



Defence Research and
Development Canada

Recherche et développement
pour la défense Canada



On Database Synchronization & Logistics Adaptive Planning Framework

*Net-Centric Logistics Framework Development for Adaptive
Operational Planning*

M. Debbabi et al.
TrustSeer Corporation

The scientific or technical validity of this Contract Report is entirely the responsibility of the Contractor and the contents do not necessarily have the approval or endorsement of the Department of National Defence of Canada.

Defence Research and Development Canada – Valcartier

Contract Report
DRDC Valcartier CR 2011-624
March 2011

Canada

On Database Synchronization & Logistics Adaptive Planning Framework

*Net-Centric Logistics Framework Development for Adaptive
Operational Planning*

M. Debbabi et al.
TrustSeer Corporation

Prepared by:
TrustSeer Corporation
C.P. MacKay, P.O. BOX 55002
Montreal, Québec, H3G 2W5

PWGSC contract number: W7701-9-3030

The scientific or technical validity of this Contract Report is entirely the responsibility of the Contractor and the contents do not necessarily have the approval or endorsement of the Department of National Defence of Canada.

Defence Research and Development Canada – Valcartier

Contract Report
DRDC Valcartier CR 2011-624
March 2011

IMPORTANT INFORMATIVE STATEMENTS

The scientific or technical validity of this Contract Report is entirely the responsibility of the Contractor and the contents do not necessarily have the approval or endorsement of the Department of National Defence of Canada.

- © Her Majesty the Queen in Right of Canada, as represented by the Minister of National Defence, 2011.
- © Sa Majesté la Reine (en droit du Canada), telle que représentée par le ministre de la Défense nationale, 2011.

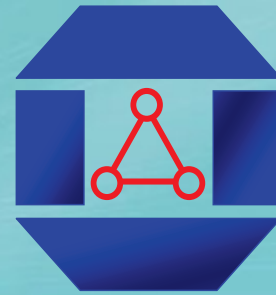
Abstract

This document contains information on the design and implementation of the database synchronizer test-bed, along with a centralized planning framework over it. The purpose of the document is to present the software life cycle according to the IEEE 12007 standard. Here we illustrate the requirements analysis followed by software design, architecture and implementation details. The main intent of the project is to integrate and synchronize these two systems such that any change of information in one of these systems is directly and automatically reflected in the other system in near real-time. This involves the elaboration of a centralized framework and related procedures for the integration and synchronization of two databases, namely, LogBase (the database behind EVE application) hosted on Microsoft SQL-Server and NMDS hosted on Oracle. Such a net-centric logistics framework will allow access and will synchronize multi-level information sources while constantly monitoring/tracking assets information in order to present it to the users. On top of the data synchronization framework, we have designed and implemented net-centric logistics framework enabled by various information sources at different levels. This information is enabling us to create and simulate logistics military missions in relation to contingency planning, monitoring and analysis. This proof-of-concept extensively demonstrates the implementation of theoretical logistic planning using vehicle routing problems on a specialized client-server framework.

Résumé

Ce document contient de l'information sur la conception et la mise en œuvre du banc d'essai du synchroniseur de base de données, ainsi que d'un cadre de planification centralisé. Le but de ce document est de présenter le cycle de vie du logiciel selon la norme IEEE 12007. Nous illustrons ici l'analyse des besoins suivie des détails de la conception, de l'architecture et de la mise en œuvre du logiciel. L'objectif principal du projet est d'intégrer et de synchroniser ces deux systèmes de sorte que tout changement d'information dans l'un de ces systèmes se reflète directement et automatiquement dans l'autre système en temps quasi réel. Ceci comprend l'élaboration d'un cadre centralisé et des procédures connexes pour l'intégration et la synchronisation de deux bases de données, à savoir LogBase (la base de données derrière l'application EVE) hébergée sur Microsoft SQL Server et le SNDM hébergé sur Oracle. Un tel cadre logistique réseaucentrique permettra l'accès et synchronisera les sources d'information à plusieurs niveaux tout en surveillant et en suivant constamment l'information sur les biens afin de la présenter aux utilisateurs. En plus du cadre de synchronisation des données, nous avons conçu et mis en œuvre un cadre logistique réseaucentrique reposant sur diverses sources d'information à différents niveaux. Cette information nous permet de créer et de simuler des missions logistiques militaires liées à la planification d'urgence, à la surveillance et à l'analyse. Cette validation de principe démontre de manière approfondie la mise en œuvre de la planification logistique théorique au moyen de problèmes de routage de véhicules sur un cadre client-serveur spécialisé.

This page intentionally left blank.



TrustSeer Corporation



Task-6: Technical Documentation

On database Synchronization & Logistics Adaptive Planning Framework

Project: Net-Centric Logistics Framework Development
For Adaptive Operational Planning

| | |
|-----------------------------|--------------------|
| Solicitation No. | W7701-093030/A |
| Client Reference No. | W7701-9-3030 |
| GETS Reference No. | PW-\$QCL-028-12527 |
| File No. | QCL-9-30750 (028) |
| Date. | March 31, 2011 |

TrustSeer Corporation
C.P. MACKAY, P.O. BOX 55002
Montreal, Quebec,
H3G 2W5 Canada
E-mail: info@trustseer.com
Web site: www.trustseer.com



EXECUTIVE SUMMARY

This document contains information on the design and implementation of the database synchronizer test-bed along with centralized planning framework over it. The purpose of the document is to present the software life cycle according to the IEEE 12007 standard. Here we illustrate the requirements analysis followed by software design, architecture and implementation details. The main intent of the project is to integrate and synchronize these two systems such that any change of information in one of these systems is directly and automatically reflected in the other system in near real-time. This involves the elaboration of a centralized framework and related procedures for the integration and synchronization of two databases, namely, LogBase (the database behind EVE application) hosted on Microsoft SQL-Server and NMDS hosted on Oracle. Such a net-centric logistics framework will allow access and will synchronize multi level information sources while constantly monitoring/tracking assets information in order to present it to the users. On top of the data synchronization framework, we have designed and implemented net-centric logistics framework enabled by various information sources at different levels. This information is enabling us to create and simulate logistics military missions in relation to contingency planning, monitoring and analysis. This proof-of-concept extensively demonstrates the implementation of theoretical logistic planning using vehicle routing problems on a specialized client server framework.

TABLE OF CONTENTS

| | |
|--|-----------|
| EXECUTIVE SUMMARY..... | 2 |
| INTRODUCTION | 7 |
| SCOPE | 7 |
| REQUIREMENTS ANALYSIS..... | 8 |
| REQUIREMENTS ELICITATION | 8 |
| APPROACH DERIVATION | 10 |
| ARCHITECTURE & METHODOLOGY | 12 |
| SYSTEM ARCHITECTURE | 12 |
| INTEGRATION ARCHITECTURE | 13 |
| MONITORING ARCHITECTURE..... | 14 |
| PLANNING ARCHITECTURE | 15 |
| SOFTWARE DESIGN | 17 |
| SERVER SIDE DESIGN..... | 18 |
| CLIENT SIDE DESIGN..... | 27 |
| IMPELEMENTATION & SCREENSHOTS | 35 |
| TECHNOLOGY STACK..... | 35 |
| DATABASE INTEGRATION..... | 35 |
| CLUSTERING | 38 |
| PLAN RE-ROUTING..... | 41 |
| SIMULATION | 42 |
| ADAPTIVE PLANNING..... | 43 |
| CONCLUSION..... | 45 |
| REFERENCES | 46 |
| APPENDIX..... | 48 |
| TECHNOLOGY STACK..... | 48 |
| DEFINITIONS & ACRONYMS..... | 50 |

LIST OF FIGURES

| | |
|--|----|
| Figure 1: Requirement analysis using Fishbone diagram..... | 9 |
| Figure 2: Main activities of net-centric logistics framework for adaptive planning..... | 10 |
| Figure 3: Overview of DB-Synchronizer Client-Server architecture | 12 |
| Figure 4: DB-Synchronizer Client-Server Integration architecture..... | 13 |
| Figure 5: Operational architecture of DB-Synchronizer Software..... | 14 |
| Figure 6: Step by step methodology for data synchronization within two databases..... | 15 |
| Figure 7: Adaptive planning architecture for plan generation | 16 |
| Figure 8: Design of core software modules interaction | 17 |
| Figure 9. Server Side Web Service Methods..... | 20 |
| Figure 10. DDP Mission Processing Logic | 22 |
| Figure 11. Prism Results Parsing | 23 |
| Figure 12. Extended Expression Parser and Evaluator Supporting Arithmetic, String and Mixed expressions..... | 25 |
| Figure 13: Server-side database subscription module..... | 26 |
| Figure 14: Class diagram for database integration module | 27 |
| Figure 15: Design level classes for the data reporting module responsible for data analysis and statistics | 28 |
| Figure 16. ATO Mission Structure Representation | 30 |
| Figure 17: Class diagram for implementing the Geographic Information System..... | 32 |
| Figure 18. Node Clustering Generation | 33 |
| Figure 19: DB-Synchronizer data integration module interface..... | 36 |
| Figure 20: DB-Synchronizer Reporting module interface | 37 |
| Figure 21: DB-Synchronizer: Synchronization designer module | 37 |
| Figure 22. Node Clustering Interface..... | 38 |
| Figure 23. Identification of Cluster Heads / Distribution Centers..... | 39 |
| Figure 24. Strategic Node Identification | 39 |
| Figure 25. Disabling of Strategic Node..... | 40 |
| Figure 26. Larger Example of Clustering, Cluster Head and Strategic Node Identification | 41 |
| Figure 27: Display of mission DDP on GIS system..... | 42 |
| Figure 28: Display of alternative routes(in green) to a resource allocation plan. | 42 |
| Figure 29: Simulation of mission plans and alternatives | 43 |
| Figure 30: Display of adaptive planning using MDSDVRP planning logic in existence of two clusters. | 43 |

Figure 31: Display of same adaptive planning using MDSDVRP planning logic in existence of three clusters.....44

Revision History

| Name | Date | Reasons for Changes | Version |
|----------------|-------------------|---|---------|
| TrustSeer Team | February 22, 2011 | Report Draft | 0.1 |
| TrustSeer Team | March 25, 2011 | Spelling & Grammar Correction | 0.2 |
| TrustSeer Team | March 29, 2011 | Modification from internal suggestions. | 0.3 |
| TrustSeer Team | March 30, 2011 | Final Report | 1.0 |

INTRODUCTION

The proposed research and development project of Net-Centric Logistics Framework Development for Adaptive Operational Planning is a joint effort of Defence Research and Development Canada - Valcartier and CANOSCOM towards superior logistic support. It aims to enhance the current military planning capabilities for logistic, functional support inputs to the operational plan (Op Plan) and day-to-day logistics/movement support for ongoing operations. Moreover, this initiative will help establishing new agreements by quickly aggregating the current requirements in concert with the operational command. In this setting, the goal of this project for the proof-of-concept software development is threefold. First, a comprehensive research and development effort is required as a foundation for automated and intelligent access, use, composition, aggregation and exploitation of a variety of data/information sources and applications as per the requirements of CANOSCOM planners and operators. Second, state-of-the-art technology and tools are to be used in an *integrated platform* in order to process the information for effective plan generation, editing as well as for monitoring expected and unexpected events during the plan execution. Finally, plan and resource information should be allowed for partial analysis and simulation to understand the consequences through what-if scenarios.

Scope

CANOSCOM planners and operators require information from CF databases to update CF resources status. CANOSCOM presently uses two major systems to access such information, namely: National Materiel Distribution System (NMDS), Logistic Functional Area Services (LOGFAS). NMDS represents an elaborated system that is presently used by DND for assigning the movement of their own assets in different locations of the world. NMDS gets always updated from DND applications and other data sources. The primary interest of the software application prototype is to get updated information from this vital data sources for the Canadian Forces and synchronize with the underlying databases of LOGFAS system. LOGFAS is a NATO application and currently being maintained through a back-end SQL-Server database, named LOGBASE. Once the synchronization will be achieved, any update of NMDS data-source will be automatically update LOGBASE and therefore will be reflected to LOGFAS applications

The second phase of the project was aimed to generate logistic plan related tasks by retrieving information from these data-sources. Different vehicle routing models are implemented in the server side to generate tasks from the information retrieved from underlying two databases, namely: NMDS and LOGBASE. In this document we present the design and implementation of the system to produce logistic planning tasks. The tasks are represented in NATO military standard Air Tasking Order (ATO).

The proof of concept is a client-server project that comes with the following modules: DBSynchronizer Client, Synchronization Web Service, Planning Web Service and Localised Map Service.

REQUIREMENTS ANALYSIS

This section elaborates briefly to our notion on the requirements as understood from several discussions with DND officials, scientific authorities, developers and others. We tried to include these requirements for the adaption to the net-centric framework into the proof of concept. First we present in the following the mapping of the solution requirements that was done in the beginning of the project.

Requirements Elicitation

After a careful analysis of the related work and background information, we understood two main requirements for the implementation in this project:

- **Information Integration & Synchronization:** The integration of databases, web-services and legacy systems, as initiated in the direction of NCO framework is of paramount importance in order to achieve the desired framework functionality. Data monitoring and synchronization will play vital role in synthesizing information and in presenting it to the decision makers.
- **Data Analysis & Planning:** Research and development of planning toolbox that will allow the users to plan and collaborate with other users and update their plan. This requires efficient planning algorithms that may be used by users to create plan using integrated and synchronized data from the underlying system. The adaptive planning algorithms are required to be more focused to logistics movement and that fit well into the distributed architecture on a collaborative environment.

Actually, these two points explicitly explain how the information needs to be integrated and, what are the requirements of integration and the reasons behind creating net-centric logistics framework for adaptive planning.

In the context of CANOSCOM operations, the underlying gap in the collaboration between NATO and DND are in the integration and synchronization of their information sources. ADAMS and EVE are used as standalone applications, meaning that manual intervention is required for the input of information from NMDS in order to update and display the scheduling of asset movement to other international defence organizations, mainly NATO. On the other hand, NMDS is directly updated by DND, Canada officials and the information is received by CANOSCOM. CANOSCOM manually transmits Canadian assets distribution information into LOGBASE through the LOGFAS applications.

Figure 1 represents the two basic components of the framework that are divided into three sub-components for each branch. Data Integration & Synchronization is further elaborated in Data & Application integration section. Wrapping application information into web-services and using APIs to access databases allows integrating remote information sources to our application in NCO framework setting. In the next sections, we will highlight how these requirements are met through the architecture and design.

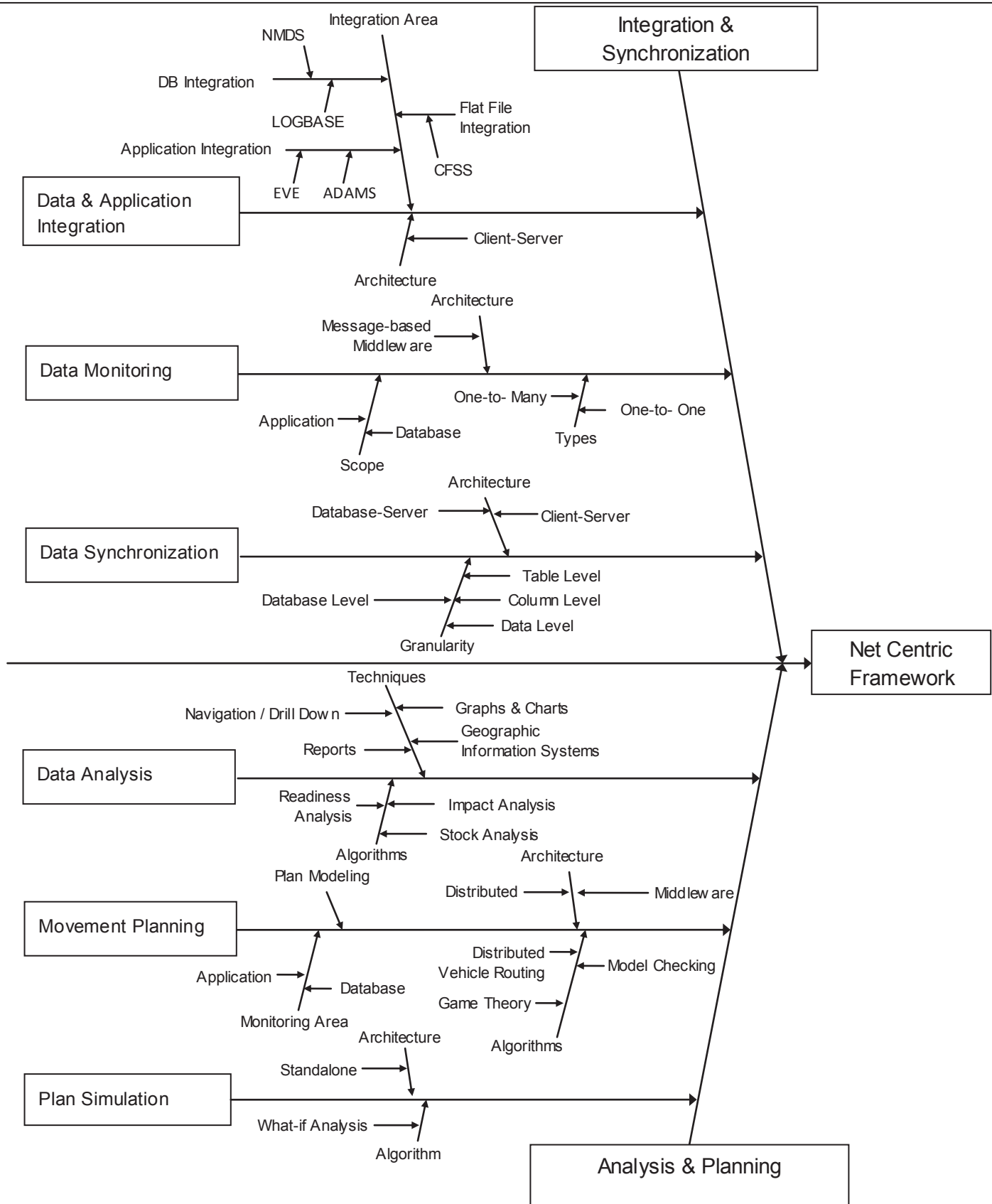


Figure 1: Requirement analysis using Fishbone diagram

Approach Derivation

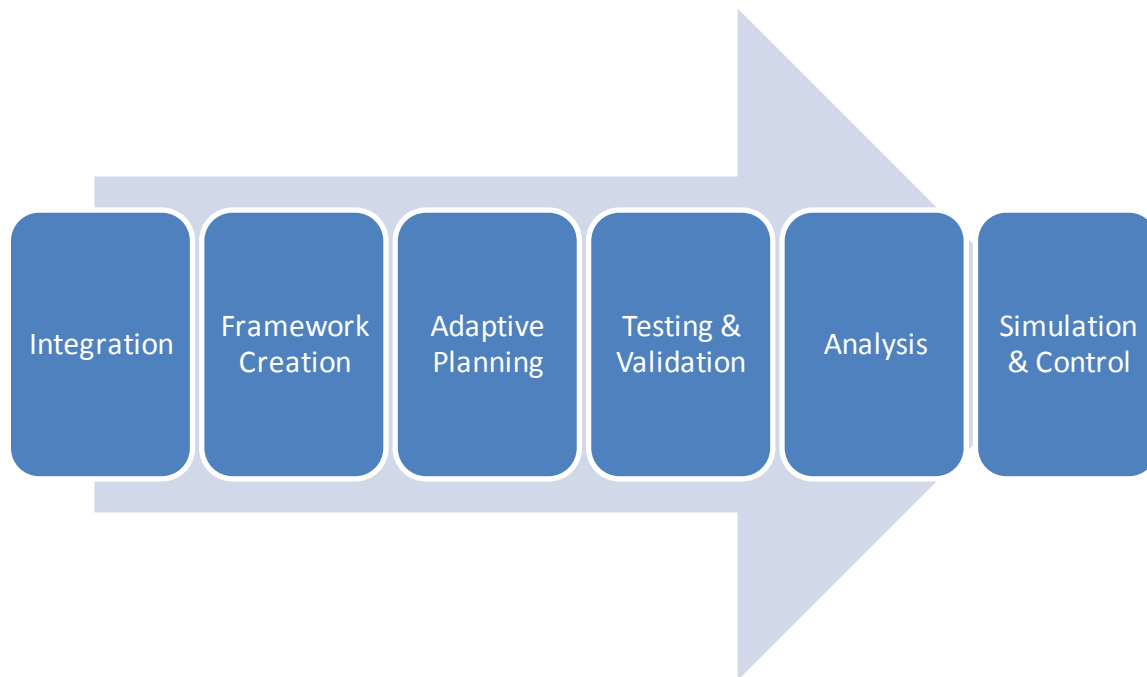


Figure 2: Main activities of net-centric logistics framework for adaptive planning.

