https://doi.org/10.20965/jdr.2016.p0255

Paper:

Simulation Data Warehouse for Integration and Analysis of Disaster Information

Jing Zhao, Kento Sugiura, Yuanyuan Wang, and Yoshiharu Ishikawa

Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
E-mail: {zhao,sugiura,yuanw}@db.ss.is.nagoya-u.ac.jp, ishikawa@is.nagoya-u.ac.jp
[Received October 1, 2015; accepted January 21, 2016]

Studies on disaster countermeasures utilize extensive simulations of earthquake, tsunami, people evacuation, and other targets, generating enormous amounts of data. The continuing development of computational capability has facilitated the increase of the simulation data size and the utilization of such "big data" has become a serious problem. With this background, the present study proposes, from the viewpoint of information science, the simulation data warehouse approach for the interactive analysis of large simulation data and describes a method of realizing a data warehouse. An objective of this study is to integrate different simulation data sets and enable exploratory analysis of multiple accumulated simulation data with high-speed response by data preprocessing. Further, the developed prototype system architecture and a case example of its use are explained.

Keywords: data warehouse, disaster information, simulation data, spatio-temporal databases, interactive analysis

1. Introduction

In scientific fields, the amount of data has been increasing enormously owing to the development of information technologies, high performance computers, and high capacity storage. The science of managing large data is frequently called **data science** and has attracted attention as a new direction of science. In this study, we focus on simulations conducted in different fields. A massive amount of data is generated in these simulations and **data-intensive computing**, which aims to use these data, now receives attention as the "fourth paradigm" of science [1].

Many simulations are conducted to analyze earthquakes, tsunamis, and other disasters. Because the simulations are performed many times using different conditions and parameters, the data volume becomes large. The simulation results are frequently compared or integrated with other simulation data for advanced analysis, which could also be utilized in future studies. The idea of managing the simulation data corresponds to the above-mentioned fourth paradigm. The amount of data produced from today's simulations is increasing and hence, technical development for the effective use of big simulation data is strongly required. In particular, support for exploratory analysis, which is characteristic to science, and support for instantaneous interactive responses to analysis requests for big data are desired. With this in mind, a system technique is developed in this study.

For **spatio-temporal simulations** addressing temporal and spatial information, the present study focuses on the technical development of a data warehouse where a large amount of simulation data is managed and analyzed. Earthquake and tsunami simulations [2] are selected as examples and a system is designed and developed considering the requirements of this field. In the following, a system architecture and implementation technique is described and an interface and functionality of the developed prototype for data analysis are explained.

The paper is organized as follows. In Section 2, the requirements for simulation data analysis are explained. Section 3 is devoted to a description of the data warehouse that is used as a base for the present study. The architecture of a prototype system and a realization method for the system are presented in Section 4. In Section 5, the development approach for the simulation data warehouse for the target simulation data is provided. Case examples of queries and visualization are presented in Section 6 and the problems identified in the development of the prototype are explained in Section 7. In Section 8, related studies are introduced and a summary is offered in Section 9.

2. Simulation Data Analysis

The majority of disaster simulations have a simulation target of a particular area and calculate temporal variations in the area after the disaster. For example, a tsunami simulation calculates the flood depth at each site by time period. An evacuation simulation calculates the location where evacuees can be found by time period. The captured simulation data links the temporal information such as the time of occurrence of a disaster to the spatial information such as the coordinate of an evacuee's location.

Let us consider damage status analysis as an example of an analysis using disaster simulation data. For the analysis, it is assumed that an analyzer specifies the scale and location of an earthquake and integrates multiple sets of earthquake simulation data. Suppose we have a request

for the integrated analysis of earthquake intensity in the center of Tokyo, damage from a tsunami, and evacuation status of the people in the event of an earthquake at the seismic center x offshore Chiba Prefecture of magnitude y that occurs at noon on a weekday in April. To respond to this request, it is necessary to identify and extract the simulation data from the database that matches the conditions.

Analysis, however, is usually made in an exploratory manner. Analysis is not made initially on the details of the simulation data directly. Data are observed from multiple viewpoints to identify the important data and then, for a more detailed analysis, the data are further investigated. For instance, the query "Calculate the number of people in each cell of a rough mesh of 1 km $\times$ 1 km by aggregating evacuee data in the specified area after the occurrence of the earthquake" can be instantaneously executed, if the functionality of grasping the people distribution in each cell is available by techniques like visualization, which is useful for analysis.

In business, data warehousing is known as a technology for supporting the analysis of big data in an exploratory and interactive manner [3–5]. Data warehousing is a system to realize efficient interactive analysis by data preprocessing in the database. Unlike an ordinary database, data in a data warehouse does not require updating and hence, the data can be reorganized for analysis.

Knowing the demand and background, in the present study we utilize the data warehouse technology for the integrated and interactive analysis of simulation data. In particular, with an emphasis on spatio-temporal data such as disaster information, system techniques are developed addressing the characteristic data and distinctive analysis properties in this field. We therefore use a new term, *simulation data warehouse*, for a data warehouse specialized for simulation data. Because temporal and spatial information are both involved in the spatio-temporal simulation, it is important to provide supports corresponding to them, such as constructing spatial index that aims to optimize query processing. Further, because it holds more exploratory aspect in scientific domain comparing to the business field, it is necessary to analyze data interactively using a trial and error process.

