UTILIZATION OF COMPUTER SIMULATION FOR THE LOCALIZATION OF ROLLING STOCK WITH UTILIZATION TECHNOLOGY ORACLE SPATIAL AND DYNAMIC DATABASE VIEWS

Jan Fikejz[(a)] , Jan Merta[(b)]

[(a)] Department of Software Technologies, FEI, University of Pardubice, Pardubice, Czech Republic
[(b)] Department of Software Technologies, FEI, University of Pardubice, Pardubice, Czech Republic

[(a)] Jan.Fikejz@upce.cz ,[(b)] Jan.Merta@student.upce.cz

ABSTRACT

This article deals with utilization of computer simulation for the localization of rolling stock on the railway network using dynamic views. Attention is focused on the description of the location of rolling stock in the designed model of railway network. Further attention is aimed on optimizing search operations ORACLE database and designing of the optimization using dynamic views.

Keywords: Railway infrastructure models, train positioning, railway traffic simulation, database optimization, database dynamic views

1. INTRODUCTION

Rolling stock localization is a constantly discussed topic involving a lot of subjects. The problem of rolling stock localization may be divided in two main areas of interest: localization for the needs of (i) security technology and localization for the needs of (ii) information and telematics systems. In the first case the reliability and safety are emphasized, nevertheless these systems are often connected with higher realization costs due to the fact that they often require adding further communication or identification elements/devices to railway infrastructure.

In the latter, a certain scope of inaccuracy or reduced reliability can be accepted, which on the other hand often results in a significantly lower implementation of such solutions. Rolling stock localization has recently been connected with the use of satellite navigation system (GNSS – Global Navigation Satellite System).

2. POSSIBLE TYPES OF LOCALIZATION

Rolling stock localization can be divided into three main parts:

- localization without the use of GNSS,
- GNSS using localization,
- GNSS-based, involving further support systems.

2.1. Rolling stock localization without the use of GNSS

This type of rolling stock localization often requires complementing the rail network infrastructure with additional construction elements, which entails higher costs of the actual implementation. On the other hand, this type of localization shows a high accuracy and reliability and is often used in the railway signalling technology. Essentially, it relates to the system of:

- ETCS (Ghazel 2104; Lieskovský and Myslivec 2010),
- Automatic train control (Chudacek and Lochman 1998; Lieskovský 2004),
- Track circuits (Dorazil 2008),
- RFID.

2.2. Rolling stock localization using GNSS

If we use GNSS for various application levels, we need to take into account an indicated position error, which is generally based on the nature of the satellite navigation. If we use systems that operate with the position information only on an informative level, we can tolerate a certain error; however, such inaccuracy is unacceptable in the railway signalling technology. However, various additional systems can be implemented to eliminate the error (completely or at least partially), thus making the position of the tracked object more accurate. The following systems can be listed in this group:

- EGNOS (Senesi 2012)
- Differential GPS (O'Connor 1997)

2.3. GNSS based localization involving additional support systems

As mentioned above, precise localization of rolling stock using GNSS, especially for the needs of signalling technology, is a priori impossible. Nevertheless, the position of a rail vehicle can be put significantly more precisely with the use of additional systems. This concerns especially the solutions using inertial systems

(Standlmann 2006), but also less known systems such as those based on GNSS and contactless eddy current measurement (Becker and Poliak 2008)

3. RAILWAY NETWORK MODEL

Undirected graph, as defined graph theory, is a natural candidate for a railway network model. Based on an analysis of data provided by the company SŽDC-TUDC (consisting of service regulations, passports and codebooks), sets of algorithms were subsequently created, with which it was possible to generate a three-layer model of the rail network (Fikejz and Kavička, 2011). Roughly speaking, the track can be divided into individual so called supertracks, which consist of definition supra-sections (TDNU), where each supra-section contains track definition sections (TUDU) with mileposts (in hectometres). Basic aspects of the description of the rail network are collectively shown in Figure 1.