Figure 2 presents the proposed main activities of the Net-Centric logistics framework for adaptive planning.

The *integration* module is responsible for data integration needed to connect all the information and service sources within and across the organization, for the purpose of information sharing. Integration of information systems is one of the key issues for the net-centricity. We defined interfaces and communication protocols based on a message-based middleware in three phases: data integration, service integration, and messaging service. Data integration is related with the retrieval of data from the structured, semi-structured, or unstructured sources of information inside and outside the organization. As of the service integration, we created three web-services responsible for a) data integration and synchronization of two databases, b) logistic planning service for creating tasks from the plan related information from databases and input from client c) Localized map service to provide map images to be presented in a GIS in order to help planners in creating plans using geographic locations.

The *net-centric logistic framework* is created in two levels: Data Integration Framework and Planning framework. The frameworks are designed on top of a message based middleware implemented using Java message service. NMDS and LOGBASE are two integrated data sources.

In relation to the *adaptive planning*, various planning techniques are used including Model checking techniques and heuristics based solution to vehicle routing problems. In the military context, an important aspect is related to the resource allocation and the distribution of assets and personnel over a transport network. In this setting, the Vehicle Routing Problem (VRP) needs to be addressed for

adequate routing procedures that are required in order to efficiently utilize the transport network. Of a particular significance is the movement of aerial assets such as transport planes in pursuit of resupply or strategic airlift operations. In this pursuit, adaptive planning techniques are investigated in various domains from model checking, clustering to different solution techniques vehicle routing models. The Vehicle Routing Problem Analysis in the Context of Adaptive Planning can be approached using the probabilistic model checker PRISM¹ and/or the symbolic model checker NuSMV² for the analysis of vehicle routing. This approach provides an overall assessment of the system, yielding the near-minimum cost for moving many vehicles or assets across a shared transport network. Planning humanitarian logistics is also strongly related to the Vehicle Routing Problem (VRP). This type of problem is encountered in a variety of commercial, military and government applications. We researched and developed different solution techniques for solving such problems of adaptive planning. Those techniques are integrated to the proof of concept.

The testing and validation of the developed prototype is a vital part of the project. Specific techniques are tested and reported in the context of adaptive planning. Furthermore, the test bed is run in the development environment and demonstrated to the Scientific Authorities.

The application framework supports analysis of retrieved information and the plans. A separate reporting tab has been added to the application for showing data with curve and to perform simple data analysis on a report created from the integrated databases. Analyzed similarity among the tables can be later exploited for the synchronization purposes. For the case of plan analysis, we allow users to input various planning problems through a very user-friendly way. The plan parameters get processed in the server side and create a set of tasks for each plan that can be saved locally. The tasks are saved in most commonly used military tasking format named as ATO. User may visually analyze multiple plans on top of the same geographic information systems.

Simulation and Control activities are directed to the optimization of the designed and developed tasks. The application presents "what-if" scenarios for the used processes, methods and strategies. The simulation tool can be used to present decision makers the present composed effect of the created adaptive plans for assets movement at a particular moment of execution.

¹ <http://www.prismmodelchecker.org/>

² http://nusmv.irst.itc.it/NuSMV/papers/sttt_j/html/index.html

ARCHITECTURE & METHODOLOGY

The main intent of this section is to present important architectural components and their associations. We describe high-level methodology to implement such operational architecture. The name of the tool is chosen as DB-Synchronizer throughout the description. DB-Synchronizer represents a powerful and easy-to-use software prototype that offers an elaborated user interface that allows database exploration and cross synchronization between them the explored databases. It is mainly intended for conducting automatic transformation and transfer of information from one database to another over coalition multinational boundaries in an integrated and interoperable manner. Furthermore, the tool provides several other important features and operations including database exploration, report creation, near real time information monitoring etc. in a network centric environment.

We also extend DB synchronizer tool to planning horizon. The integration of the planning architecture is kept centralized in synchrony to previous design decisions. In this setting, the dients choose their planning concern and send respective parameters to the server to create set of tasks by processing their plans. The respective plans are processed centrally and stored locally by the clients for later analysis and simulation.

System Architecture

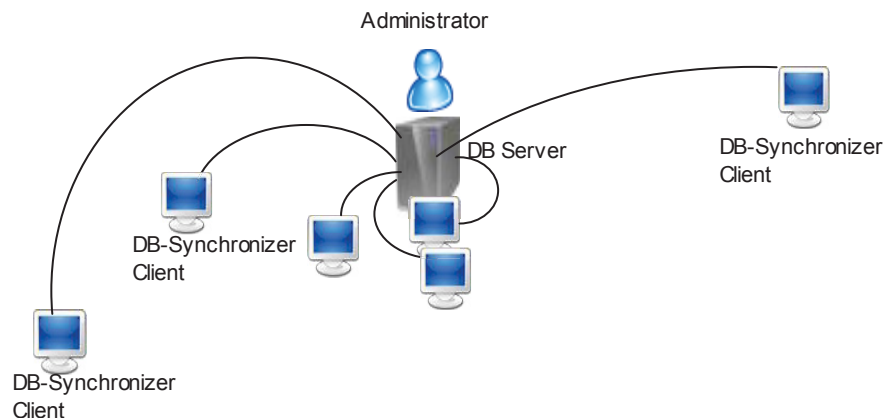


Figure 3: Overview of DB-Synchronizer Client-Server architecture

The DBSynchronizer tool employs a client server paradigm wherein the clients can be distributed over a large geographic area. From various remote locations, each client is maintaining active connection to a central server. This allows for a common visual inspection perspective accompanied by near real time monitoring of the integrated information in a network centric environment as depicted in the Figure 3.

From an architectural point of view, the DBSynchronizer uses a hybrid architecture where the user is allowed to interact directly with the databases using JDBC connections in order to explore the databases. Moreover, the dient application employs a powerful network centric message-oriented middleware in order to synchronize and monitor the underlying databases. Figure 1 depicts the corresponding architecture.

Integration Architecture

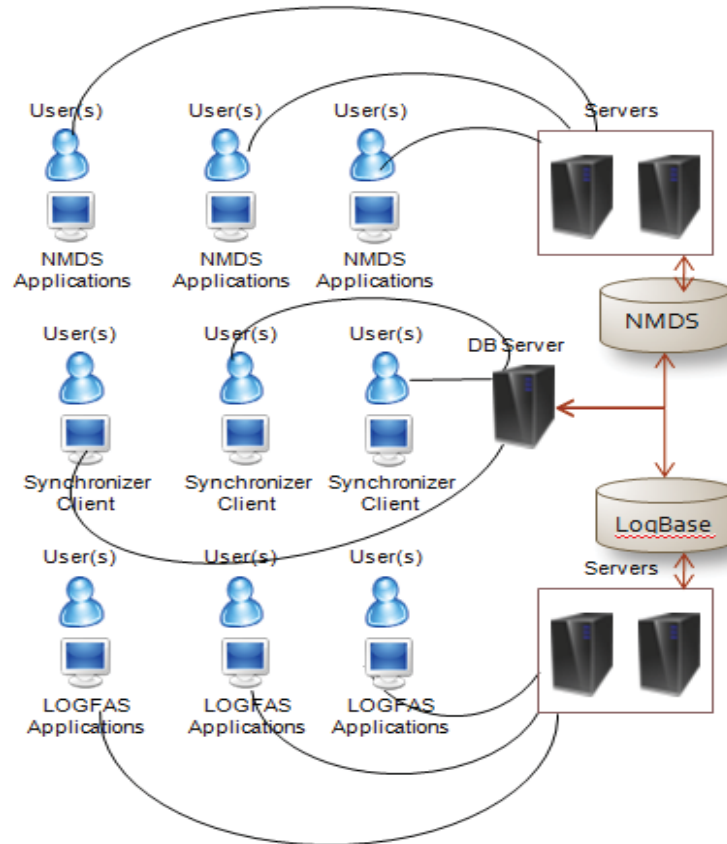


Figure 4: DB-Synchronizer Client-Server Integration architecture

Figure 4 presents the envisioned big-picture of the integration architecture of the proposed Synchronizer Client in the DND framework of applications. It presents our developed architecture where the DB-Synchronization Server hosts the middleware framework explicitly on top of the two databases and allows sharing of information. However the other NMDS and LOGFAS applications can be working similarly as it was before. Integration of the DB-Synchronization server will make difference as the inside synchronization module will allow the desired information to be shared automatically among two databases and thereby will be seen also from the NMDS and LogFas applications.

Data integration is one of the most important features of the proposed framework. The integration is meant to gather information from several distributed databases (possibly located in different geographic areas). This is transparent by hiding the complexity and heterogeneity of these databases. Moreover, it ensures that the data is exchanged reliably. The preliminary version of the developed DB-Synchronizer server prototype supports two cutting-edge database oriented application solutions namely: LOGBASE having the underlying Microsoft SQL-Server database for LOGFAS 6.0 application suite and NMDS, an Oracle database maintained by the Department of National Defence (DND), Canada. The DB-Synchronizer client may opt for retrieving the information directly from data-sources or through JMS for monitoring purposes.

Monitoring Architecture

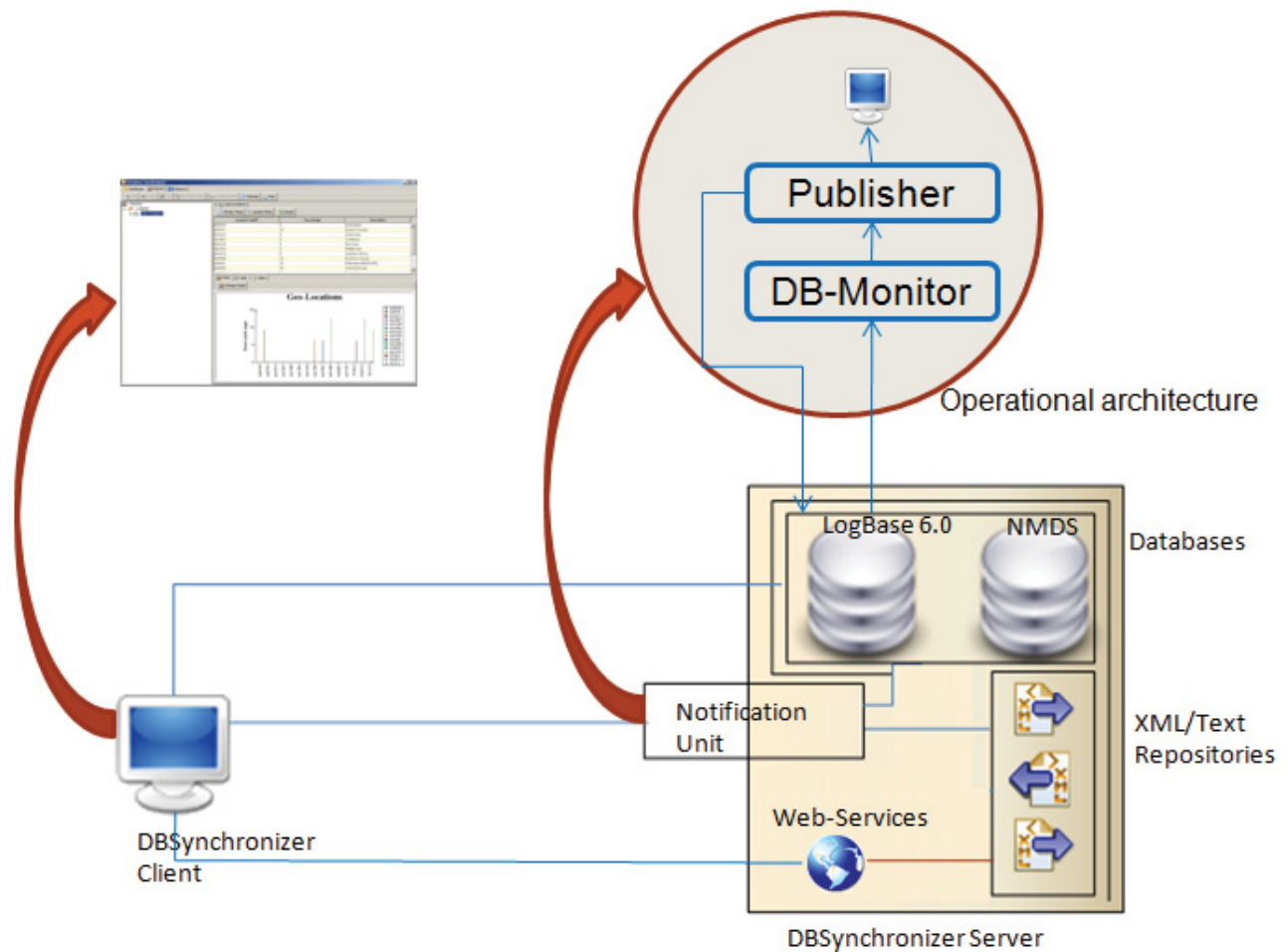


Figure 5: Operational architecture of DB-Synchronizer Software

The monitoring architecture of the DB-Synchronizer tool is presented in Figure 5. Within the client-server framework, the application runs a message based middleware that have three major components:

- **DB-Synchronizer Publisher:** DB-Synchronizer Publisher is a server side software component programmed to capture the subscription requests and to notify multiple users of a particular change either from the client or inside the databases.
- **DB-Synchronizer Monitor:** DB-Synchronizer Monitor is also a server side software component and an important part of the server side. The monitor module works as an interface between the database and the server that gathers the changes in the database and sends the changes to the publisher module.
- **XML Data & Flat Files:** Several supporting XML and other data files are kept in the DB-Synchronizer server. These files are used to hold the connection properties and other synchronization information that are delivered to a thick client application if needed.

Synchronization of information systems is a necessity for two basic reasons. The first reason is to gather accurate information at the clients' side and to be informed with most current situation at a certain course of operations during the execution of a plan that may span over a long time.

The notification mechanism is used in the framework in order to enable the remote clients to share a common perspective with respect to the data subscribed for notification. The framework issues near real-time notifications to the users, asynchronously, upon the occurrence of changes in data sources while the particular data-set is subscribed by the user. Secondly, it has been seen in practice that several databases in various institutional facilities gather similar information but remain out of any automated synchronization. Therefore manual insertion, deletion and update of information are required to synchronize information that is primarily stored in single database.

The data synchronization module guarantees that specific NCO server module will update multiple databases automatically if there is pre-defined synchronization policy in place for the particular information and the security credentials are respected. Figure 6 depicts step by step methodology for data synchronization within two databases.

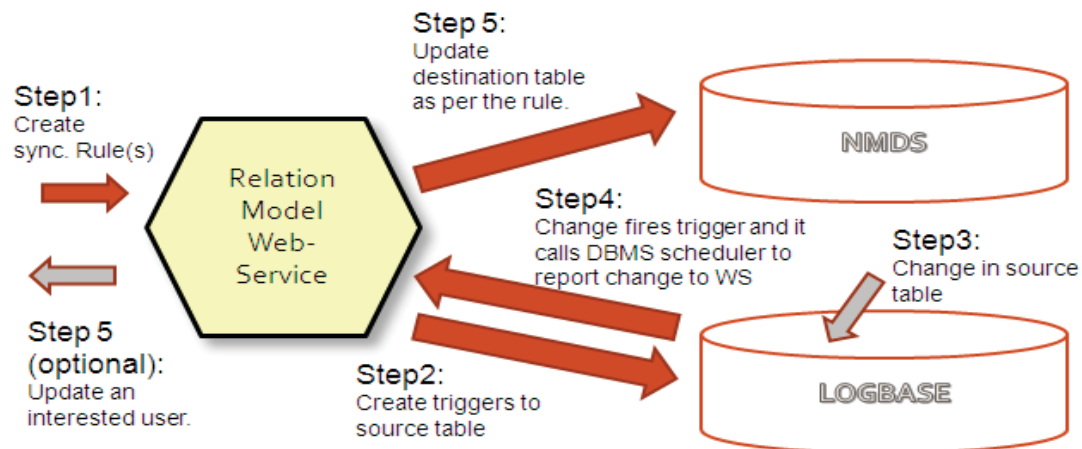


Figure 6: Step by step methodology for data synchronization within two databases.