3. Data Warehouse Technology

Unlike a conventional database that is updated by adding or deleting data, a data warehouse requires data reorganization in an appropriate form for the analysis of accumulated data and the data in the data warehouse must be configured in such a manner that interactive analysis can be realized from various perspectives. This requirement was pointed out in 1993 by E. F. Codd, a proponent of relational data models. This type of analysis, called *OLAP* (On-Line Analytical Processing) [6], requires a multi-dimensional-structured database. Data warehouse (DWH) [3–5] was proposed at approximately the same time. It shares the motivation and purpose with OLAP.

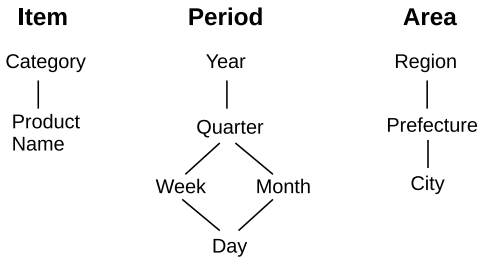


Fig. 1. Example of concept hierarchy.

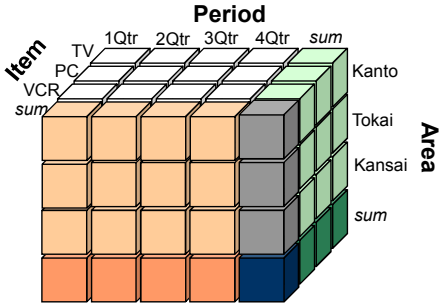


Fig. 2. Example of data cube.

However, DWH is more system-oriented.

One of the basic concepts often used for data warehouse is **concept hierarchy**. A single concept hierarchy is assigned to each domain. For example, the concept hierarchies presented in **Fig. 1** are used in a data warehouse for collecting and analyzing the sales of a company that sells electric appliances throughout the country. For "Item," there is a hierarchy of category and product names. For "Period," there is a hierarchy of year, quarter, week, month, and day. For "Area," there is a hierarchy of region, prefecture, and city. For example, an analysis command such as "Analyze the data of total sales amount of LCD televisions in the Prefectures in 2014, and aggregate the total sales amount by various combinations of prefecture and month." can be made based on the concept hierarchy. A variety of analyses can be executed by changing the levels and combinations of the concept hierarchies in a flexible manner or using various calculation functions.

Significant research and numerous data warehouse and OLAP developments have been undertaken. In particular, the **data cube** created by Gray et al. had an important impact [7]. A data cube is a data model used for analysis. It is obtained by arranging the requirements of a multi-dimensional data analysis and target data conceptually expressed as a multi-dimensional cube. **Fig. 2** presents sales information in the form of a three-dimensional data cube where each cell contains a statistical quantity such as amount of sales. Using calculation tools to process the data cube, one can make a variety of analyses.

In today's commercial relational database management systems (RDBMS), the basic functions of a data warehouse exist, making this appropriate to be used as the basis for the development of a simulation data warehouse system. *MDX* (MultiDimensional eXression) [5, 8] is fre-

quently used in the interface for RDBMSs. It is implemented in Microsoft SQL Server [9] and has become the commercial system de facto standard. The format of MDX is similar to that of SQL and is written as follows.

```
SELECT
  Member selection ON COLUMNS,
  Member selection ON ROWS
FROM <cube name>
[WHERE conditions specification]
```

This instructs a multi-dimensional cube to create a two-dimensional tables. `ON COLUMNS` specifies the content along the vertical axis and `ON ROWS` is the content along the horizontal axis. `Member selection` specifies a dimension and level from the concept hierarchy. In the `WHERE` field, conditions for the data cube are set. In the present study, MDX is implemented in the system.

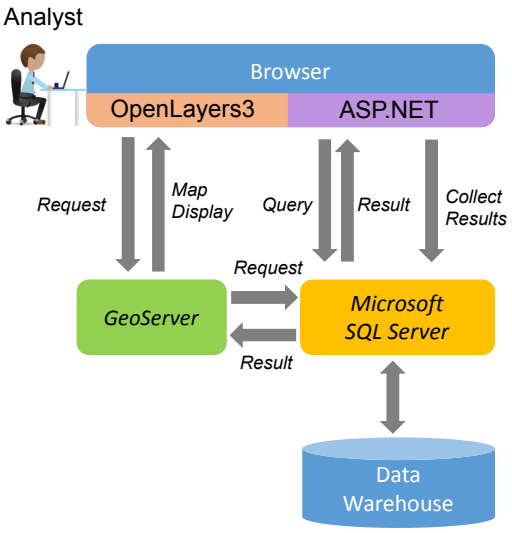


Fig. 3. Prototype system architecture.

4. Configuration of Prototype System

4.1. System Architecture

The architecture of the constructed prototype system is illustrated in **Fig. 3**. All of the simulation data are stored in the data warehouse. For the management of the data warehouse, a commercial RDBMS, Microsoft SQL Server [9], is employed. Because the system includes the basic functionalities of the data warehouse that contains a multi-dimensional data cube, interactive analysis can be performed efficiently using these built-in tools.

GeoServer [10] is an open source server software for sharing and editing geographic information. Access to the spatial database is made upon request and geographic information stored in the database is extracted as a diagram in vector or raster model. The software also has various functions to address geographic or spatial data.

