

Information Design for Urban Traffic Simulation

Manuj Darbari¹ , Bhaskar Karn², Prof.Dr. P. K. Bansal³ , Rajesh Goel⁴

1.Assistant Professor, BBDNITM, A-649, Indira Nagar, Lucknow, manujuma@rediffmail.com

2.Coordinator, B.I.T, Mesra Ranchi, bhaskar203001@rediffmail.com

3.Principal, Malout Institute of Management and Information Technology(MIMIT),Malout-Punjab.
India-152107, pawankumarbansal@yahoo.co.in

4. Associate Professor and Head, Electronics and Communication Engineering Department, Ambala College of engineering and applied Research, Post Office- Sambalkha, Ambala Cantt-133101-Haryana India.
ecedept2003@yahoo.com

Abstract

This paper focuses on the present Indian Traffic conditions in Sub-metro city of India of Lucknow. We will be focusing on all the design aspects of Information Flow Model which is divided into two parts namely Urban Decision Subsystem and Urban Physical subsystem. The main focus is on the issues where change in any of the above subsystem should coordinate well with other subsystem. There should be high degree of information flow between these two components.

Keywords: UML, XMI, UTS.

1.0 Introduction :

The rapid growth of UTS applications in recent years is generating an increasing need for tools to help in system design and assessment. Traffic simulation models have proven to be one of the main cost-effective tools to reach these objectives. Tools are needed to improve the increasing complexity and rapidly deteriorating transportation systems of today. The ability of traffic simulation models to quantify the urban transportation system is still missing even today. Currently there are 80 simulation models available out of them CORSIM and ITEGRATION appear to have the highest probability of success in real world applications. But both of them suffers from the disadvantage of high complexity of the system, which makes it difficult to be enhanced or customized according to specified user's requirements.

Our focus is mainly on the design methodology which highlights UML for developing the entire modeling structure which can be stored in XMI format for further up gradation at later stage. In this paper we have shown some of the class libraries that can be represented into two forms Decision subsystem libraries and Physical subsystem libraries. These approaches that can be used to achieve consistent software evolution such as to develop a comprehensive language[1] that covers all the dimensions of software or to develop a mechanism that integrates tools for different dimension.

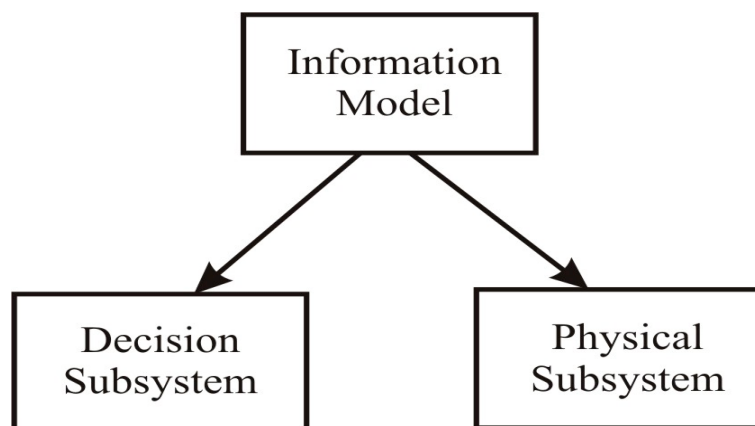
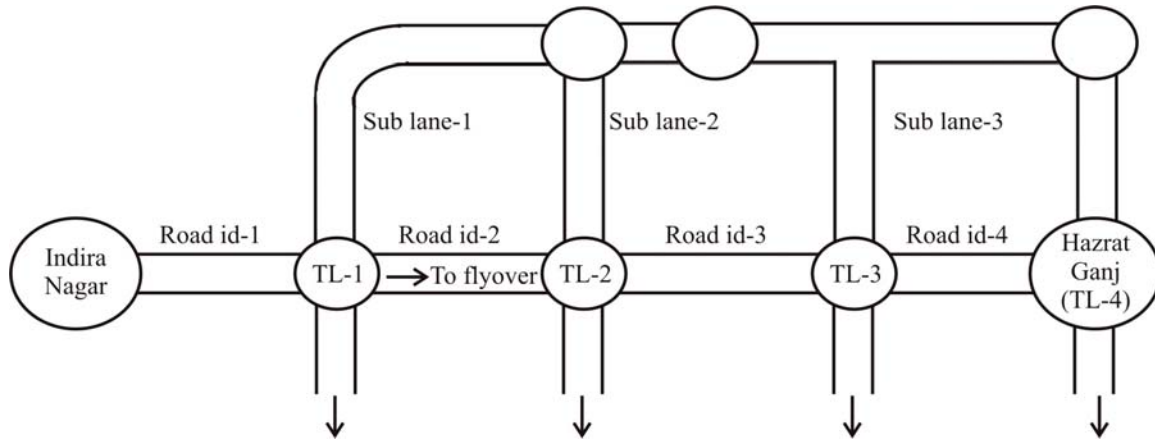


Fig. 1 . : Information Model components

2.0 Building Graphical Road Model :

In order to analyze the information flow model we have taken the graphical models of a section of road network. We will take the case of road Model with two points Indira Nagar (Residential Area) to Hazratganj (Commercial Area). of Lucknow City (India).