Mileposts (in hectometres) are shown in Figure with the distance in kilometres and are graphically represented using grey points. TUDU is recorded using a six-digit code (163105, 163106, 16307, 173202) and are graphically represented using solid lines (red, black, orange, brown). Individual supra-sections (CLS 007, CLS008, REG023) are shown in light blue and supertracks (B421021 1 and B421021 1A) are shown in dashed lines. A place significant in terms of transportation (branch line) is symbolized by a green square.

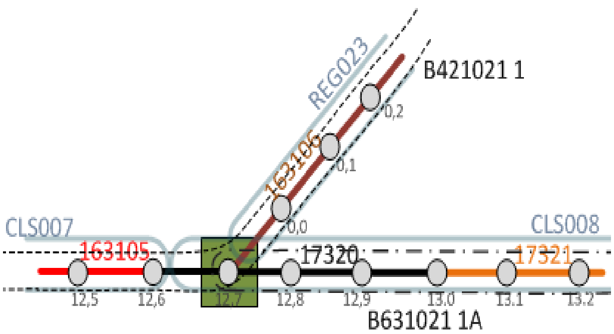

Figure 1: Basic aspects of the description of the rail network

The algorithm of railway network model (Fikejz and Kavička, 2011; Fikejz and Řezanina 2014) was implemented directly on the database level using PL/SQL language. However, the algorithm had to be adjusted and generalized several times since there are various nonstandard conditions in the data, such as jumps in the mileposts (nonlinear growth of the kilometre succession between the mileposts) or change of an increasing kilometre sequence into a decreasing one and vice versa. The final model include three data layers:

- **Data-micro**, consisting of vertices and edges,
- **Data-macro**, containing super-vertices and super-edges,
- **Data-macro**, include mezo-vertices and mezo-edges.

The data structure non-oriented graph was finally implemented directly in the ORCLE database using the ORACLE Spatial Network Data Model (Kothuriat al. 2007) technology. This technology enables the user to build a various network representations, involving also the object scheme and the communication interface API.

The objects scheme includes metadata and network tables. The interface contains on the server side PL/SQL API (an SDO_NET packet) for the creation, control and analysis of the database network, and a middle layer Java API (on client´s side) for the network analysis. The actual network is then defined by means of two compulsory tables:

- Node table,
- Link table.

Concept of the technology is described in Figure 2.

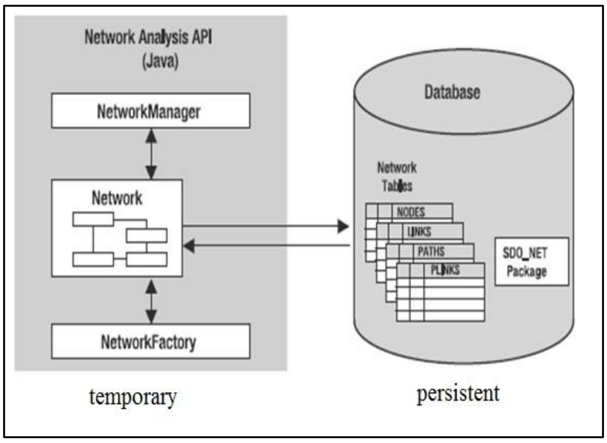

Figure 2: Oracle Network Data Model

For the work with spatial data, ORACLE with Spatial technology defines a special object data type SDO_GEOMETRY, which enables its user to store a number of spatial information and geometric types, such as various points, arcs, linear chains or polygons.

4. VISUALIZATION

For the purposes of visualization of proposed railway infrastructure model visualization tool MapViewer developed in the JAVA language (Murray et al. 2010) was employed. MapViewer represents J2EE service for displaying maps using the spatial data (e.g. using object data type SDO_GEOMETRY) managed by ORACLE Spatial. This technology enables to compose extensive map layers with various levels of details of information displayed (Fikejz and Kavička 2012) of the described information (Fikejz and Kavička, 2011). The basic concept of the MapViewer architecture is demonstrated in Figure 3.

