Florida Public Hurricane Loss Model (FPHLM): Research Experience in System Integration

[1]Shu-Ching Chen, [2]Min Chen, [1]Na Zhao, [3]Shahid Hamid, [1]Khalid Saleem, [1]Kasturi Chatterjee

[1]Distributed Multimedia Information System Laboratory
School of Computing and Information Sciences
Florida International University, Miami, FL 33199, USA
{chens, nzhao002, ksale002, kchat001}@cis.fiu.edu

[2]Department of Computer Science
University of Montana, Missoula, MT 59812, USA
chen@cs.umt.edu

[3]Department of Finance
Florida International University, Miami, FL 33199, USA
hamids@fiu.edu

ABSTRACT

The Florida Public Hurricane Loss Model (FPHLM) developed in the State of Florida offers an open, public and effective tool for the government to regulate the insurance ratemaking process. It is also open to public scrutiny and provides an understandable baseline to check the windstorm insurance rate. Currently, the model has finished its development on probabilistic assessment of insured hurricane wind risk to residential properties and has successfully passed the audit of Florida Commission on Hurricane Loss Projection Methodology. As a multi-disciplinary large scale project, its system development and integration faced numerous challenges varying from technical factors to project management aspects. In addition, different from the general practice of software system development, FPHLM is essentially a research oriented project where the algorithms are constantly revised, validated and improved along with the research progress, which inevitably poses additional challenges upon the system development and integration cycle. With the help of systematic project management and advanced software development theories, techniques and tools, the FPHLM team overcame these challenges while maintaining delivery schedules, meeting budget constraints, and offering good software development practices. This paper addresses the challenges and experiences associated with the integration process of the FPHLM software system, which we believe will benefit the related digital-government applications.

Categories and Subject Descriptors

D.2.m [Software Engineering]: Miscellaneous.

General Terms

Management, Documentation, Design.

Keywords

System integration, digital government, Florida public hurricane loss model.

1. INTRODUCTION

Hurricanes are one of the most devastating and costly natural perils in U.S. For nearly a decade, commercial modeling companies have developed several hurricane risk models [8] [11] to assist the insurance industry in the ratemaking process. However, as the parameters that affect storm damage forecasts have been kept under wraps, the public and regulators were left to accept the scenarios presented by the insurers, without being able to closely examine the assumptions they made. Some rates, especially for windstorm insurance, have doubled or tripled in many vulnerable states and the affordability and availability of insurance and reasonableness of the rates have become important economic and social issues. On the other hand, the existing public systems, such as HAZUS [9] developed by Federal Emergency Manage Agency, possess insufficient system portability, have limited accessibility as stand-alone systems and do not address insurance issues.

To improve the regulation of insurance ratemaking process and offer better public accessibility, a multi-disciplinary cutting edge public model, called Florida Public Hurricane Loss Model (FPHLM) [2], has been developed in the State of Florida, funded by the Florida Office of Insurance Regulation. This model is a web-based modeling and simulation system; designed and developed by an interdisciplinary team of researchers and scientists from meteorology, statistics, financial, actuarial, wind engineering, and computational sciences disciplines. It is open to public scrutiny and provides a benchmark to compare and validate against the insurance losses and risks produced by more secretive industry models. The model design and development for insured hurricane wind risk to residential properties were completed successfully this year. Additionally, Florida Commission on Hurricane Loss Projection Methodology have recently conducted and approved a rigorous audit of our model. Currently, we are working to extend the model for handling commercial buildings. The main focus of this paper is to address the challenges and experiences associated with the integration process of the software system for our model which can assist in the design, development and integration of components for future large scale inter-disciplinary digital governance projects.

System integration remains an open and challenging issue especially for a complex digital government project like FPHLM. In the literature, numerous research studies have been presented to address this issue via theoretical analyses and/or case studies. In terms of theoretical analysis, system integration was studied from different perspectives. For instance, Nilsson et al. [4] addressed system integration in four aspects. In [5], web services are studied as a potential solution to the application integration problem. As far as the real application is concerned, architectural reasoning approach is adopted in [3] to integrate three software systems. In [6], two case studies are presented to illustrate the idea of enterprise application integration from the Enterprise Component/CAS (Complex Adaptive Systems) perspective. In this paper, we present our efforts and experience in integrating the FPHLM system by introducing a novel integration approach which intelligently combines and customizes the existing techniques to suit the specific requirements of the project.