Planning Architecture

Figure 7 depicts the architecture of adaptive operational planning system. The operational architecture is presented under multiple layers with higher focus on the 'Planning' layer. The other layers of the architectural diagram are Communication, Analysis and Visualization. The main idea of this architecture is to propose the inter-component relationships among the operational components within server and clients. As illustrated in the figure, the underlying communication layer provides a client node with input information on plan parameters and events upon their requests. The system also reads input from the human or other applications through its visual interface. Once a planning task is requested to the server, the planning layer retrieves information from the environment. Information can be related to plan and such information is directly transferred to the planning level through the 'Data Exchange' service.

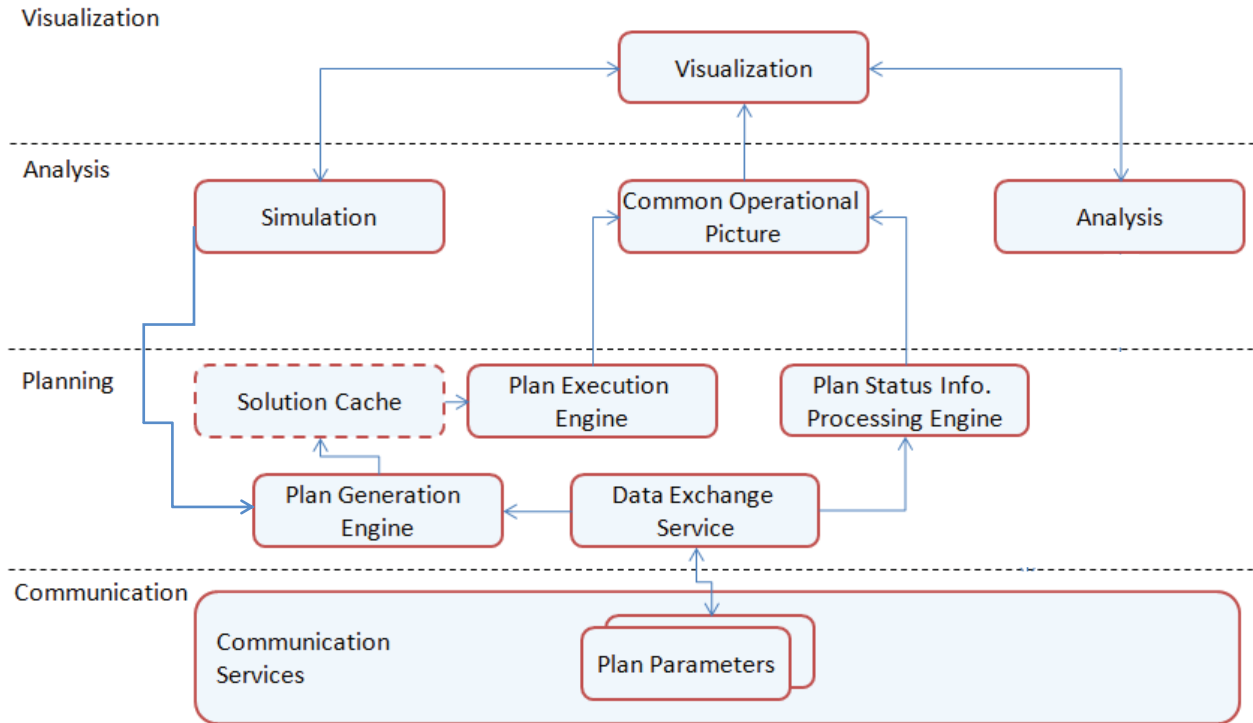


Figure 7: Adaptive planning architecture for plan generation

The centralized planning module invokes plan generation engine that tries to solve an instance of distinguished VRP problems either using model-checking or heuristic algorithms. Solution Cache is used to create multiple solutions within a time limit. The best solution is provided to the plan execution engine at the client side. The information related to the plan status is also passed. At this stage, it is difficult to comment on the details of visualization or analysis components. However, within analysis layer, the job of the application is to primarily create a common operational picture from planned task and related information and to present it to the users. The specific scope of the plan execution analysis is on top of a given state space generated from a chosen plan. Analysis of information allows planners to answer some of their questions and may 'intelligently' reveals incumbent problems. However, it does not answer any what-if scenarios that involve re-planning. Simulation, on the other hand, offers re-planning possibility to generate a new set of tasks and to match with current plan-related information.

SOFTWARE DESIGN

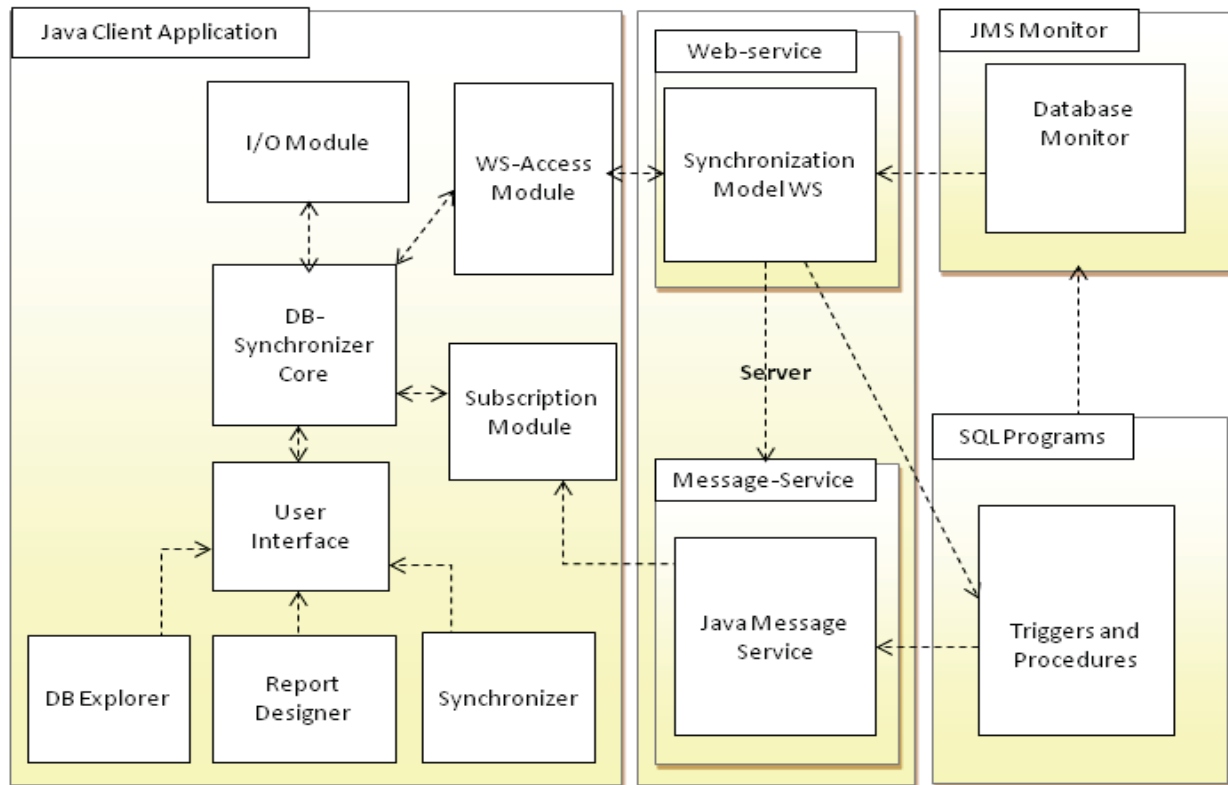


Figure 8: Design of core software modules interaction

The core of the DB-Synchronizer application represents a powerful and easy-to-use software prototype that offers an elaborated user interface that allows database exploration and cross synchronization between them the explored databases. It is mainly intended for conducting automatic transformation and transfer of information from one database to another over coalition multinational boundaries in an integrated and interoperable manner. The main design of the core of DB-Sync client application represents a thick Java application that uses a Java JMS client. The DB-Synchronizer client may retrieve and/or update mission related information stored in LOGBASE and NMDS databases using the connection to the central server. Moreover, it can monitor database information related to the movement schedules of military assets in a network centric environment. Using the monitoring capability, the application can be notified and updated upon any change performed by other clients on the server databases related to the mission application.

Figure 8 depicts the design of the system core components for the data-source integration. The designed and developed user interface has three main components, namely: DB-Explorer, Report Designer and Synchronizer. The first one is essential to integrate a new database to the Synchronizer application architecture. It allows user to explore the database and to understand the relationships among the database components. The Report Designer is used for data-mining on some of the required tables/views using the power of SQL queries that are used to generate and persist the corresponding reports. These persisted reports can be presented graphically. Furthermore, they can be updated, sorted

and filtered as per the user's choice. The third and the final component of the user interface is the Synchronizer where user can 'program/design' synchronization entries with different particularities among two tables A and B that are select from the databases to be synchronized. The scope of the synchronization is discussed later in a tabular format. Once the tables A and B are synchronized, change of any data from Table A will be automatically transferred to Table B without user intervention. The active Synchronizer client will also be remotely notified about the change. It has been clearly shown in the figure that the client system core uses web-service for instantaneous data communication between server and clients, while rely on JMS-based middleware for data monitoring and synchronization. The later is achieved in an established asynchronous way.

The implementation of the proof-of-concept is related to a development of considerably large number of classes. It is not possible to present them within this documentation one by one. Therefore we have chosen some of the most important modules to be presented through class diagrams.

Server Side Design

In Figure 9 we have a Class Diagram containing three Web Service Classes that are exposing a number of web methods at the server side. Below, we present in a tabular form each of the exposed web-methods.

| Location | [Class] Web Method | Comments |
|-------------|------------------------------|---|
| Server Side | [RelationModelWSImpl] | |
| | test | Test for checking the model generation capability |
| | deleteModel | Used to delete an existing model |
| | createNewModel | Used to create a new model |
| | getDataMiningFile | Used to retrieve a data mining file |
| | serverInformed | Used to notify the server |

| Location | [Class] Web Method | Comments |
|-------------|-----------------------------|--|
| Server Side | [LogBaseWrapperImpl] | |
| | getDDPMission | Retrieves a DDP mission based on its ID |
| | getMissionAlternatives | Retrieves mission alternatives based on existing mission ID, a list of comma separated ICAO and suggestion count |
| | getAllDDP | Retrieves all DDP mission IDs |
| | getAllAirports | Retrieves all airports |
| | getAllAirportsByPattern | Retrieves all airports based on regular expression pattern |
| | zipGepAllMsnAirports | Retrieves a zip-compressed structure of all mission airports |
| | getAirportLocation | Retrieves the latitude/longitude location of an airport |
| | getAirportLocList | Retrieves the latitude/longitude location of a list of airports |
| | invokeGLPSol | Invokes the Integer-Linear Programming Solver for VRP model |

| Location | [Class] Web Method | Comments |
|-------------|-----------------------------|--|
| Server Side | [MapCashManagerImpl] | |
| | getTile | Returns a tile based on specified parameters |
| | loadTileFromSite | Fetches a tile from the network |
| | getTileFromDiskCash | Reads a tile form the local disk cache |
| | getLocation | Returns current location |
| | saveTiletoDiskCash | Stores a tile in the local disk cache |
| | getSatelliteTileNameCode | Returns the satellite tile name code |
| | horizontalTileNumber | Returns the horizontal tile number |
| | verticalTileNumber | Returns the vertical tile number |

| RelationModelWSImpl |
|---|
| Attributes |
| Operations |
| <pre> public RelationModelWSImpl() public String test(String param) public void deleteModel(String name) public void createNewModel(String name, String desc, String from_table, String to_table, String relation, String DataminingGroupTreeNodeName) public String getDataMiningFile() public boolean serverInformed(String db, String schema, String table, String data, String hint) </pre> |
| LogBaseWrapperImpl |
| Attributes |
| Operations |
| <pre> private String xmlizer(XMap xmap) public String getDDPMission(String mission) public String[] getMissionAlternatives(String mission, String commaSepDisabledICAOs, String commaSepOptionalICAOs, int suggestionCount) public String[] getDDPMissions(String missionArray[0..*]) public String getAlIDDP() public String getAllAirports() public String getAllAirportsByPattern(String pattern) public byte[] zipGetAllMsnAirports() public String getAirportLoc(String icao) public String[] getAirportLocList(String icaoList[0..*]) public String invokeGLPSol(String problid, String patICAO, String patDmnd, String patCap, String patEstbCost) public void main(String args[0..*]) </pre> |
| MapCashManagerImpl |
| Attributes |
| <pre> public boolean fullMapCashingProcess = false public int IMT_MAP = 0 public int IMT_SATELLITE = 1 public int IMT_TRACE = 2 package Hashtable memoryCash = new Hashtable() </pre> |
| Operations |
| <pre> public String getTile(double tileX, double tileY, double zoomFactor, int ImageType) public String loadTileFromSite(double tileX, double tileY, double zoomFactor, int ImageType) public String getTileFromDiskCash(int tileX, int tileY, int zoomFactor, int imageType) public String getLocation() public String saveTiletoDiskCash(BufferedImage tile, int tileX, int tileY, int zoomFactor, int imageType) public String getSatelliteTileNameCode(double x, double y, double zoomFactor) public double horizontalTileNumber(double zoom) public double verticalTileNumber(double zoom) package void cashTile(int tileX, int tileY, int zoomFactor, int targetZoomFactor) </pre> |

Figure 9. Server Side Web Service Methods

In Figure 10 we have the DDP Mission Processing Logic Class Diagram containing 7 Classes, including the LogBaseWrapperImpl Class that exposes some of the server side web methods. We present below a brief description of each of the other 6 classes.

- **HMSTime:** Helper Class with no methods used for time manipulation based on hour, minute and second
- **YMDDate:** Helper Class with no methods used for date manipulation based on year, month and day
- **LogBaseWrapperCore:** Class providing the core logic for the LogBaseWrapperImpl web-methods
- **DDPMissionBuilder:** Class providing functionalities for building ATO mission structure from DDP mission
- **DDPMissionBuilderVRP:** Class providing functionalities for building ATO mission structure from DDP mission and VRP solution result
- **Leg:** Helper class for leg manipulation in DDPMissionBuilderVRP

In Figure 11 we have the Class Diagram for the Prism Results Parsing Logic. We present below a brief description of each of the classes.

- **PData:** Helper Class used for manipulating prism data
- **LineParser:** Interface containing the *parse* design contract method
- **PrismStateListParser:** Parser Class implementing specific *LineParser* logic for parsing States description
- **PrismTransListParser:** Parser Class implementing specific *LineParser* logic for parsing Transitions description
- **PrismStateRewardListParser:** Parser Class implementing specific *LineParser* logic for parsing State Rewards description