Analyzers use a web browser as the analysis interface. To display a map on the browser, JavaScript for managing geographic data on the browser and OpenLayers [11], a library of CSS (Cascading Style Sheets), are used. For the management of information other than maps on the browser, the ASP.NET framework of Microsoft is included. This makes queries to the database in the backend and collects the results from the database. It also saves the data results to the database for reference by GeoServer.

4.2. Data Warehouse Functions of Microsoft SQL Server

In this section, the data warehouse functions of Microsoft SQL Server used to realize the prototype system are introduced. The data warehouse functions of SQL Server are called multi-dimensional modeling functions and are included in the SQL Data Server Tools [12]. These basic functions are also provided by other commercial RDBMSs and hence, the implementation approach proposed in the present study can be used for other RDBMSs.

For the development of a multi-dimensional data cube, it is necessary to specify a target database and then develop a **fact table** that presents the collected information and **dimension tables** that present the dimensions for analysis. The fact table is uniquely determined once the target data are specified. Therefore, a major task is to develop the dimension tables considering concept hierarchies. Then, the cube wizard is used to develop a data cube and attributes of each dimension are added or deleted to develop the desired structure of the data cube. Deployment is required to permit an actual analysis using the data cube.

One of the analysis methods with a data cube is to use the OLAP function, which is a standard function of SQL. However, although this function is standardized as SQL/OLAP [13–15], there are only a small number of RDBMSs that implement the extended functions. Therefore, to implement OLAP in the prototype system, MDX [5, 8] is provided as the query language for the SQL Server.

Because queries of this prototype system contain "narrowing-down" processing of data on a map, spatial indexing, a function of the SQL Server, is applied to the data as required. Consequently, a significant improvement of the response speed, in particular in the analysis of narrow areas, can be expected.

5. Development of Simulation Data Warehouse

5.1. Objective Data Sets and System Requirements

For the development of the prototype system, we obtained two sets of simulation data, earthquake and tsunami. One is the flood depth data of a tsunami in Kochi City after the occurrence of an earthquake, provided by Koshimura's group at Tohoku University. The other set of data is the evacuation simulation data of the people flow in the same place, Kochi, and obtained from Sekimoto's group of the University of Tokyo. The sample of

the people flow that we use in the present study was obtained from the simulations of approximately 40 thousand people conducted based on person trip data after the occurrence of the earthquake.

Because there was a difference in the target area and simulation length between the two sets of data, preprocessing and adjustment of the data sets were performed. The flood depth was simulated every 30 seconds on a $3,504 \times 2,364$ grid and three-hour data of the flood depth were recorded starting from the occurrence of the earthquake. The people flow data were those collected for six hours under the conditions where the earthquake occurred at 9:00AM and the peak of the evacuation was 60 minutes after the occurrence of the earthquake. The data were recorded every ten seconds. To address the flood depth data areas, the finest area grid of the data cube was set to $4,096 \times 4,096$. (Hence, some grid cells do not contain data.) Each grid corresponded to a size of 10 m $\times$ 10 m in real space. The coordinate values of the people flow data was easily assigned to each grid cell. The starting time was set to 9:00AM as the people flow started at that time and the ending time was set to three hours after that because the flood depth data were for three hours. The minimum time division was set to ten seconds. Hence, the same flood depth value was repeated three times.

In order to meet the requests for earthquake and tsunami analysis, the following requirements for the data set were applied.

1 Integrated analysis must be permitted for multiple sets of data. In the present study, two simulation data sets (flood depth and evacuee flow) were used. This is an essential requirement because multiple sets of data must be integrated to analyze and forecast damages.

2 Interactive analysis must be permitted using user interfaces including a visualization functionality. In particular, parameters and restriction conditions of the simulation data must be adjustable. For the visualization, a function of zooming in/out on a map is also necessary.

3 To realize interactive analysis, response time must be in an appropriate range, even for large data. In the present implementation of the system, the target response time must be one second or less.

In this study, Requirement 1 is satisfied by developing a data cube on the assumption of an integrated analysis scenario. The Requirement 2 is addressed using software (GeoServer) to display the geographic and spatial data. To meet Requirement 3, a prior arrangement of the data was undertaken for constructing the data cube. This ensures a compact statistical data set and a short response time for the expected basic processing.

5.2. Development of Data Cube

Although many possibilities can be considered for the integrated analysis of the data, the present prototype uses

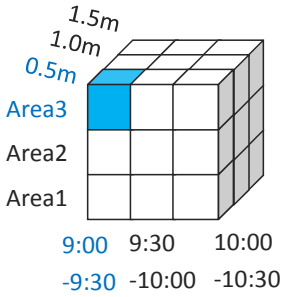


Fig. 4. Data cube for the prototype system.

