Modeling Methodology for Component Reuse and System Integration for Hurricane Loss Projection Application

Kasturi Chatterjee[1], Khalid Saleem[1], Na Zhao[1], Min Chen[1], Shu-Ching Chen[1], Shahid S. Hamid[2]
[1]Distributed Multimedia Information System Laboratory
School of Computing and Information Sciences
Florida International University, Miami, FL 33199, USA
[2]Department of Finance
Florida International University, Miami, FL 33199, USA
[1]{kchat001, ksale002, nzhao002, mchen005, chens}@cs.fiu.edu, [2]hamids@fiu.edu

Abstract

Hurricanes are one of the deadliest and perilous natural calamities on the face of earth having a severe impact both on the lives of the people and economy of a nation. Attempts have been made to mitigate hurricane aftermath, by utilizing research and tools that can analyze hurricanes and estimate projected losses. The need for such research methodologies and tools stimulated the development of a multi-disciplinary cutting edge public hurricane model called Public Hurricane Risk and Insured Loss Projection Model (PHRLM). The complex and diverse nature of the application raises the need for module abstraction, seamless integration and effective reusability to create a uniform generic environment. Providing efficient interaction between these complex multi-disciplinary modules using different abstractions, formalisms, data formats and communications and making each module transparent enough to be reusable becomes a complicated task. This paper presents a UML based formal Modeling Methodology, enabling component reuse and integration of the application.

1. Introduction

Hurricanes pose one of the greatest environmental threats to human beings and economy of the nations. For the past few years, hurricanes have been more frequent in U.S due to increased warming of water in the Atlantic Ocean causing loss of human lives and destruction of properties in the coastal regions. Over the period of time, different commercial modeling companies have developed risk and assessment models to assist insurance companies in formulating their rating policies. These models lack transparency in describing the assumptions and rationales used to estimate the losses and leave the general public and regulators at the whims of the companies, to accept the scenarios as presented without being able to judge their credentials and accuracy. The public systems like HAZUS [6], though have the transparency of information lack portability and easy maintenance. Due to their PC-based implementation, new hardware needs to be deployed with the increase of users and installations need to be repeated. Moreover, the different modules of the application execute rather independently making information exchange, reuse and efficient resource management a hurdle. In order to address all the above stated problems, a multi-disciplinary public model was developed called Public Hurricane Risk and insured Loss projection Model (PHRLM) [9] by the State of Florida, funded by Florida Office of Insurance Regulation. The model has a component-based approach where each sub-application corresponding to a different genre has been modularized. The model was developed by an interdisciplinary team of experts belonging to meteorology, statistics, finance, actuarial science, engineering and computer science disciplines. It is an open model facilitating ease of maintainability and portability. The complexity and diversity of the system makes system integration a challenge. Different modules carrying domain specific tasks need to seamlessly interact with one another which necessitate semantic maintenance and data integrity as well as the presence of efficient interfaces enabling abstraction. Moreover, to enhance the utility of the application, the individual components should be reusable by different applications with minimum system customization.

The diverse and complicated structure of the PHRLM application necessitates the introduction of a modeling technique to achieve the desired abstraction required both for efficient integration as well as independent component reuse. A model is an abstract representation of a system that enables us to answer questions about the system [2]. Modeling is a way to deal with complexity by ignoring irrelevant details. Efficient system integration and reuse requires an object-oriented and component based approach,