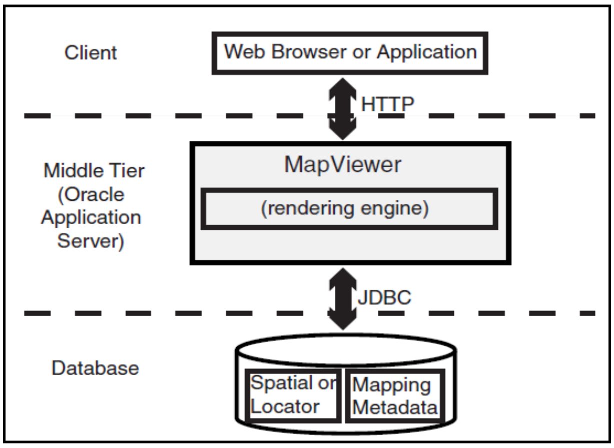

Figure 3: Basic concept of MapViewer service

It is possible to use various operators and functions above spatial objects of ORACLE database with option component Spatial (Kothuriat al. 2007). One of them is SDO_NN (Near Neighbour) that enables to specify the nearest geometry (so called neighbour), in our case the nearest vertex or rather edge of non-oriented graph. If we have GPS information about actual position of a rolling stock, it is possible to use this operator for specifying the nearest vertex/edge and subsequently for example to visualize the location on a map composed by means of MapViewer technology. The figure 4 shows the overall concept of the data and visualization layers of the railway network model.

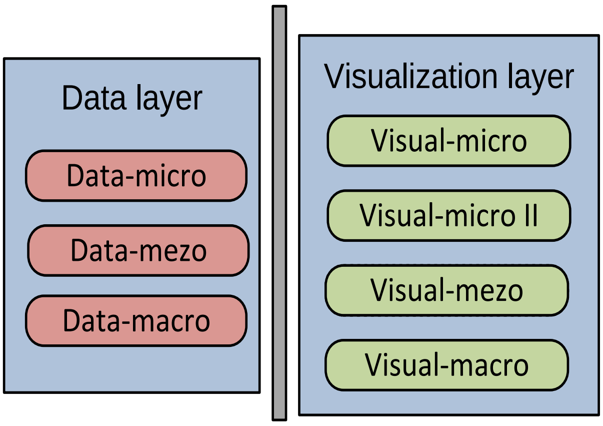

Figure 4: Overall concept of the railway infrastructure network model

5. SIMULATION OF TRAFIC

The selected train vehicles are equipped with communication terminals, which broadcast data including current GPS coordinates of the rolling stock. When the vehicle is in motion, this communication terminal sends information about its position every 30 seconds. Table 1 contains the data obtained from the communication terminals.

Designed simulation model contains the core of discrete simulation utilizing standard calendar of process messages, which were, during the simulation, executed based on their time stamp. This model was consequently implemented into the software demonstrator InfraRail that is intended for the additional support of dispatching control.

Simulation of traffic of rolling stock can be divided into two parts. Simulation based on:

- Real historical data (emulation of operation),
- Generated data.

The main idea of the localization of rolling stock position is based on the finding the nearest vertex in the data micro layer of model from the current GPS position of moving train. This algorithm include the base tests of GPS validity and is used Oracle SDO_NN operator for the searching the **N**earest **N**eighbour respectively the vertex.

6. SQL OPTIMIZATION QUERIES

If we use the database and SQL language there should be focused on the optimization. In this case we use SDO_NN operator so that we focus on optimization of SQL queries which are called directly from JAVA application. We can use the following:

- Statement Object,
- Prepared Statement Object,
- Calling functions that are stored in the database.

The Statement Object is base of universal query but its main disadvantage is periodical assembling of query. Use of Prepared Statement Object and PL/SQL function brings the use of so called bind variables. In this case the database gets still the same SQL statement (with still same hashcode) but with different values. This means that in the database machine is used always the

Table 1: Recorded time-stamped train movements data

Train number	Latitude	Longitude	Speed	Azimuth	Train vehicle identifier	Time
48701	50.02274	15.33554	78	57	91547123022	14.02.11 04:34:25
48701	50.02654	15.34246	80	43	91547123022	14.02.11 04:34:55
48701	50.03077	15.34873	68	43	91547123022	14.02.11 04:35:25
48701	50.03495	15.35505	61	45	91547123022	14.02.11 04:35:55

same execution plan.