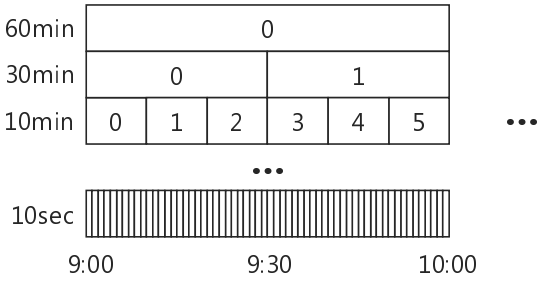


Fig. 5. Concept hierarchy for time dimension.

three dimensions, i.e., time, area, and flood depth of the tsunami, to assume a scenario of analyzing the number of evacuees. **Fig. 4** illustrates the data cube created based on these considerations. Each cell of the data cube can be accessed using a key of the three dimensions, time, area, and flood depth of the tsunami, and each cell contains the number of evacuees as a fact. It should be noted that not all three dimensions are necessary for the analysis. For example, one can perform an analysis using only the time and area dimensions.

In order to extract strategic knowledge from a data cube, it is necessary to view its data at several levels of detail. In our case, an analyst may want to view the distribution of evacuees on the map at a finer granularity or a coarse granularity. Hierarchies of dimensions enable it by defining a sequence of mappings relating different dimension levels.

In the following segment, we explain how the concept hierarchies of each dimension of the data cube were set. Details of the developed database schema are presented in Appendix A. First, for the time dimension, the maximum time period was set to one hour and the minimum time period to ten seconds. Between them, a hierarchy of time periods of 30 minutes, ten minutes, five minutes, one minute, and ten seconds was set. **Fig. 5** is a representation of the hierarchy. At each level of the concept hierarchy, sequence numbers starting from zero are assigned.

For the area dimension, the space was divided into grid cells to which sequence numbers are assigned. In the concept hierarchy, a single cell of the entire dimension is the highest class and the cell is divided to 2×2 to obtain the second level. The cells of each level are further divided until $4,096 \times 4,096 = 2^{12} \times 2^{12}$ cells are obtained. This

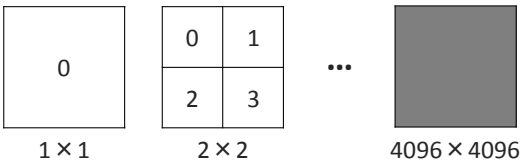


Fig. 6. Concept hierarchy for area dimension.

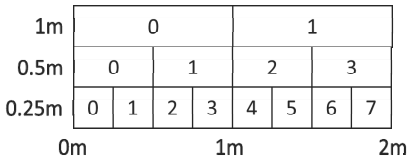


Fig. 7. Concept hierarchy for flood depth dimension.

process creates 13 levels, which are indicated in **Fig. 6**. The cell number in each level is determined by the z-order method [16], which assigns relatively similar numbers to the cells located close to each other in a two-dimensional plane for easy calculation of the numbers.

For the flood depth dimension, a concept hierarchy of three levels was created as displayed in **Fig. 7**. In this study, the levels are defined by the divisions 1 m, 0.5 m, and 0.25 m, respectively.

Based on the above setting of the dimensions, a query in the MDX language setting the level of each dimension can be issued. For example, the query could be "Obtain the total number of evacuees in each cell with a 10 minute division for the time dimension, 16×16 divisions for the area dimension, and 0.5 m interval for the flood depth dimension." One can use a part of the dimensions (e.g., not use the flood depth), set a range to the dimensions (e.g., calculate only from 9:00AM to 10:00AM), or change the aggregate functions (e.g., average, maximum value, or other).

An RDBMS with data warehouse functionality has a function of partial prior data arrangement according to possible patterns of queries in order to accelerate response processing when queried [3, 5]. Further, high-speed query processing can be realized by combining values that are partially calculated in advance.

Regarding data warehouse size, the data cube's main body (fact table) is dominant in size in the data warehouse. Because the time division is ten seconds in the finest division level, and the number of divisions is $3 \times 60 \times 60 \div 10 = 1{,}080$ for three hours. The spatial grid size is $4{,}096 \times 4{,}096$ and the flood depth dimension is divided to eight divisions. Therefore, if the value of a single cell is presented in 4 bytes, we have $4 \times 1{,}080 \times 4{,}096 \times 4{,}096 \times 8$ (bytes) $= 540$ (GB). However, the actual measurement data size in the entire data warehouse was 8.2 GB. This is because not all cells have values and data compression is performed by the SQL Server. In the present implementation, fine divisions were used. However if ten-second divisions or 10 m $\times$ 10 m divisions are not necessary, the data cube can be made more compact.

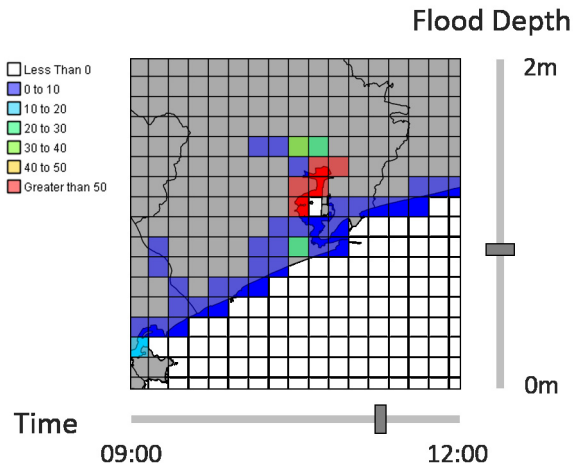


Fig. 8. Image of the user interface.

