Journal of Applied Mechanics Vol.13, pp.691-698 (August 2010) JSCE

Real-Time Hazard Map as an Application of Enhanced Integrated Earthquake Simulation (IES) with High Performance Computing Technique

Gholamreza Sobhaninejad*, Muneo Hori**,
Tsuyoshi Ichimura*** and Lalith Maddegedara****

*Graduate School Student, Earthquake Research Institute, University of Tokyo (1-1-1 Yayoi, Bunkyo, Tokyo, 113-0032
**Ph.D., Earthquake Research Institute, University of Tokyo (1-1-1 Yayoi, Bunkyo, Tokyo, 113-0032)
***Ph.D., Earthquake Research Institute, University of Tokyo (1-1-1 Yayoi, Bunkyo, Tokyo, 113-0032)
****Ph.D., Department of Civil Engineering, University of Tokyo (7-3-1 Hongo, Bunkyo, Tokyo, 113-8656)

Integrated Earthquake Simulation (IES) is enhanced with High Performance Computing (HPC). The development of an automatic generation of a real-time hazard map is studied as a potential application of IES enhanced with HPC. This map is generated for urban areas immediately after wave records of a large earthquake are available and possible hazard and damage caused by the same earthquake are visualized in the web. The system is designed so that the real-time hazard map is automatically generated by means of IES. As an example, the generation of the hazard map is studied for Tokyo Metropolis, and it is shown that the map can be made within a half day when a PC cluster of moderate size is used.

Key Words : *Hazard Map, High Performance Computing Earthquake Disaster, Information and Communication Technology, Object Oriented Design*

1. Introduction

Integrated Earthquake Simulation (IES) is enhanced with High Performance Computing (HPC) that takes advantage of parallel computation [1),2),3)]. The simulation of earthquake hazard and disaster for a given earthquake scenario is made for a larger area in a shorter time. The simulation with higher spatial and temporal resolution as well as higher reliability can be made by implementing more advanced seismic response analysis programs.

Even though IES is enhanced with HPC, its primary objective does not change, i.e. to estimate the possible hazard and disaster for a given earthquake scenario. Such an estimate serves as fundamental data for the earthquake disaster preparation. Also IES enhanced with HPC could be used to make an urgent evaluation of earthquake disaster immediately after a large earthquake attacks an urban area. The users of IES for the primary objective are local government and private sectors. Another objective of using IES is considered in this paper, together with seeking a potential user of such an application.

Residents' awareness of an earthquake disaster should be kept at a suitable level to promote earthquake disaster mitigation. A so-called hazard map of an earthquake is a good tool to alert residents' awareness. It is possible to make a dynamic hazard map, or 2D or 3D images of earthquake hazard and disaster which are simulated in an urban area model, in real time using the visualization tool of IES. This is a hazard map of the next generation type and is called a *real-time hazard map* in this paper.

The real-time hazard map is generated immediately after a large earthquake happens. It is composed of images which show the synthesized strong ground motion and the resulting seismic responses of all structures in a target area, assuming that the same earthquake happens there. HPC that is enhanced for IES is essential in order to compute a larger area in a shorter time. A higher computing performance is required for IES. Increasing the spatial resolution of the IES simulation is important as well providing more realistic images of a possible earthquake disaster to each of the residents.

The potential user of the real-time hazard map is local resident. There could be another potential user, such as a government official or a private sector executive, who may use this map to make a drill of running business continuity plan right after an earthquake happens. The requirements of the real time hazard map for them are the speed of generating a map and the reliability of predicting damage distribution; the instantaneous generation of the map is important if the drill of running business continuity plans is planned to be carried out immediately after a large earthquake.

In order to meet the user requirements, IES must have a function for automatic generation of the real-time hazard map. Monitoring IES[1] is needed so that there is no malfunction in the automatic generation. The use of *web technology* as well as the top-down design method is considered; the design method consists of the three phases, namely, defining functionalities and resources, designing the system, and implementing the system to the web.

