Modeling Intelligent Transport System using OPAT: A Language To Transform Multi Agent System In Object Oriented Notation

Manuj Darbari, Bhaskar Karn, Hasan Ahmad

Abstract— This highlights Intelligent Transport system using a tool OPAT. It is a combination of three basic concepts OPEN, Activity Theory and Artificial Intelligence. OPAT tool provides a dynamic binding methodology linking various OPEN diagrams with Activity theory diagrams giving a clear picture of Urban Traffic Network. It also emphasize on using the planning concept of sequence of activities by using the concept of Artificial Intelligence. Lastly, it provides an XML documentation of outline AOM Model which is helpful in formalizing the Model.

***Index Terms*— OPEN (Object Process Environment Notation). AOM (Activity oriented Modeling).**

I. INTRODUCTION

OPEN is a process –focused, third generation full lifecycle object oriented (00) methodology. It is a third generation object- oriented software development methods, providing strong support for process modeling by using a tailorable, objectified lifecycle. One of the prime purposes of developing OPEN was to provide a useful and usable "standard" software development process framework. OPEN has many elements : Process modeling and management. These are three levels of Process in OPEN : the business finished "product lifecycle the software engineering process (SEP) and the modelling process. The modeling process assists with identifying how things change with time and what work products should be created and when.

In OPAT Tool we have tried to integrate the concept of Activity Theory with OPEN Notational language i.e. OML 1.0. OPAT stands for OPEN + Activity Theory Modeling Tool. It defines a generic metamodel to represent the abstract syntax and semantics of Agent Based Modeling. The tasks and method are then stored in a reporrting for further references. OPAT Tool recognizes four meta-models that describe the system views. OPAT proposes the use of combinational language consisting of OPEN Notations with AT. OPAT is an OPEN profile based on the activity theory (AT) framework which includes. The concepts to describe societies of actors that are both autonomous and intentional. The Activity Theory object Model makes the integration process independent of any given methodology. Considering AT as the foundation for this intermediate language comes from multi agent traffic systems and their collaborations.

Manuscript received December 20, 2007, Modeling intelligent transport system using OPAT.

M. D. is with the Babu Banarsi Das Institute of Technology and Management, phone: 09105222311551; e-mail:manujuma@ rediffmail.com.

B. K., is working as Coordinator Department of Information Science, B.I.T, Mesra, Ranchi.

H. A. is working as Sr. Design Engineer, Nokia India Limited, Bangalore.

II. MODEL DESCRIPTIONS

In order to completely describe a particular traffic related case the developer has the options of using the following models.

A. Class Model

In OPEN concept the class diagram is tear apart to get the two complementary icons : the types and the class implementation. An object is an instance of a class as well as conforming to its type (s). The main focus is to represent the class in CIRT format (Class Instance, Role and Type) as show in Figure 1.

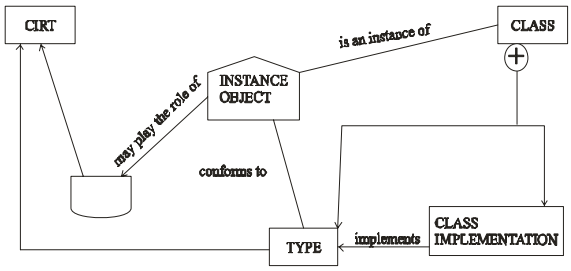

Figure 1 : Notation of Class with CIRT concept.

B. Collaboration Model

Collaboration diagram have been graph structure similar to semantic nets. They can be shown with or without message and are typically used to provide summary information particularly in terms of coupling. The major advantages is the ability to handle exceptions and the availability of logic boxes to handle branching, looping, critical regions and interlearning due to concurrency thereby greatly reducing the number of diagrams.

C. Aggregation Model

In OPEN terms it is also described as "Whole- part relationship". The aggregation modelling used in UML (Unified Modeling Language) uses Black and White diamond but in OPAT we have used the OPEN concept to remove this drawback for Agent Collaboration and Association. It is shown by unique nodel point connections with various objects located at in the form of boxes. They are connected via these nodal points.

D. *Activity Oriented Modeling (AOM)*

It is intended to combine all the agents which are used in Urban Traffic Modelling and are classified into Activity Theory terms.

At the element are assigned in terms of Urban Traffic Components and are then modelled. We have used the expert systems concept of "Projections" and "Planning". The plannar coordinates between "Subject", "Activity" and "Tools (terms of Activity Theory) and Projection predicts the" Outcome" (Term of Activity Theory).