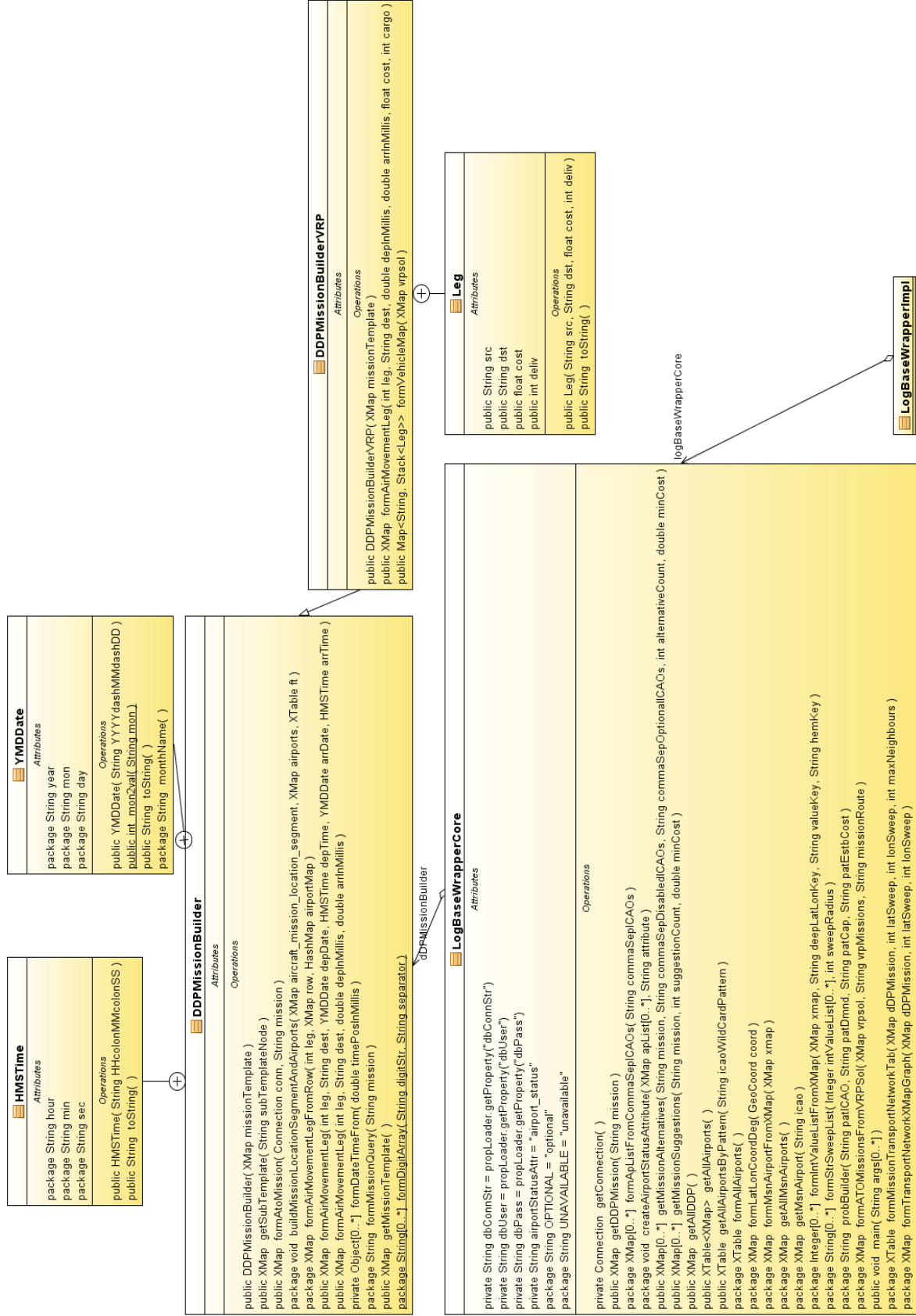


Figure 10. DDP Mission Processing Logic

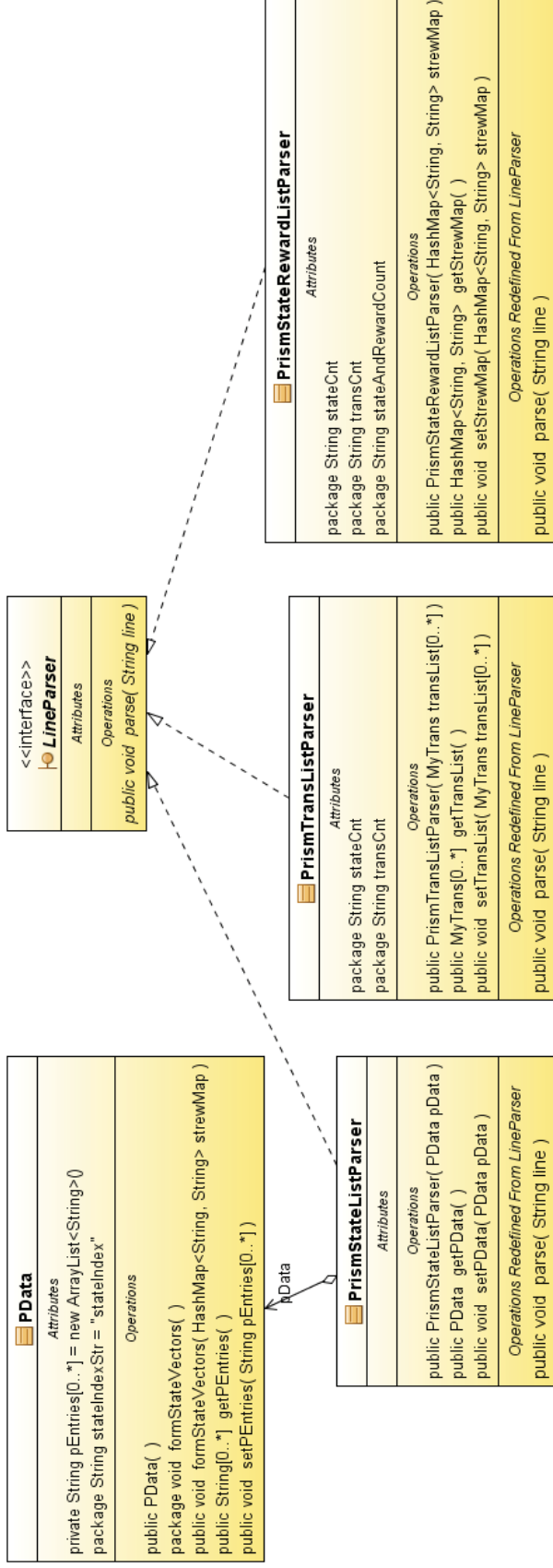


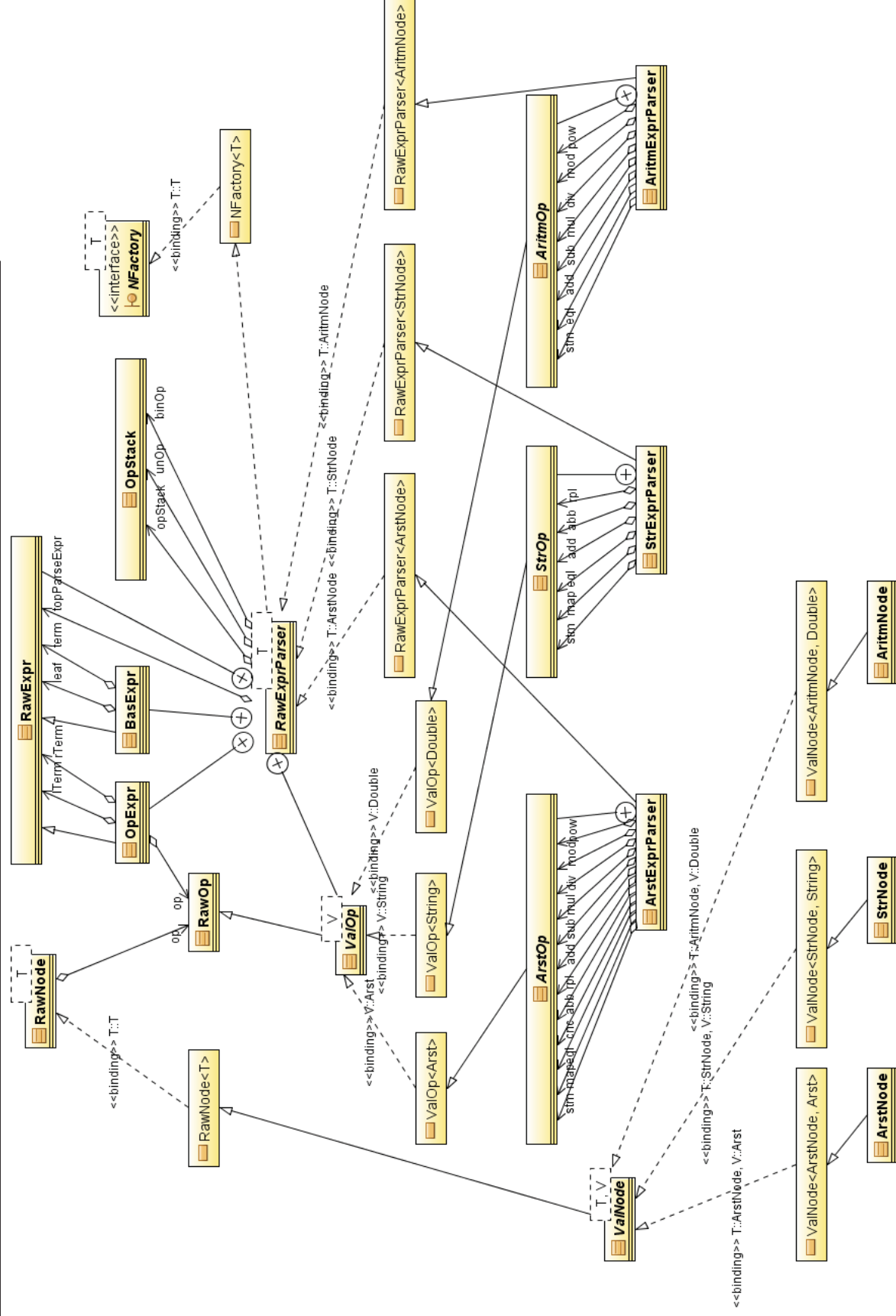
Figure 11. Prism Results Parsing

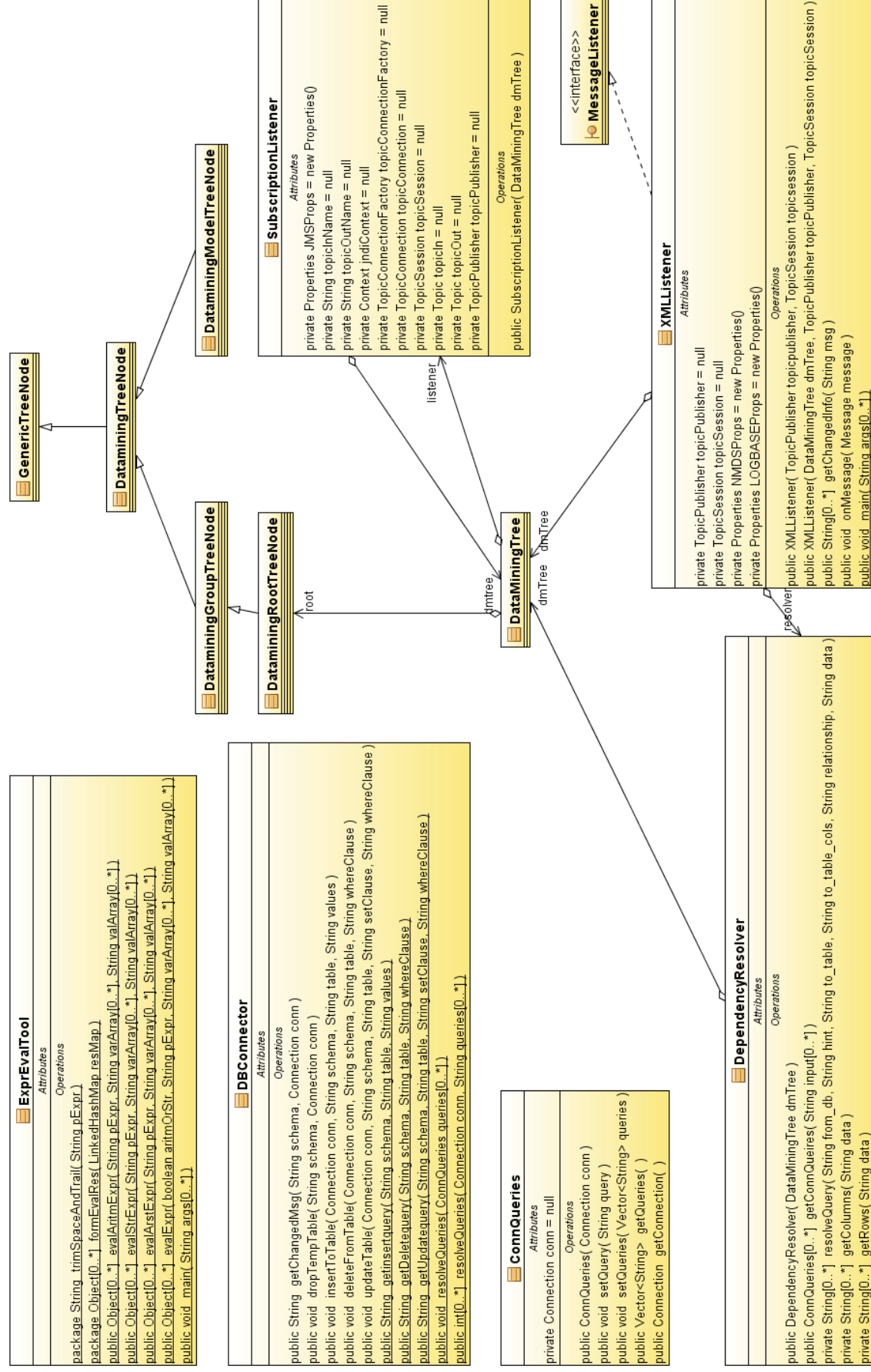
In Figure 12 we have the Class Diagram corresponding to an extended expression parser and evaluator supporting Arithmetic, String and Mixed Expressions. We present below a brief description of each of the classes.

- **RawExpr**: Class containing basing parsing logic
- **RawNode**: Class containing basic node manipulation logic
- **RawOp**: Class containing basic operator logic
- **BasExpr**: Class for parsing basic expressions
- **OpExpr**: Class for parsing operator expressions
- **OpStack**: Helper Class for holding unary and binary operator references
- **NFactory**: Node Factory Class

- **RawExprParser:** Stub Class containing anchor points for basic expression parsing
- **ValNode:** Specialization of RawNode for value holding
- **ValOp:** Specialization of RawOp for evaluating expressions
- **AritmNode:** Specialization of ValNode for value holding double values
- **AritmOp:** Specialization of ValOp for evaluating arithmetic expressions
- **StrNode:** Specialization of ValNode for value holding string values
- **StrOp:** Specialization of ValOp for evaluating string expressions
- **ArstNode:** Specialization of ValNode for value holding mixed arithmetic and string values
- **ArstOp:** Specialization of ValOp for evaluating mixed arithmetic and string expressions
- **AritmExprParser:** Specialization of RawExprParser containing parsing logic for arithmetic expressions
- **StrExprParser:** Specialization of RawExprParser containing parsing logic for string expressions
- **ArstExprParser:** Specialization of RawExprParser containing parsing logic for mixed arithmetic and string expressions

Figure 13 depicts the database subscription module of the server side. The class diagram presents a number of classes in relation to subscribing and subscription listening. From a more analytical viewpoint, the DataMiningTree is the main class that is holding inter-database relationships. Addition of a new relationship to the tree invokes required connection queries using ConQueries. DependencyResolver is an important class that actually resolves the database dependencies. In other word, it contains logic to guide what effect of one database change may bring to other database.