It consists of mesh road network with nodes as junctions .



TL- Traffic Light

Fig 2. Graphical Road Model.

From the above graphical road model[2,4] we have developed Object diagram of the entire grid

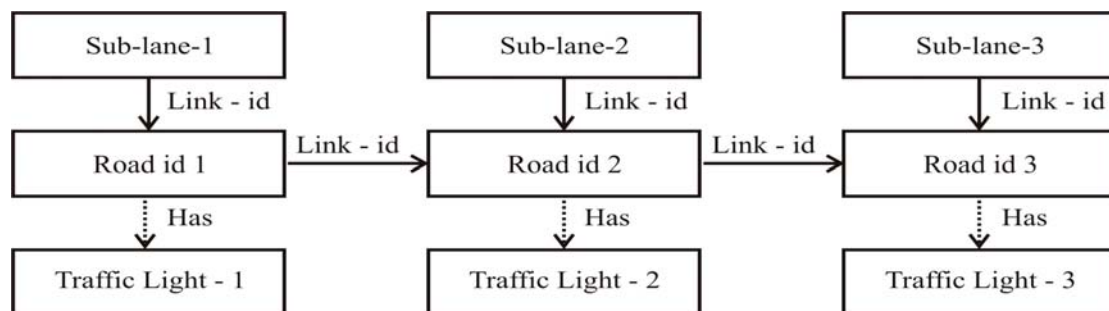


Fig. 3 : Object Diagram for Urban Traffic System

These diagram describes the system structure and dependencies on Decision sub-system based on Physical sub-system. The main focus is to show how the system unit coordinate with each other. For example let us consider the case of an accident between TL-1 and TL-2 in the stretch of Road-id2.

Based on the physical subsystem a decision sub system has to take necessary action accordingly. The meta data provided by single UML[3,5] diagram are many and above all refer to different part of code since main elements of a sequence diagram are objects and messages from them it is possible to extract information regarding the instances of a class. We adopt XML Meta data interchange XMI[1,3] for design information. XMI provides a text based support of UML diagram. We will show the coupling between physical subsystem and decision subsystem[5] and later on represent it with XML tags. Fig. 4 shows the coupling between the two systems how during the case of traffic jam in one section i.e. TL-1 to TL-2 Traffic is being diverted by information Flow center.

