



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

2.3 FUNCTIONAL MODEL

Functional model shows how values are computed independent of the sequencing, object structure. Functional dependency is shown through data flow diagrams. Functions are represented through pseudo code. The processes drive an object or relationship through its lifecycle and can be derived from the Dynamic Model. Data flow diagrams in their simple DeMarco form are used in this step.

The processes on the data flow diagrams will correspond to activities or actions in the state diagrams of the classes. The flows on a data flow diagram will correspond to objects or attribute values in an object diagram.

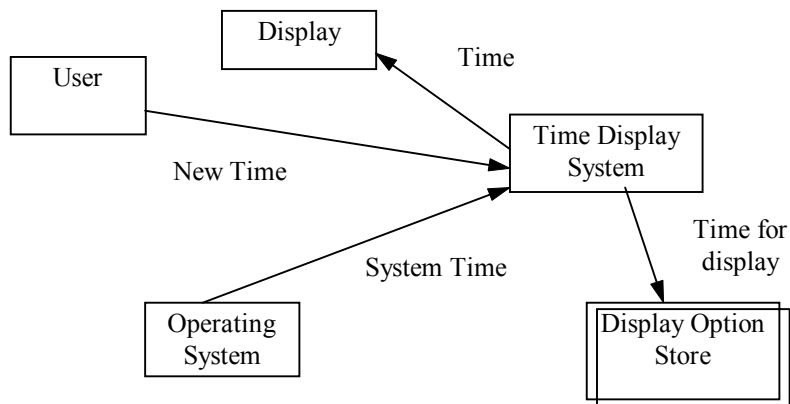
The following steps will result in the Dynamic Model

- Identifying Input and Output Values
- Developing Dataflow diagrams
- Adding operations



Step 1. Identifying Input and Output Values

Following diagram shows the input and output values to the Time Display System. It also shows where the inputs come from and where the output goes.

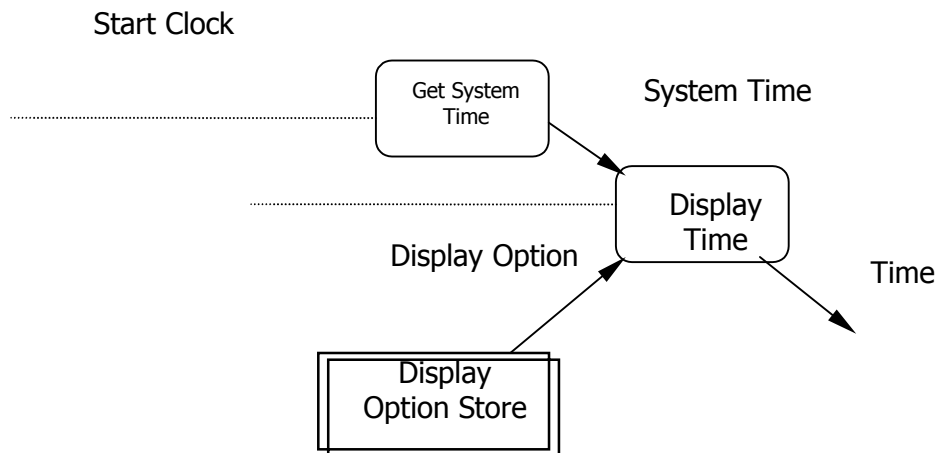




Step 2. Developing Data Flow Diagrams

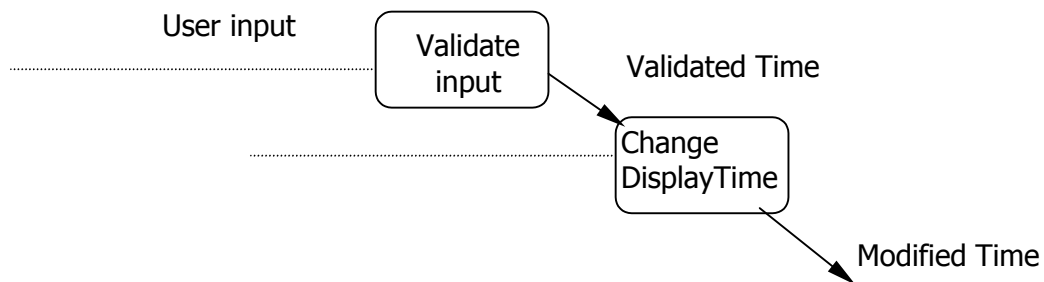
We start with making a top-level data flow diagram.