6.1. Design of the optimization using the dynamic views

Despite the advanced optimization techniques of database ORACLE, the question arises as to whether is necessary to perform queries to the whole bases of vertices respectively edges. Whereas we are inquiring on the position of rolling stock (RS) within railway network which from the logical view, are not able to change the position by more than tens of meters, then we can reduce the base of vertices/edges by using the dynamic view. The main ideas are based on the following assumptions:

- For each new RS is created an initial dynamic view,
- Queries are periodically performed into the current view,
- If the RS is not already in the current view then the dynamic view is recalculated regarding to the azimuth of a moving train.

This means SQL query is performed into the relevantly reduced base of vertices/edges. This leads to optimization and the overall time saving of query.

7. TESTING

Testing of proposed optimization of selected search operations should show whether the using of dynamic view are correct. For the purpose of testing there has always been generated thousand points with GPS coordinates in the selected area, wherein the base of data consists of about one hundred thousand entries. For the time measuring of each query there were used information from database system tables ORACLE *v$sql* and values from column *ELAPSED_TIME*. For secondary comparison there was used time measured directly in application that reflect the time overhead of communication between application and database machine. For each test the time was measured as following:

- Average time,
- Max / min time,
- Median / modus / deviation,
- Total time in database / application.

7.1. Technique of queries

Firstly, there were tested all query techniques without dynamic views, that means:

- Statement Object,
- Prepared Statement Object,
- Calling functions that are stored in the database.

As it was expected, the Statement Object showed approximately 3.5x worse time of processing then Prepared Statement Object and query in the function. The results are shown in table 2.

Table 2: The comparison of different techniques queries in second

Method	Statement	Prepared Statement	Oracle function
Average	0,1208972	0,0091850	0,0124396
Min	0,059907	0,0035530	0,0047680
Max	3,28629	1,640714	1,607601
Median	0,0994355	0,0066205	0,0097315
Modus	0,066694	0,0040350	0,010057
Diversion	0,1151091	0,0520075	0,0514502
Total time DB	120,897181	9,185022	12,439586
Total time App	195,717069	53,824611	60,93599

7.2. Optimization by the dynamic views

The second area of tests used the dynamic views for Prepared Statement Object and query in the function for different sizes variants of dynamic views. The tests were performed within the three different sized areas:

- 10 x 10 km,
- 40 x 20 km,
- 50 x 50 km.

The comparison of result times for different size views within Prepared Statement shows that using of views brings the better results than without them. The best result comes with using of dynamic views 40 x 20 km. There is the time saving approx. 0.003 second. The all results we can see in the table 3.

Table 3: The results of times for Prepared Statement within different size of area in second

Size of view	10x10	40x20	50x50
Average	0,007647564	0,007537129	0,007978
Min	0,002743	0,002815	0,003319
Max	1,454968	1,157581	1,261327
Median	0,0049415	0,005347	0,006121
Modus	0,004397	0,003494	0,004287
Diversion	0,046213794	0,03691007	0,039899127
Total time DB	7,647564	7,537129	7,977629
Total time App	39,759844446	39,70151928	40,368371538

The next testing was used the function that is stored in the database. In this case was detected as the fastest query of dynamic views for the area 10 x 10 km. In comparison against PreparedStatement this way brings additional time savings but only on the database level. The total time in the application is on the other hand less favourable. The all results we can see in the table 4.

Table 4: The results of times for function within different size of area in second

Size of view	10x10	40x20	50x50
Average	0,007312738	0,007474627	0,007914767
Min	0,003082	0,003242	0,003108
Max	1,370451	1,354342	1,568563
Median	0,0052415	0,005647	0,005894
Modus	0,003895	0,005491	0,004772
Diversion	0,043302936	0,0427506	0,049460025
Total time DB	7,312738	7,474627	7,914767
Total time App	51,351476774	54,835111842	59,058458159

The next table (Table 5) only summarizes different approaches (Oracle function and Prepared Statement) for using the dynamic database view for the area sized 40 x 20 km.