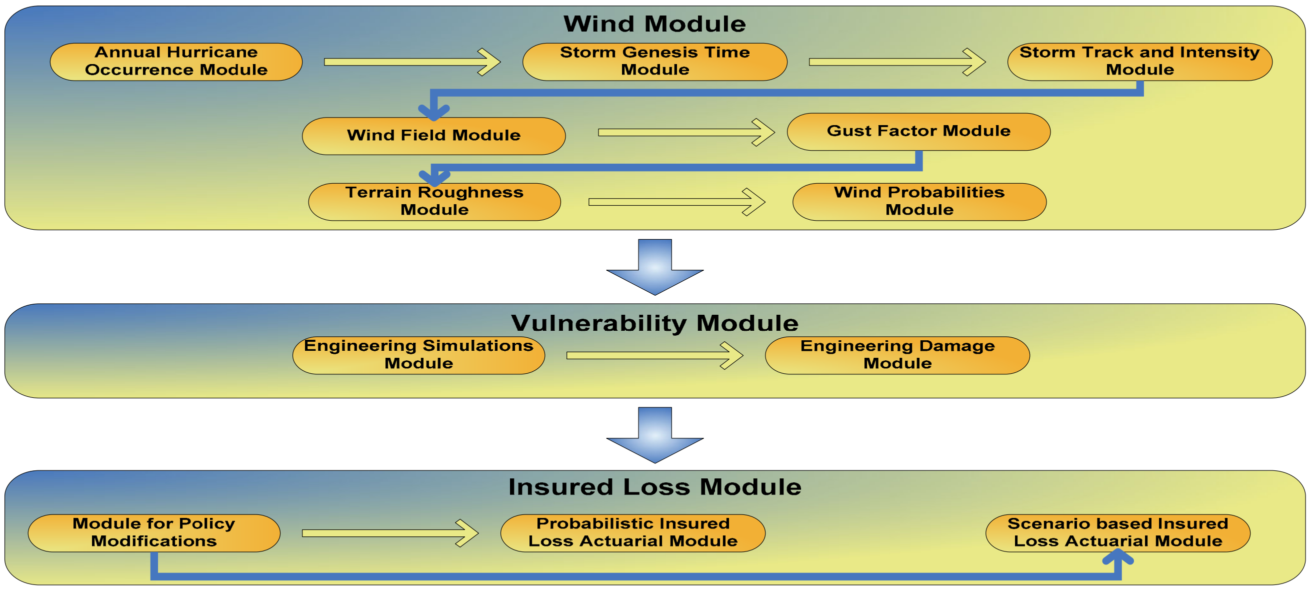
Wind Module
Annual Hurricane Occurrence Module
Storm Genesis Time Module
Storm Track and Intensity Module
Wind Field Module
Gust Factor Module
Terrain Roughness Module
Wind Probabilities Module
Vulnerability Module
Engineering Simulations Module
Engineering Damage Module
Insured Loss Module
Module for Policy Modifications
Probabilistic Insured Loss Actuarial Module
Scenario based Insured Loss Actuarial Module

Figure 1. System Architecture of PHRLM

as the main essence of object oriented paradigm is reuse and encapsulation. Hence, to model the application, UML was found to be the appropriate language as the main goal of UML is to provide a standard notation that can be used by all object-oriented methods and to select and integrate the best elements of precursor notations [2].

Model driven reuse and integration differs from integration and reuse by programming. Programming approach concentrates on a particular scenario and inextensible solution to a specific challenge. Model-driven integration focuses on abstracting the information content into a model that describes the application's information resources and thus makes it easy to be applicable to any system within the same genre.

This paper proposes a UML based modeling technique to enable and enhance efficient system integration and component reuse for the PHRLM application and serves as a prototype to solve similar problems for multi-disciplinary large-scale research applications.

The rest of the paper is organized as follows. Section 2 presents a brief overview and System Architecture of the PHRLM Project, Section 3 discusses the Modeling methodologies undertaken to successfully formalize the application with a focus to enable component reuse and system integration modeling techniques followed by Section 4 which presents the conclusion.

2. Overview of PHRLM

The PHRLM model is a probabilistic model designed to estimate damage and insured losses due to the occurrence of hurricanes in the Atlantic Basin. The PHRLM estimates the full probabilistic distribution of damage and loss for any significant storm event. The modeling methodology of PHRLM can be partitioned into three major components: (a) Wind Module, (b) Vulnerability Module and (c) Insured Loss Module. The major components are developed independently before integration. The Wind Hazard Module is a meteorological module which deals with estimating the number of hurricanes and storm genesis time, generating storm track, open terrain wind speeds and ultimately the wind probability for each zip code. The next module is the Vulnerability Module which is an engineering module which simulates the wind damages, generates the damage matrices and the vulnerability functions for mitigated structures. The final module is the Insured Loss Module which models the wind deductibles, generates the annual loss and the expected loss cost for a specific hurricane. The model can estimate losses in a particular zip code and can also simulate a potential storm. Each module can be considered as a business object component which provides a service-oriented interface to external systems and allows interoperability between systems.

2.1. System Architecture

An appropriate system architecture provides flexible yet robust infrastructures to build, extend and maintain any application. The PHRLM system uses a modular component-based architecture. PHRLM being a multi-disciplinary project, different modules relate to one or more domains. Due to the research nature of this application, each module is subject to constant revision and update. Thus, the modules are made as loosely coupled as possible to meet the flexibility and adaptability requirements. In Figure 1, the system architecture of the application is illustrated.

