Chapter 40
SEDRIS Transmittal Storing and Retrieval System Using Relational Databases

Yongkwon Kim
Korea Advanced Institute of Science and Technology (KAIST), Korea

Heejung Yang
Korea Advanced Institute of Science and Technology (KAIST), Korea

Chin-Wan Chung
Korea Advanced Institute of Science and Technology (KAIST), Korea & Chongqing University of Technology (CQUT), China

ABSTRACT

Modeling and simulation (M&S) are widely used for design, analysis, and optimization of complex systems and natural phenomena in various areas such as the defense industry and the weather system. In many cases, the environment is a key part of complex systems and natural phenomena. It includes physical aspects of the real world which provide the context for a specific simulation. Recently, several simulation systems are integrated to work together when they have needs for exchanging information. Interoperability of heterogeneous simulations depends heavily on sharing complex environmental data in a consistent and complete manner. SEDRIS (Synthetic Environmental Data Representation and Interchange Specification) is an ISO standard for representation and interchange of environmental data and widely adopted in M&S area. As the size of the simulation increases, the size of the environmental data which should be exchanged between simulations increases. Therefore, an efficient management of the environmental data is very important. In this paper, the authors propose storing and retrieval methods of SEDRIS transmittals using a relational database system in order to be able to retrieve data efficiently in the environmental data server cooperating with many heterogeneous distributed simulations. By analyzing the structure and the content of SEDRIS transmittals, relational database schemas are designed. To reduce query processing time of SEDRIS transmittals, direct storing and retrieval methods which do not require the type conversion of SEDRIS transmittals are proposed. Experimental analyses are conducted to show the efficiency of the proposed approach. The results confirm that the proposed approach greatly reduces the storing time and retrieval time compared to comparison approaches.

DOI: 10.4018/978-1-4666-9845-1.ch040

1. INTRODUCTION

With the rapid development of computing technologies, modeling and simulation (M&S) has become an integral part of the modern research and development process (Günal & Pidd, 2011; Stopford & Counsell, 2008). A common representation of the physical environment is a critical element in M&S and is a necessary precondition for the interoperability of heterogeneous simulations (Sedris, 1994). Recently, integration of heterogeneous simulations increases to work together with different simulations. Especially, in the defense industry, integration of simulations is widely used. For example, flight simulations for training several pilots and a simulation for command-level training where officers are trained to manage complex situations and command thousands of simulated participants can be integrated (Moller, 2012). Interoperability of heterogeneous simulations depends heavily on sharing complex environmental data in a consistent and complete manner (Ryu, 2009).

The environmental data of distributed simulations is managed by a separate environmental data server that is connected in a network. The environmental data server stores the environmental data before other simulations are conducted, and the environmental data in the environmental data server is not changed after the data is stored. The stored data is utilized as the reference data of a specific environmental condition. A simulation in a network sends a request to the environmental data server in order to get some environmental data of a specific location with a specific condition. Between the environmental data server and other simulations, the fast data retrieval is essential. If the data retrieval is not fast enough, receiving the environmental data can become a bottleneck of simulations. Thus, for the environmental data server, the performance of the storing side is not important as compared with that of the retrieval side.

The *Synthetic Environmental Data Representation and Interchange Specification* (SEDRIS) is an ISO standard for representation and interchange of environmental data such as ocean, terrain, atmosphere, and space, etc. SEDRIS is based on five core technology components which are the SEDRIS Data Representation Model (DRM), the Environmental Data Coding Specification (EDCS), the Spatial Reference Model (SRM), the SEDRIS interface specification (SEDRIS API), and the SEDRIS Transmittal Format (STF). The DRM, the EDCS, and the SRM are used to achieve an unambiguous representation of environmental data. The SEDRIS API and the STF allow an efficient sharing and interchange of the environmental data represented by the other three components.

The STF is a conceptual file format that is developed to store SEDRIS transmittals. A SEDRIS transmittal is composed of one or more files containing environmental data. The STF provides a platform-independent interchange mechanism for SEDRIS transmittals. As the simulation becomes more complex, the size of the STF file increases. For example, an infantry simulation requires only atmospheric data on the ground. However, if a fight simulation cooperates with the infantry simulation, the STF file for these simulations should include atmospheric data in the sky over infantry as well as environmental data on the ground. As the size of an STF file about the environmental data increases, finding the environmental data with a specific condition becomes more difficult because there is no way to divide an existing STF file into many sub-files. A huge STF file should be accessed although a simulation in a network wants just one value. For example, assume that there is a tank simulation and it wants the temperature, the air pressure, and the humidity of the simulated tank's current location in the rain. The location of the simulated tank changes as the simulated tank moves, and its moving direction cannot be known to the environmental data server. Thus, the environmental data server should send the atmospheric data corresponding to the current location of the simulated tank. The atmospheric data of some region in the rain is usually stored in an STF file together because it is impossible to store the atmospheric data of every location in separate STF files, and the entire STF file should be scanned to find a value whenever

Related Content