Client Side Design

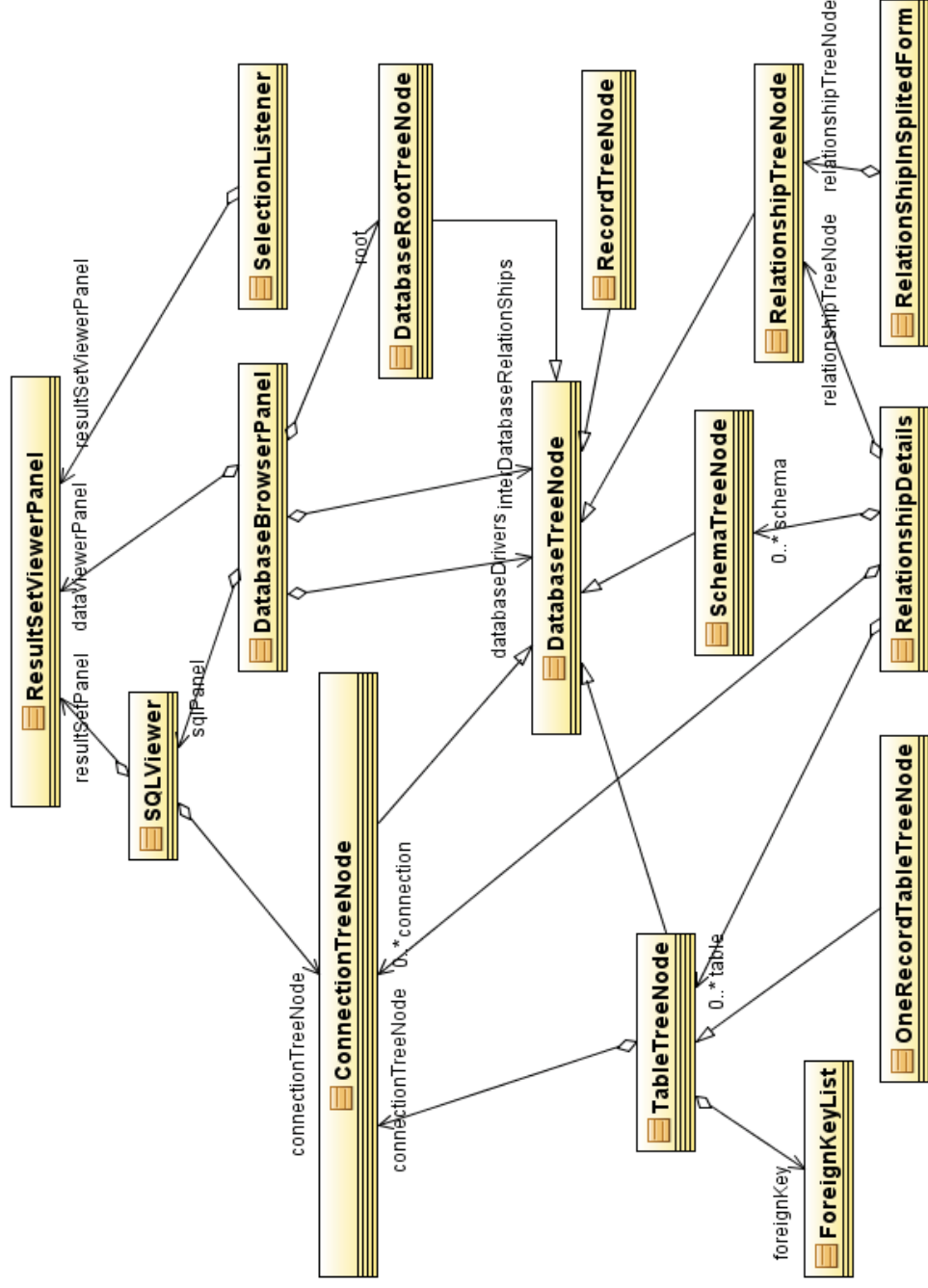


Figure 14: Class diagram for database integration module

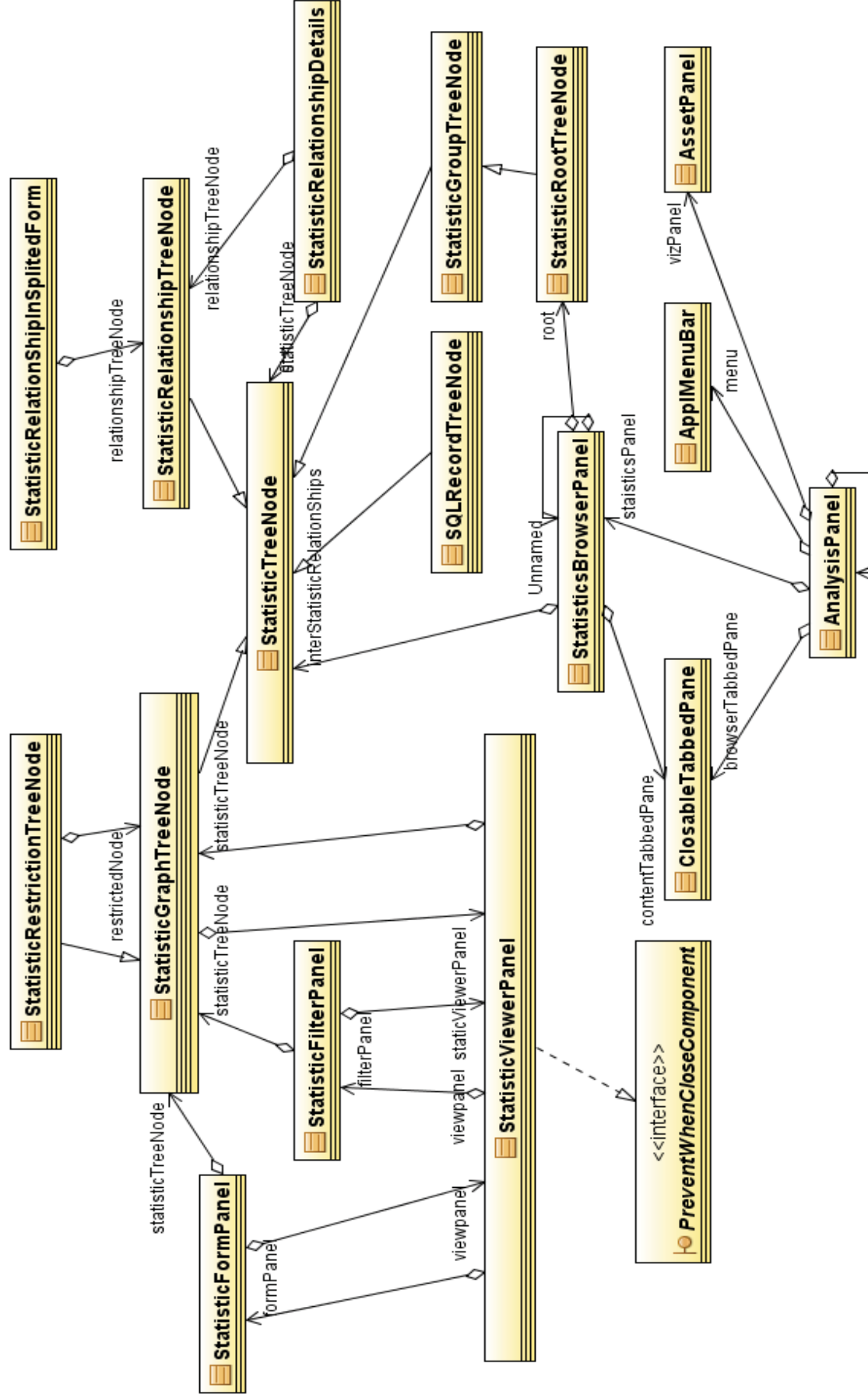
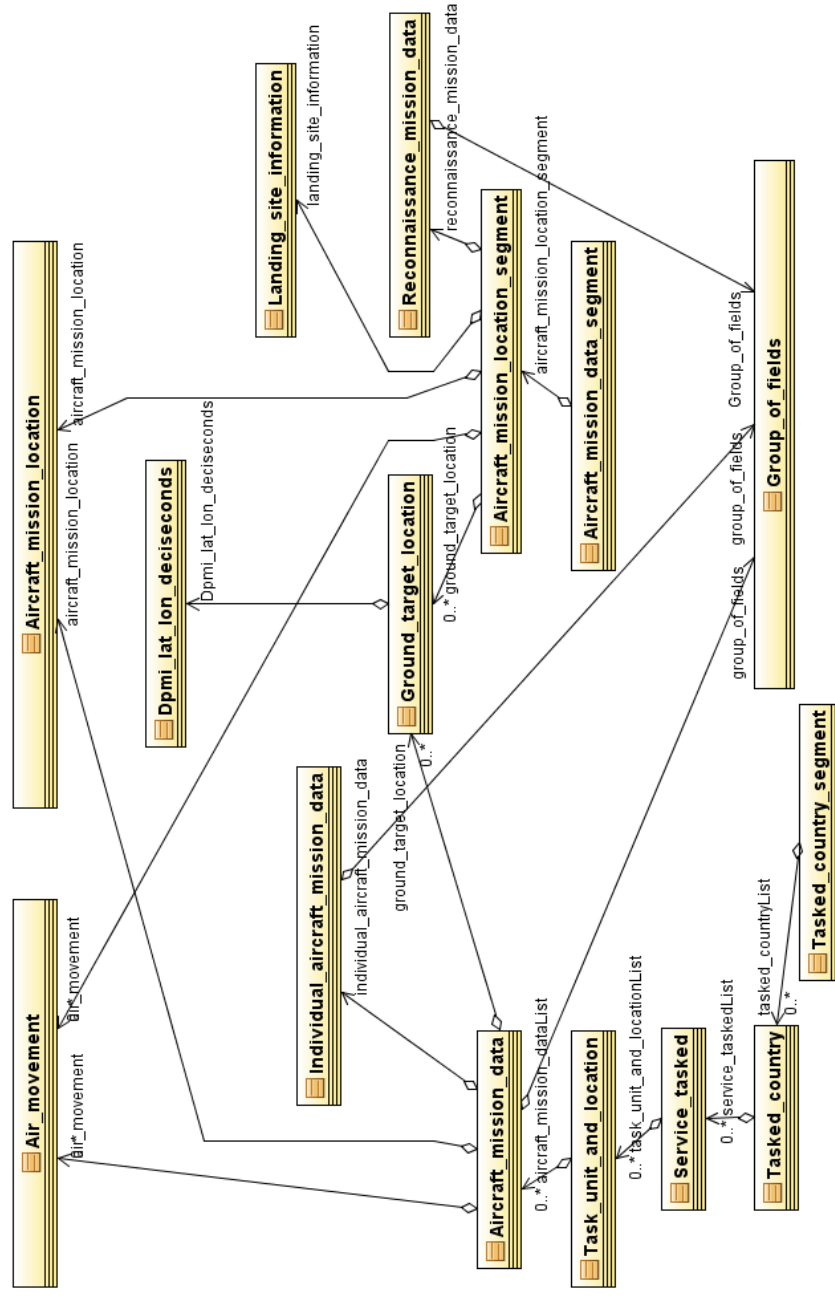


Figure 15: Design level classes for the data reporting module responsible for data analysis and statistics

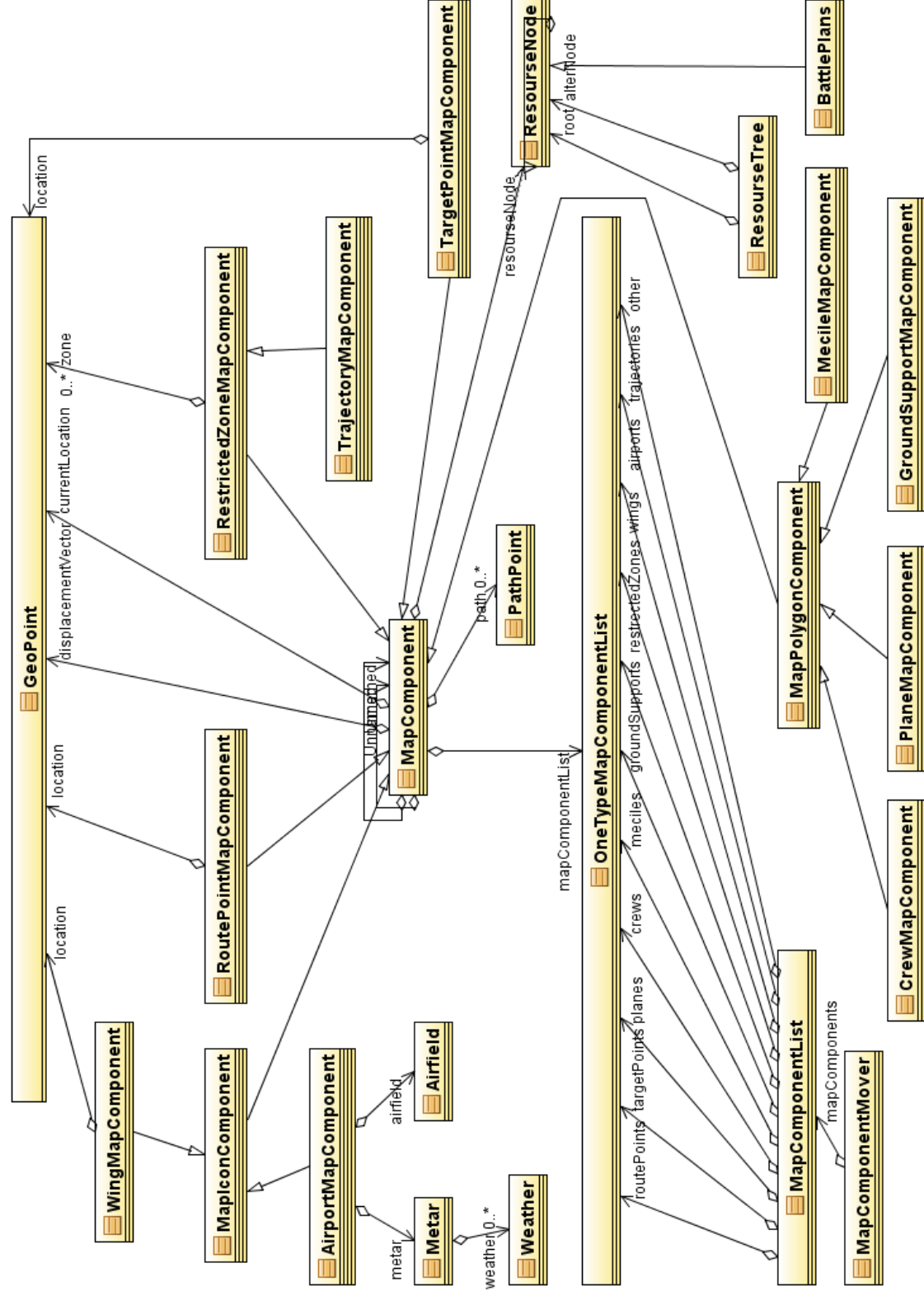
In Figure 16, we have the Class Diagram corresponding to the ATO mission structure representation. We present below a brief description of each of the classes.

- **Air_movement:** Describes the movement of the aircraft with respect to specified locations
- **Aircraft_mission_location:** Describes locations associated with the mission
- **Dpmi_lat_lon_deciseconds:** Holds latitude/longitude data to tents of a degree of precision
- **Individual_aircraft_mission_data:** Hold mission data specific to a particular assigned aircraft
- **Landing_site_information:** Contains landing site information
- **Ground_target_location:** Aggregates *Dpmi_lat_lon_deciseconds*
- **Reconnaissance_mission_data:** Contains reconnaissance mission information
- **Aircraft_mission_data:** Aggregates *Air_movement*, *Aircraft_mission_location* and *Individual_aircraft_mission_data*
- **Aircraft_mission_location_segment:** Aggregates *Ground_target_location*, *Air_movement*, *Aircraft_mission_location*, *Landing_site_information* and *Reconnaissance_mission_data*
- **Aircraft_mission_data_segment:** Aggregates *Aircraft_mission_location_segment*
- **Task_unit_and_location:** Aggregates *Aircraft_mission_data*
- **Service_tasked:** Aggregates *Task_unit_and_location*
- **Tasked_country:** Aggregates *Service_tasked*
- **Tasked_country_segment:** Aggregates *Tasked_country*



In Figure 17, we have the Class Diagram corresponding to the design of the Geographic Information System (GIS) used for rendering. We present below a brief description of each of the classes.

- **GeoPoint:** Describes the latitude / longitude of a geo-location corresponding to where the point should be plotted
- **MapComponent:** General purpose component responsible for rendering and holding common properties
- **PathPoint:** Aggregates mission specific *MapComponent*
- **ResourceNode:** Holds most general purpose properties and functionalities of objects in relation to the GIS.
- **ResourceTree:** Tree Structure Containing *ResourceNode* entities as leafs.
- **BattlePlans:** Specialized representation of *ResourceNode* containing battle plan representation
- **MapIconComponent:** Specialized *MapComponent* with an unique icon to be displayed on the GIS
- **WingMapComponent:** Specialized *MapIconComponent* representing location of AirForce Wings
- **AirportMapComponent:** Specialized *MapIconComponent* representing location of military and/or civilian airports
- **RouteMapComponent:** Specialized dynamic *MapComponent* with route information
- **RestrictedZoneComponent:** Specialized static *MapComponent* highlighting a specific zone on the GIS (using a *GeoPoint* collection)
- **TrajectoryMapComponent:** Specialized *RestrictedZoneComponent* with trajectory information
- **Metar:** Representation of weather information in standardize METAR format corresponding to *AirportMapComponent* aggregation
- **Weather:** Collection of weather information corresponding to *Metar* aggregation
- **Airfield:** Represents AirField related information corresponding to *AirportMapComponent*
- **MapPolygonComponent:::** Specialized dynamic *MapComponent* highlighting drawing specific structure on the GIS around a specified *GeoPoint*. This class is further specialized to draw planes, crews, ground support or missiles.
- **OneTypeMapComponentList:** Aggregates common type of *MapComponent* entities
- **MapComponentList:** Aggregation of *OneTypeMapComponentList*
- **MapComponentMover:** Animates dynamic map components for simulation purposes



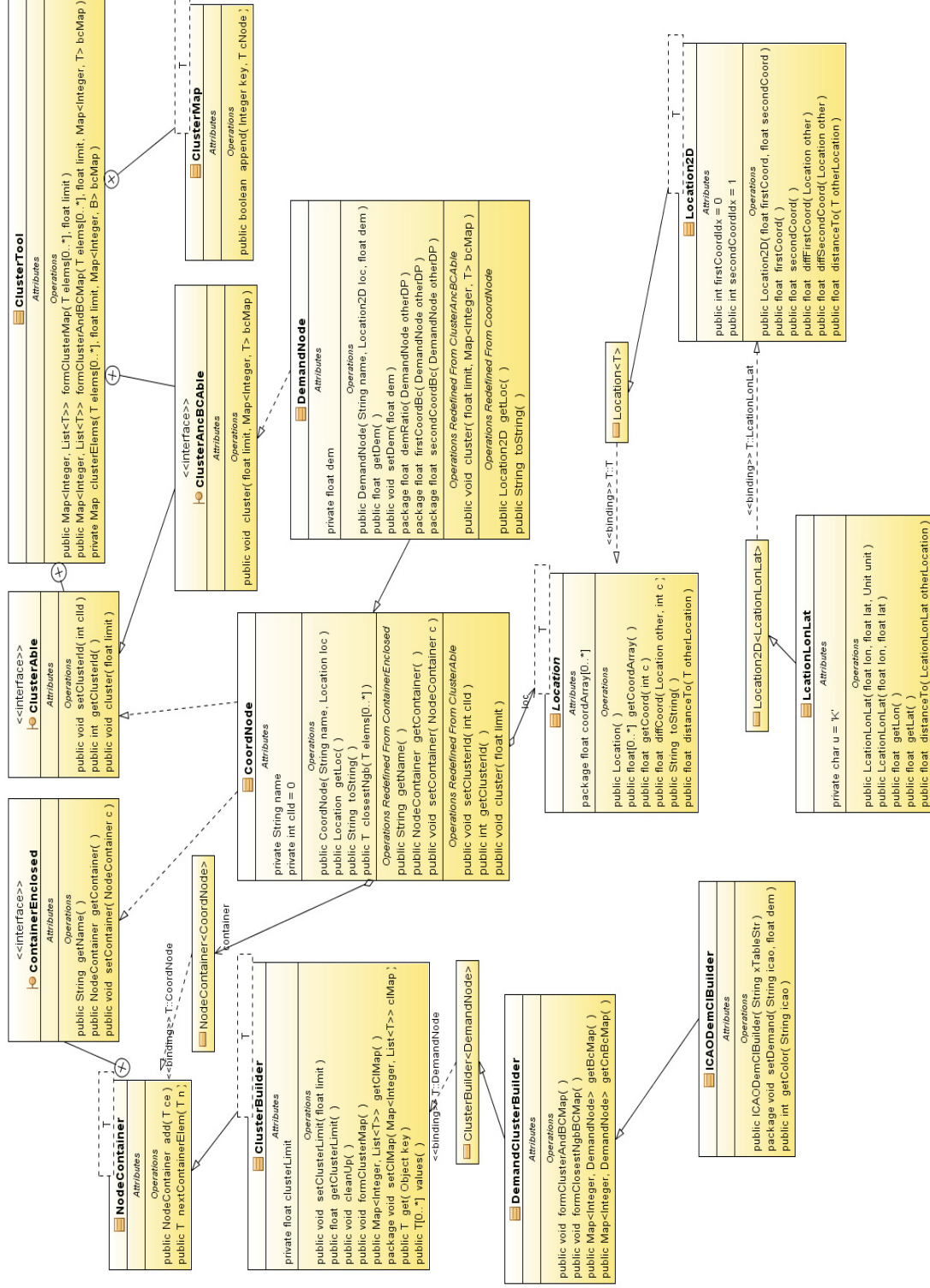


Figure 18. Node Clustering Generation

In Figure 18, we have the Class Diagram corresponding to the Node Clustering Generation. We present below a brief description of each of the classes.

- **ClusterTool**: Contains clustering interfaces and provides support for the clustering logic and barycenter identification
- **ClusterAble**: Interface required to be implemented in order to use the clustering logic provided in *ClusterTool*
- **ClusterAndBCAble**: Specialization of *ClusterAble* required to be implemented in order to use the barycenter logic provided in *ClusterTool*
- **ClusterMap**: Specialized Map used for holding cluster data
- **ContainerEnclosed**: Interface used for grouping elements
- **NodeContainer**: Contains *ContainerEnclosed* interface and provides functionality for successive element iteration
- **ClusterBuilder**: Extends *NodeContainer* and provides generic clustering logic
- **DemandClusterBuilder**: Extends *ClusterBuilder* and provides weighed barycenter logic
- **ICAODemCIBuilder**: Specialization of *DemandClusterBuilder* tailored for ICAO identified airport locations
- **CoordNode**: Holds Location and other node specific information and implements *ContainerEnclosed* and *ClusterAble* interfaces; aggregates *Location*
- **DemandNode**: Extension of *CoordNode* containing level of demand and implementing *ClusterAndBCAble* interface
- **Location**: Holds generic coordinates and related functionalities for distance calculation
- **Location2D**: Specialization of *Location* tailored for 2D coordinates
- **LocationLonLat**: Extension of *Location2D* specific for geographic coordinates, providing corresponding distance calculation function

IMPELEMENTATION & SCREENSHOTS

User Interface

The user interface has three main components, namely: DB-Explorer, Report Designer and Synchronizer. The first one is essential to integrate a new database to the Synchronizer application architecture. It allows the user to explore the database and to understand the relationships among the database components. The Report Designer is used for data-mining on some of the required tables/views based on the power of SQL queries that are used to generate and persist the corresponding reports. The persisted reports can also be presented graphically. Furthermore, they can be updated, sorted and filtered as per the user's choice. The third and the final component of the user interface is the Synchronizer where user can 'program/design' synchronization entries with different particularities among two tables A and B that are select from the databases to be synchronized. The scope of the synchronization is discussed later in a tabular format. Once the tables A and B are selected to be synchronized, change of any data from Table A will be automatically transferred to Table B without user intervention. The active Synchronizer client will also be remotely notified about the change.

