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

LIG Nex1 Jongsu Hwang
Content

1. Naval Combat System
2. Track Management System
3. Main Memory Database
4. Moving-Object Database
5. J-Track Management
6. Performance
1. Combat System

System for command and control, fire control, and command support by integrating the available sensors, weapon, navigation and communication systems.
## 2. Combat System Architecture

### Application S/W

<table>
<thead>
<tr>
<th>CMS Interface Layer</th>
<th>HMI (Human-Machine Interface)</th>
<th>MMU (Machine-Machine Interface)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tactical Operation HMI (Surface Ship/Submarine) CSCI</td>
<td>Sensor Interface Server (Surveillance Radar, Tracking Radar, EOTS, IFF, Navi-Radar) CSCI</td>
<td>Weapon Interface Server (Chaff, Gun, SAAM, Light-weight torpedo, TACM, ...) CSCI</td>
</tr>
<tr>
<td>System Management HMI CSCI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simulation Control HMI CSCI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tactical Situation Display HMI CSCI</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### CMS Application Layer

<table>
<thead>
<tr>
<th>Mission Plan Management CSCI</th>
<th>Track Management CSCI</th>
<th>TEWA CSCI</th>
<th>Engagement Control CSCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Motion Analysis CSCI</td>
<td></td>
<td></td>
<td>Fire Control (Gun, Missile, Engagement CSCI)</td>
</tr>
<tr>
<td>Tactical Data Management CSCI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Analysis CSCI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simulation CSCI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status Monitoring CSCI</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### CMS Support Layer

<table>
<thead>
<tr>
<th>Access Control CSCI</th>
<th>Navigation Support CSCI</th>
<th>Spatial Info Management CSCI</th>
<th>Video Management CSCI</th>
<th>Interface Data Analysis CSCI</th>
</tr>
</thead>
</table>

### Common Service S/W

<table>
<thead>
<tr>
<th>Digital Chart Management CSCI</th>
<th>System Management CSCI</th>
<th>Alarm/Warning Management CSCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication Middleware (DDS)</td>
<td>DBMS</td>
<td>Network Management</td>
</tr>
<tr>
<td>Network Time Management</td>
<td>Redundancy</td>
<td></td>
</tr>
</tbody>
</table>

### Infra S/W

<table>
<thead>
<tr>
<th>TCP/UDP-IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System (VxWorks, Windows)</td>
</tr>
</tbody>
</table>
2. Combat System Architecture

Russia Stock Market Cap.

Apple Stock Market Cap.
2. Combat System Architecture

+ High speed Transaction processing Database

- Memory structures
  - System Global Area (Buffer Cache)
  - Background processes
  - Database files

- Process structures
  - Data / Log / CKPT

- Storage structures
  - Database files

- Memory
  - Database file (Table / Index)
  - Background processes
  - Log / CKPT
3. Track Manager
3. Track Fusion

ASW Track 1 → Sensor A
ASW Track 2 → Sensor B
ASW Track 3 → Sensor C
ASW Track 4 → Sensor D

Fusion

Fusion condition Parameter

Track Management
4. Aegis Modernization (AMOD)

AMOD, DDG-1000 (US Navy)

OACE-CAT 4
- Common Services and APIs
- Flexibility to Support and Back-Fit

OACE-CAT 3
- Standards-based Interface
- Commercial Mainstream
- Products and Technologies

Application SW:
- Common Reusable Components
- Platform Specific Components

Non - LM Software

1. Real Time Operating System
2. Pub-Sub Communications
3. High Availability Middleware
4. Enterprise System Management
5. Human-Systems Software
6. Network Management Tools
5. Open Architecture TM

Joint Track Management Alignment
overview

- Align AMOD and SSDS Track Management to a Common Architecture
  - Provide Consistent Functional Allocation, Data Representation and Attributes
  - Incorporate Reusable System Track Manager and Track Server Components

- Provides Hierarchical Track File (System Level – Source Level)
  - Track Server Standard Access Interface for Client Applications
  - Track Manager Integrates Track Data Source via Common Interface
  - Extensible for New Track Data Sources

- Provides Two Complete Versions of Live Training Tracks
  - Allows Training Override of Multiple Attributes
  - Training Tracks Can be Physically Relocated From Live Location.