The contents of this paper are as follows: In Section 2, we discuss required functionalities and available resources for a system which generates a real-time hazard map based on IES simulation. Section 3 demonstrates the design of such a system, which is made by taking advantage of modern software engineering concepts, e.g., design pattern and decentralization. Section 4 shows the results of implementing such a system to generate real-time hazard maps for Tokyo. The concluding remarks are made in the last section.

2. Definition of Functionalities and Resources

The automatic generation of the real-time hazard map takes the following steps: 1) detecting the occurrence of an earthquake; 2) executing the simulation for the detected event by managing the computation resources; and 3) visualizing the results as video clips and images and putting them on the web. The soundness of the system for the automatic generation should be monitored by an administrator of the system.

A strong ground motion network is used to detect the occurrence of an earthquake; waves recorded by the network are used as input of the IES simulation. Such usage of the network data could be regarded as a new service of the strong ground motion network. In this study, the K-NET (Kyoshin-Net) is used as a strong ground motion network. This network consists of 1028 observation stations and covers the whole Japanese Islands with average spacing of 25 km. When a large earthquake occurs, the waveform records of all K-NET stations are automatically sent to the data management center of National Research Institute for Earth Science and Disaster Prevention (NIED).

Executing the IES simulation is a core part of the automatic generation of the real-time hazard map. The time required for this part is easily increased if available computer resources are poorly used. Taking the advantage of the scalability of IES enhanced with HPC, we seek to establish a resource management strategy so that the efficiency of using available computer resources is increased.

A free-ware software, `Visit`, is used as a visualization tool of IES for a real-time hazard map. This is because `Visit` offers the off-line rendering facility which accelerates the visualization process and can use raw visualization data in a *Visualization Toolkit* (VTK) format which IES is able to output [4)].

During the operation, the soundness of the system for the automatic generation should be monitored; monitoring is inevitable for a system with less use. The soundness is checked for each component of the system separately.

In order to provide all the functionalities, the design of the automatic generation system for the real-time hazard map is illustrated as a use case diagram shown in **Fig. 1**. For simplicity, the system is called IES_WEB, and it consists of the three components, namely, `Event_Manager`, `Simulation_Manager`, and `Visualization_and_Web_Manager`, for the automatic generation, and has one component, `Monitoring_Room`, for the system soundness monitoring.

Each component of IES_WEB has a huge work-

[1] It is required to examine the results of the automatically generated hazard map. Errors in computing the earthquake hazard and disaster would lead to fatal mistakes in running the drill of business continuity plans.

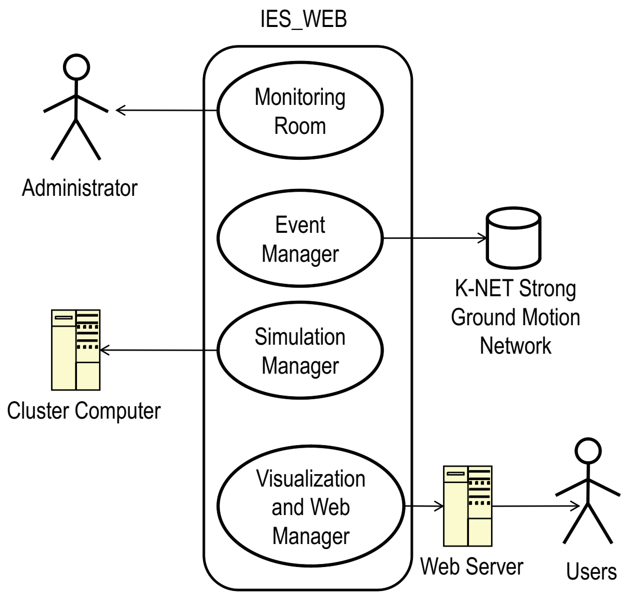

Fig. 1 IES_WEB use case diagram.

load, and hence decentralization is needed; components are allocated in different machines; see [5] for the decentralization of the system. Better fault tolerance is provided to IES_WEB's components by setting looser conjunction between components. The Internet protocol is used for the communication among the components. The TCP/IP socket is used to establish the communication channel, and the *Secure Socket Layer* (SSL) is used to support the communication; see [6]. The communication between `Visualization_and_Web_Manager` and the web server is established through the *File Transfer Protocol* (FTP).