6. Case Example of Query and Visualization

The user interface (web browser) used for the analysis in the prototype system can easily set parameters and the narrowing-down condition (e.g., start time, end time). An image of the user interface is presented in **Fig. 8**. It is a two-layer image with people flow data in a heat map format, which is the result of the query, presented on an administrative division map in Kochi Prefecture. An approximate number of evacuees in each cell can be found using two slide bar to interactively change the time and flood depth. The area level can be changed using the zoom-in/out function. Every time the designation changes, a new query in MDX language is issued to the backend SQL Server and new total values are calculated and presented. Analyzers can perform analyses using these functions and gradually narrowing-down the areas from a wide map to a narrow map.

Figure 9 is an example of the zoom-in function. In this example, the maximum number of evacuees in each area in the specified time period from the start time to the end time is presented as a heat map. In this case, we assume that the analyzer reviewed the upper map first and was interested in the central portion. The analyzer uses the "+" icon on the upper left to zoom in and a result is presented immediately. The area is made finer by one level and the adjustment is automatically made by the system. In the prototype system, the response time from when the query was issued to when the result was displayed was approximately 0.7 seconds.

There is also another function realized implemented in the prototype. This creates a motion picture by visualizing a temporal change in the number of evacuees in the target area. Presentation with a motion picture is effective for understanding a change in a situation.

The system requirements described in Section 5.1 are examined based on the experiment results. "Integrated analysis must be permitted for multiple sets of data." (Requirement 1) was satisfied by the development of the data cube. It was also verified that the system satisfied "Inter-

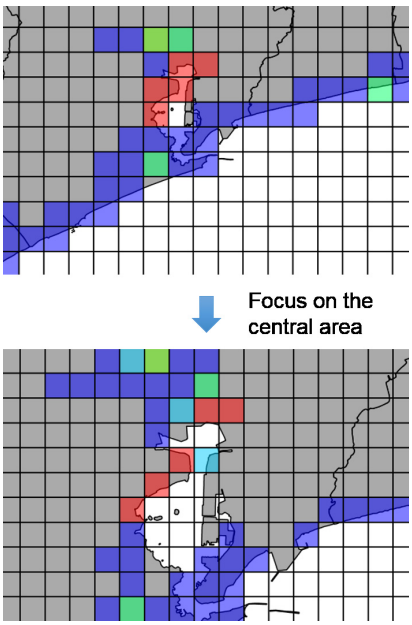


Fig. 9. Zoom-in feature.

active analysis must be permitted using user interfaces including a visualization function." (Requirement 2), which was a relatively simple requirement for the analysis. For "To realize interactive analysis, response time must be in an appropriate range, even for large data." (Requirement 3), a response time less than the preset condition, one second, was achieved. Therefore, it must be considered that the proposed prototype system satisfied all the requirements. Because the developed system was a prototype with incomplete functionality, it was relatively easy to meet these requirements. When functions are added and expanded in the future, the requirements must be reviewed and a verification of the fulfillment of the requirements elaborated.

7. Discussions and Future Work

Through the development of this prototype, issues to be resolved in the future were identified. The followings are some of the problems.

- **Selection of dimensions and facts:** In the business field, it is not difficult to choose dimensions and facts. For example, the dimensions could be "period" and "area" and the fact could be "amount of sales." However, in the analysis of disaster data, flood depth could be used as dimension as in this paper or as fact to issue the query "Visualize flood depth data under a specified condition." Because the combination of dimensions and facts is not fixed, a flexible setting of the dimensions and facts must be possible in a simulation data warehouse. Further, there could be a request for visualizing damage situations using certain damage indices that are determined from a combination of the number of evacuees and flood depth. In this case, a function of incorporating a user-defined function is required.
- **Conceptual hierarchy:** In the present implementation, a concept hierarchy was set as described in Section 5.2. However, a method to construct a concept hierarchy when other types of simulation data are used requires future study. In a spatio-temporal simulation, space and time are the major dimensions and their meaning is clear. It is, therefore, not difficult to develop a static concept hierarchy in advance. For the dimensions of measurement values and indices, such as flood depth dimension in this paper, the setting of a concept hierarchy itself could be a function of the analysis. Hence, it could be necessary to specify a part of the concept hierarchy dynamically when the analysis is made. Because this increases the query processing cost, the development of an implementation technique would be necessary.
- **Handling multiple parameters:** This was not a problem in the present prototype because the data set was fixed. However, in general, because simulations use many parameters, the method of addressing them is a problem. For example, suppose a request "Want to know the change of flood depth at site p when the seismic center gradually moves from x to y." In this case, multiple sets of simulation data created with different seismic center parameters must be integrated and used. A method to accomplish this could be a problem within the framework of the data warehouse.
- **System architecture:** For the simulation data warehouse, we used a commercial RDBMS and utilized the data warehouse functions and spatial database functions that it provided. This policy is effective for the analysis presented in this paper and realizes instantaneous responses. However, it is not sufficient for the above-mentioned combination of flexible dimensions and facts. Google BigQuery [17] or others that use parallel processing on many machines to realize a real-time response without preprocessing could present a computer resource or cost problem. Dynamic data cube query processing remains under study [18]. A database management system (DBMS) specialized for the use of large array data, such as SciDB [19], could be considered.
- **Flexible visualization functionality:** As discussed previously, a method for visualizing an answer to a query in a form suitable for analysis is an important factor of realizing interactive analysis. In the prototype system, a simple interface was developed for each type of analysis. However, if the number of analysis types or query types is considerable, the cost would be excessive to develop interfaces for each one. Therefore, the development of a flexible visualization technique is required. As an alternative, an approach of embedding DBMS as a component of the visualization system [20] could be considered.