2. OVERVIEW OF FPHLM PROJECT

FPHLM started by statistically analyzing the historical hurricane records in HURDAT data set [10] and simulating tens of thousands of hurricanes in a series of years over the ocean. Then, we forecast possible tracks for each storm and how each would decay once it hits land. We also evaluated the effect of three-second gust wind over various types of terrain, which appear to cause most of the damage. Thereafter, the meteorological data were combined with extensive information on the constructions of homes and apartments and their vulnerabilities to windstorm damage. To this extent, the model can forecast losses in a particular zip code or a portfolio of properties using data of insured property coverage and past homeowner insurance claims. It can also simulate a potential storm season or calculate losses caused by a specific hurricane, such as hurricane Katrina or Rita.

As a multi-disciplinary large scale project, FPHLM faced numerous challenges varying from technical factors to project management aspects in system development and integration. In addition, unlike the general practice of software system development, FPHLM is essentially a research oriented project where the algorithms are constantly revised, validated and improved along with the research progress, which inevitably poses additional challenges upon the system development and integration cycle. Therefore, to ensure the successful completion of the project, we adopted the idea of spiral model [1], an iterative activity-centered model to manage the project development. Meanwhile, we meticulously analyzed the specific characteristics and unique requirements of FPHLM project as there is always a discrepancy between concepts and reality.

The spiral model presented in [1] focuses on addressing risks incrementally and can be divided into the following rounds: Concept of Operation, Software Requirements, Software Product Design, Detailed Design, Code, Unit Test, Integration and Test, Acceptance Test, and Implementation. Here, each round is composed of four phases. In brief, the first phase aims at exploring alternatives, defining constraints, and identifying objectives and the second phase manages risks associated with the solutions defined during the first phase. Then during the third phase, developers realize and validate a prototype or the part of the system associated with the risks addressed in this round and finally the fourth phase focuses on planning the next round based on the results of the current round. As the focus of this paper is upon the FPHLM system integration process, in the next section, we will briefly discuss the issues considered in the system design and development phase which are critical for the system integration.

3. SYSTEM DESIGN

Generally, a large portion of the development cost of a complex information system is attributable to the decisions made in the upstream portion of the software development process; namely requirements specification and design which refers to the first four rounds in the spiral model. As for the FPHLM project, this stage is of essential importance due to the following two reasons.

- As a multi-disciplinary project, knowledge from multiple technical and functional domains is a necessity for FPHLM. However, individual team members do not have all of the knowledge required for the project and must acquire additional information before accomplishing productive work.
- As a research oriented project, the algorithms are constantly revised, validated and improved along with the research progress. Such evolution of system in real time requires an effective integration plan which manages the system complexity as well as its flexibility and adaptability.

Correspondingly, we identified three main issues as detailed below which were essential and critical for the success of the system integration.

3.1 Knowledge Acquisition, Sharing and Integration

Generally speaking, knowledge acquisition, sharing, and integration are time-consuming yet significant activities that to some level determine the development, implementation and completion of a design plan [7]. Here, the source of such knowledge can be relevant documentation, formal training sessions, other team members and the results of trial-and-error behavior, etc. In FPHLM, the knowledge can be broadly classified into four general categories: 1) requirement clarification, where the requirements refer to both the external requests from the project shareholders and internal requests from the group members; 2) background introduction in both technical and application aspects; 3) design approach; and 4) system requirements. Various approaches were adopted for the group members to acquire and share knowledge. For instance, a project website was developed to serve as a forum and an information center for the group members to air the opinion and to share the information.

The Computer Science (CS) group works very closely with domain experts in different groups (statistics, meteorology, engineering, actuarial, and financial) in design, development, and validation phases to ensure that the multidisciplinary knowledge are well captured, integrated and realized in the system. The team members of each group continue giving feedback to CS group to improve the system and at the same time often receive valuable opinions and suggestions from CS group. The CS team thus plays a pivotal role by explaining the cause and effect of the results and the possible ways to improve the model to get more accurate results.