3. IES_WEB Design

First of all, we design `EventInfo` and `NetInfo` classes to keep the simplicity and integrity in implementing the IES_WEB components. The `EventInfo` class is used for intra- and inter-component communication. Earthquake information and wave data are stored in this class, and they are sent and received by different components. When the data of strong ground motion networks need corrections, such as baseline correction or drift correction, the `EventInfo` class conducts a necessary data processing which uses a `Process( )` function. The `NetInfo` class stores information related to network communication. This class is used to establish the connection, to disconnect the connection and to send and receive information through a communication medium. For the soundness monitoring, a `MonitorListener` class is introduced so that it controls communication with the `Monitoring_Room` component, using the `EventInfo` and `NetInfo` classes.

Next, we take advantage of a pair of classes, standard Java Observer and Observable classes; see [7),8)]. This pair is de facto standard in the software engineering when an object or component is continuously monitored by other components and is used to take some action. Inheritances of these classes for the components of IES_WEB are made according to the Observer-Observable design pattern. It should be noted that the inheritances of the Observer and Observable classes are forced to work on two separate threads so that their processes become smoothest by taking advantage of the Java standard `Runnable` class.

The `Event_Manager` component is designed to monitor the occurrence of an earthquake recorded in the K-NET. It consists of `NetworkObserver`, `SimulationManagerCommunicator`, and `MonitorListener` classes. The design of the `Event_Manager` component is illustrated in **Fig. 2**.

As mentioned, the `NetworkObserver` and `SimulationManagerCommunicator` classes are inheritances of the standard Java Observer and Observable classes. The `NetworkObserver` class checks[2] a new event using `CheckNewEvent()` function, and, if there is a new event, notifies it to the `SimulationManagerCommunicator` class and others using `Query()` and `GetSignal()` functions. The `SimulationManagerCommunicator` class receives and stores the event information. It uses a `SendEvents()` function to send the latest event information to the `Simulation_Manager` component and deletes the oldest event information from the events queue.

The `Simulation_Manager` component is designed to execute the IES simulation for an event which the `Event_Manager` component detects. It consists of `EventListener`, `SimProcessManager` and `MonitorListener` classes. The schematic design of this component is presented in **Fig. 3**.

The `Simulation_Manager` component uses

[2] The current IES_WEB executes the simulation, regardless of a main shock or an after-shock. The type of an earthquake will be considered in order to execute the automatic generation. Such a consideration will be made by the `NetworkObserver` class.