- **Realtime response for interactive analysis:** In the experiment, the response time from when the query was issued to when the result was returned was approximately 0.7 seconds, which was sufficient to realize interactive analysis. In the case of a longer simulation time, the data cube domain that the system accesses for visualization processing does not change significantly, and hence, a similar response time could be expected. In the case of larger people flow data, the data are arranged when the data cube is constructed and hence, there is no influence on the response time. More dynamic and advanced queries, however, could be influenced by an increase of the amount of data. An improvement and expansion of queries for assisting data exploration and development of efficient processing techniques are required.
- **Advanced analysis functionality:** To consider a more advanced analysis function, let us use the query example of "Identify all seismic centers having a flood depth more than 1 m at site x with the parameters other than the seismic center being fixed to specified values." It is possible to answer this query because the data necessary required for the answer exist in the data warehouse. However, the problem is how to express the query and how to process it efficiently. A query language that can describe such a query in a simple manner could be necessary.
- **Cooperation with other systems** Some analyses could require cooperation with other systems. An example is cooperation with visualization processing. For some queries, advanced visualization is required for presenting the analysis process or analysis result. In these cases, coordination with a dedicated visualization system would be a practical solution. Further, if advanced statistical processing were required for a part of the target data, coordination with a statistical processing such as R would be necessary. To incorporate domain-dependent analysis processing into a system, a function of incorporating into the system and executing a user-defined function provided by a user would be necessary.

8. Related Studies

Studies on data warehousing and OLAP have been primarily in the business field. However, data warehousing has also been developed for spatio-temporal data closely related to the present study. Explanations and surveys can be found in chapter 11 of [5] or in [21]. The method of developing a concept hierarchy using the inclusion relation between spatial areas, used in the present study, was also employed in conventional spatial data warehouses. A spatial database technology was developed and has been realized as an implementation technique for the efficient utilization of spatial index [22, 23]. Further, efforts to expand a data model [24] to incorporate the semantics of a geographical space or to introduce a more flexible space division method [25] have been made.

The majority of the studies on spatial data warehouses use geographical data and aim to analyze statistical information (e.g., population distribution). In addition to the spatial information, time information may also be utilized. From this perspective, the data warehousing of movement tracking data such as in [26] is technically closely related to the present study as it integrates and arranges continuous spatio-temporal information. One of the characteristics of simulations is that they use not only spatial information but also time information. Further, it is sometimes difficult to determine whether a certain kind of information, such as flood depth in this paper, should be treated as a dimension or fact. Because various experimental parameters and conditions are set in simulations, how to analyze in an integrated manner the simulation experiments conducted with different parameters could be a problem. Previous conventional studies did not encounter this issue.

The present study aimed to support exploratory analysis. For the exploration of data cubes with data warehouse and OLAP, studies have been made primarily in the business field [27–29]. An example of the studies was to identify an exception from a data cube to support a user's data exploration. In [27], indices of exceptions are calculated in advance to guide analyzers for data exploration. In a different approach [28], interestingness is calculated based on the context of the user. In [29], advanced operators were proposed for user's data exploration. The aim of the operators was to determine an exception, as in previous studies, without excessive manual exploration.

Unlike these studies that attempted to identify "exceptions" defined by considering general business situations, our study focused on earthquake and tsunami simulation data on the assumption that users would be interested in indices, such as intensity of damage, which were dependent on the target domain. In the development, we addressed the available content of the simulation data and focused on a combination of the flood depth and the number of evacuees, in particular on their visualization. In future, we would like to introduce more advanced indices, in addition to the above, to assist the user's data exploration for earthquake and tsunami damage analyses.

In the present system development, the implementation is realized by utilizing the advanced data warehouse function of Microsoft SQL Server. However, it is expected that dynamic elements could increase in the future as stated in the previous section. In the data cube, computational acceleration is realized by preprocessing and the prior arrangement of the data. Another option is to use the hybrid architecture of an array DBMS and processing system that has been developed in recent years. Array DBMS systems such as SciDB [19, 30] can efficiently manage extensive array data and are suitable for storage and query of data provided on a grid, such as flood depth data.

In this study, the simulation data were visualized. There are many studies on visualization systems. Although a commercial system is available, processing that closely cooperating with database and data warehouse needs to be studied more. For example, [31] used the

MapReduce technology to develop an efficient visualization system of spatio-temporal satellite data.

9. Summary

In this paper, we demonstrated the concept of simulation data warehouse, architecture of a prototype system, and case example of an interactive analysis of disaster simulation data. As mentioned in Section 7, there remain several problems with this study. We would like to not only develop a new system technique but also accelerate joint studies with specialists of earthquake and tsunami analyses, use other simulation data, analyze the requests for integrated analysis, and accumulate usage cases.

Acknowledgements

The authors would like to thank Koshimura's group of Tohoku University and Sekimoto's group of the University of Tokyo for providing the data sets. Part of this study was supported by the CREST – "Creation of innovative basis of big data analysis for earthquake and tsunami disaster reduction by coordination and data integration among large high-resolution numerical simulations" – project commissioned by the Ministry of Education, Culture, Sports, Science and Technology, "DIAS-P," and Grant-in-Aid for Scientific Research (25280039).

