

Moving-Object management method of combat system using main memory DBMS

Método de gestión de objeto en movimiento de un Sistema de combate usando la memoria principal DBMS

Jongsu Hwang ¹

Abstract

The naval combat management system conducts a comprehensive analysis of obtained information and integrates it with on-board sensors, weapons, and other equipment. Furthermore, it automatically performs all processes such as engagement plans, weapon assignments, and target detection using already pre-deployed database information. Thus, it is important to manage moving objects from many different kinds of sensors. In this paper, we will introduce the Moving-Object management method of combat systems using the In Moving Object DBMS. This is based on the main memory database technology provided by the high-speed transaction processing performance.

Key words: MO DBMS, Moving Object, Combat system, Track management.

Resumen

La gestión del sistema de combate naval conduce a un análisis comprensivo de información obtenida y se integra con sensores a bordo, armas y otros equipos. Además, ejecuta automáticamente rosos los procesos tales como planes de combate, asignaciones de armas y detección del objetivo usando información de bases de datos pre-desplegadas. Así, es importante gestionar los objetos en movimiento de diferentes tipos de sensores. En este artículo, presentaremos el Método de gestión de objeto en movimiento de un Sistema de combate usando la memoria principal DBMS. Esto es con base en la memoria principal de la tecnología de base de datos proveída por el desempeño de procesamiento de transacción de alta velocidad.

Palabras claves: MO DBMS, Objeto en movimiento, sistema de combate, gestión de rastreo.

Date Received: October 16th 2014 - *Fecha de recepción:* Octubre 16 de 2014

Date Accepted: December 15th 2014 - *Fecha de aceptación:* Diciembre 15 de 2014

¹ Research engineer. Lignex. Seoul, Korea. Email: jongsu.hwang@lignex1.com

Introduction

The naval combat management system conducts a comprehensive analysis of obtained information and integrates it with on-board sensors, weapons, and other equipment. Furthermore, it automatically performs all processes such as engagement plans, weapon assignments, and target detection using already pre-deployed database information.

Most combat management systems are using the database to manage obtained sensors, weapons data and information needed to command. But due to the data management system performance limitations, combat management utilizes only stores management. And the data management system can't process real-time data. In addition, there is difficulty with the sharing and integration of data because the data management system stores non-real-time data through the file system. So, it has been required quickly access the database and reduce network overload between the database and the applications.

This paper will propose a method for the moving object database used by the target fusion of combat management systems. It uses the moving object database and provides high speed transaction processing performance and storage features of moving object location in real time. It also processes a feature of spatio-temporal query. The paper is organized as follows. Section 2 describes the track management features. Section 3 presents

the main-memory database management system. Section 4 describes the moving objects database features. Section 5 introduces the Kairos moving object database. Section 6 details our application method. Section 7 concludes the paper with directions for future work.

Track Management

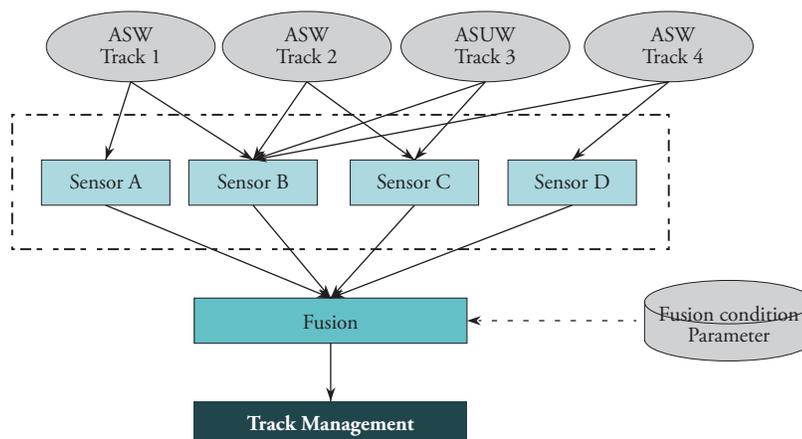
The track manager is an improved component within combat systems that receives and translates information from air, surface and subsurface sensors to create an integrated picture of the locations and paths of aircraft, ships and submarines in a battle space area.

The process of combining data has been called sensor correlation and fusion, or simply data fusion. [1]Data fusion is a multilevel, multifaceted process dealing with the registration, detection, association, correlation, and combination of data and information from multiple sources to achieve a refined state and identity estimation, and complete and timely assessments of the situation.