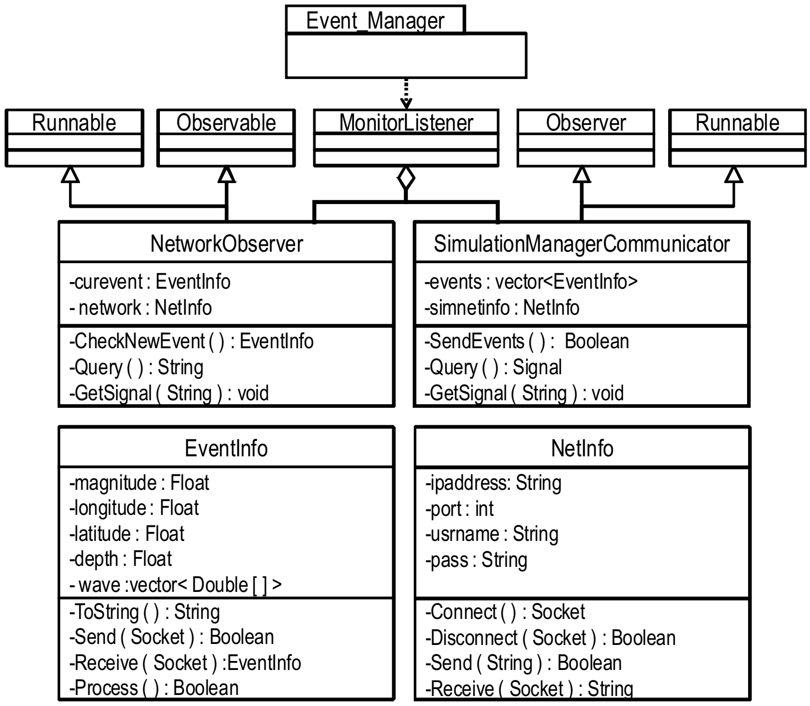

Fig. 2 Schematic view of Event_Manager component.

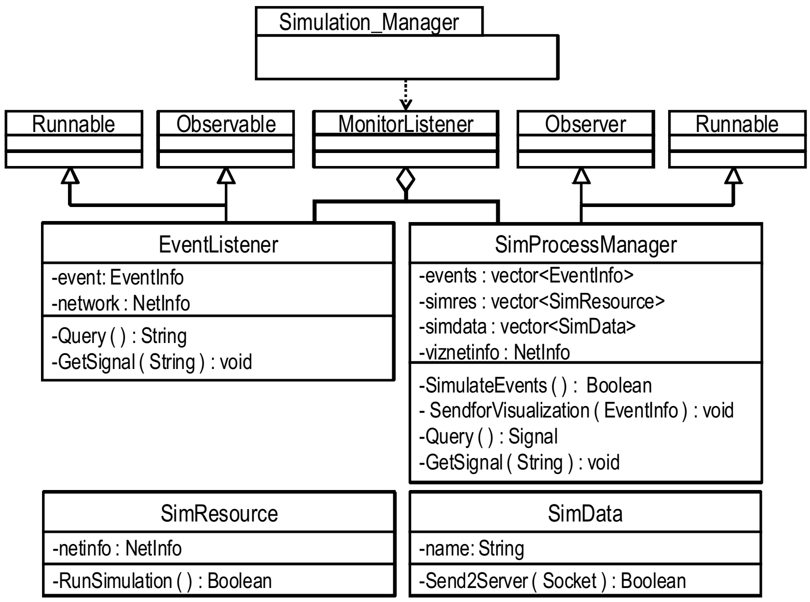

Fig. 3 Schematic view of Simulation_Manager component.

`SimResource` and `SimData` classes for the information related to computational resources and GIS data. The `SimResource` class represents a computation resource which executes the IES simulation for a given input strong ground motion; an instance of this class is a node of a cluster computer in the present IES. The `SimResource` class uses a `RunSimulation()` function to execute the IES simulation on a specific computation resource which is designated by the `SimResource` class.

The `EventListener` and `SimProcessManager` classes are inheritances of the standard Java Observable-Observer classes. The `EventListener` class is responsible for establishing a connection to the `Event_Manager` component. When a new event arrives, it notifies the `SimProcessManager`

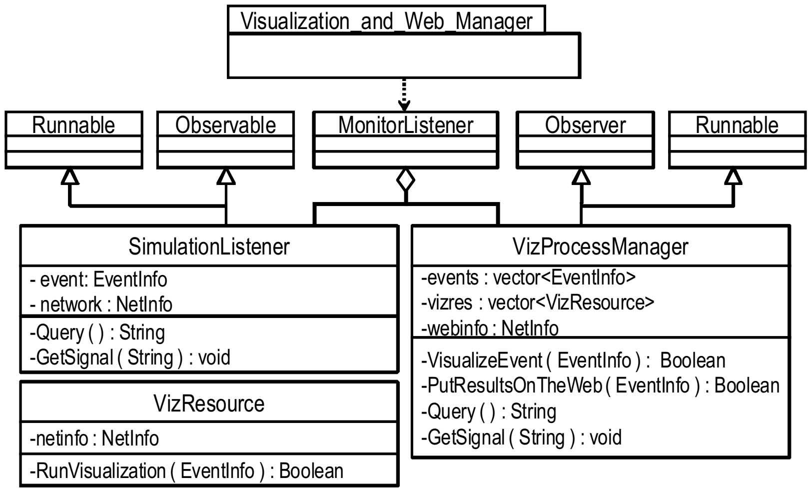

Fig. 4 Schematic view of Visualization_and_Web_Manager component.