References:

[1] T. Hey, S. Tansley, and K. Tolle (Eds.), "The Fourth Paradigm: Data-Intensive Scientific Discovery," Microsoft Research, 2009.

[2] S. Hayashi and S. Koshimura, "The 2011 Tohoku tsunami flow velocity estimation by the aerial video analysis and numerical modeling," Journal of Disaster Research, Vol.8, No.4, pp. 561–572, 2013.

[3] J. Han, M. Kamber, and J. Pei, "Data Mining: Concepts and Techniques," Morgan Kaufmann, 3rd edition, 2011.

[4] W. H. Inmon, "Building the Data Warehouse," John Wiley & Sons, 3rd edition, 2002.

[5] A. Vaisman and E. Zimányi, "Data Warehouse Systems: Design and Implementation," Springer, 2014.

[6] E. F. Codd, S. B. Codd, and C. T. Smalley, "Providing OLAP to user-analysis: An IT mandate," E.F. Codd and Associates, 1993, http://www.minet.uni-jena.de/dbis/lehre/ss2005/sem_dwh/lit/Cod93.pdf [accessed December 21, 2015]

[7] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatarao, F. Pellow, and H. Pirahesh, "Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals," Data Mining and Knowledge Discovery, Vol.1, No.1, pp. 29–53, 1997.

[8] MultiDimensional eXpressions, http://en.wikipedia.org/wiki/MultiDimensional_eXpressions [accessed December 21, 2015]

[9] Microsoft SQL Server, http://www.microsoft.com/en-us/sqlserver [accessed December 21, 2015]

[10] GeoServer, http://geoserver.org/ [accessed December 21, 2015]

[11] OpenLayers 3, http://openlayers.org/ [accessed December 21, 2015]

[12] Multidimensional modeling (Adventure Works tutorial), https://msdn.microsoft.com/en-us/library/ms170208(v=sql.120).aspx [accessed December 21, 2015]

[13] A. Eisenberg and J. Melton, "SQL standardization: The next steps," ACM SIGMOD Record, Vol.29, No.1, pp. 63–67, Mar. 2000.

[14] A. Eisenberg, K. Kulkarni, J. Melton, J.-E. Michels, and F. Zemke, "SQL:2003 has been published," ACM SIGMOD Record, Vol.33, No.1, pp. 119–126, Mar. 2004.

[15] J. Melton, "Advanced SQL:1999 – Understanding Object-Relational and Otehr Advanced Features," Morgan Kaufmann, 2003.

[16] H. Samet, "Object-based and image-based object representations," ACM Computing Surveys, Vol.36, No.2, pp. 159–217, June 2004.

[17] Google BigQuery, https://cloud.google.com/bigquery/?hl=en [accessed December 21, 2015]

[18] N. Kamat, P. Jayachandran, K. Tunga, and A. Nandi, "Distributed and interactive cube exploration," in Proc. ICDE, pp. 472–483, 2014.

[19] M. Stonebraker, P. Brown, D. Zhang, and J. Becla, "SciDB: A database management system for applications with complex analytics," IEEE Computational Science & Engineering, Vol.15, No.3, pp. 54–62, 2013.

[20] A. Aiken, J. Chen, M. Stonebraker, and A. Woodruff, "Tioga-2: A direct manipulation database visualization environment," In Proc. ICDE, pp. 208–217, 1996.

[21] L. Gómez, B. Kuijpers, and B. Moelans, "A survey of spatio-temporal data warehousing," International Journal of Data Warehousing and Mining, Vol.5, No.3, pp. 28–55, 2009.

[22] D. Papadias, P. Kalnis, J. Zhang, and Y. Tao, "Efficient OLAP operations in spatial data warehouses," In Proc. SSTD, pp. 443–459, 2001.

[23] D. Papadias, Y. Tao, P. Kalnis, and J. Zhang, "Indexing spatio-temporal data warehouses," In Proc. ICDE, pp. 166–175, 2002.

[24] L. Gómez, B. Kuijpers, and A. Vaisman, "A data model and query language for spatio-temporal decision support," GeoInformatica, Vol.15, No.3, pp. 455–496, 2011.

[25] S. I. Gómez, L. A. Gómez, and A. A. Vaisman, "A generic data model and query language for spatiotemporal OLAP cube analysis," in Proc. EDBT, 2012.

[26] L. Leonardi, G. Marketos, E. Frentzos, N. Giatrakos, S. Orlando, N. Pelekis, A. Raffaetà, A. Roncato, C. Silvestri, and Y. Theodoridis, "T-warehouse: Visual OLAP analysis on trajectory data," In Proc. ICDE, pp. 1141–1144, 2010.

[27] S. Sarawagi, R. Agrawal, and N. Megiddo, "Discovery-driven exploration of OLAP data cubes," in Proc. EDBT, pp. 168–182, 1998.

[28] S. Sawaragi, "User-adaptive exploration of multidimensional data," in Proc. VLDB, pp. 307–316, 2000.

[29] S. Sarawagi and G. Sathe, "I^3: Intelligent, interactive investigation of OLAP data cubes," in Proc. ACM SIGMOD, 2000.

[30] M. Stonebraker, P. Brown, A. Poliakov, and S. Raman, "The architecture of SciDB," in Proc. SSDBM, pp. 1–16, 2011.