- Provides Dual Ownship – Tactical and Training
  - Allows Training View to be Repositioned with No Impact to Tactical View

Ref. The Modernization of the Aegis Fleet with Open Architecture, Lockheed Martin, ’11. 09
6. Main Memory Database

Main Memory Database

- Application
- Checkpoint Files
- Transaction Logs
7. MMDB & DRDB

DATA FLOW
- Application
  - DBMS
  - Memory Data

DATA FLOW
- Application
  - DBMS
  - Buffer
  - Disk Data
## 7. MMDB & DRDB

<table>
<thead>
<tr>
<th></th>
<th>MMDB</th>
<th>DRDB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data model</strong></td>
<td>Relational</td>
<td>Relational</td>
</tr>
<tr>
<td><strong>System Structure</strong></td>
<td>Client/Server and inner DB</td>
<td>Client/Server</td>
</tr>
<tr>
<td><strong>Server Structure</strong></td>
<td>Multi-Thread</td>
<td>Multi-Thread / Multi-Process</td>
</tr>
<tr>
<td><strong>CPU Use Rate</strong></td>
<td>lower CPU use rate using simple search Algorithm</td>
<td>Higher CPU use rate using complicate search Algorithm</td>
</tr>
<tr>
<td><strong>DISK I/O</strong></td>
<td>Minimum disk I/O in order to Recovery</td>
<td>Normal disk I/O in order to Select, Insert, Update, Delete</td>
</tr>
</tbody>
</table>
8. Moving Object DataBase

- Spatio-Temporal data types
- Spatio-Temporal Operators
- Spatio-Temporal indexing

Diagram showing layers:
- MO DB
- Spatial DB
- DR DB

Leaf node:
- A
- E
- F
- B
- C
- D

Diagram on the right shows a 3D representation with axes x, y, t.
9. Kairos Database Overview

- MO (Moving Object) Data Model
  - MO Data Types, Spatio-temporal Operators
  - Spatio-temporal Indexes, MO SQL

- Spatial Data Model
  - Spatial Data Types, Spatial Operators
  - Spatial Indexes, Spatial SQL

- Relational Data Model, Transaction Processing
  - Query Processing
  - Transaction Management
  - Recovery, ...

- Replication Functions
- Caching Functions
### DR DB Data Type

<table>
<thead>
<tr>
<th>ID</th>
<th>Gender</th>
<th>Name</th>
<th>Current position(X,Y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>man</td>
<td>Bred</td>
<td>(120,150)</td>
</tr>
<tr>
<td>02</td>
<td>woman</td>
<td>Ray</td>
<td>(130,140)</td>
</tr>
<tr>
<td>03</td>
<td>woman</td>
<td>Ray</td>
<td>(160,150)</td>
</tr>
</tbody>
</table>

### MO DB Data Type

<table>
<thead>
<tr>
<th>ID</th>
<th>Gender</th>
<th>Name</th>
<th>Current position(X,Y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>man</td>
<td>Bred</td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>woman</td>
<td>Ray</td>
<td></td>
</tr>
</tbody>
</table>
B. Spatio-Temporal Index

- Spatio-temporal index of a 3D R*- Tree Structure
- Use minimum bounding rectangle information

<table>
<thead>
<tr>
<th>ID</th>
<th>Phone</th>
<th>Name</th>
<th>MPoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>010-xxxxxx</td>
<td>anton</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>010-xxxxxx</td>
<td>robert</td>
<td></td>
</tr>
</tbody>
</table>
# C. Spatio-Temporal Operator

<table>
<thead>
<tr>
<th>Operator group</th>
<th>Description</th>
<th>Sample</th>
</tr>
</thead>
</table>
| **Spatio- Temporal Relation Operators** | Analysis spatial phase relationship between moving spatial object- moving spatial object or moving spatial object-spatial object  
Operator : Intersects, Overlaps, Crosses, Within, Contain, Disjoint, Equals, Touches | ![Intersects](image1) ![Difference](image2) ![Intersection](image3) ![Contains](image4) ![Overlaps](image5) ![Precedes](image6) |
| **Set Operators**               | - One moving spatial object or moving spatial objects.  
- Operator : Union, Difference, Intersection                                                      | ![Union](image7) ![Difference](image2) ![Intersection](image3) |
| **Trajectory Relation Operators** | Analysis phase relationship of trajectory of moving object.  
Operator : Enter, Leaves, Meets, Passes, Insides                                                     | ![Enter](image8) ![Leaves](image9) ![Meets](image10) ![Passes](image11) ![Insides](image12) |
| **Temporal Relation Operators** | - Analysis time phase relationship between temporal object – temporal object  
Operator : Contains, Overlaps, Precedes                                                                | ![Contains](image4) ![Overlaps](image5) ![Precedes](image6) |
## C. Spatio-Temporal Operator