class, and the `SimProcessManager` class executes several or many simulations simultaneously on available computational resources using a `SimulateEvents()` function; the `SimProcessManager` class does not start the simulation until sufficient computational resources become available. It notifies the `Visualization_and_WebManager` component when the simulation finishes.

The `Visualization_and_Web_Manager` component is designed to visualize the results of the IES simulations and to put them in the web. It consists of `SimulationListener`, `VizProcessManager` and `MonitorListener` classes. The schematic design of this component is presented in **Fig. 4**.

The `Visualization_and_Web_Manager` component uses a `VizResource` class. This class is for a resource for visualization, such as a visualization program for the IES simulation results.

The `SimulationListener` class receives a message from the `Simulation_Manager` component about the finishing of the IES simulation and notifies the `VizProcess_Manager` class. This class has a queue of events, which are to be visualized using a `VisualizeEvents()` function and to be uploaded in the web server using a `PutResultsOnTheWeb()` function.

The `Monitoring_Room` component consists of Graphic User Interface (GUI) and the aggregation of three classes, which inherit an abstract class of `Monitor_able_Object`, namely, `EventManagerMonitor`, `SimManagerMonitor`, and `VizManagerMonitor`. Figure 5 shows the class diagram of the `Monitoring_Room` component.

The `Monitor_able_Object` class has three functions, namely, `Query()`, `Interpret()`, and `SendSignal()`. The `Query()` function sends a query request to other components and receives a reply. The `Interpret()` function uses the reply to check an error in the inquired component. The `SendSignal()` function sends a signal to other components to restart, stop, or terminate the execution processes.

4. IES_WEB Implementation

In the present subsection, the implementation of IES_WEB, or the automatic generation of the real-time hazard map using K-NET, is discussed. The `Event_Manager` component as well as the `EventInfo` class are designed so that the implementation becomes straightforward.

Through the `Event_Manager` component, IES_WEB accesses the K-NET data. This component monitors the occurrence of a new event by checking the K-NET site in a fixed time interval. It downloads event information when it detects a new event; the information consists of wave data which are recorded on several observation stations. The information is processed to select the wave data that have the maximum geometrical mean acceleration in the horizontal components. The `Event_Manager` component instantiates an entity of the `EventInfo` class for the selected wave data, and sends this entity to the `Simulation_Manager` component.

Two earthquake events, which are recorded in the

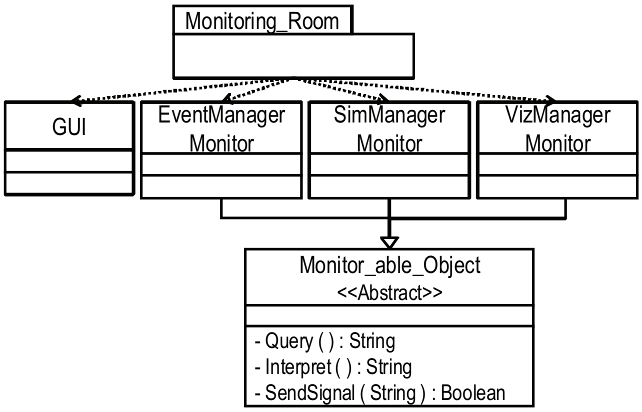

Fig. 5 Schematic view of Monitoring_Room component.