[31] A. Eldawy, M. F. Mokbel, S. Al-Harthi, A. Alzaidy, K. Tarek, and S. Ghani, "SHAHED: A MapReduce-based system for querying and visualizing spatio-temporal satellite data," in Proc. ICDE, pp. 1585–1596, 2015.

Appendix A. Schema of the Simulation Data Warehouse

Figure 10 presents the schema description of the simulation data warehouse developed in the present prototype. `RecordID` is a major key ID uniquely assigned. `EvacuationRecordTable` corresponds to the fact table. `TimeKey`, `PlaceKey`, `DepthKey` present foreign keys corresponding to the time, area, and flood depth dimensions, respectively. `Number` is the number of evacuees in each cell. That is, this table provides the number of evacuees from the viewpoint of time, area, and flood depth.

`TimeTable` is the dimension table. `TimeKey` is a major key ID. `hour`, `min30`, `min10`, `min5`, `min`, and `sec10` contain the sequence numbers at their respective level of the concept hierarchy.

`PlaceTable` and `DepthTable` are both dimension tables. Their details are omitted here.

```
CREATE TABLE TimeTable(
  TimeKey int NOT NULL PRIMARY KEY,
  hour int,
  min30 int,
  min10 int,
  min5 int,
  min int,
  sec10 int
)

CREATE TABLE PlaceTable(
  PlaceKey int NOT NULL PRIMARY KEY,
  AreaID1 int,
  AreaID2 int,
  AreaID4 int,
  AreaID8 int,
  AreaID16 int,
  AreaID32 int,
  AreaID64 int,
  AreaID128 int,
  AreaID256 int,
  AreaID512 int,
  AreaID1024 int,
  AreaID2048 int,
  AreaID4096 int,
)

CREATE TABLE DepthTable(
  DepthKey int NOT NULL PRIMARY KEY,
  cm100 int,
  cm50 int,
  cm25 int
)

CREATE TABLE EvacuationRecordTable(
  RecordID int NOT NULL PRIMARY KEY,
  TimeKey int FOREIGN KEY REFERENCES TimeTable(TimeKey),
  PlaceKey int FOREIGN KEY
    REFERENCES PlaceTable(PlaceKey),
  DepthKey int FOREIGN KEY
    REFERENCES DepthTable(DepthKey),
  Number int
)
```

Fig. 10. Schema of simulation data warehouse.

Name:
Jing Zhao

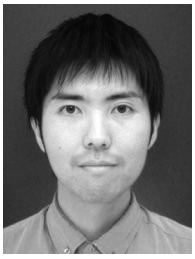
Affiliation:
Ph.D. Candidate, Graduate School of Information Science, Nagoya University

Address:
Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan

Brief Career:
2013- Master Student, Graduate School of Information Science, Nagoya University
2015-Ph.D. Candidate, Graduate School of Information Science, Nagoya University

Selected Publications:
• "A Density-based Approach for Mining Movement Patterns from Semantic Trajectories," The IEEE Region 10 Conference (TENCON 2015), November 2015.

Academic Societies & Scientific Organizations:
• Database Society of Japan (DBSJ)



Name:
Kento Sugiura

Affiliation:
Ph.D. Candidate, Graduate School of Information Science, Nagoya University

Address:
Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan

Brief Career:
2013- Master Student, Graduate School of Information Science, Nagoya University
2015- Ph.D. Candidate, Graduate School of Information Science, Nagoya University

Selected Publications:
• "Grouping Methods for Pattern Matching in Probabilistic Data Streams," The 20th International Conference on Database Systems for Advanced Applications (DASFAA 2015), pp. 92-107, April 2015.

Academic Societies & Scientific Organizations:
• Information Processing Society of Japan (IPSJ)
• Database Society of Japan (DBSJ)

Name:
Yuanyuan Wang

Affiliation:
Research Associate, Graduate School of Information Science, Nagoya University

Address:
Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan

Brief Career:
2014- Researcher, Faculty of Computer Science and Engineering, Kyoto Sangyo University
2015- Research Associate, Graduate School of Information Science, Nagoya University

Selected Publications:
• "Will Mail-Based Disaster Information Systems Work Well?: A Case Study in Japan," Proceedings of the 7th International Conference on Intelligent Interactive Multimedia Systems and Services (KES IIMSS 2014), pp. 293-303, 2014.

Academic Societies & Scientific Organizations:
• Information Processing Society of Japan (IPSJ)
• Database Society of Japan (DBSJ)

Name:
Yoshiharu Ishikawa

Affiliation:
Professor, Graduate School of Information Science, Nagoya University

Address:
Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan

Brief Career:
1994- Assistant Professor, Nara Institute of Science and Technology
1999- Assistant Professor, University of Tsukuba
2003- Associate Professor, University of Tsukuba
2006- Professor, Nagoya University

Selected Publications:
• "Probabilistic Range Querying over Gaussian Objects," IEICE Transactions on Information and Systems, Vol.E97-D, No.4, pp. 694-704, April 2014.

Academic Societies & Scientific Organizations:
• Association for Computing Machinery (ACM)
• IEEE Computer Society
• Information Processing Society of Japan (IPSJ)
• Institute of Electronics, Information and Telecommunication Engineers (IEICE)
• Database Society of Japan (DBSJ)