Table 5: Times of two different approaches for view 40x20 km

Method	Oracle function 40x20	PreparedStatement 40x20
Average	0,007475	0,007537
Min	0,003242	0,002815
Max	1,354342	1,157581
Median	0,005647	0,005347
Modus	0,005491	0,003494
Diversion	0,0427506	0,036910
Total time DB	7,4746270	7,537129
Total time App	54,835112	39,70152

Although the total times in database are quite similar the different situation is in the total time in JAVA application. The ORACLE function against Prepared Statement has significantly worse time. The reason is that the ORACLE function requires additional time to the calling and takeover result.

7.3. Result of tests

The above mentioned tests and their results proved that the set preconditions of time saving with using of dynamic views were correct. On the figure 5 is shown graph which demonstrated gradual acceleration of SQL queries with utilization of the dynamic views. Also it is shown that the using of Statement queries significantly lags behind the queries with optimization.

The average times for different approaches is shown in Figure 5.

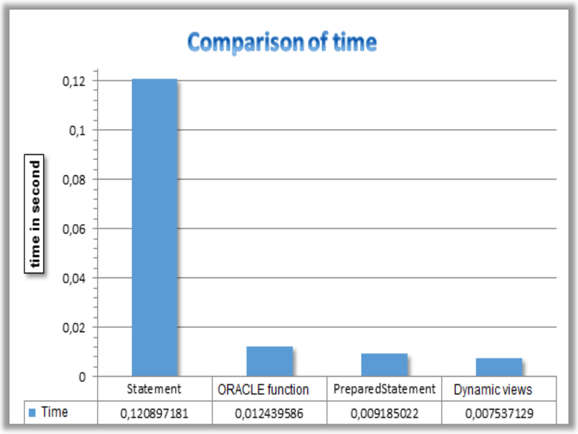

Figure 5: All tested approaches

The optimization of SQL queries based on dynamic views was subsequently integrated into a demonstration application InfraRAIL for localization of rolling stock. The running application using dynamic views within a railway network model is illustrated in Figure 6.

Within the final evaluation of results was found that for hundreds active trains on the railway network when the each train sends information about position each ten seconds is total time savings up to a few seconds.

CONCLUSION

This paper is focused on localization of rolling stock on the railway network and on the optimization of database queries by using dynamic views. In the introductory part, the paper shortly describes the existing systems in the localization area. This entire problem is divided into three parts. Further is described three-layer of railway network model, which is built as undirected graph directly in the ORACLE Spatial Network Data Model technology.

The second part is devoted to describing of different approach of optimization of database queries. Further is there described the design and implementation of the optimization database queries using a dynamic database views. In addition, there are shown the results of various tests which for hundreds active trains on the railway network proves that the using the optimization database queries (by the dynamic views) brings within a running application an additional timesavings.