### Trajectory Relation Operators

<table>
<thead>
<tr>
<th>Return Value</th>
<th>Operator name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Int</td>
<td>ST.ENTER(A,B)</td>
<td>Mgeometry</td>
<td>The operator will be evaluated that “A” moving-object entered into “B” moving-object from outside to inside</td>
</tr>
<tr>
<td>Int</td>
<td>ST.LEAVES(A,B)</td>
<td>Mgeometry</td>
<td>The operator will be evaluated that “A” moving-object entered into “B” moving-object from inside to outside</td>
</tr>
<tr>
<td>Int</td>
<td>ST.PASSES(A,B)</td>
<td>Mgeometry</td>
<td>The operator will be evaluated that “A” moving-object penetrated “B” moving-object</td>
</tr>
<tr>
<td>Int</td>
<td>ST.MEETS(A,B)</td>
<td>Mgeometry</td>
<td>The operator will be evaluated that they met at the border of “A” and “B” object</td>
</tr>
<tr>
<td>Int</td>
<td>ST.INSIDES(A,B)</td>
<td>Mgeometry</td>
<td>The operator will be evaluated that “A” moving-object is staying in the “B” moving-object inside.</td>
</tr>
</tbody>
</table>
10. J-TM Design

- Sensor A
- Sensor B
- Sensor C

Track Management Application Software

API

- Fusion condition
- Kairos MO DB
- Fusion Result
11. J-TM Procedure

++ Joint track Management Procedure using moving-object database

01. Create table with a reference to parameter for the target fusion’s available condition

02. Database API (application program interface) calls using triggers when sensor track information is input

03. Determine fusion availability according to the target’s location using the moving object database function

04. Insert results into the fusion table

05. Return the fusion table value from the database
11. J-TM API Design

- Application Software
- Human Display Software
- Library
- Kairos MO DB
- Procedure

API connections between Application Software and Human Display Software, as well as between the Library and other components.
11. J-TM API List

- Set Fusion
- Get Confirm
- Get Alert
- IFF Code
- Initial Schema
- Delete Track
- Get Conflict Table
- Get Confirm
- Get Alert
- Set Correlation
- Set DeFusion
- Set Track
- Save Track
- Set Parameter
- Reset Schema
- Get Correlation

Provide About 100 API
12. Performance

**Spatio Query Performance**

- **Test Environment**
  - Windows 2003/Xeon * 16CPU, 64G Memory

- **Test Model**
  - Performance of Spatio Relation Operators
  - Data : 248,115 Polygons

- **Result**
  - MM DBMS(Kairos DBMS) is 10 times faster than other DR DBMS
13. Conclusion

Many Benefit of Implementation of Joint-TM

- 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

<table>
<thead>
<tr>
<th>Benefit</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduction in Time to Field</td>
<td>- Decreased development and acquisition cycle times to field new capabilities</td>
</tr>
<tr>
<td></td>
<td>- Faster integration of open standards based systems</td>
</tr>
<tr>
<td>Improved Interoperability</td>
<td>- Use of common warfighting applications (joint-TM)</td>
</tr>
<tr>
<td>Reduction in Risk</td>
<td>- Leverage proven reusable components</td>
</tr>
<tr>
<td></td>
<td>- Test early and often in the developmental cycle to minimize risk of delivering non-interoperable products</td>
</tr>
<tr>
<td>Cost Avoidance</td>
<td>- Cost avoidance from software re-use and use commodity COTS products at optimum prices</td>
</tr>
<tr>
<td></td>
<td>- Reduced training and streamlined lifecycle support</td>
</tr>
</tbody>
</table>
Thank you
Appendix
A. DataBase Backup

1. 백업명령
2. Check point
3. DB 파일 복제 수행

KAIROS
Memory
Memory DB
DB 파일

11-13 DE MARZO DE 2015
B. Database Recovery

- Memory DB
- Query Engine
- Transaction Logs
- Log Buffer
- Memory DB changes
- DML 연산

- Physical memory
- Memory DB
- Checkpoint Thread
- Periodically write
- DB backup files
- Log files
- Write during Commit Op.

- Log Manager
C. Dual Replication
D. RD-DBMS SYNC

Diagram showing the synchronization process between KAIROS and Disk DBMS, with steps for sync/async propagate, flush/download, full autorefresh, and load/refresh.