Technology Stack

The prototype solution is mostly using open source and standardized technologies. The details of the technology stack can be found in the Appendix at the end of this document.

Database Integration

The client application has four modules namely: DB-Explorer, Reports, Synchronizer and visualization. The first three modules are related to database integration & synchronization.

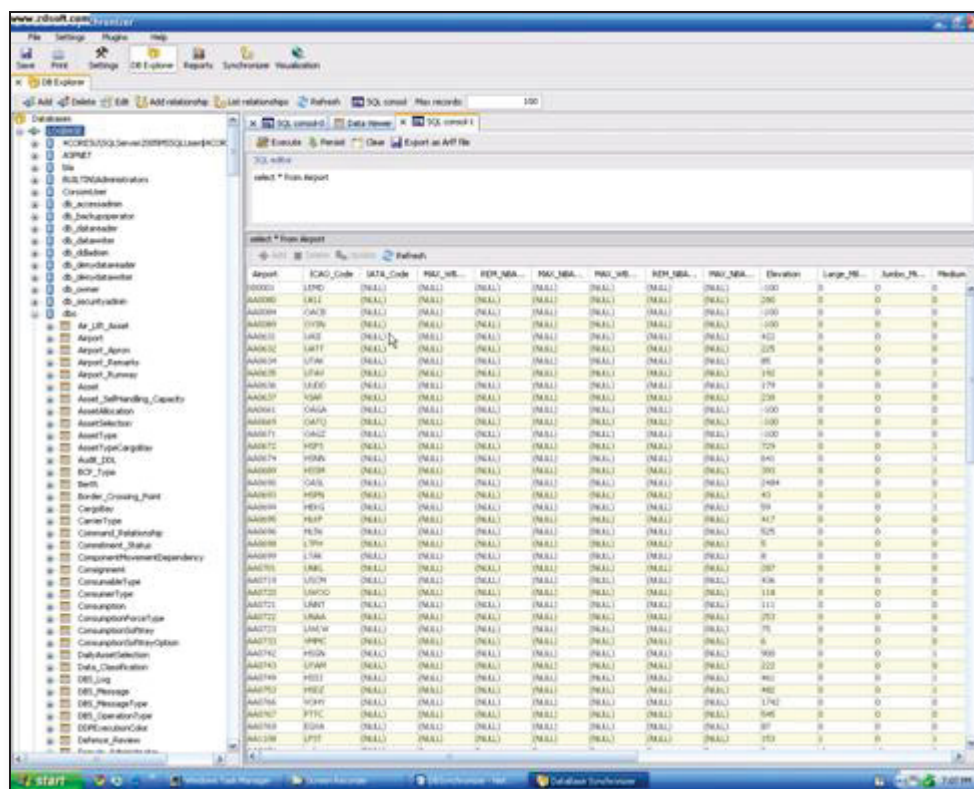


Figure 19: DB-Synchronizer data integration module interface

Figure 19 presents a screenshot from an easy to use software module where NMDS and LOGBASE databases are integrated and the related tables can be shown in the interface by the use of simple queries.

Figure 20 presents a reporting scenario from the LOGBASE database. Reports are named in a tree structure at the left of the plug-in. There are many features available for the report. The edit option allows for adding or editing charts to the reports. The chart may be altered by using the “Change Graph” feature using the corresponding button in the toolbar of the chart-tab. Moreover, in the window that opens with the information of the current report, there is a “Has a Chart” option. By selecting it, one can choose the chart type and change the X and Y value with the required column names (the ‘value’ column should have numerical or time/date values). The user may also filter a part from the main report according to his/her need. The filtering capability allows the user to see specific information from a table by selecting the desired columns to be displayed and by specifying ranges of values for a column.

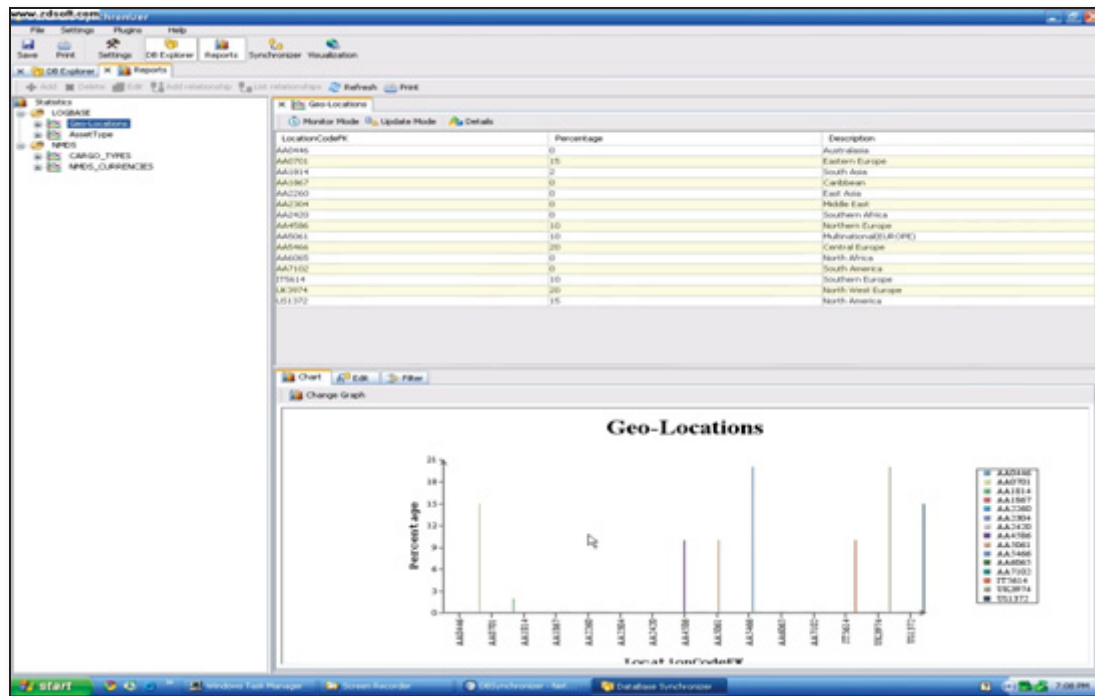


Figure 20: DB-Synchronizer Reporting module interface

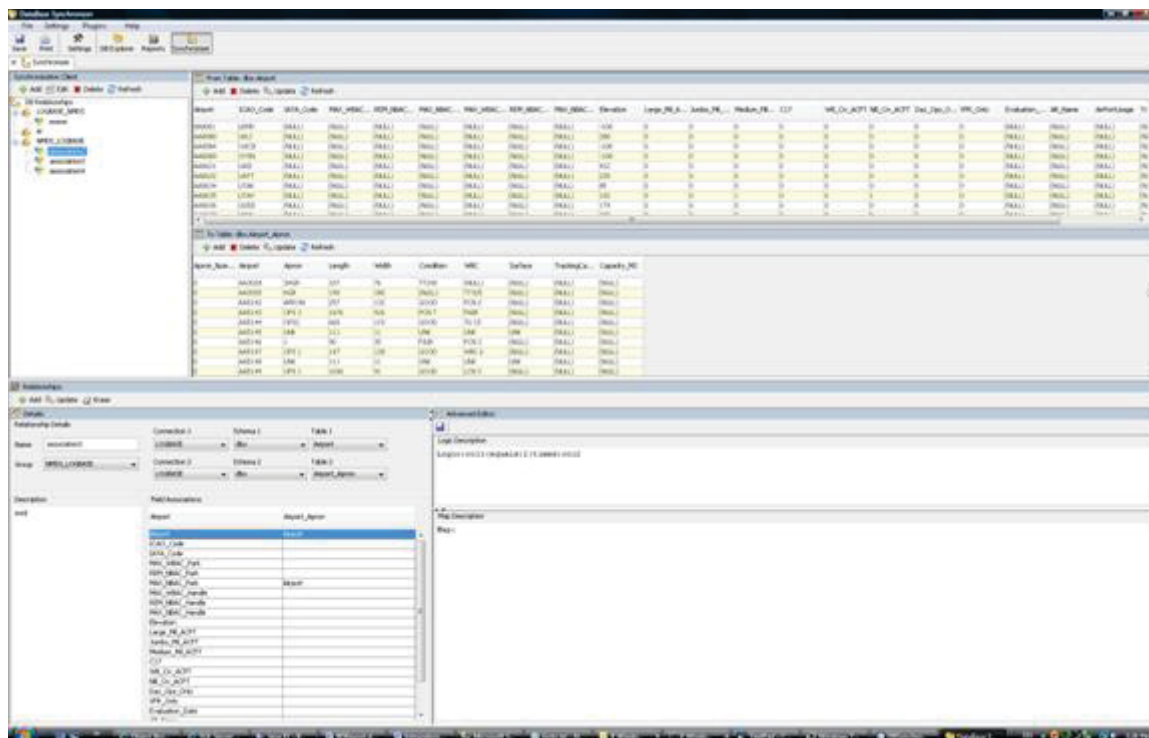


Figure 21: DB-Synchronizer: Synchronization designer module

Figure 21 depicts the visual interface of the DB-Synchronizer Synchronization Design module where users may specify the synchronization parameters between two tables in two databases. The two tables are shown in the right side. Moreover, at the bottom, the interface of the synchronization engine is provided. The user can use the interface in order to provide advanced synchronization mapping of values including the use of arithmetic, string and mixed expressions. This is especially useful in the case where the columns of the tables to be synchronized are of different type (e.g. numeric vs. string) or require the use of enumerated values of abbreviations that are distinct among the tables. The design of the parser and expression evaluator was presented in Figure 12, along with an accompanying description of the related design components.

Clustering

In this section, we present a number of screenshots that have been taken from the developed client application interface. In Figure 22, we can see the feature of cluster formation based on the clustering distance adjustable parameter. We see the forming of clusters while sliding the configurable cluster distance interface control. In this context, it must be mentioned that the nodes that are further away from any other node than the clustering distance, do not form proper clusters (are not coloured with a specific cluster color in the map) and can be viewed as degenerated clusters of a single node.

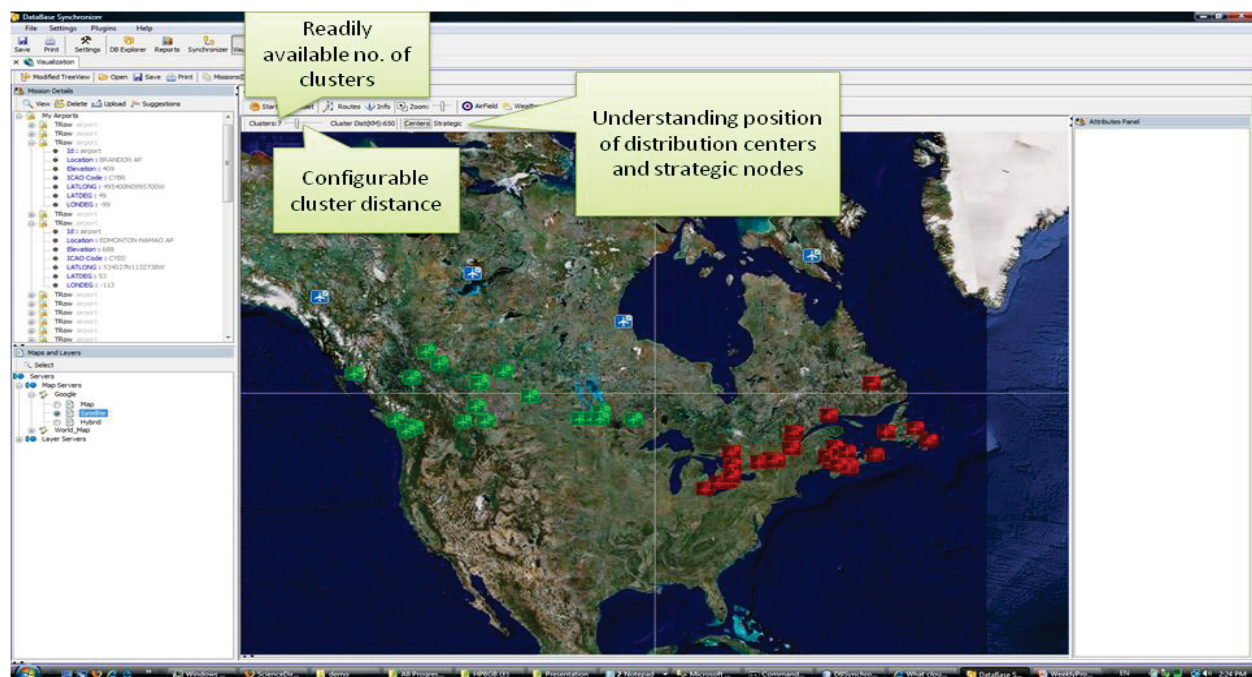


Figure 22. Node Clustering Interface

Figure 23 shows cluster heads identification corresponding to the potential location for the placement of the distribution centers. In the map, we can see the cluster heads surrounded by blue circles. The cluster heads are identified in a manner that is similar to that used for identifying a barycenter by taking into account both the distance and the demand level of the cluster nodes. In this setting, it must be noted that each nodes that forms a degenerated cluster represents its own barycentre.

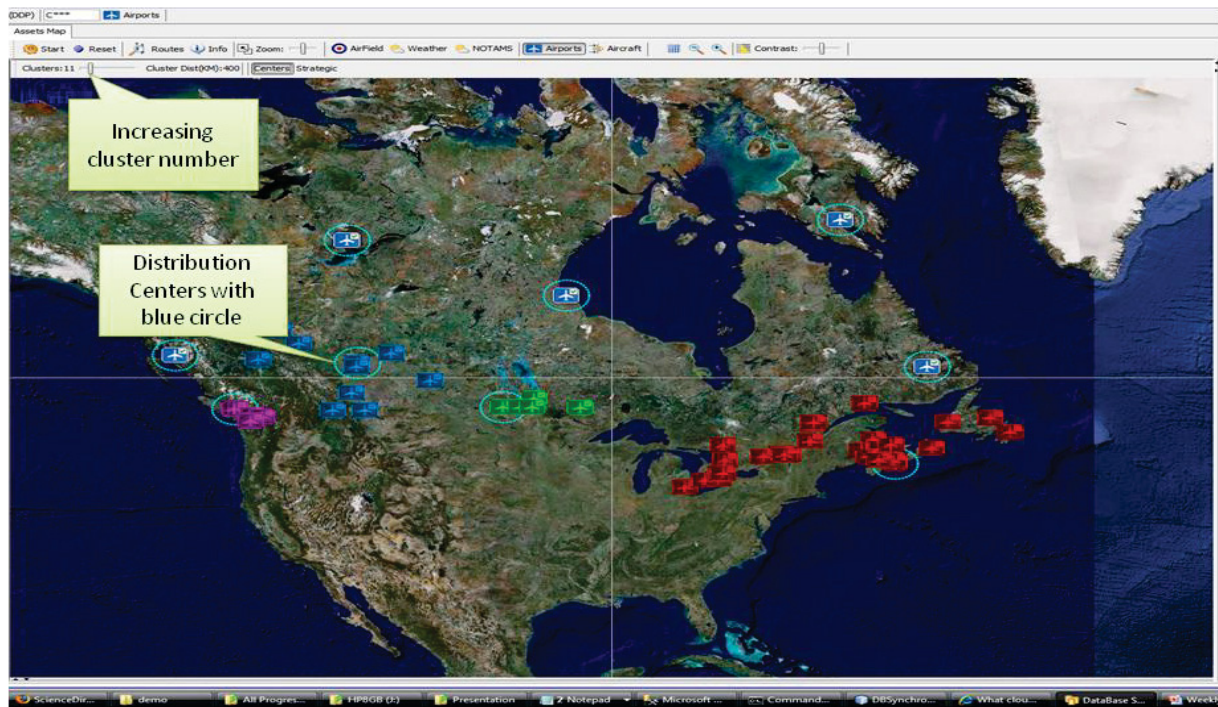


Figure 23. Identification of Cluster Heads / Distribution Centers

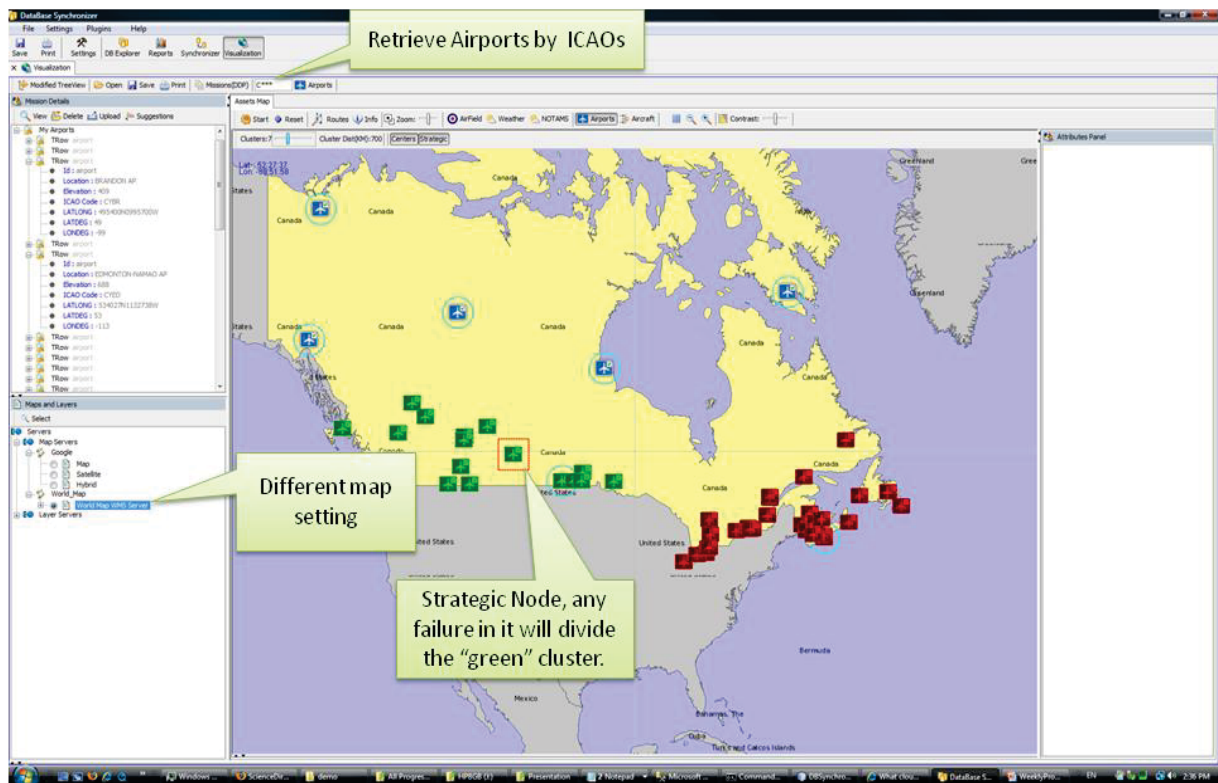


Figure 24. Strategic Node Identification

In Figure 24 we can see along with the identification of the cluster heads, the identification of the strategic node in the green cluster, surrounded by a red square in a different map setting. A strategic node in a cluster is essentially any node that has the property that if removed from the cluster, this would cause the cluster to break in two or more smaller clusters. Strategic nodes are generally very important nodes in a cluster as they may represent potential single-point of failure for communications or transport. The design of the clustering solution was presented in Figure 18 along with accompanying explanations of the related component parts.