Sensors produce individual observations or measurements (source track) that must be placed in proper context first to create organized data sets, and then evaluated to infer higher-level meaning about the overall content for the information.

The process of fusion which partitions data into associated categories includes correlation and

Fig. 1. Track Fusion.



association stages. In a typical problem, the data from different sensors are partitioned to associate all measurements from common targets into individual target categories. Sensor measurements are compared with correlation metrics that use temporal, spectral or spatial properties to score each alternative assignment hypothesis. The sensor data is then associated with the corresponding data from other sensors and is assigned to categories.

[2]Track Management has been developed by open architecture. It will potentially be installed on many surface and air platforms. OA (Open Architecture) framework will assist the Navy with integrating a single, cost-effective track manager that can be easily maintained and upgraded to meet emerging customer requirements.

In this paper, we put emphasis on the OA framework track management using the COTS (commercial on-the-shelf) database.

Main memory Database management system

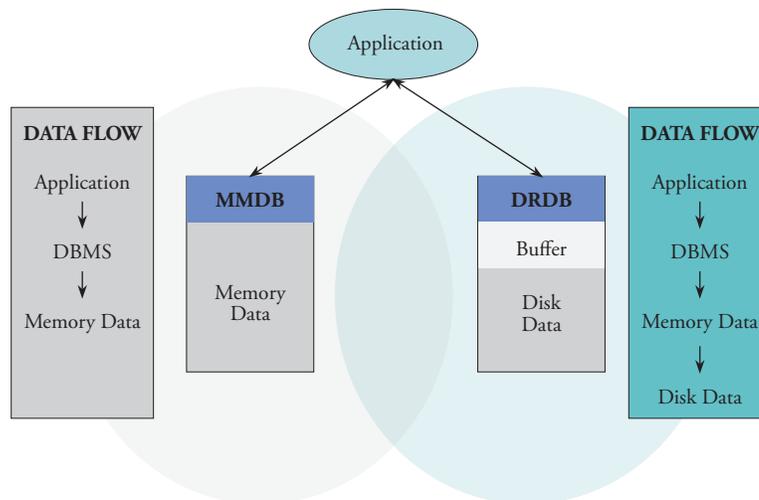
[3]In a main memory database system (MMDB) data resides permanently in the physical memory; in a conventional database system (DRDB) it is disk resident. In a DRDB, disk data may be cached into memory for access; in a MMDB the memory resident data may have a backup copy

on disk. So in both cases, a given object can have copies both in memory and on disk. The key difference is that with the MMDB, the primary copy lives permanently in the memory. This is because data can be accessed directly through the memory, MMDBs can provide fast response times and transaction throughputs, when compared to DRDBs This is especially important for real-time applications where transactions have to be completed by their specified deadlines.

A computer's main memory clearly has different properties from that of magnetic disks, and these differences have profound implications on the design and performance of the database system. Although these differences are well known, it is worthwhile reviewing them briefly.

- A) The access time for the main memory is magnitudes less than for the disk storage access.
- B) The main memory is normally volatile, while disk storage is not. However, it is possible (at some cost) to construct a nonvolatile main memory.
- C) Disks have a high, fixed cost per access that does not depend on the amount of data that is retrieved during the process. For this reason, disks are block-oriented storage devices. The main memory is not block oriented.
- D) The layout of data on a disk is much more critical than the layout of data in the main

Fig. 2. MMDB and DRDB.



memory, since sequential access to a disk is faster than random access. Sequential access is not as important in main memories.

E) The main memory is normally directly accessible by the processors, while disks are not. This may make data in main memory more vulnerable than disk resident data to software errors.

Table 1. Compare DRDB and MMDM.

	MMDM	DRDB
Data model	Relational	Relational
System Structure	Client/Server and inner DB	Client/Server
Server Structure	Multi-Thread	Multi-Thread / Multi-Process
CPU Use Rate	lower CPU use rate using simple search Algorithm	Higher CPU use rate using complicate search Algorithm
DISK I/O	Minimum disk I/O in order to Recovery	Normal disk I/O in order to Select, Insert, Update, Delete

These differences have effects on almost every aspect of database management, from concurrency control to application interfaces.

