An Approach of UML for an Elevator System by Implementing Stereotypes

Clarence J M Tauro[1], Dr. N Ganesan [2], Rashmi N C[3], Janet Agnes N C[4]

[1] Centre for Research, Christ University
Hosur Road, Bangalore, India
[2] Director (MCA), RICM
Bangalore, India
[3 4] Department of Computer Science, Christ University
Hosur Road, Bangalore, India

***Abstract* — this paper studies the extension mechanisms of the UML in the object oriented application and it introduces a new approach of using Stereotypes and shows its usefulness by implementing in an elevator control system. This paper also discusses the process of specifying, translating and verifying UML specifications for an elevator system. Different kinds of existing UML Meta-models used in an elevator system are analyzed UML elevator system by implementing the Stereotypes will be created based on the real time system using the analysis report.**

Keywords-component; formatting; UML; Real Time object oriented systems;Stereotypes,

I. Introduction

Motto of real time application is to develop a good software model to reduce the complexity of the system. Unified Modeling Language (UML) simplifies complex process of software design by making blueprint of software architecture [1]. UML brings a set of notations and concepts that meets the needs of typical software modelling projects, but in some scenarios users found UML unable to express their modeling needs and its limitations are removed in UML by three extension mechanisms Stereotypes, Constraints and tagged values. Elevator control system is a complex real time multi task system. This report is on the basis of the system design describes elevator implementation using Stereotypes.

This paper consist of four sessions: The UML extension is given in the first section. The second section would cover the existence of Stereotypes in the UML. The Elevator control system is been analyzed with the diagram in third section and implemented with the Stereotypes in the fourth sections.

UML Extension Unified modeling language (UML) is the defacto standard object oriented modeling language for constructing the real time object oriented application. UML has a set of diagrams to describe the structure and behavior of the system. It allows obtaining an appropriate model to the different domains. All the UML extensions should follow the standard proposed by the OMG [3]. There exist four common mechanisms that can be used consistently throughout the language - specifications, common divisions, adornments, and extensibility mechanisms. The main focus of this work will be the extensibility mechanisms.

II. Extensibility Mechanisms

The extensibility mechanisms allow customizing and extending the UML by adding new building blocks, creating new properties, and specifying new semantics in order to make the language suitable for problem domain [4]. There are three common extensibility mechanisms that are defined by the UML:

- Stereotypes: Adding new model element
- Tagged values: Creating new properties
- Constraints: specify new constraints

1. stereotypes

Stereotypes allow extending the vocabulary of the UML so that we can create new model elements, derived from existing ones, but which have specific properties that are suitable for the problem domain. They are used for classifying or marking the UML building blocks in order to introduce new building blocks. Stereotypes also allow us to introduce new graphical symbols for providing visual effects to the models that speaks the vocabulary of a specific domain.

2. Tagged values

Tagged values are properties for specifying keyword-value pairs of model elements, where the keywords are attributes. They allow you to extend the properties of a UML building block so that you create new information in the specification of that element. Tagged values can be defined for existing model elements, or for individual stereotypes so that everything with that stereotype has that tagged value.

3. Constraints

Constraints are properties for specifying semantics and/or conditions that must be held true at all times for the elements of a model. They allow extending the semantics of a UML building block by adding new rules or modifying existing ones.

A. Existence of the Stereotypes

Even though the UML is very well-defined there might be situations in which it might lack in wanting to bend or extend the language in some controlled way to control it to specific problem domain in order to simplify the communication of objective. In such situation the UML Extensions mechanism would come into picture to overcome the limitation of UML.

Stereotypes are used to extend the UML notational elements, to classify and extend associations, inheritance relationships, and classes. It could also be used to indicate a meaning or different use betweens two elements with identical structure and also specify a geometric icon to be used to present elements with the stereotype.

III. Real Time Domain system – The elevator control mechanism

The Elevator control system discussed in the paper would be the ideal elevator which consists of only one elevator cabin, which travels between each floor in the building; there is a single button on each floor to call the lift. Inside the elevator cabin there are a series of buttons, one for each floor. Requests are definitive, i.e., they cannot be cancelled and they persist, thus they should eventually be serviced. The arrival of the cabin at a floor is detected by a sensor. The system may ask the elevator to go up, go down or stop. The system may ask the elevator to open its door. The system will receive a notification when the door is closed. This simulates the activity of letting people on and off at each floor. The door closes automatically after a predefined amount of time. In order to ensure safety, emergency brakes will be triggered and the car will be forced to stop under any unsafe conditions. When an elevator has no request, it remains at its current floor with its door closed. Based on the above scenario the UML diagram is been implemented in the Figure 1 and Figure 2. Use case diagram captures the functional requirement of the system and it's the interaction between the actor the system.

Consider Figure 1 which describes the Use case diagram with basic function of elevator using the stereotypes.
Use case description:

- Request Elevator
- Indicate Position
- Emergency
- Open/Close
- Indicate Moving direction

Earlier Use Case diagram was successful in capturing the system interaction how exactly system and actor communicate but was unsuccessful in showing the characteristics of system that's whether its input device, output device, mandatory or optional, in this scenario stereotypes came into picture. `<<include>>, <<extend>>` are stereotypes used in the Figure1. `<<include>>` stereotype indicates a necessary relationship between two Use Case `<<extend>>` stereotypes indicates optional relationship between two use cases. With the help of stereotypes any users can understand the behaviour of the system.