Figure 25 shows the disabling of the strategic node and the subsequent splitting of the green cluster.

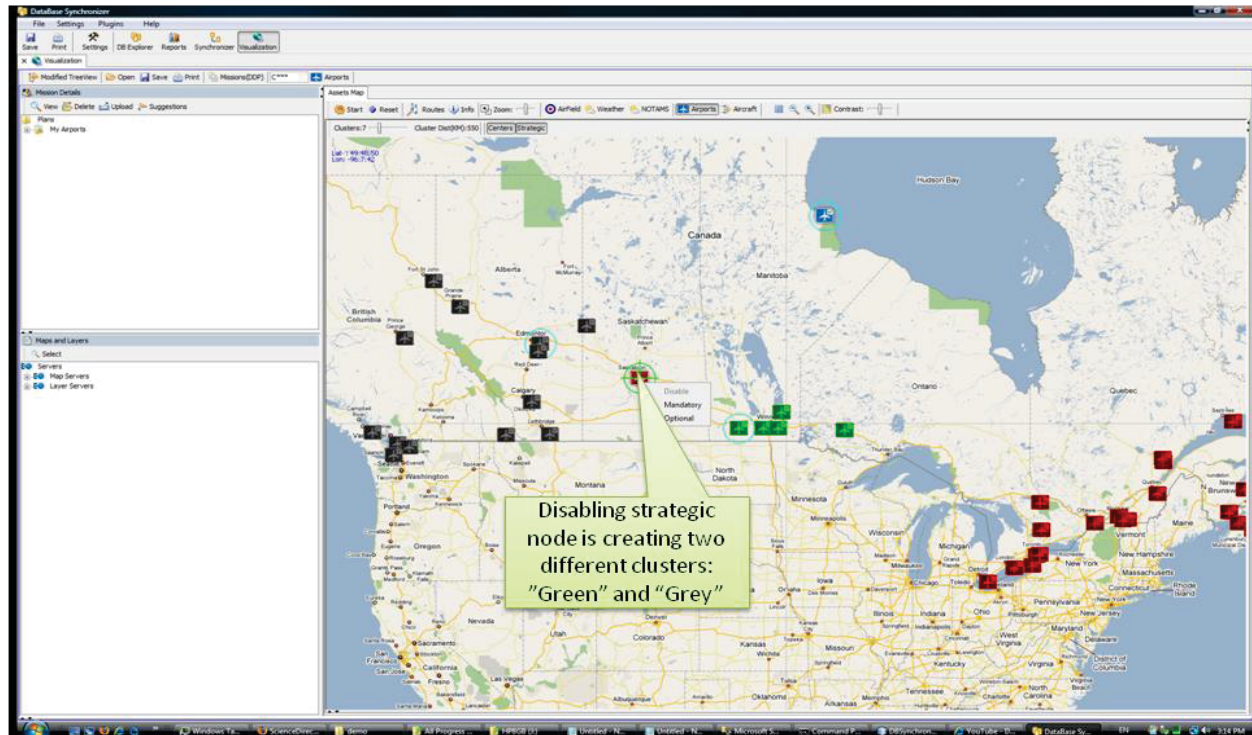


Figure 25. Disabling of Strategic Node

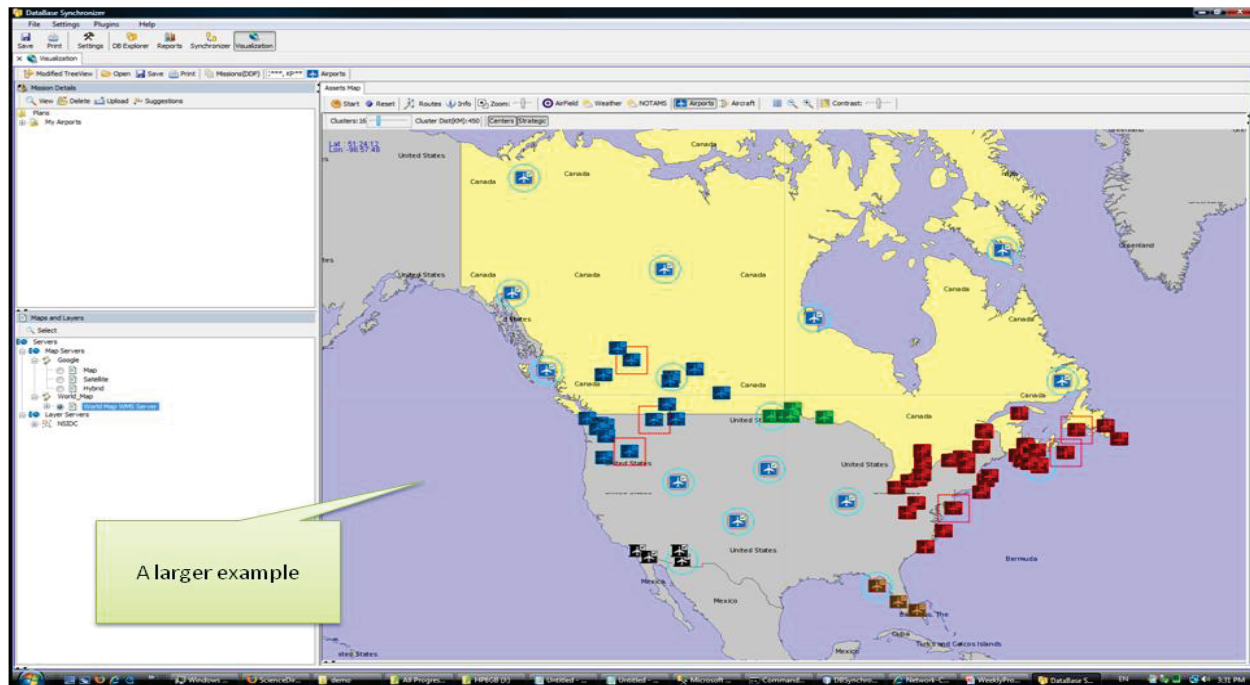


Figure 26. Larger Example of Clustering, Cluster Head and Strategic Node Identification

In Figure 26 we can see a larger example of node clustering, along with the identification of the corresponding cluster heads and strategic nodes. In this example, we can observe from the figure that the red cluster has two adjacent strategic nodes in the right side.

Plan Re-routing

In Figure 27 we can see an example of DDP mission displayed on the GIS. The mission involves the movement (with intermediate stop-overs) of a plane from a source airport (ICAO code: EBBR) to a destination airport (ICAO code: XMMX) and back. An intermediate stop is scheduled at the airport with the ICAO code: LGSA. In this context, we consider that the LGSA airport becomes unavailable due to an exogenous event (e.g. severe weather) and is marked in red after the corresponding status has been set by the user. After that, the user can ask the framework for mission suggestions and inspect them as shown in Figure 28. The system is then fetching information from the data sources, in this case, the LOGBASE database and is searching for airports nearby to the location of the disabled one. After retrieving the information, an adaptive planning model is constructed based on the initial and retrieved information. The model encodes the movement dynamic and the cost associated to the distances. Then, the model is assessed and a number of alternated routes are found and ranked based on their corresponding cost.

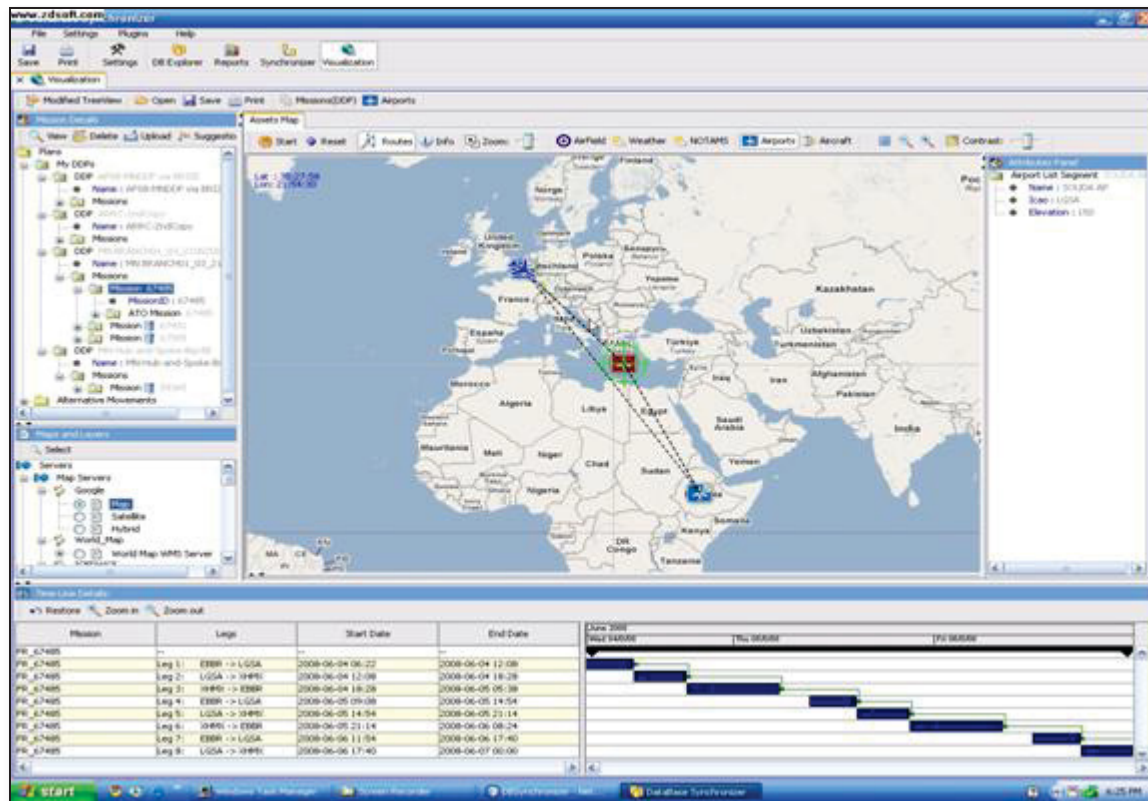


Figure 27: Display of mission DDP on GIS system



Figure 28: Display of alternative routes(in green) to a resource allocation plan.

Simulation

In case of a plan alteration, the user has the possibility to select its mission alternative of choice (usually the one with the smallest cost) and simulate it as seen in Figure 29. However, if needed, this feature can also be used on the initial plan as well. We can see in the bottom part of figure the scheduled legs of the flight in the left along with a Gantt chart at the right side showing the corresponding projected time line.

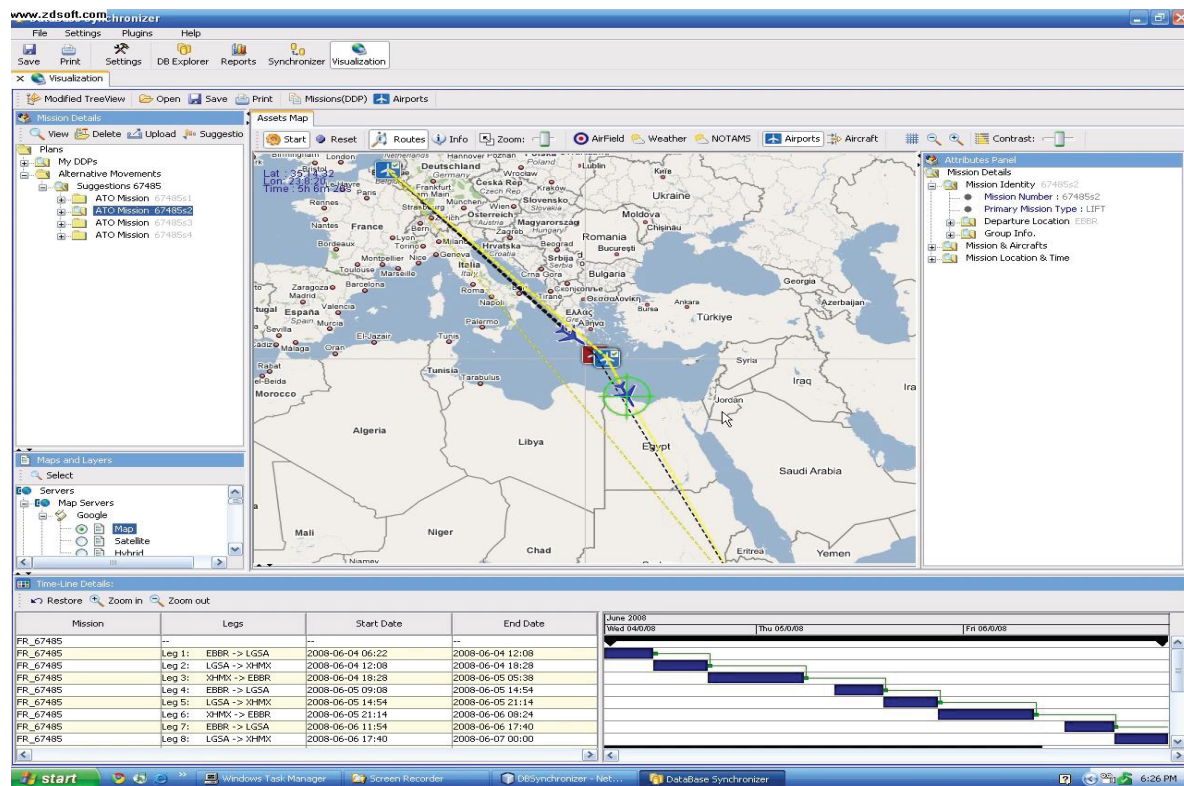


Figure 29: Simulation of mission plans and alternatives

Adaptive Planning

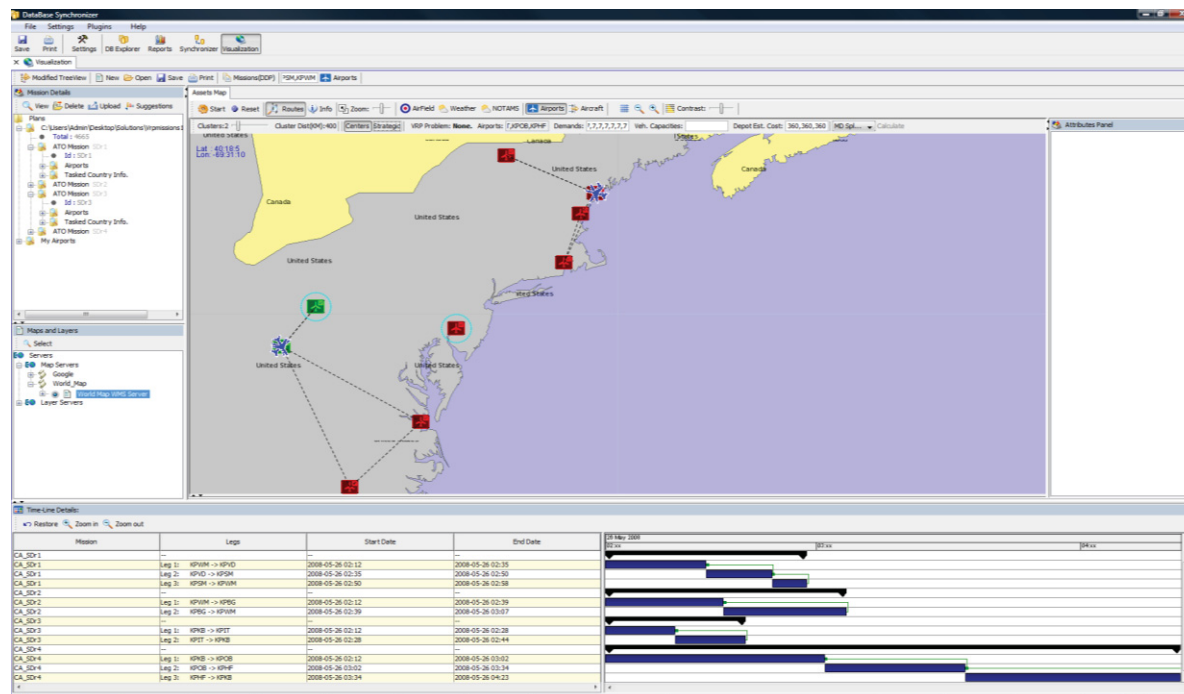


Figure 30: Display of adaptive planning using MDSDVRP planning logic in existence of two clusters.