Moving Object Data management system

[4]The moving object databases stores all kinds of data related to a moving object, such as location, direction, and other temporal data. Therefore, this kind of database considers how real world phenomena represented in the database is positioned in a given spatial and temporal framework, and requires defining new data types and operations. The moving object database utilizes not only spatial database concepts, but also temporal database concepts.

The major motivation for studying spatial databases is for the support of geographic information systems. Early GIS made only limited use of the DBMS technology. However, commercial spatial

databases offer spatial extensions now. Hence, it can be stored all the data of a digital map in the database. All diverse entities can be modeled by three fundamental abstractions which are point, line and region.

On the other hand, the goal of the temporal database research has been to integrate temporal concepts deeply into the DBMS data model and query language and to extend the system accordingly to achieve efficient execution.

In most of the applications, it is possible to visualize the world as a digital map, with altitude and other data. New indexing techniques are implemented on spatial and temporal data in order to efficiently query this kind of data. Thus, R-tree indexing and its derivatives are used in moving object databases. Moving object databases have fully been implemented. For this paper, we used a Kairos MO database.

The Kairos MO is a main memory database that provides high-speed transaction processing performance. In addition to high performance, the Kairos MO database provides a storage feature for moving object location in real time, and processing feature of spatio-temporal query.

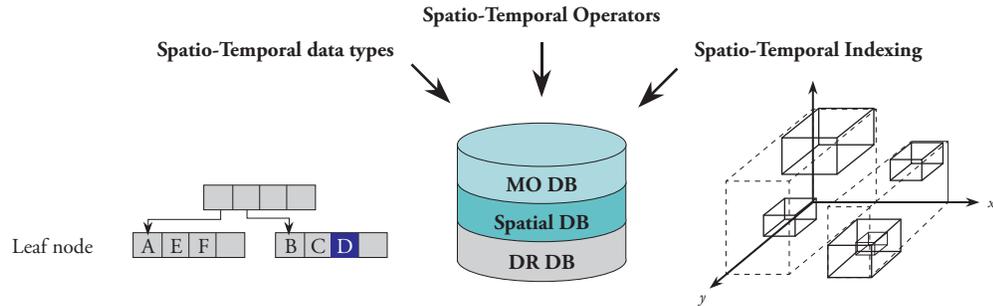
The Kairos MO database is based on the Kairos spatial database.

In order to express moving object, it provides spatio-temporal data type, spatio-temporal operators, an index for spatio-temporal data, and processing for uncertainty. Thus, it supports a function for the processing of moving objects which change their positions or shapes constantly over time. It also supports the Kairos Spatial object processing function.

Kairos moving object database system overview

[5]The usage of the Kairos moving object database in this paper, gives features regarding about data type, operator, and index.

Fig. 3. Kairos moving object database overview.



Spatio-Temporal Data Type

The Kairos MO database supports three type data types: temporal data type, spatial data type, and moving object type. The Moving object type comprises an MBase type for representing value which changes over time, and MGeometry for representing moving objects which change their positions or shapes over time.

Spatio-Temporal Operator

The Kairos MO database supports various spatial-temporal operators: moving object creation and updates, attribute information analysis and extraction, as well as the phase relationship between each spatio-temporal object.

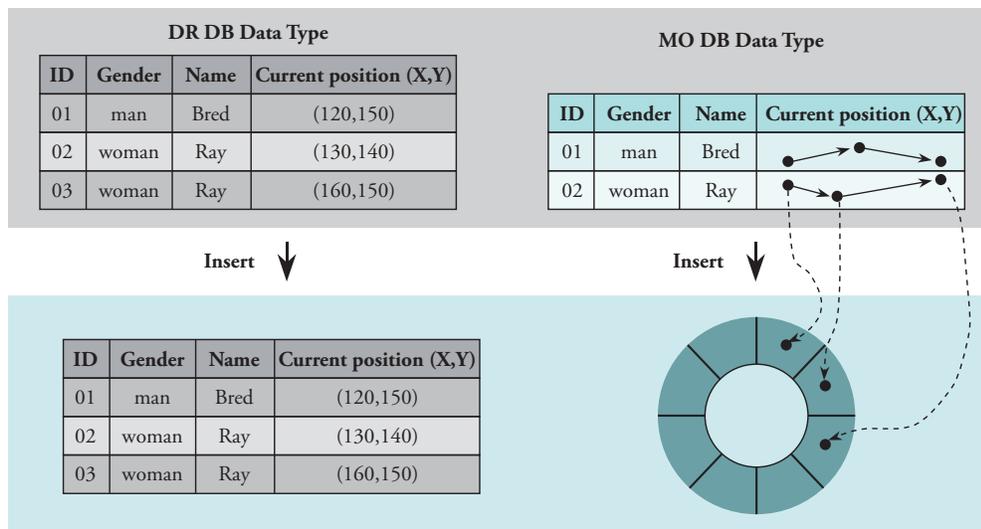
Table 1. Compare DRDB and MMDB.