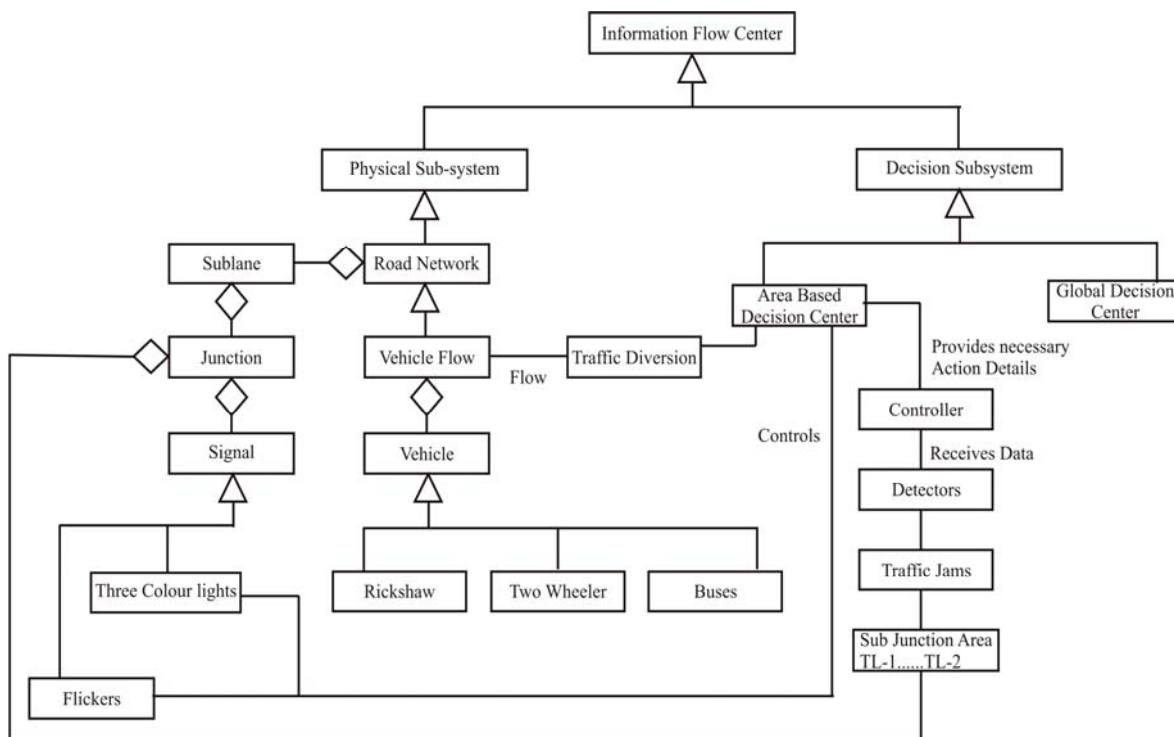


Fig 4. Association Diagram of Physical and Decision Subsystem

The flow of information is linked by the help of area based decision center, which receives the necessary information from the detectors. Role of physical subsystem is to accept the information from the decision subsystem and divert the traffic to various sub lanes linked with TL-1 and TL-2. The entire information flow model can be simulated by the help of software modeling as discussed in section 3.

3.0 Software Modeling :

The main focus of this section is to develop and simulate the UML model and store it into XML format. We have used the concept of Meta Data derived from UML diagrams and have used VB.Net [6] as a simulation language.

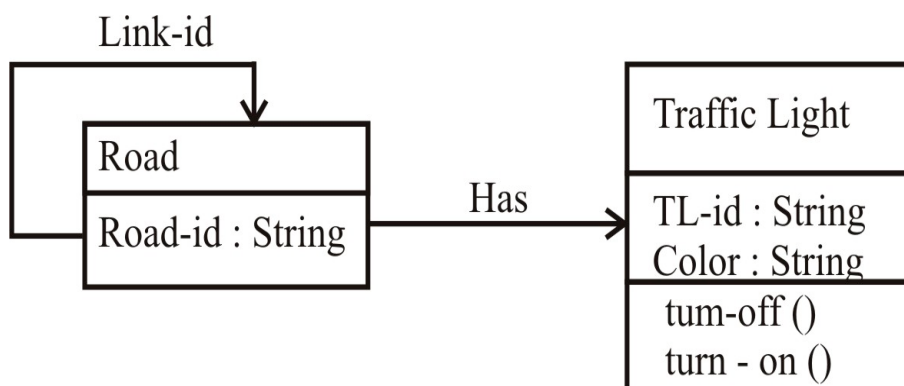


Fig. 5 : Sample Class Diagram

To start with the development of the form we can create a class road having the following buttons :

1. Txt RoadName
2. Txt RoadId

3. Txt TotalConnecting Lane

4. Txt Traffic Light_number

5. Txt Traffic Light_timings

We will create a button named between save.

```
Public Class Road

    Inherits SerializableData

    ' members...

    Public RoadName As String

    Public RoadId As String

    Public TotalConnecting Line As String

    Public Traffic Light_number As String

    Public Traffic Light_timings As String

End Class
```

Now to have the data to be saved as an XML file we must have serializable Data Class For that we have to create two classes :

Road and RoadCollection. Road will be used for storing a single instance of a Road – id in the RoadCollection. Road Collection will store the entire listing of the Road Name.

The RoadCollection should look like.

```
<VisualStudioProject>
  <VisualBasic
    ProjectType = "Local"
    ProductVersion = "7.10.3077"
    SchemaVersion = "2.0"
    ProjectGuid = "{F6804154-049D-4FEA-8ACD-FA313C7E6F24}"
  >
  <Build>
    <Settings
      ApplicationIcon = ""
      AssemblyKeyContainerName = ""
      AssemblyName = "RoadCollection"
      AssemblyOriginatorKeyFile = ""
      AssemblyOriginatorKeyMode = "None"
      DefaultClientScript = "JScript"
      DefaultHTMLPageLayout = "Grid"
      DefaultTargetSchema = "IE50"
      DelaySign = "false"
      OutputType = "WinExe"
      OptionCompare = "Binary"
    >
  </Build>
</VisualStudioProject>
```

```
OptionExplicit = "On"
OptionStrict = "Off"
RootNamespace = "RoadCollection"
StartupObject = "RoadCollection.Form1"
>
<Reference
  Name = "System.Windows.Forms"
  AssemblyName = "System.Windows.Forms"
/>
<Reference
  Name = "System.XML"
  AssemblyName = "System.Xml"
/>
</References>
</Imports>
</Build>
<Files>
  <Include>
    <File
      RelPath = "Road.vb"
      SubType = "Code"
      BuildAction = "Compile"
    />
    <File
      RelPath = "RoadCollection.vb"
      SubType = "Code"
      BuildAction = "Compile"
    />
    <File
      RelPath = "AssemblyInfo.vb"
      SubType = "Code"
      BuildAction = "Compile"
    />
    <File
      RelPath = "Form1.vb"
      SubType = "Form"
      BuildAction = "Compile"
    />
    <File
      RelPath = "Form1.resx"
```

```
        DependentUpon = "Form1.vb"

        BuildAction = "EmbeddedResource"

    />

    <File

        RelPath = "SerializableData.vb"

        SubType = "Code"

        BuildAction = "Compile"

    />

</Include>

</Files>

</VisualBasic>

</VisualStudioProject>
```