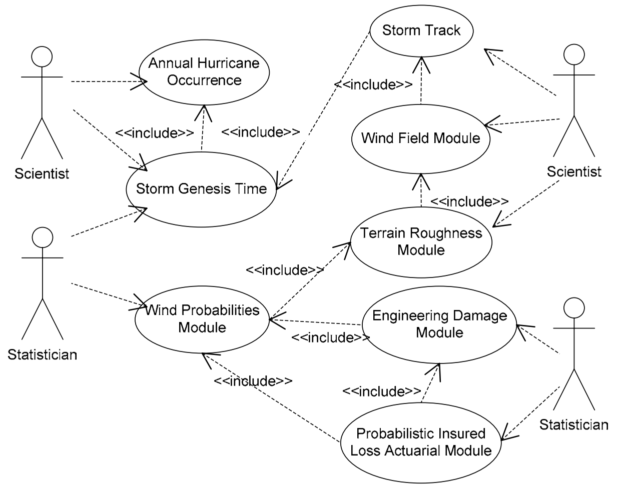

Figure 2. Use Case Diagram of PHRLM

3. Modeling Methodology

The multi-disciplinary nature of the project makes integration and component reuse a challenging job as it increases the complexity of the system manifold. Thus, arose the need of modeling to provide the necessary abstraction required to achieve seamless integration and reuse efficiently. The modular nature of the system and the reusability requirement makes object-oriented modeling apt for the job.

3.1. Enabling Component Reuse Using UML

With the increase of complexity and size of applications, development costs have increased manifold. Thus, efficient reuse of existing technologies has become very important. Object-oriented paradigm enables easy reuse of components. An object-oriented framework is defined as a set of classes that embodies an abstract design for solutions to a family of related problems [2].

Since, our application is a public-benefiting open model; one of the main requirements was reuse by other similar applications. As explained earlier, there are several modules in the system performing specific tasks. For example, the Vulnerability Module calculates the damage ratios with the supplied values of vulnerability statistics and wind speed. The same module has the potential to be extended for being reused in calculating the damage caused by other kind of wind based natural calamity like Tornados etc. by appropriate control of the input parameters and customization. Moreover, currently the model can estimate hurricane losses for only residential structures. However, the model is highly extensible and allows for the addition of components to estimate losses for commercial structures and high rise buildings. The model can also be used as a reference to develop similar models for other vulnerable coastal areas and can be extended to develop general Disaster Control and Management models.

For effective reuse several criteria should be satisfied by each module as well as by the entire system. The module should have a proper abstraction to make the complex functionalities transparent to the user, it should be portable and should be able to exist and co-operate independently outside the existing framework. Efficient component reuse also needs the entire system to be loosely coupled so that the dependence of each module with one another is as minimal as possible. To fulfill all the above requirements, a modeling technique is required. UML provides the required abstraction and a complete standardized description of the system. Each UML diagram is designed to let developers and customers view a software system from a different perspective and varying degree of abstraction. UML diagrams commonly include Use Case Diagrams, Class Diagrams, Interaction Diagrams, State Diagrams, Activity Diagrams and physical Diagrams.

Major reuse scenarios can be modeled by three generalization relationships supported by Use Case Diagrams [5] viz. <extend>, <include> and <inheritance>. An extending use case is an alternate course of the base use case. An include dependency is a generalization relationship denoting the inclusion of the behavior described by another use case. The third way of reuse is inheritance of use cases where one use case is inherited from other use cases and the inheriting use case would completely replace one or more of the courses of action of the inherited use case. The introduction of these generalization techniques allows reuse both within the application as well as by other external applications.

Based on the component reuse frameworks proposed in [7, 8], two major characteristics need to be considered: class generality assessment and relations among classes. As proposed in [7, 8], a class may be marked as *general* or *specific*. A *general* class is expected to be reused but a *specific* class is intended for a particular application. This concept is extended and used in modeling of our system. Since every component is aimed to be reusable, hence classes belonging to different modules should be a *general* class. However within a particular module, the class can be *specific*. Figure 2 denotes the Use Case diagram of the entire system which depicts the implementation of <include> relationship among different modules. To maintain the generality of the classes and to make each module independent, <extend> or <inheritance> relationship is not used within the system. Thus, the individual modules can be easily extended or inherited in other applications without any constraint of inter-modular dependability. Figure 3 explains one such scenario where the Vulnerability Module could be extended to different types of application specific estimation scenarios like the