3.2 Model Architecture Design

When developing model architectures, the main goal is to achieve the necessary functionality and quality properties in accordance with the specified requirements and identified constraints. In this paper, we particularly discuss the development of model architecture as a significant part of our integration activities, which led to a relatively effortless integration for FPHLM.

As the knowledge acquisition and integration is a long-term process, FPHLM was designed using a top-down approach together with the critical module identification method. Consequently, the overall model flow was first defined, the critical modules were identified, and the model architecture was designed as shown in Figure 1.

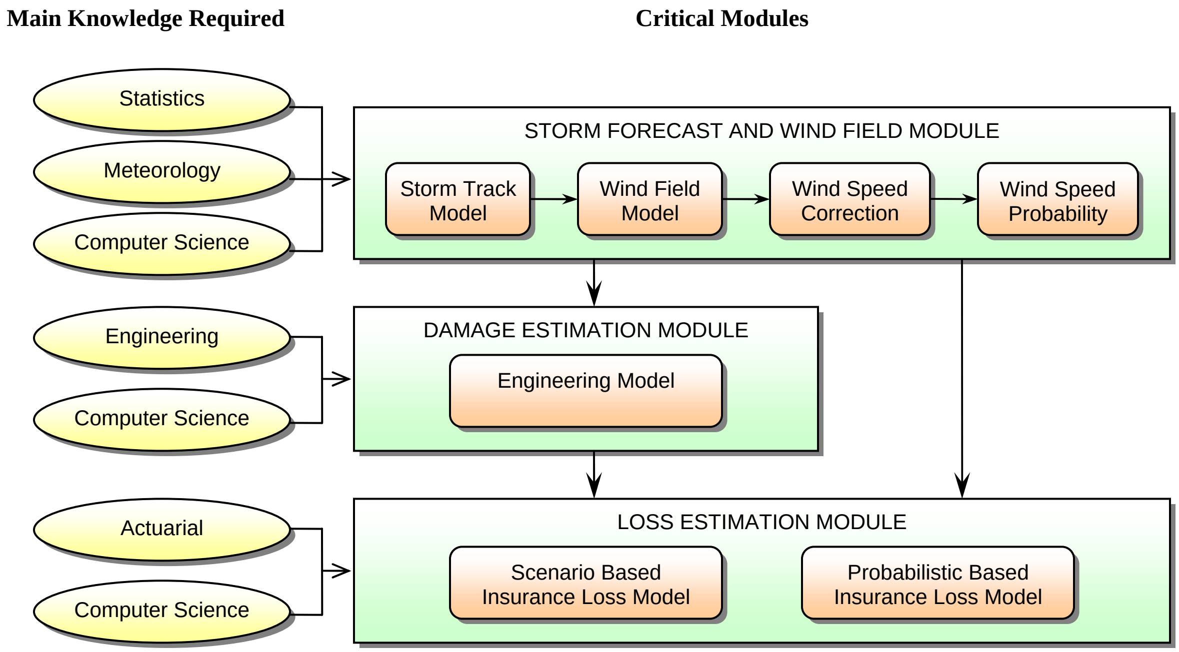

Figure 1. PHRLM model architecture

Table 1. Module functionalities

Module	**Component**	**Functionalities**
Wind Field Module	Storm Track Model	Generate the storm tracks for simulated storms based on random historical initial conditions (from HURDAT database) and stochastic algorithms
	Wind Field Model	Calculate wind speed time series for each of the zip codes affected by the storms
	Wind Speed Correction	Refine open terrain wind speed produced by the Wind Field Model with respect to the actual terrain (based on land use/land cover)
	Wind Speed Probability	Calculate the probabilities of the 3-second gust wind speeds affecting each of the targeted zip codes
Damage Estimation Module	Engineering Model	Develop the exposures and vulnerability model to estimate the physical damage for different exposures and wind speeds
Loss Estimation Module	Scenario Based Insurance Loss Model	Calculate the expected losses of a certain insurance portfolio caused by a specific hurricane
	Probabilistic Based Insurance Loss Model	Calculate the annual expected loss of a certain insurance portfolio

Here, FPHLM is assembled from a set of loosely-coupled modules to meet the flexibility and adaptability requirements. The functionalities of the modules are summarized in Table 1. Each module can be considered as a business object component which provides a service-oriented interface to external systems and allows interoperability between systems. In addition, each module was independently developed and required a minimum amount of knowledge from other domains. For example, Storm Forecast Module is constructed from a statistical point of view. At this point, little knowledge from actuarial and engineering disciplines is required. Moreover, the Wind Field Module, Damage Estimation Module and Loss Estimation Module are mainly related to the knowledge of a specific domain, namely meteorology, engineering, and actuarial, respectively.