III. IMPLEMENTATION WITH OPAT :

In order to illustrate the features of these modeling tool we will be describing each of these model in detail. Let us begin with our first model.

A. *Class Model*

In all the models we have taken Urban Traffic System as our example. The Snapshot shown below gives the picture of Urban Traffic Model. This modelling tool focuses on introduction of fully abstract classes into a design to increase the "plug in" compatibility of code components, focussing sense by composition. Another important thrust in this model is to maintain a fully connected class hierarchy which makes navigating hierarchy maintenance, further inheritance and instantiation easier.

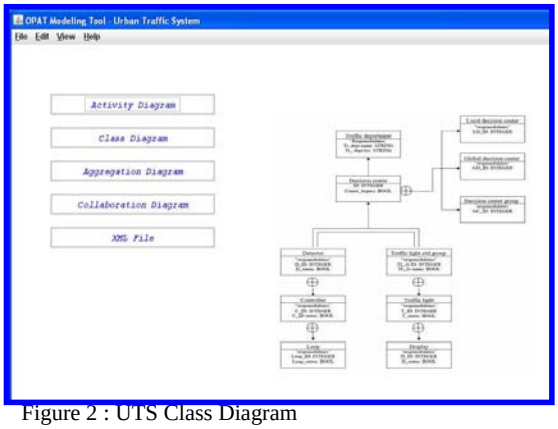

Figure 2 : UTS Class Diagram

What we have done is that, we have identified just the primary methods that are implemented different in each descendant class, and construct deference's classes retrospectively, based on grouping the shared secondary methods that use the small set of primary methods.

In the Fig. 2, we have shown "Traffic Department" as main class and "Decision centre" as the sub class derived from it. Some of the responsibilities are transferred from Traffic Department to the Decision Center and then they are transferred or inherited by the sub classes also. The basic job of the decision center is to get the information from Detector hierarchy and give the command for – traffic light control accordingly. The responsibility are assigned at each level which the particular class or sub class has to follow at the given time frame.

The factor of "TIME" can also be shown by OPAT model in the class diagram in the form of "VERSION". VERSION specifies the object retention requirements that should be modelled to provide the required functionality. If we redefine decision center than any previous information is not lost, but instead when they are next accessed an updating method is invoked which recreates the object according the old information retained and the new class definition with version (v.2) is specified.

A particular instance of Traffic light changes with time giving different versions of the generic object.

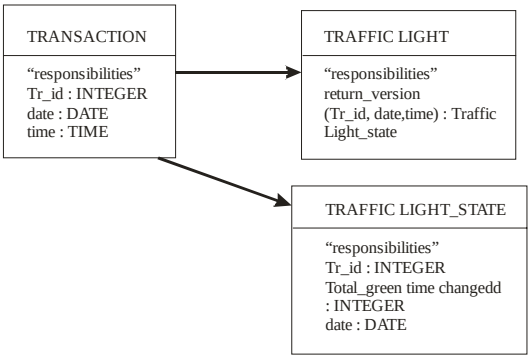

Figure 3 : An object Model of Version

In Figure 3 due to sudden rise in traffic volume there is a decision taken by the decision center to change the green light timings. The Traffic Light (v.2) shows the new state of Traffic Light- State object. Similarly we can define it for all the agents.

B. *Collaboration Diagram*

In OPAT tool, the collaboration diagram is able to show the exception handling and message passing capability. They generally represent the summary information in case of coupling. We can show the class diagram in the form of collaboration. Here the start of the diagram is shown by "Loop" and its responsibilities and interaction with other objects are written at the junction which describes the responsibilities and its affect on other objects. Fig. 4 shows the OPAT framework of collaboration diagram.

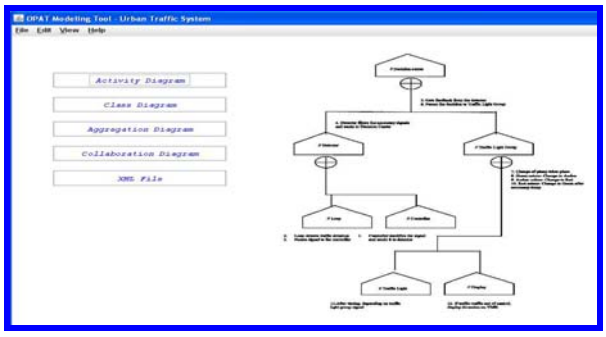

Figure 5 : Aggregation Model

C. *Activity Oriented Modeling (AOM)*

This model is being derived from Ruben and Pavon on Agent Oriented development. It is divided into the phases, first framework is same as UML-AT Model where we shown Traffic Department as community and Decision center as

subject and rest of the objects according to the A.T. format as shown in Figure 6.