Class diagram describes the static designs which in turn help us to understand the functional requirements of the system, how the system is composed from the description of classes. A class diagram shows a set of classes, interfaces collaborations and their relationships. Figure 2 describes the significance of stereotypes in elevator implementation. Class diagram captures the basic functional aspects of the elevator like elevator button, floor button, sensor, door, floor light. These are the stereotypes `<<electronic system>>, <<input device>>, <<output device>>` which describes the behaviour of classes.

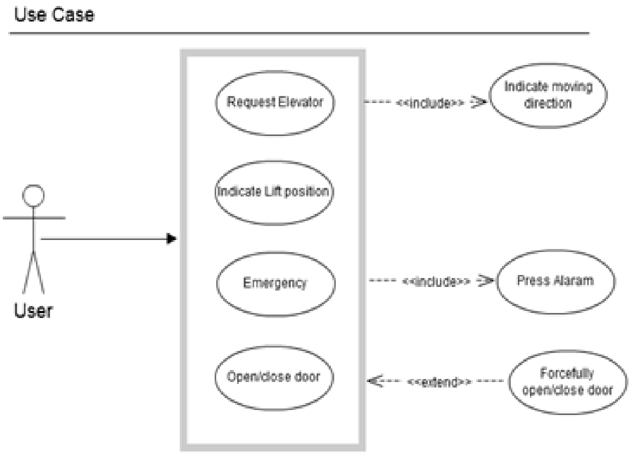

Figure 1: Use Case Diagram of Elevator System

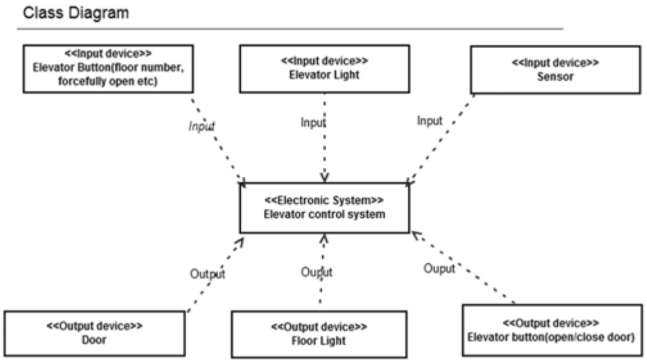

Figure 2: Class Diagram of the Elevator System

IV. Significance of Stereotypes in Real Time System

According to *Kopetz*, a real time computer system is a computer system in which the correctness of the system behaviour depends not only on the logical results of the computations, but also on the physical instant at which these results are produced[2].

Stereotypes play significant role in real time design, there are different kinds of objects each may represent the most common kinds of objects that can be found in real-time systems design as well as a set of relationships that can be defined among classes belonging to these stereotypes. A stereotype class identifies some common characteristics for a set of classes. Inheritance restrictions, synchronization constraints, implicit behaviour, method protocols are the examples of the Stereotype classes. A common set of stereotypes for real-time system design includes the following methods:

- *Subsystem*: A subsystem is an abstraction of a complex component.
- *Passive Class:* passive classes can only change their state when they are requested by other objects, by means of a method or operation invocation.
- *Protected Class:* protected classes can only Change their internal state through atomic operations. They represent data or other objects which are used by more than one active object
- *Cyclic Classes:* Cyclic objects are used to represent periodic behaviour. They have an independent thread of control,
- *Resource Classes*: Data resources can be represented with protected classes, but there are other resources, like CPU, input output devices, or memory, that impose constraints on the system execution.

V. Conclusion

In this report, a detailed UML documentation for elevator control system using stereotypes is given. The UML diagram used in the documentation are Use Case and Class diagrams. Every UML diagram should be just a graphical representation of system and having its own significance. An UML diagram is combined to capture the behaviour of the real system.

Stereotype describes the behaviour/state of individual objects. Hardware and software features of an elevator system are explained in this paper by using the stereotypes.

However, functions of the elevator system described in this report are limited to basic functionality.

Acknowledgment

We would like to thank Prof. Jiberial, Department of Computer Science, Christ University, for his constant guidance and support which helped us to complete this paper successfully.

References

[1] Lu Lu, “UML Documentation for an Elevator System,” Distributed Embedded Systems, Fall 2000, December 2000.

[2] Hermann Kopetz. Real-Time Systems, Design Principles for Distributed Embedded Applications.UML 2003-the unified modeling language: modeling languages and applications: 6th international conference, San Francisco, CA, USA, October 20-24, 2003: proceedings

[3] UML 2003-the unified modelling language: modelling languages and applications : 6th international conference, San Francisco, CA, USA, October 20-24, 2003 : proceedings

[4] Jasmine Farhad, “The UML Extension Mechanisms”, Dept of computer science, university college London, 13-Dec-02.

[5] Miguel A, Alejandro Alonso, Juan A. de la Puente, Object-oriented Design of Real-time Systems with Stereotypes, E. T.S. I. Telecomunicacio'n, Universidad Polite'cnica de Madrid Ciudad Universitaria, s/n, Madrid, E-28040 Spain