3.3 Integration Plan Definition

A systematic integration plan is essential for efficient integration, which was designed to cover the following three major aspects in FPHLM.

- Define common representation (i.e., semantic integration): it aims at addressing the semantic content of the data to ensure that the same concept represents the same meaning and vice versa in different modules.
- Specify common interface between modules: it enables each isolated module to communicate with other modules.
- Implement data sharing mechanism: its goal is to ensure that the complementary information about the same data items is consistent, coherent and correct.

The aforementioned issues were taken into consideration throughout the FPHLM integration process and are detailed in the next section.

4. SYSTEM INTEGRATION

Integration Technology plays an important role in ensuring an effective and efficient integration process for software systems. In particular, the FPHLM system used a wide variety of data structures, software components, tools and languages as shown in Figure 2, to effectively estimate the cost of insured hurricane losses [2]. To accommodate the specific research needs and some of the existing development modules, FORTRAN, IDL and Matlab were used to implement the Storm Track Model, Wind Field Model, and the Damage Estimation Module, respectively. It was then the responsibility of the CS group to design and develop the remaining modules and to integrate them into the integrated system. As for the module development, an Oracle9i Application Server (Oracle9iAS) was deployed, whose OC4J container embeds a web server that handles the information receiving, interpretation, and responding tasks. For the sake of efficiency, C++/C along with the IMSL Mathematical and Statistical Libraries were invoked by Java Beans and used to conduct the complex scientific computation tasks such as statistical modeling and simulation. The client communicates with the web server via HTTP and SSL protocol, and Java Server Page (JSP) is exploited to generate dynamic web pages to show the modeled and/or simulated results. Oracle9i database was adopted and an object-relational database schema was designed to manage huge amounts of hurricane data and simulation results, where the communications with the web server are handled by Database Java Bean.

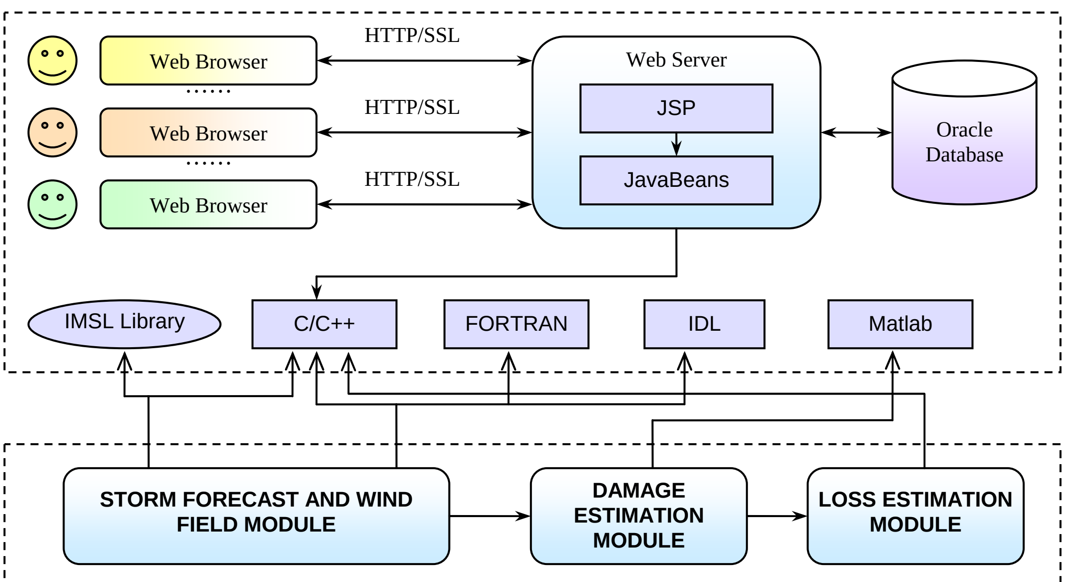

Figure 2. Overview of the techniques in PHRLM

Due to its diverse nature, the integration of the FPHLM couldn't be achieved by following any one of the popular integration techniques like File Transfer, Shared Database, Remote Procedure Invocation or Messaging in their absolute forms but rather we used a combination and in many places a customization of the existing techniques to develop a novel integration technique which suited the requirements of the projects without introducing any conflict.