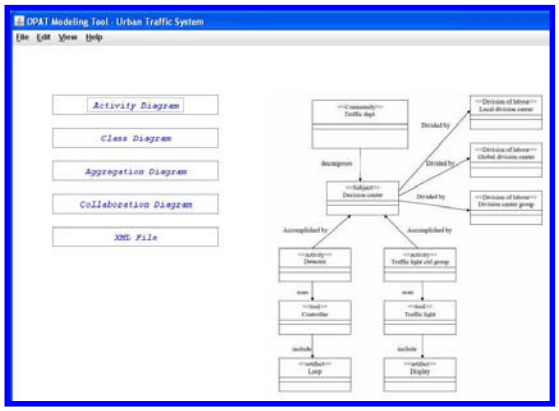

Figure 6 : Activity Oriented Model (AOM)

In the second phase we have integrated our software tool with USP? to derive a planning and projection to predict what the Decision Center will do in specific situations. The common algorithm for them will be

```
( plan
objects  :  name1 formula 1
           name2 formula 2
           ……..
steps  :  name1 act 1
           name2 act 2
           ……..
order  :  (Seq name mame )
           (overlap name name)
protect  :
           State1 init 1 fin 1
           State2 init 2 fin2
        ……)
```

There is an option in OPAT Tool which can state the plan depending on the <<subject>>, << Activity>> and <<tool>>. For example consider detector which is an activity which does the framing for detection of traffic jam through the <<tool>> controller and loop. The procedure is

```
( plan- for :  (detect _traffic?loop?controller)
    objects   : Aritifact (list artifect loop)
      …….

      Steps : detect traffic
              (repeat until detect traffic flow?sensor)
            (sense?Trafficflow? Nil)
```

The entire coding and behavioural planning of various agents and categories of activity theory classification is shown in side windowpane. The planning decision and coordination among various Agents can be achieved and programmed by clicking on each of the components one by one the software will automatically coordinate their working by taking help from OPEN class and collaboration Model.

In this window we can also program for protection violation which records the interval for which that should be protected. In this we have also linked it to advance version of the class diagram.

In the third stage we have introduced a small window in which the entire activity oriented model is being converted into XML format as shown in the fig. 7.

Figure 7 :. XMF Generation Text Box

The main feature is that it automatically detects the root element by the help of behavioral planning being done in the previous stage. We have used XMF (Executable meta-modeling Language) which exploits the features like :

- The derivation of an abstract syntax model
- Description of the semantics.
- A presentation of the profits concrete syntax.

Using this we can generate the entire model into XMI format.

This helps in creation of skeleton for designing a web application and later on they are embedded with Java, to provide a formal verification of the entire model which is not yet done in Activity oriented UML-AT Modeling framework.

IV. CONCLUSIONS & FUTURE SCOPE

This paper presents an extension of UML-AT concept by way of integrating it with OPEN concept. OPEN provides a much better and precise Agent oriented Modelling framework than UML. The Class Diagrams and collaboration diagrams provides a new concept of versions which tells about the instance of the particular class with time. This extra feature is very much required when we one dealing with modelling real time applications like Urban Traffic Modelling.

Secondly the migration of responsibilities are very well described by the aggregation diagram which is unique feature in OPEN as compared to UML notation.

Lastly, we have developed AOM which are partially extension of UML-AT concept with additional feature of linking with aggregation, Class and collaboration diagrams. It is also supported by the concept of planner system in which it is integrated with LISP for necessary planning of sequence of Activities. AOM also highlights the conversion of AOM models into XML format which can be formalised by embedding it with java platform various new concepts can be introduced in the tool by adopting its metamodel specification. The future scope could be extending the XML to XMF for more realistic and effective modeling.

ACKNOWLEDGMENT

We are grateful to Lucknow Traffic police department for disclosing their Hierarchical structure.

REFERENCES

[1] Henderson-sellers, B, , "*OPEN Toolbox of Techniques*", ACM Press, 2004, USA.
[2] Handerson-Sellers, B, "Agent-Oriented Methodologies, London, UK, Idea Group publishing 2005.
[3] Bratman , M.E. "Intentions, Plans and Practical Reason, Cambridge, MA : Harrard University Press, 1987.
[4] N.J. Nelsson, "Artificial Intelligence – A new synthesis", Harcourt Asia Ltd, 1998.