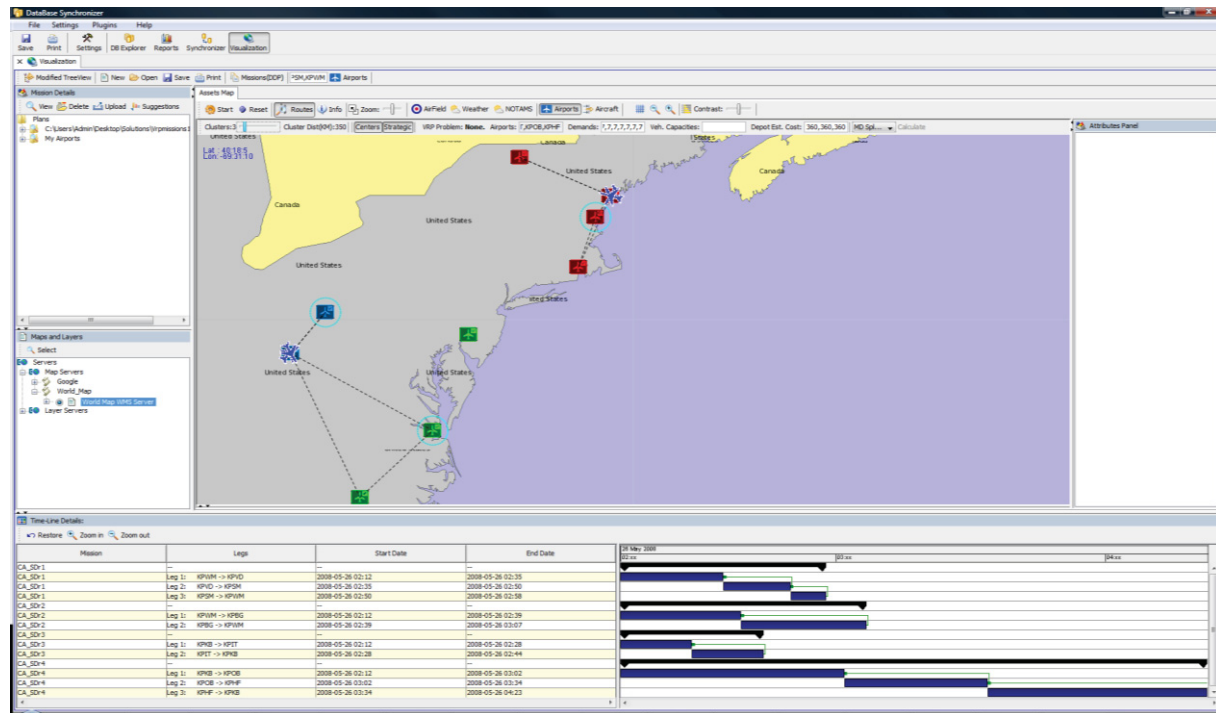


Figure 31: Display of same adaptive planning using MDSDVRP planning logic in existence of three clusters.

Figure 30 and Figure 31 are presenting an important example of an adaptive planning within two different clustering possibilities. The plan is created by using the Multi-Depot Split-Delivery Vehicle Routing Problem (MDSDVRP) model for the transportation. The figure is clearly showing that by solving the model it is possible to select the depots much better (optimally) than by using the usual clustering techniques. To that effect, we can note that the dashed blue circled cluster heads are not corresponding to the actual position of depots (where the planes are located) in this case. Moreover, another benefit is that the MDSDVRP model can provide solutions involving the serving of the same customer(s) by more than one vehicle without necessitating the serving vehicles to start from the same depot. This can mitigate potential issues that can arise in certain cases along the cluster borders.

ATO Mission Structure Generation

In order to have a mission structure that can be versatile and suitable for different persistence and retrieval mechanisms, potentially across many platforms, we opted to use the XML-based ATO structure commonly utilized for coalition or joint flying missions. In this context, it is important to be able to have a common mission view. Moreover, the ATO mission structure allows for an effective mission description by having specific sections in the XML structure related to each country that may participate in a joint operation. Also, the aircraft mission is described at specific levels such as Air Movements that contain information about the mission legs. This particular feature was used in the generation of the alternative mission suggestions, which can be inserted in the corresponding place within the XML structure of the ATO data. This way, the main structure of the mission does not need to be regenerated each time from the DDP mission that is fetched from the LOGBase database. The design of the ATO mission generator and parser has been presented in Figure 10 and Figure 16 respectively.

CONCLUSION

The main emphasis of this project was put on the requirements analysis and design considerations related to the Net-Centric Logistic Framework for Adaptive Operational Planning prototype. The design and architecture of the prototype was considered based on extensive research for a decision support platform that envisions near real-time, asynchronous data integration and monitoring along with automatic information synchronization. Furthermore, during this phase research activities have been conducted with respect to the adaptive planning in the context of vehicle/asset movement within a shared transport network. The rationale for using automatic model assessment techniques, namely model checking and heuristic based vehicle routing problem solving along with the details of a structured procedure for achieving optimal decision making support in the presence of uncertainty was considered. In this pursuit, further research efforts have been dedicated for refining the proposed procedure at different levels of abstraction. In this document we provide detailed information on the designed and implemented prototype to achieve the primary objectives.

The present DND initiative is of paramount importance and in the context of information sharing and military planning and is aiming to provide automatic synchronization of military data sources in pursuit of enhanced situational awareness by employing automatic synchronization of information hosted in two databases that were otherwise updated manually. It will definitely provide a significant edge to the military decision makers that will benefit by receiving timely and updated information. Moreover, it will allow the military planners to use advanced adaptive planning techniques for improved resource allocation and distribution as well as personnel movement in the presence of uncertainty generated by exogenous events. Furthermore, the presented architecture of the platform is able to achieve such capabilities in a generic way and allows to be further tailored to meet more specific requirements and constraints.

References

- [1]. Hassan Issa, Chadi Assi, Mourad Debbabi, Sujoy Ray. "*QoS-aware middleware for web services composition: a qualitative approach*" Published in Enterprise Information Systems, Volume 3, Issue 4, pages 449 - 470, November 2009.
- [2]. Yosr Jarraya, Andrei Soeanu, Luay Alawneh, Mourad Debbabi, and Fawzi Hassane; "*Synergistic Verification and Validation of Systems and Software Engineering Models*", Journal of General Systems, Taylor & Francis Group Publishers, 2009.
- [3]. Vitor Lima, Chamseddine Talhi, Djedjiga Mouheb, Mourad Debbabi, Lingyu Wang, Makan Pourzandi; "*Formal Verification and Validation of UML 2.0 Sequence Diagrams using Source and Destination of Messages*". Electr. Notes Theoretical Computer Science 254: 143-160, 2009.
- [4]. Christel Baier, Joost-Pieter Katoen, "*Principles of model checking*", MIT Press, 2008.
- [5]. Doron A. Peled, Michael J. Wooldridge; "*Model Checking and Artificial Intelligence*", 5th International Workshop, Patras, Greece, Springer, 2008.
- [6]. Bruce L. Golden, S. Raghavan, Edward A. Wasil; "*The Vehicle Routing Problem: Latest Advances and New Challenges*"; ISBN: 978-0-387-77777-1, 2008.
- [7]. Hadi Otrok, Mona Mehrandish, Chadi Assi, Mourad Debbabi, Prabir Bhattacharya; "*Game theoretic models for detecting network intrusions*". Computer Communications 31(10), 2008.
- [8]. Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay Vazirani; "*Algorithmic game theory*", Cambridge University Press - 2007.
- [9]. James N. Webb; "*Game theory: decisions, interaction and evolution*", Springer-Verlag, London 2007.
- [10]. Dr Allan English, Colonel (retired) John Westrop. "*Canadian Air Force Leadership And Command: Implications For The Human Dimension Of Expeditionary Air Force Operations*", Contract Report, Defence R&D Canada – Toronto, 2006.
- [11]. L. Alawneh, M. Debbabi, F. Hassaine, Y. Jarraya, P. Shahi, A. Soeanu. "*Towards a Unified Paradigm for Verification and Validation of Systems Engineering Design Models*". In the Proceedings of the International Conference on Software Engineering, SE'2006, February 14-16, 2006, Innsbruck, Austria, Pages 282-287, ACTA Press.
- [12]. Prasad Sistla and Min Zhou, "*Combining Static Analysis and Model Checking for Systems Employing Commutative Functions*", in FORTE, LNCS, 2005.

- [13]. A.Benssam, A.Boukhtouta, M. Debbabi, H. Issa, S. Ray and A. Sahi, "*A Message-Based Middleware for Asynchronous Operations: Issues and Experience*", in: Proceedings of NOTERE-'05, Ottawa, Canada, pp. 51-60, 2005.
- [14]. Kurt Jensen, Andreas Podelski, "*Tools and algorithms for the construction and analysis of systems*", TACAS 2004, Barcelona, Spain, April 2004.
- [15]. Rajive Joshi, Real-Time Innovations, Inc., "*A comparison and mapping of Data Distribution Service (DDS) and Java Message Service (JMS)*", White Paper, [online], <http://www.dsp-fpga.com/articles/id/?1852>, December 2004.
- [16]. NATO Standardization Agency, "*Allied Joint Logistic Doctrine*", [online] <http://www.nato.int/docu/stanag/ajp4/ajp-4.pdf>, 2003.
- [17]. Sonic Software Corporation, "*JMS Performance Comparison: Publish Subscribe Messaging*", White paper, [online], http://www.sonicsoftware.com/products/whitepapers/docs/jms_comparison_tibco.pdf, November 2003.
- [18]. Joachim Rosenmüller, "*Game theory: stochastic, information, strategies, and cooperation*"; Kluwer Academic Publishers, ISBN: 0792386736, 2000.
- [19]. M. Clarke, Orna Grumberg, Doron Peled, "*Model checking*", MIT Press, ISBN 0-262-03270-8, 1999.

Appendix

Technology Stack

| Level | Challenges | Software/Application | Providers |
|--|---|---|---|
| Sources of Information (Server Side) | <ul style="list-style-type: none"> • Heterogeneity of data sources. • Distant locations of data sources. | <ul style="list-style-type: none"> • SQL-Server database (LOGBASE) • Oracle 10G database (NMDS) / Oracle 10G XE for development (NMDS) | DND, Canada (Scientific Authority and other contact persons) |
| Data & Services Integration (Server Side & Client Side) | <ul style="list-style-type: none"> • Heterogeneity of platforms, sources, network, applications. • Distributed nature of the system | <ul style="list-style-type: none"> • Java 2 Enterprise Edition. • Geronimo application server. • Web services stack of Axis API applications integration. • JMS service for data integration. | <p>Java 1.6 from Sun Microsystems (Server)</p> <p>Apache Foundation (Geronimo server and AXIS API).</p> <p>ActiveMQ JMS comes with Geronimo.</p> |
| Information Display and Monitoring | <ul style="list-style-type: none"> • Data visualization. • Real-time refresh. • Performance. | <ul style="list-style-type: none"> • Graphs and Charts APIs (EspressChart) • egantt Gantt Chart API • JMS listeners • Plugin-based user interface (Java Plugin Framework). | <p>Quadbase Inc.</p> <p>Map tile pictures are available from Google, Microsoft, Mapsolute, etc.*</p> <p>JMS Client from Apache Foundation.</p> <p>JPF is available at Sourceforge.*</p> |
| Message Oriented Middleware | <ul style="list-style-type: none"> • Heterogeneity of data sources • Proposer message format identification • Interfacing with the databases and the client software application | <ul style="list-style-type: none"> • Message-based middleware implemented using Java Message Service is for the development purpose. | ActiveMQ Java Message Service from Apache Foundation. |
| Information Display and Monitoring | <ul style="list-style-type: none"> • Data visualization. • Real-time refresh. • Performance. | <ul style="list-style-type: none"> • Core Java In-House applications • Graphs and Charts APIs (EspressChart) | <p>Java 1.6 from Sun Microsystems (Client)</p> <p>Quadbase Inc.</p> <p>ActiveMQ JMS Client</p> |

| | | | |
|----------------------|---|---|---|
| | | <ul style="list-style-type: none"> • JMS listeners • WSDL2Java API for Web Service Access | <p>from Apache Foundation.</p> <p>WSDL2Java API from Apache Foundation.</p> |
| Adaptive Planning | <ul style="list-style-type: none"> • Use of Model Checking • Computation Optimization • Faster reaction on exogeneous events • Combinatory explotion of state-space | <ul style="list-style-type: none"> • PRISM • NUSMV • Core Java software modules | <p>University of Oxford, Prismmodelchecker.org</p> <p>Joint effort IRST-ITC (Italy), CMU (USA)</p> <p>TrustSeer Model Checking Research</p> |
| MDSDVRP model Solver | <ul style="list-style-type: none"> • Mixed Integer Linear Programming Modeling (optimization problems) | <ul style="list-style-type: none"> • OpenSource GLPSol | <p>GLPK (GNU Linear Pogramming Kit)</p> |

Definitions & Acronyms

| Abbreviation | Meaning |
|--------------|---|
| API | Application Programming Interface |
| C2 | Command and Control |
| C4ISR | Command, Control, Communications, Computers, Intelligence Surveillance & Reconnaissance |
| CAAT-Xi | Canadian ACO ATO XML interpreter |
| CF | Canadian Forces |
| COTS | Commercial off the Shelf |
| DND | Department of National Defence |
| DRDC | Defence Research and Development Canada |
| EVE | Effective Visible Execution |
| JMS | Java Message Service |
| LOGFAS | Logistic Functional Area Services |
| NATO | North Atlantic Treaty Organisation |
| NMDS | National Movement Distribution System |
| PC | Personal Computer |
| RFID | Radio Frequency Identification |
| SQL | Structured Query Language |
| TCP/IP | Transport Control Protocol/Internet protocol |
| TD | Technology Demonstration |
| WS | Web-Services |
| WSDL | Web Services Description Language |
| XML | Extended Mark-up Language |

| DOCUMENT CONTROL DATA | | |
|---|--|--|
| (Security markings for the title, abstract and indexing annotation must be entered when the document is Classified or Designated) | | |
| 1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.) TrustSeer Corporation C.P. MacKay, P.O. BOX 55002 Montreal, Québec, H3G 2W5 | 2a. SECURITY MARKING (Overall security marking of the document including special supplemental markings if applicable.) UNCLASSIFIED | |
| | 2b. CONTROLLED GOODS (NON-CONTROLLED GOODS) DMC A REVIEW: GCEC APRIL 2011 | |
| 3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.) On database Synchronization & Logistics Adaptive Planning Framework : Net-Centric Logistics Framework Development For Adaptive Operational Planning | | |
| 4. AUTHORS (last name, followed by initials – ranks, titles, etc. not to be used) Debbabi, M. | | |
| 5. DATE OF PUBLICATION (Month and year of publication of document.) March 2011 | 6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.) 60 | 6b. NO. OF REFS (Total cited in document.) 19 |
| 7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.) Contract Report | | |
| 8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.) Defence Research and Development Canada – Valcartier 2459 Pie-XI Blvd North Quebec (Quebec) G3J 1X5 Canada | | |
| 9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.) 15ae | 9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.) W7701-9-3030 | |
| 10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.) | 10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.) DRDC Valcartier CR 2011-624 | |
| 11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.) Unlimited | | |
| 12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.) Unlimited | | |

13. **ABSTRACT** (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

This document contains information on the design and implementation of the database synchronizer test-bed, along with a centralized planning framework over it. The purpose of the document is to present the software life cycle according to the IEEE 12007 standard. Here we illustrate the requirements analysis followed by software design, architecture and implementation details. The main intent of the project is to integrate and synchronize these two systems such that any change of information in one of these systems is directly and automatically reflected in the other system in near real-time. This involves the elaboration of a centralized framework and related procedures for the integration and synchronization of two databases, namely, LogBase (the database behind EVE application) hosted on Microsoft SQL-Server and NMDS hosted on Oracle. Such a net-centric logistics framework will allow access and will synchronize multi-level information sources while constantly monitoring/tracking assets information in order to present it to the users. On top of the data synchronization framework, we have designed and implemented net-centric logistics framework enabled by various information sources at different levels. This information is enabling us to create and simulate logistics military missions in relation to contingency planning, monitoring and analysis. This proof-of-concept extensively demonstrates the implementation of theoretical logistic planning using vehicle routing problems on a specialized client-server framework.

Ce document contient de l'information sur la conception et la mise en œuvre du banc d'essai du synchroniseur de base de données, ainsi que d'un cadre de planification centralisé. Le but de ce document est de présenter le cycle de vie du logiciel selon la norme IEEE 12007. Nous illustrons ici l'analyse des besoins suivie des détails de la conception, de l'architecture et de la mise en œuvre du logiciel. L'objectif principal du projet est d'intégrer et de synchroniser ces deux systèmes de sorte que tout changement d'information dans l'un de ces systèmes se reflète directement et automatiquement dans l'autre système en temps quasi réel. Ceci comprend l'élaboration d'un cadre centralisé et des procédures connexes pour l'intégration et la synchronisation de deux bases de données, à savoir LogBase (la base de données derrière l'application EVE) hébergée sur Microsoft SQL Server et le SNDM hébergé sur Oracle. Un tel cadre logistique réseaucentrique permettra l'accès et synchronisera les sources d'information à plusieurs niveaux tout en surveillant et en suivant constamment l'information sur les biens afin de la présenter aux utilisateurs. En plus du cadre de synchronisation des données, nous avons conçu et mis en œuvre un cadre logistique réseaucentrique reposant sur diverses sources d'information à différents niveaux. Cette information nous permet de créer et de simuler des missions logistiques militaires liées à la planification d'urgence, à la surveillance et à l'analyse. Cette validation de principe démontre de manière approfondie la mise en œuvre de la planification logistique théorique au moyen de problèmes de routage de véhicules sur un cadre client-serveur spécialisé.

14. **KEYWORDS, DESCRIPTORS or IDENTIFIERS** (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

data synchronization; net-centric logistics framework; Adaptive Operational Planning; IEEE 12007 standard; LogBase; EVE application; Microsoft SQL-Server; NMDS; Oracle; contingency planning; client-server framework

Defence R&D Canada

Canada's Leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale



www.drdc-rddc.gc.ca