According to the design architecture of FPHLM, our main integration target was the data that traverses around the entire framework of the project. Since the project is an ensemble of loosely coupled modules, each module behaves as an independent body and there is no sharing of functionality between them. This eliminates the need for Remote Procedure Calls to communicate between the modules. Moreover, some modules like the *damage estimation module* is subject to constant evaluation, changes and improvements based on the outputs of other modules and research outcomes, which prevents us from formalizing it into any standard service package. Correspondingly, we conducted the integration with the combination of three main techniques viz. Semantic Integration, Data Integration through Asynchronous Message Passing, and Data Integrations using File Transfer and Centralized Databases. The main aim is to make the modules communicate with one another and exchange information without affecting the individual functionality, data content and security.

Each integration technique used in FPHLM complements one another to make every loosely coupled module integrated to the system as illustrated in Figure 3. In brief, semantic integration was applied in the earliest possible stage to help achieve and maintain the consistent representations and semantic meanings of the defined terminologies throughout the software life cycle. In addition, the major integration plans mainly focus on the connection and data exchange between consecutive modules. Generally, data can be transferred in various ways according to the data size, format and the variety of intended recipients. In FPHLM, we employ multi-modal data integration techniques to take care of all the intricacies based upon different circumstances and distinct objectives. For better system portability and accessibility, a web-based approach was introduced wherein almost all the modules were developed into a web based application. The data values and commands with relatively smaller size for the main purpose of communication are mostly transferred with asynchronous messaging to make the interface transparent to the users. In case that the data is too voluminous or it is required to be stored as intermediate results, file transfer or centralized database is used for better control over the simulation process (e.g. pause, resume, etc.) and further data analysis. Meanwhile, a security control framework is applied by using the centralized database to further ensure confidential data access. Discussed below are the different techniques of integration implemented in FPHLM for smooth communication, secure data transfer and effective transparency among the modules.