Flow of data and related processing for display of time is shown in the following diagram:





Flow of data and related processing for changing the display time is shown in the following diagram:



Top level diagrams gives the details about the input of data and how it is processed and converted by the top-level processes.

Next level diagrams expands each of the bubble in the top level diagrams and explains how processing is done as we go deep.



Step 3. Addition Operations

In this step we find all the possible operations traced during object, dynamic and functional model. These operations are listed below along with their identifying model.

Operation Name	Identifying Model
Initialize User Interface()	Object Model
Initialize System Time Control()	Object Model
Initialize Time()	Object Model
Initialize Time display()	Object Model
	Object Model
Handle Display Option()	Dynamic Model
Validate User Input Time()	Dynamic Model
Change System Time()	Dynamic Model
	Dynamic Model
	Dynamic Model
Get System Time()	Functional Model



- Object Design

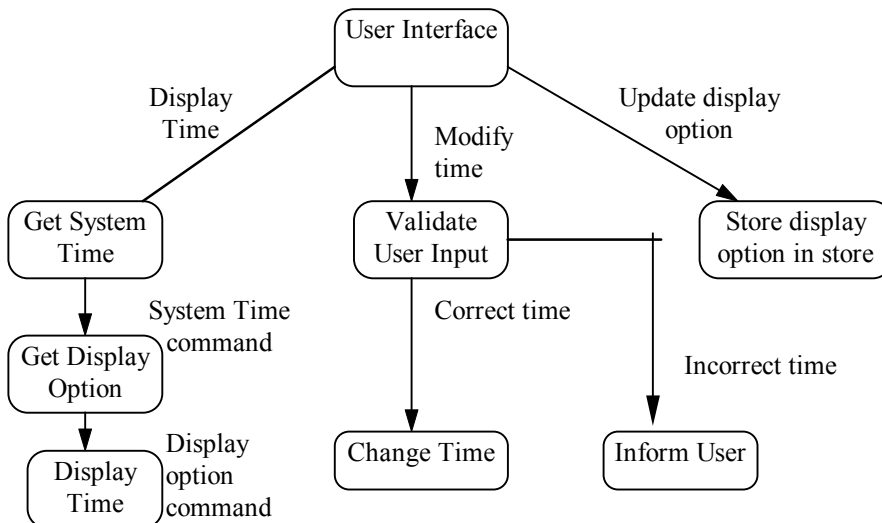
Design phase is built upon the analysis phase. The information we have obtained in the analysis phase will be used to design architecture for the Time Display System.

The following steps will result in Object Design.

- Implementation of Control
- Combining the three models
- Explaining classes including attributes and operations

Step 1. Implementation of Control

One important step that we have delayed so far is the implementation of control. The implementation of control including the concurrency control is shown in the following diagram:

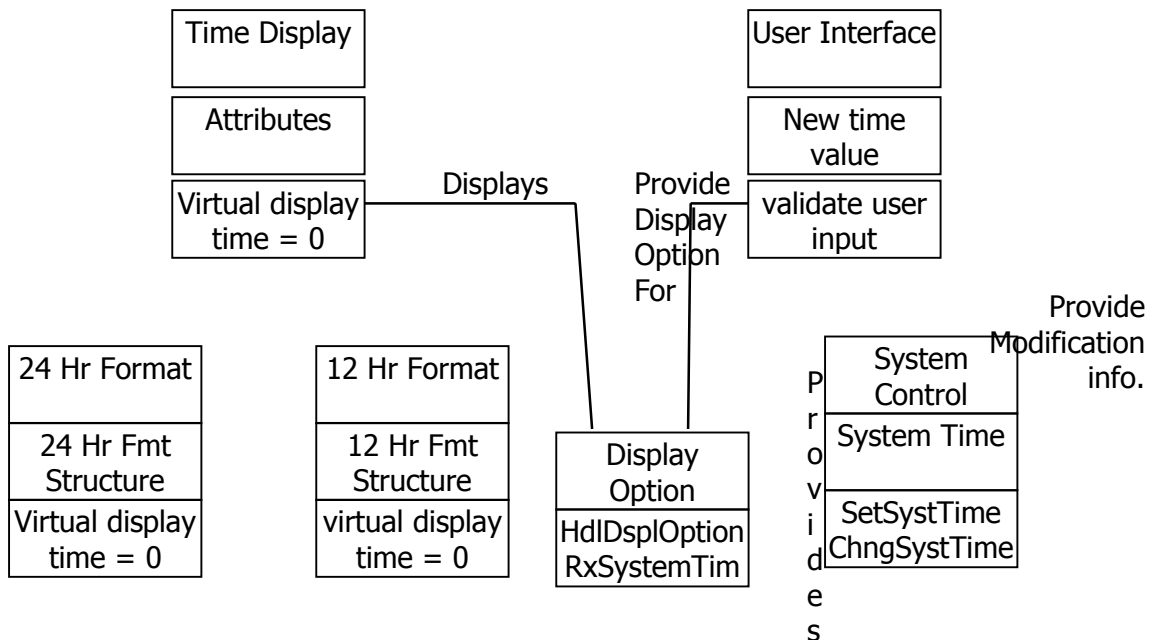




Following is the pseudo-code related to the control of Time Display System.

Step 2. Combining the 3 models

The most important step in the object design phase is to combine the three models to obtain operations on classes. This will provide us with the final object oriented model which will serve as the architecture of the Time Display System. In the functional model we found the operations from the three models and in step 1 of object design we found the control operations. We combine these operations with others required to provide a complete look to the objects and develop the final design as follows:





Step 3. Explaining Classes including Attributes and Operations

Class Time

Attribute Description:

DISPLAY OPTION

Origin Class: Time

Purpose: This attribute keeps the information regarding what format user needs time to be displayed.

Possible Values: {12HrFormat, 24HrFormat }

Value Provider: Display Option Store (In fact this value is provided by user and written to this store)

Description: To find which display format user has specified, Time Display system looks into the Display Option store and inputs the value into this attribute. Every time, when the time has to be displayed, the display format is determined by the information provided by this attribute.

SYSTEM TIME

Origin Class: Time

Purpose: To store the time obtained from operating system.

Possible Values: Any value of time

Value Provider: Operating System

Description: This attribute is used to store the time obtained from the operating system.



Operation Description

- ***handleDisplayOption()***

Origin Class: Time

Description: This operation is used to handle the display option provided by the user. This information is written to persistent storage called Time Display Store.

Input: display option

Returns: OK flag; which is a character

Algorithm:

Get display option from user()
write display option to the Display Option Store()

- ***receiveSystemTime()***

Origin Class: Time

Description: This operation is used to get the system time from class system time control which in turn gets it from the operating system..

Input: void

Returns: OK flag; which is a character

Algorithm:

***send** get time message to system time control ()*
***write** system time to local attribute()*



ACRONYMS

CFD	:	Control Flow Diagram
DDE	:	Data Dictionary Entry
DFD	:	Data Flow Diagram
FRS	:	Functional Requirements Specification
GQM	:	Goal, Question, Metric
HLDD	:	High Level Design Document
MSpec	:	Module Specification
PDCA	:	Plan, Do, Check, Act (Shewart-Deming Cycle)
PSpec	:	Process Specification
REQB	:	Requirements Book
RMF	:	Review Meeting Form
RMNF	:	Review Meeting Notice Form
RF	:	Reviewer's Form
RDQAF	:	Review Disposition and Quality Assurance Form
SPI	:	Software Process Improvement
SSTE	:	Software System Test Engineer
SSTP	:	Software System Test Plan
TQM	:	Total Quality Management
UTE	:	Unit Test Engineer
UTP	:	Unit Test Plan

END OF LECTURE 5

For More Information,
Contact: Shailaja R. Nandyal
Email: shailaja_nandyal@SITARATECH.com