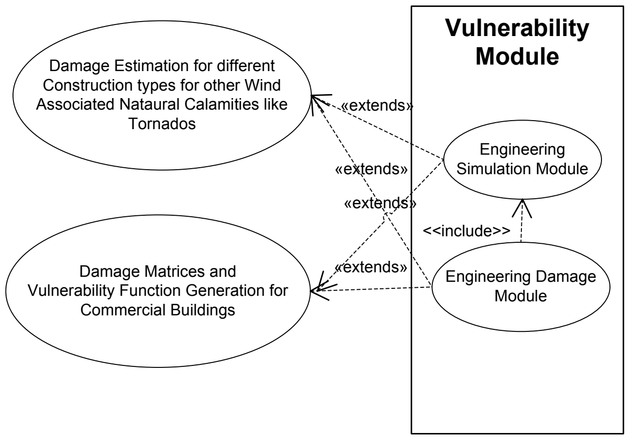
Vulnerability Module
Damage Estimation for different Construction types for other Wind Associated Nataural Calamities like Tornados
«extends»
Engineering Simulation Module
«extends»
«extends»
<<include>>
Damage Matrices and Vulnerability Function Generation for Commercial Buildings
«extends»
Engineering Damage Module

Figure 3. Component Reuse Modeling

Damage Estimation for other Wind Associated Natural Calamity and the generation of Damage Matrices and Vulnerability Function for Commercial buildings. The modeling technique applied to PHRLM aiming at component reuse can be formalized as:

Definition 1:
If, there is a loosely coupled system with self sufficient interacting modules, each module will be modeled using a separate Use Case.

Definition 2:
Suppose there exist two Use Cases UC_A and UC_B for modules A and B respectively. The only generalization relation that can exist between A and B is UC_A <include> UC_B due to the loosely coupled nature of the system. However, in order to reuse a generic Use Case UC_A, it should have a generalization relationship of <extend> with any customized external Use Case.

Once the Use Cases have been defined, the class diagrams are required to model classes. Class Diagrams describe the structure of the system in terms of classes and objects. The reuse potential of class depends on the extent to which class to class relationship or coupling is defined [7, 8]. Classes are related to other classes if they expect to use and reuse other classes in present implementations and future applications [5]. The class to class relationship is very useful to plug in a component from one application to another. One of the ways to design the class relationship is by formulating the use case to a class relationship, i.e., by keeping track of the classes included in each use case and then using the use case relationship to determine if the classes are related. Figure 4 denotes the Class Diagram of a component of the PHRLM model which explains the class relationships and hierarchy in detail. For example, the class *fitDistriBean* is related to *SimuSelection.* Hence in order to reuse *fitDistriBean* in some other application, the class *SimuSelection* needs to be present in that particular application too. Thus, if a component of an application was to be reused, the detailed class relationship gives the flexibility of functional application-nature dependent modification and customization. The rationale used behind modeling the classes of PHRLM to help efficient component reuse is as follows:

Definition 3:
Within the classes of the same Use Case, <extend> relationship can exist but not between classes of different Use Cases.

The three definitions described above form the basic rationale by which efficient component reuse was modeled for a diverse loosely coupled system where modules communicate with each other using Data Flow. The approach provides a modeling prototype for any Disaster Management Application consisting of several components performing different domain specific jobs. Thus, with proper UML modeling using Use Case Diagrams and Class Diagrams, the different components of the PHRLM system can be customized and plugged in by other applications efficiently.

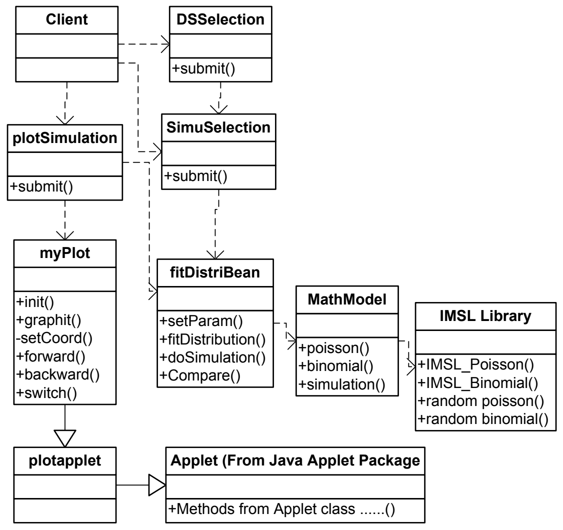

Figure 4. Modeling of Class level Coupling of a PHRLM Module

3.2. Enabling Component Integration Using UML

The multi-disciplinary nature of PHRLM makes system integration an indispensable but a very complex task. Maintaining semantic uniformity of the data as well as their seamless communication among the different modules becomes a challenging job. Due to the diverse nature of the project, the integration of PHRLM could not be achieved by following any one of the popular integrating techniques like File Transfer, Shared Database, Remote