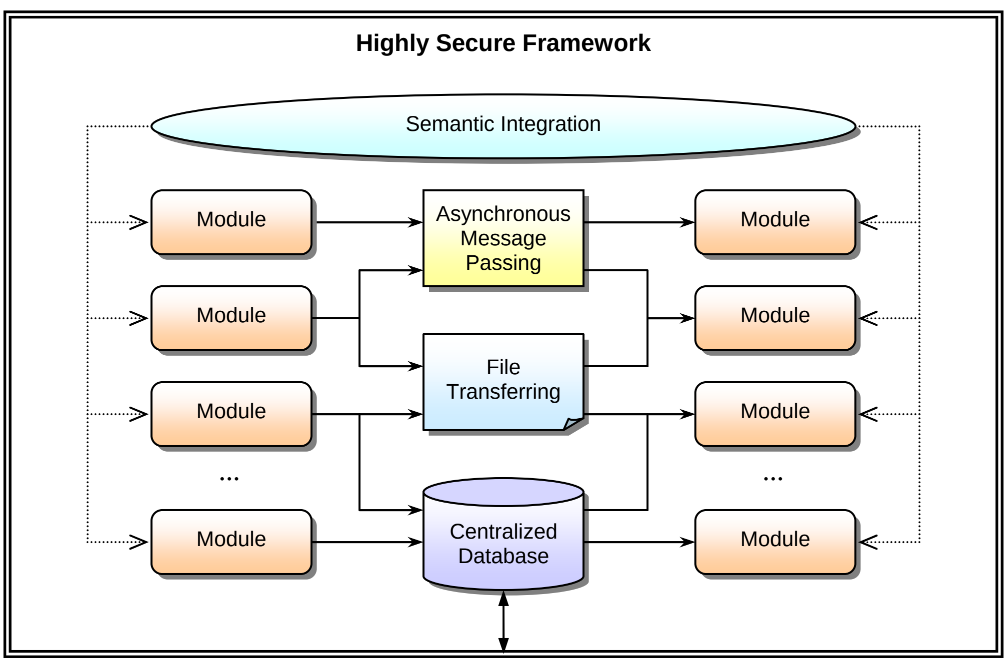

Figure 3. The secured multi-model integration plan of PHRLM project

Table 2. FPHLM terminologies and their semantics

Terminology	Semantics
Hurricane	Storm with wind speed >=74 mph or 33.012 meters/sec
Wind Speed	Hurricane wind speed in miles/hour
Pressure	Storm Pressure in mb
Damage	Extent of Physical Damage
Content Loss	Content Loss in Dollar value due to damages
Structure Loss	Structure Loss in Dollar value due to damages
APP Loss	Appurtenant Loss in Dollar value due to damages
ALE Loss	Additional Living Expenses Loss in Dollar value due to damages

4.1 Semantic Integration

As a multidisciplinary application, the terminologies used throughout the system play a vital role in contributing a common platform for understanding the underlying principles for flawless execution. Hence, Semantic Integration, which addresses the semantic content of the data in different modules, is an important part in integrating the entire project. When a system is semantically integrated, there is a common set of concepts used by all applications in the system and the user interface. Since FPHLM has several domains involved in the project like Statistics, Meteorology, Computer Science, Engineering and Actuarial, it cannot be expected that every module will use a common set of terminologies to implement the ideas for every scenario. Thus, the challenge was to select the key ideas across the entire system and make them uniform to achieve Semantic Integration without introducing unnecessary complexities. The key terms selected consist of the basic ideas upon which the entire implementation schema of the system relies. In certain cases, due to implementation prerequisites, some of the terms have different local meanings. A mapping technique was implemented which can map the global terms to their local meanings without altering the concepts. Table 2 lists a set of terms which were attached to a specific semantic meaning throughout the system.

4.2 Integration through Asynchronous Message Passing

Asynchronous messaging architectures have proved to be a good strategy for integrating loosely coupled systems since they overcome the limitations of remote communication, such as latency and unreliability. The approach is to let each isolated system communicate with one another through messages in a messaging channel. The advantage of using messaging in FPHLM is that it helps in communicating asynchronously and makes the communication more reliable because the two applications do not have to be running simultaneously. Messaging makes the system responsible for transferring data from one application to another, so the applications can focus on what data they need to share rather than how to share it.

We integrated our system using messaging by first converting each module into a web-based application and then communicating between the web-based units through messages over a secured communication channel. This approach allows us to encapsulate individual system components and make their functional complexities transparent to others. It also provides an interface to each module to enable the communications without interfering with the functionalities of each unit. In FPHLM, we mainly use two different types of Messaging viz. *Event Messaging* and *Document Messaging*.

In *Event Messaging*, when a subject has an event to announce, it creates an event object which is wrapped into a message and transferred over the message channel. The receiver on the receiving end unwraps it, detects the event and acts accordingly. *Event Messaging* was implemented in FPHLM for each module by notifying the consequent modules about an event where the sender modules play an active role. It does not invoke any functions in any module but simply informs on the status of some events within its own domain. For example, each module after its completion sends a message to activate the execution of the next module. Moreover, with event messages, one module can announce that the data is ready to be picked up from some common resource or centralized Database.

Document Messaging passes data and lets the receiver decide how to handle it. In FPHLM, *Document Messaging* was achieved by sending the data structures and/or data values wrapped in messages over the web from one unit to another. The module interface receives the message, extracts and reformats the data according to its application requirement. This technique is particularly applied when the exchanged data have simpler structures and are only used for the communication purpose, which helps in avoiding intensive file I/O operations and thus improves the overall performance of the web-based architecture.

4.3 Data Integrations using File Transfer and Centralized Databases

Asynchronous messaging is a portable solution for seamless connectivity and data communications between different modules while ensuring the transparency to the general users. However, it cannot be applied to address all the integration problems, especially in handling the large scale data transferring tasks. Moreover, some modules such as Damage Estimation Module were constantly revised and improved, and involved intensive mathematical operations which inhibited them from being an ideal candidate for messaging approach. Hence, the centralized Oracle database and file transfer approach is especially applied in FPHLM in which the data required and used is of enormous size, frequently accessed and extremely sensitive. A set of criteria were considered in this process as detailed below.

