for scenario 4.

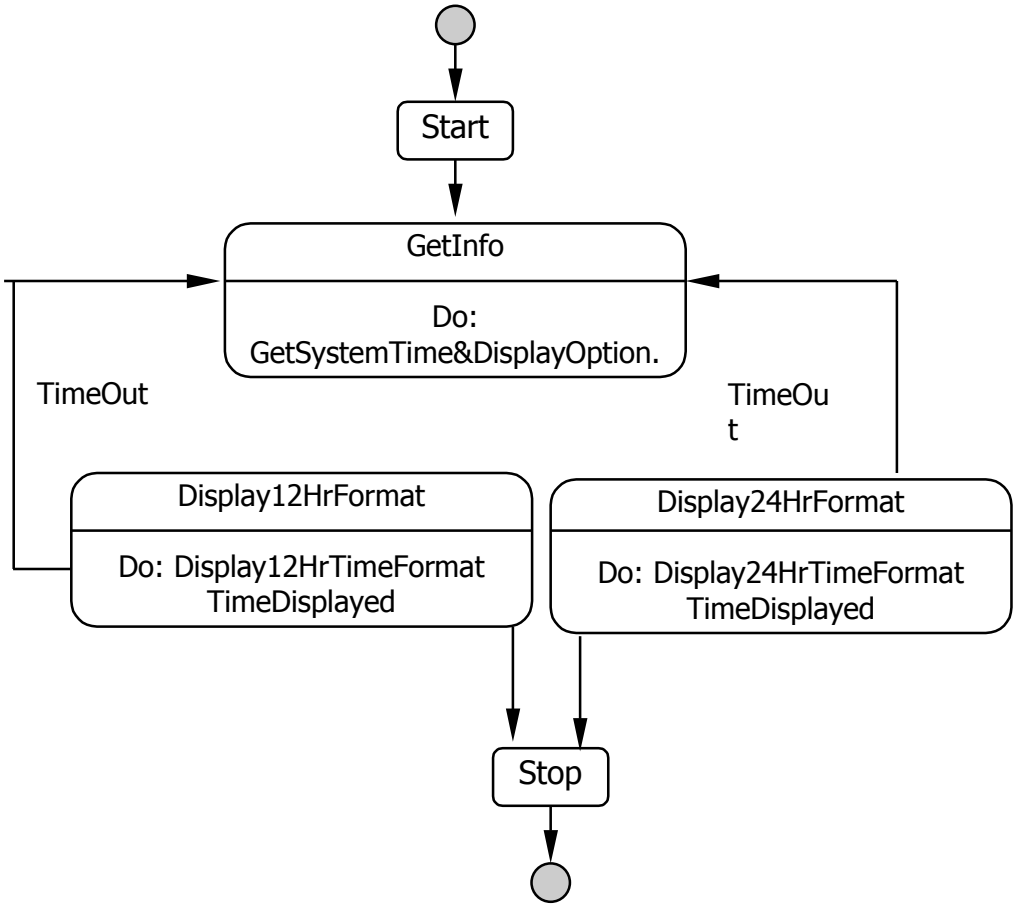
Time Display ***Time*** ***System Time Control*** ***User Interface*** ***User***



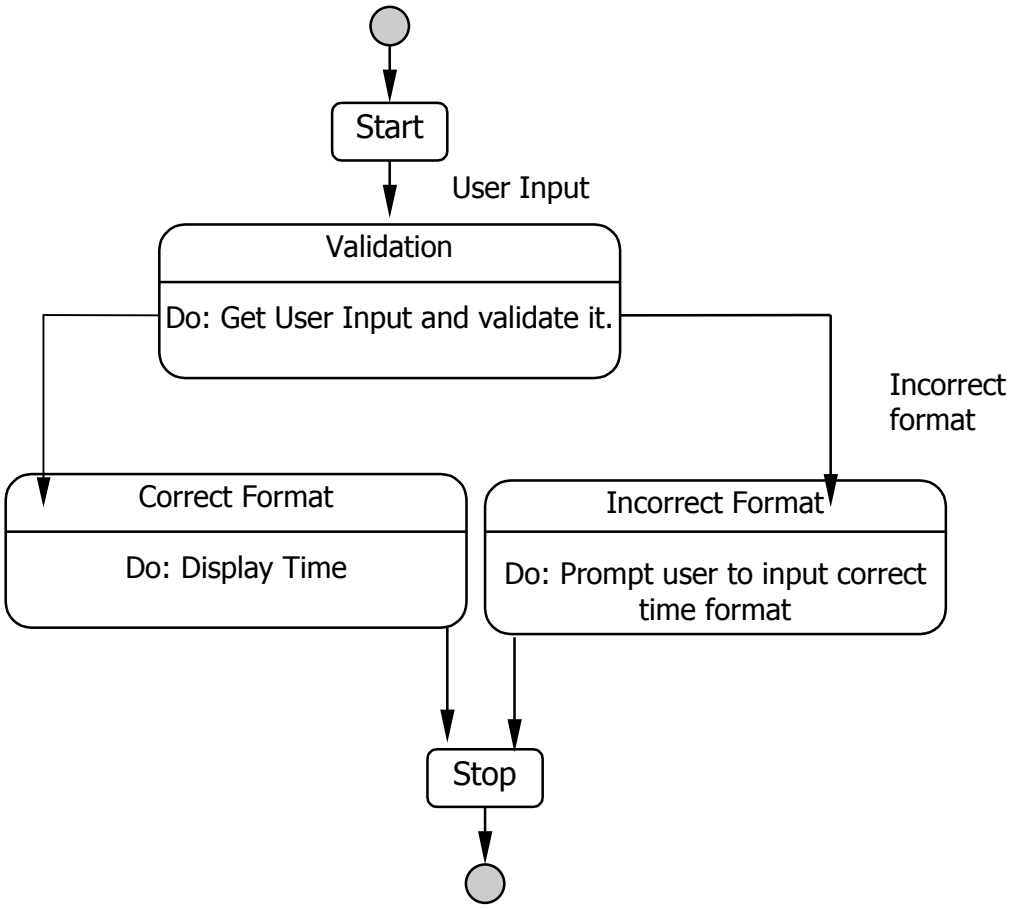
Step 3. Building State Diagrams

A state diagram is made for each object that shows considerable dynamic behavior. State diagrams helps in tracking operations on the objects.

First we make the state diagram for the Time display.



The following diagram presents the dynamic behavior of the update time.



The above two state diagrams covers the scenarios we have already discussed. The dynamic behavior of other classes is trivial and does not require any state diagram.

END OF LECTURE 4

CONTINUED ON LECTURE 5 [SITARA SE/SEPTEMBER 2001]