Both these classes will be inherited from another class called serializable inter. This base class will contain the logic needed to save the name to disk and loading them back again. In XML parlance, the saving process is known as serialization and the loading process is known as deserialization.

4.0 Building Serializable Data :

Starting with construction of a new class called as serializable Data, the coding of the class will look like :

```
Imports System.IO

Imports System.Xml.Serialization

Public Class SerializableData

    ' Save - serialize the object to disk...

    Public Function Save(ByVal filename As String)

        ' make a temporary filename...

        Dim tempFilename As String

        tempFilename = filename & ".tmp"

        ' does the file exist?

        Dim tempFileInfo As New FileInfo(tempFilename)

        If tempFileInfo.Exists = True Then tempFileInfo.Delete()

        ' open the file...

        Dim stream As New FileStream(tempFilename, FileMode.Create)

        ' save the object...

        Save(stream)

        ' close the file...

        stream.Close()

        ' remove the existing data file and

        ' rename the temp file...

        tempFileInfo.CopyTo(filename, True)

        tempFileInfo.Delete()
```

```
End Function

' Save - actually perform the serialization...

Public Function Save(ByVal stream As Stream)

    ' create a serializer...

    Dim serializer As New XmlSerializer(Me.GetType)

    ' save the file...

    serializer.Serialize(stream, Me)

End Function

' Load - deserialize from disk...

Public Shared Function Load(ByVal filename As String, _
    ByVal newType As Type) As Object

    ' does the file exist?

    Dim fileInfo As New FileInfo(filename)

    If fileInfo.Exists = False Then

        ' create a blank version of the object and return that...

        Return System.Activator.CreateInstance(newType)

    End If

    ' open the file...

    Dim stream As New FileStream(filename, FileMode.Open)

    ' load the object from the stream...

    Dim newObject As Object = Load(stream, newType)

    ' close the stream...

    stream.Close()

    ' return the object...

    Return newObject

End Function

Public Shared Function Load(ByVal stream As Stream, _
    ByVal newType As Type) As Object

    ' create a serializer and load the object...

    Dim serializer As New XmlSerializer(newType)

    Dim newObject As Object = serializer.Deserialize(stream)

    ' return the new object...

    Return newObject

End Function

End Class
```

The main purpose of taking XML as the main annotation format is to convert entire UML diagrams into XML file format. It provides best means of storing application data and makes the task of data transportation and exchange earlier.

5.0 Conclusion and Future Scope :

This paper has described how object Oriented Technology (UML) can provide a new way of describing Urban information flow models. The .Net platform provide an excellent means of annotation of entire road network we have also purposely implemented entire simulation model is Visual Basic. Net as it removes the builder of writing complex code for user interface. It also allows the programmers to draw their own user interface, since UTS design extensively in values. Active X control Visual Basic. Net was probably the best choice of it.

In this paper we have shown a methodology in which the design information can be directly applied in writing the application code. We will extending our work for hypermedia extension of UTS model in which on commuter can also access it to find the latest condition of traffic in particular area and apply OCL (Object Constraint Language) developed in terms of Traffic notation for formal verification of our software model.

References :

- [1] Jacobson I, Booch. G, and J Rambaugh. VML Notation Guide, Rational Corporation,USA,1997.
- [2] L.Zhao and T. Foster, Driver Duty: A pattern for public transport systems , JOOP, 1997.
- [3] P.P Chen,The Entity-Relational model, towards a unified view of data, 9-136, 1976. OMG Press.
- [4] Yogshen Quan, Urban Traffic control, ChinaCommunication Press, 1989.
- [5] Shoufeng ma, Ynajan L, Guoguang, System Analysis for coordination modes between urban traffic control and vehicle route guidance, Journal of Management Sciences in China, 2003; 71-78.
- [6] Ullman, Professional VB.NET 2003 , SAMS Series , WROX Publication , USA.

Article received: 2007-08-19