The first criterion is regarding the data integration and correctness, which is accomplished mainly via the semantic integration. In addition, to further check the completeness and correctness of the exchanged data, the outputs of the previous module were utilized as the inputs for the experiments and testing of the next module.

Secondly, for the generation of various sets of hurricane simulations, it is essential to define their classifications and lead the next module to the correct path for the right inputs. Our solution is to organize the data in different tables or in different folders. For instance, with the execution of Annual Hurricane Occurrence (AHO) module, it generates distinct folders for the new simulation or stores the results in the existing folder, if the simulation with same parameters has been conducted before. Therefore, users are able to choose the dataset and resume the simulation process in the future.

Thirdly, it is important to protect the confidentiality of the sensitive data to ensure the recipient has appropriate access privilege. FPHLM adopts an integrated access control infrastructure to complete the identification management. The user information and password are stored in the centralized database while the website based system offers the login mechanism for identification checking. Different access control capabilities are assigned to each user and therefore only the intended recipients have access permission. In addition, to further ensure secured data integration and transfer, particular protection plan is embedded to guard the data in the transition process either from node to node basis, or at the application level. For example, the highly sensitive insurance data are normally password protected and stored in the servers. The intended recipients are notified when the files are ready and can only access the data through the server after logging into the system. All of these integration strategies make the overall system highly secure and guarantee more confidential communications between different modules. The aforementioned integration approaches can assist future research and development efforts for similar large scale multi-disciplinary digital government projects. The above approaches can be utilized to assimilate information from different domains and integrate the heterogeneous system modules/components, ensuring an effective and concrete digital governance solution.

5. CONCLUSIONS

In this paper, we presented our efforts and experiences in the development of the Florida Public Hurricane Loss Model (FPHLM) with the focus on the system integration process. This model has been adopted by the State of Florida Office of Insurance Regulation to regulate the insurance ratemaking process and is open to public. As a multi-disciplinary digital government project, system integration was one of the major challenges during our system development cycle. For the sake of efficient integration, three main issues (knowledge acquisition, sharing and integration, model architecture design, and integration plan definition) were taken into consideration during the requirement specification and design phase. In addition, we adopted a novel integration technique which was developed by the intelligent combination and customization of the existing techniques to address the specific characteristics and unique requirements of FPHLM project. Our approach can further facilitate rapid and effective development and integration of multi-disciplinary digital government project components via utilization of integration plans that mainly focused on connection and secure data exchange while ensuring system portability and accessibility. We believe the multidisciplinary large scale system integration experience will benefit the future research and system development in related areas.

6. ACKNOWLEDGMENT

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

7. REFERENCES

[1] Boehm, B. A Spiral Model of Software Development and Enhancement. Software Engineering Project Management (1987), 128-142.

[2] Chen, S.-C., Gulati, S., Hamid, S., et al. A Web-based Distributed System for Hurricane Occurrence Simulation. Software: Practice and Experience, 34, 6 (May 2004), 549-571.

[3] Land, R. and Crnkovic, I. Software Systems Integration and Architectural Analysis – A Case Study. Proceedings of the International Conference on Software Maintenance (Sep. 2003), 338-347.

[4] Nilsson, E. K., Nordhagen, E. K., and Oftedal, G. Aspects of Systems Integration. Proceedings of the First International Conference on Systems Integration (Apr. 1990), 434-443.

[5] Stal, M. Web Services: Beyond Component-Based Computing. Communications of the ACM, 45, 10 (Oct. 2002), 71-76.

[6] Sutherland, J. and Heuvel, W.-J. Enterprise Application Integration and Complex Adaptive Systems. Communications of the ACM, 45, 10 (Oct. 2002), 59-64.

[7] Walz, D. B., Elam, J. J., and Curtis, B. Inside a Software Design Team: Knowledge Acquisition, Sharing, and Integration. Communications of the ACM, 36, 10 (Oct. 1993), 63-77.

[8] EQECAT WORLDCATenterprise Model. http://www.eqecat.com/.

[9] HAZUS manuals page. http://www.fema.gov/hazus/li_manuals.shtm.

[10] HURDAT data. http://www.aoml.noaa.gov/hrd/hurdat/.

[11] RMS U.S. Hurricane Model. http://www.rms.com/Catastrophe/Models/United_States.asp#WS.