Procedure Invocation or Messaging in their absolute forms, instead it required a combination of these techniques and customization at some places to suit the requirements of the project. From the implementation point of view, semantic integration is achieved by selecting the key terms across the entire application and making them uniform. Due to domain specific requirements, these key terms have different local meanings which were resolved using a mapping technique that involved the mapping of local terms with the global terms. Asynchronous messaging technique is used to integrate the loosely coupled components of the system by overcoming the limitations of latency and unreliability. To achieve the above technique, each module is developed into a web-based application and they are made to communicate through Event Messaging and Document Messaging over a secured channel. For data that is too voluminous or too sensitive to be transmitted over communication channel, a centralized database is used for storage, maintenance and efficient access of sensitive data.

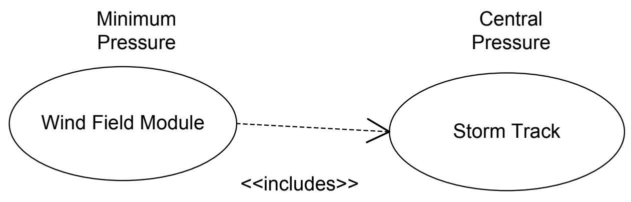

Figure 5. Semantic Integration Modeling for Global Concept "Minimum Pressure"

The above integration methodologies and planning for PHRLM explicitly depicts the use of several integration approaches which calls for an efficient modeling representation of the entire integration approach for formalization and efficient understanding. The integration methodologies adopted in PHRLM cannot be compared with any Enterprise Integration Architecture as it has an essential research approach which calls for constant revision and update of requirements as well as design thus hindering it from following the strict life-cycle of any enterprise integration architecture (e.g. CIMOSA [3] and ARIS [1]) which is composed of domain identification, concept design, requirement definitions, design specifications, implementation description, domain operation and decommission definition [10]. Here, we propose an integration modeling specification for PHRLM which can be extended for other Disaster Management and Recovery Systems. The modeling involves combination of the UML Data Modeling Technique [4] and Use Case and Class Diagrams to depict the integration plans utilized by PHRLM.

In our system, Semantic Integration is modeled using Use Case Diagrams with generalization relationship depicting the semantic uniformity. For each global term all the modules using it are related to one another by a <include> relationship to depict the flow. The method can be formalized with the following definition.

Definition 4:

If Module A, Module B and Module C use a global concept I with different local terms, then there exist a Use Case U_I for the global concept I which consists of the use cases for Module A, B and C and the local terms in each of them related to each other by a <include> relationship depicting the presence of the Global Term I in each one of them.

The above definition is explained in Figure 5 where the use of the global concept "*the minimum pressure of the hurricane"* in different modules is illustrated. In Wind Field Module, the term *Minimum Pressure* is synonymous to the term *Central Pressure* in Storm Track module, both depicting the concept of minimum pressure of a hurricane.

The integration through messaging is modeled using Sequence Diagrams for pairs of modules that communicate with one another. The sequence diagram for individual module also depicts the communication among the different sub-modules within the module. The proposed concept is conceptualized with the following definition.

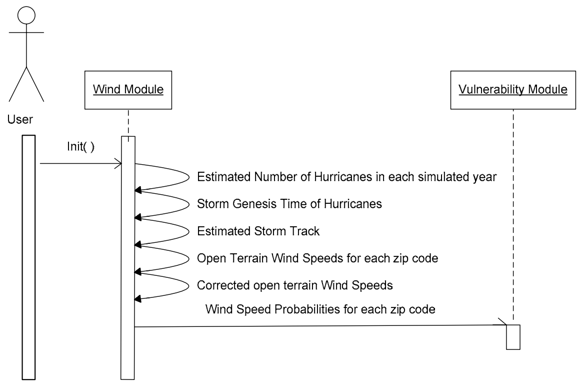
User
Wind Module
Vulnerability Module
Init( )
Estimated Number of Hurricanes in each simulated year
Storm Genesis Time of Hurricanes
Estimated Storm Track
Open Terrain Wind Speeds for each zip code
Corrected open terrain Wind Speeds
Wind Speed Probabilities for each zip code

Figure 6. Asynchronous Communication Modeling across Modules

Definition 5:

If Module A communicates with Module B, then there exists a Sequence Diagram S_{AB} which depicts the communication sequence among them.