Operator	Description
Spatio- Temporal Relation Operators	Analysis spatial phase relationship between moving spatial object-moving spatial object or moving spatial object-spatial object.
Set Operators	One moving spatial object or moving spatial objects.
Trajectory Relation Operators	Analysis phase relationship of trajectory of moving object.
Temporal Relation Operators	Analysis time phase relationship between temporal object – temporal object.

Spatio-Temporal Index

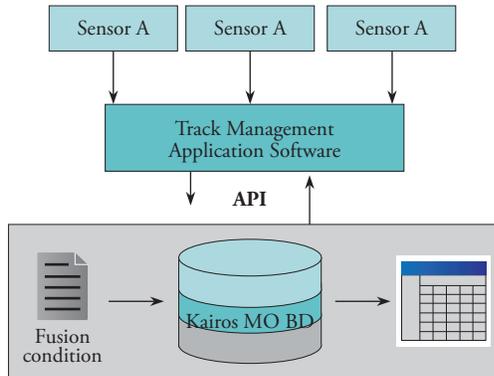
The Kairos MO database supports the spatio-temporal index of a 3D R* Tree structure. It uses minimum bounding rectangle information

Fig. 4. Moving object DB data type.



for high spatio-temporal query process performance.

Fig. 5. Database Track Management Sequence.

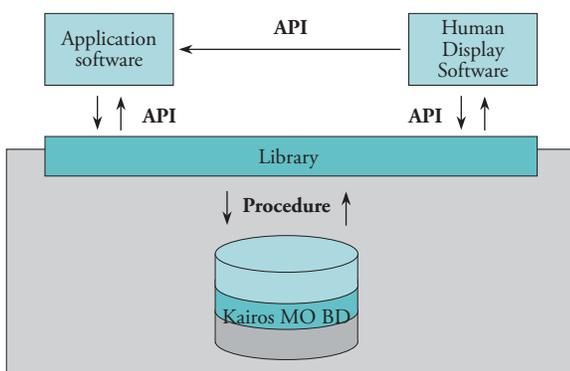


Database track management sequence is as below

- A. Create table with a reference to parameter for the target fusion’s available condition.
- B. Database API (application program interface) calls using triggers when sensor track information is input.
- C. Determine fusion availability according to the target’s location using the moving object database function.
- D. Insert results into the fusion table.
- E. Return the fusion table value from the database.

It can store, update, create track using moving object database, by the process of using a track management sequence. It can also fuse using the moving object database function.

Fig. 6. Database Track Management API.



If you look at the API interface, application software and human display software are connected with the database track management by library. It is processed through about 100 API calls. The application software obtains a fast result and response using the database procedure.

Conclusion and future

In this work, the usage of the moving object databases, which have different capabilities than traditional databases, is defined for track management. This paper proposes a method that the moving object database be used with target fusion.

The use of a commercial on-the-shelf product guarantees reliability and usability. Furthermore, it has a cost- effective track manager that can be easily maintained and upgraded to meet emerging customer requirements.

In the future, databases can replace the track-management function of combat management systems. As a common module, it can also be applied to various combat management systems.

References

- [1] YAAKOV BAR-SHALOM, PETER WILLET, and XIN TIAN, “Tracking and Data Fusion: Handbook of Algorithms”, 2011.
- [2] GENERAL DYNAMICS ADVANCED INFORMATION SYSTEMS , [web on line]. Available at : <http://www.gd-ais.com>
- [3] GARCIA-MOLINA, KENNETH SALEM, "Main Memory Database Systems: An Overview", 1992.
- [4] PERIHAN KILIMCI, OYA KALIPSIZ, "Moving Objects Databases In Space Applications", 2007.

- [5] MARTIN ERWIG, MARKUS SCHENEIDER, "*Spatio-Temporal Predicates*", IEEE Transactions on Knowledge and Data Engineering, Volume 14 Issue 4, July 2002. Page 881-901