K-NET database, are used in order to demonstrate the automatic generation of the real-time hazard map. The peak ground acceleration of the wave data that are selected for the first and second earthquake events are 1715.5 and 447.4 gal, respectively.

The `Simulation_Manager` component executes the IES simulation for urban area models, in which the structures are modeled as Multi-Degree of Freedom (MDF) models [1]. These models are constructed by using the GIS data of Tokyo Metropolis, and represented as an entity of the `SimData` class; see **Fig. 3**. There are 286 models of the urban area for the entire Tokyo Metropolis or 23 Wards.

In this study, we use a cluster computer consisting of 14 computational nodes which have 8 cores on each node. The `Simulation_Manager` component uses each node as a computation resource, assigning one entity of the `SimResource` class to each; one entity of the `SimResource` uses 8 cores to execute the IES simulation for one entity of `SimData` class. The `Simulation_Manager` component creates the same number of threads as the number of available computational resources; the number is 8 in the current setting. The simulation of all the resources starts simultaneously. The average CPU clock time which is needed for one model is 10 minutes, and the simulation time for all the models of Tokyo Metropolis is 3.5 hours.

In order to accelerate the visualization, the `Visualization_and_Web_Manager` component creates several threads using all available computer resources. Each thread runs an instant of the `Visit` software. Visualizing the IES simulation for one model takes 1 minute when eight threads are used. It takes approximately 1 hour to finish the visualization of the 286 models of the entire Tokyo Metropolis. The `Visualization_and_WEB_Manager` uploads the results in the form of bitmap images to the web server. This process takes 10 minutes for all the models when the visualization results are in the form of Portable Network Graphics (PNG) images, the average size of which is 200 kilobyte.

Table 1 Characteristics of Selected GIS tiles.

GIS Tile ID	Ward	Number of Buildings
09KD963	Nerima	14911
09LD083	Bunkyo	12755
09LD171	Shinjuku	14143
09LD262	Shibuya	9136
09LD291	Koto	8612
09LD352	Setagaya	13348
09LD472	Shinagawa	12582

As an illustration, we chose 7 models; see **Table 1** for the location of the models and the number of analyzed structures. Although static, the real-time hazard maps are generated in less than 5 hours; see **Fig. 6** for three samples of the automatically generated hazard map of the two earthquake events. The damage of structures is indicated as the degree of inter-story drift angle, which is usually used as an index of seismic structure damage. It is apparent that conducting such a massive simulation which generates detailed results for each story of a building in such a short amount of time could not be possible without leveraging the power of HPC.