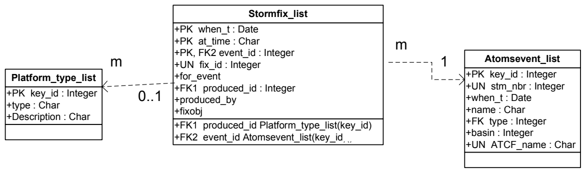

Figure 7. UML based Data Modeling

System Integration using Centralized Database is modeled using the UML Data Modeling Profile as introduced in [4]. Here, the concept of tables and relationships used in a database maps to the concept of classes and associations in UML. The database used in PHRLM has several components like Tables, Schemas, TableSpace, View, Columns, Key, Index and Constraints. Each of them is modeled using the UML Data Modeling Technique. Some of them are explained below:

1. *Tables: A table represents a set of records having a similar structure containing data. A table is represented in the UML diagram as* ***class diagrams****.*

2. *Schemas: A Schema is used to organize tables. In UML modeling, a schema is represented by a* ***package****.*

3. *Columns: Column is the field in the table where data is stored. Columns are modeled using* ***Attribute*** *representation in UML.*

4. *Constraints: Constraint is a rule imposed on the columns or on rows of a database. They are modeled through* ***stereotyped operations*** *as well.*

The data modeling techniques discussed are illustrated in Figure 7. The detailed modeling of the database using easy to understand UML diagrams helps in integration and use of the centralized database by all the participating modules. Such detailed representation helps in easy communication and adaptation of any change in the centralized database. This leads to a uniform and detailed understanding of the database schema by all the participating modules, thus facilitating the integration process without suffering from inconsistency and ambiguous interpretation.

Thus, we propose a unique modeling technique for integration of complex, multi-disciplined and loosely coupled systems by combining three types of modeling approaches applied to semantic integration, communication and centralized databases. The approach can be easily extended and used as a prototype for developing and integrating other similar applications.

4. Conclusion

This paper proposes a modeling methodology for component reuse and system integration for a diverse multidisciplinary application PHRLM, which analyzes hurricanes to predict their paths and occurrence and estimate projected losses. The proposed modeling methodology is unique and customized for a loosely coupled, multi-domain large scale research oriented application and can be used as a prototype to design Disaster Management and Recovery Systems for different kinds of natural calamities. It will also assist in solving imperative problems of system integration, extensibility and component reuse which are a critical concern for successful software engineering implementation.

5. Acknowledgement

This work is partially supported by Florida Office of Insurance Regulation (OIR) under "Hurricane Loss Projection Model." While the project is funded by the Florida Office of Insurance Regulation, the OIR is not responsible for this paper content.

References

[1] ARIS Reference, "*http://www.ids-scheer.com/international/english/products/aris_design_platform/50324".*

[2] B. Bruegge and A.H. Dutoit, "Object-oriented Software Engineering Using UML, Patterns, and Java," Second Edition, 2004.

[3] CIMOSA Reference, "*http://cimosa.cnt.pl".*

[4] Davor Gornik, "UML Data Modeling Profile", White Paper, Rational Software. May 2002.

[5] D. Needham, R. Caballero, S. Demurjian, F, Eickhoff, J. Mehta and Y. Zhang, "A Reuse Definition, Assessment, and Analysis Framework for UML", *Book Chapter in Advances in UML and XML –Based Software Evolution,* 2005.

[6] Hazus manuals page, "*http://www.fema.gov/hazus/li_manuals.shtm".*

[7] M. Price and S.A. Demurjian, "Analyzing and Measuring Reusability in Object-Oriented Design," *Proceedings of OOPSLA'97, 1997.*

[8] M. Price, S. Demurjian and D. Needham, "Reusability Measurement Framework and tool for Ada95," *Proceedings of 1997 TriAda Conf.,* Nov.1997.

[9] PHRLM manual, "*http://www.cis.fiu.edu/hurricaneloss*".

[10] Y. Zhou, Y. Chen and H. Lu, "UML-based Systems Integration Modeling Technique for the Design and Development of Intelligent Transportation Management System," *Proceedings of IEEE International Conference on Systems, Man and Cybernatics,* 2004.