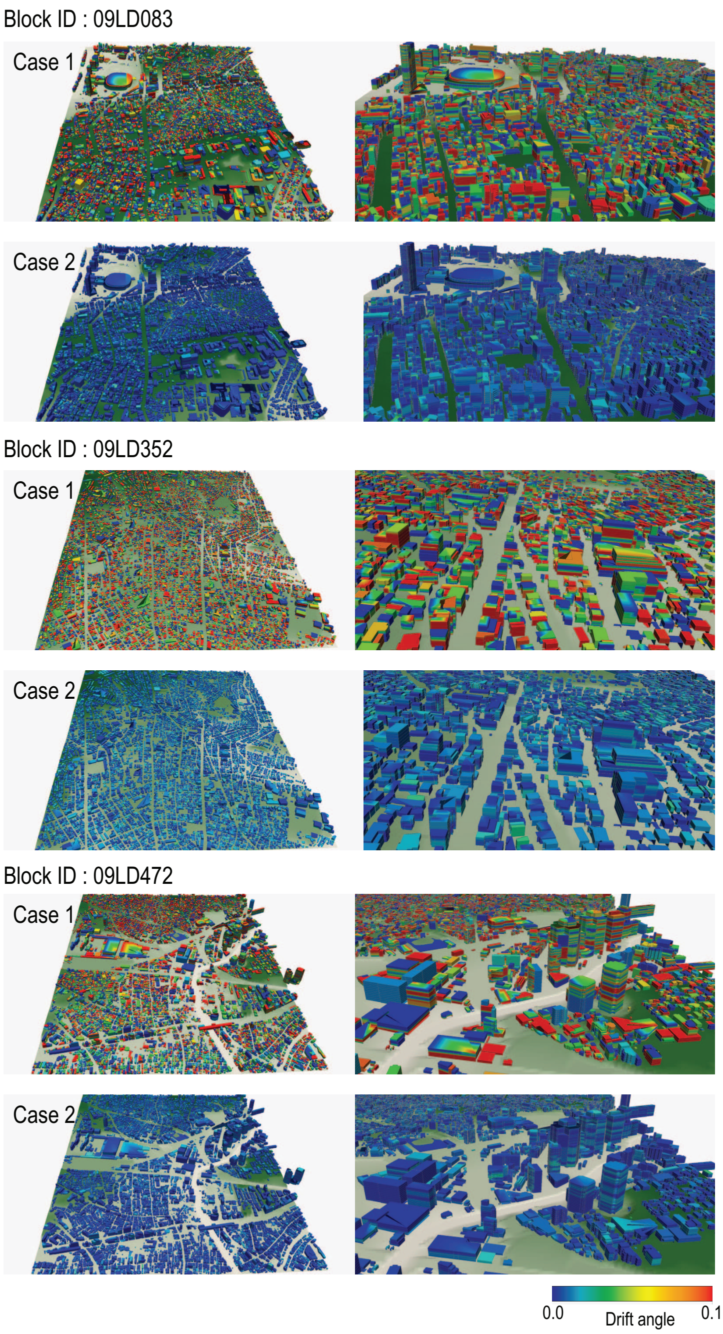

Fig. 6 A sample of generated images by IES_WEB (left column), the focused view of the maps (right column).

5. Conclusion

This paper demonstrates a potential application of IES enhanced with HPC to automatically generate a real-time hazard map. It is HPC that plays the key role of conducting large-scale numerical simulation in short amount of time, to generate the real-time hazard map. The systematic method of developing IES_WEB is presented, and the emphasis is put on software engineering aspects to mobilize the available facilities in earthquake engineering and computational technology. The example of the automatic generation of the real-time hazard map is discussed. The power of IES enhanced with HPC is utilized to finish the automatic generation less than a half day, using the computer environment of the authors.

It is straightforward to replace the linear analysis with the non-linear analysis for the simulation that is conducted in IES_WEB. Also, video clips can be used instead of static images as the output of IES_WEB. The time required for them is surely longer than that measured for the present setting. A computer environment which finishes these simulation and visualization in a reasonably short time should be found.

Acknowledgements: The authors gratefully acknowledge the financial support of Japanese government (MEXT). Also, we would like to extend our thanks to PASCO Co. Ltd. for providing the GIS data of Tokyo city.

REFERENCES

1) Gh. Sobhaninejad, M. Hori, "Basic study on enhancing IES with parallel computation," Journal of Applied Mechanics, JSCE, **12**, 623-629, 2009.
2) K. Tanaka, Gh. Sobhaninejad and M. Hori, "Data Conversion of GIS to Urban Area Model for Natural Disaster Simulation," Journal of Applied Mechanics, JSCE, **11**, 647-654,2008.
3) T. Ichimura, M. Hori, "Current state of integrated earthquake simulation for earthquake hazard and disaster," Journal of Seismology, **12**, 307-321, 2008.
4) Kitware Inc., "Visualization Toolkit (VTK) Users Guide," Kitware Inc.,www.vtk.org.
5) V. Silva, "Grid Computing for Developers," Charles River Media, 2005.
6) S. A. Thomas, "SSL and TSL essentials securing the Web," New York Wiley, 2000.
7) E. Gamma, R. Helm, R. Johnson, J. M. Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software," Addison-Wesley, 1994.
8) Sun Microsystems, "Java SE API Documentation," Sun Microsystems, http://java.sun.com/javase/reference/api.jsp.

(Received 2010, March 9)