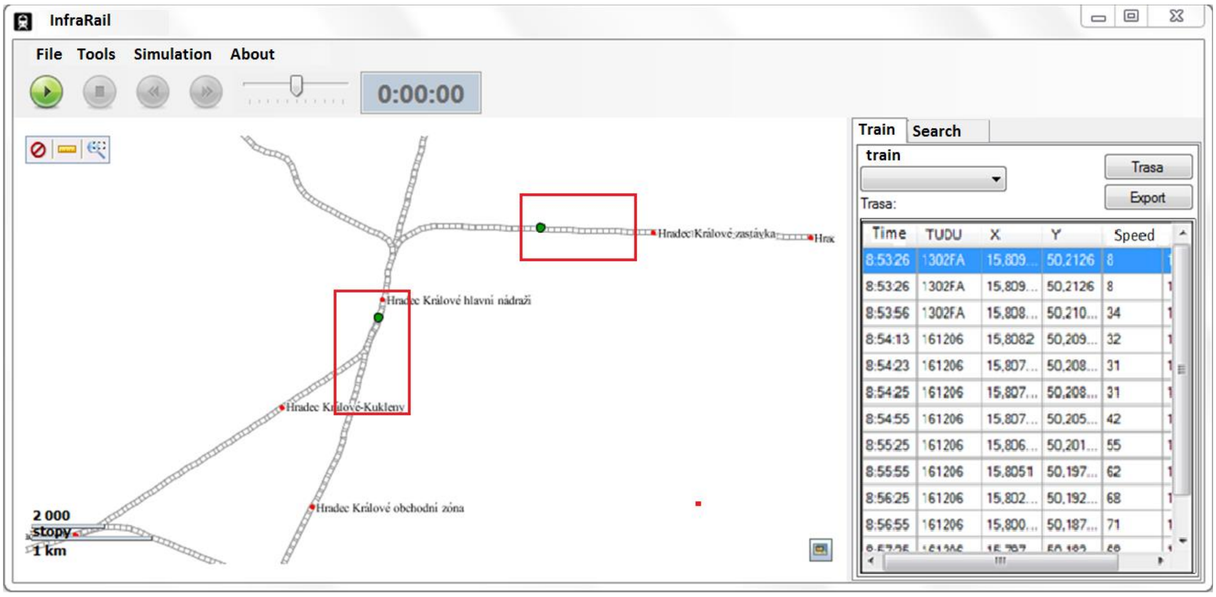

Figure 6: Running application

ACKNOWLEDGMENTS

This work has been supported by the project "SGSFEI_2015008 Models of infrastructure and operation of land transport systems" (financed by the University of Pardubice).

REFERENCES

Becker, U. and J. Poliak. DemoOrt repositions trains with satellite. In: EURAILmag Business & Technology. 18. BLUE LINE & Bro, France, 2008,s. 216-219.

Chudaček, V. and L. Lochman. Vlakový zabezpečovací systém ERTMS/ETCS. In: Vědeckotechnicky sborník ČD, č. 5/1998

Dorazil, P. Základní vlastnosti kolejových obvodů bez izolovaných styků. Pardubice, 2008. Bachelor thesis. University of Pardubice. Supervisor: Milan Kunhart.

Fikejz, J. and A. Kavička. Modelling and simulation of train positioning within the railway network. In: KLUMPP, Matthias. ESM'2012. The European simulation and modelling conference. Ostende: EUROSIS - ETI, 2012, s. 366 -376. ISBN 978-9077381-73-1.

Fikejz, J. and A. Kavička. Utilisation of computer simulation for testing additional support for dispatching rail traffic. In: European Simulation and Modelling Conference, 2011. Ostende: EUROSIS - ETI, 2011. p. 225-231. ISBN 978-90-77381-66-3.

Fikejz, J. and E. Řezanina, Utilization of computer simulation for detection non-standard situations within the new data layer of railway network model. In: The 26th European Modeling & Simulation Symposium. Bordeaux, 2014 s. 371-377, ISBN 978-88-97999-32-4

Ghazel, M. Formalizing a subset of ERTMS/ETCS specifications for verification purposes. In:Transportation Research Part C: Emerging Technologies. Elsevier Limited, 2014, pp. 60-75 ISSN: 0968-090X

Kothuri, R. et al. Pro Oracle Spatial for Oracle database 11g. New York, NY: Distributed to the book trade worldwideby Springer-Verlag New York, c2007, xxxiv, 787 p. ISBN 15-905-9899-7.

Lieskovský, A. and I. Myslivec. ETCS a AVV - poprvé společně. In: EuroŽel, Žilina, 2010

Lieskovský, A. Automatické vedení vlaků Českých drah. In: Automatizace. Praha: Automatizace, 2004, roč. 10. ISSN 0005-125x.

Murray, Ch, et al. ORACLE® Fusion Middleware : User's Guide for ORACLE MapViewer 11g Release 1 (11.1.1) [online]. 2010 [cit. 2012-07-09]. Aviable from : http://docs.ORACLE.com/cd/E14571_01/web.1111/e101 45.pdf

Murray, Ch, et al. *ORACLE® Fusion Middleware : User's Guide for ORACLE MapViewer 11g Release 1 (11.1.1)* [online]. 2010 [cit. 2012-07-09].

O'connor, M. L. Carrier-phase differential gps for automatic control of land vehicles, In: Dissertation Abstracts International, Volume: 59-06, Section: B, page: 2876.; 158 p. 1997, Stanford University, ISBN: 9780591909272

Senesi, F. Satellite application for train control systems, In: The Test Site in Sardinia, Journal of Rail Transport Planning and Managemt. Elsevier BV, 2012, s. 73-78, ISSN:2210-9706¨

Stadlmann, B. Automation of operational train control on regional branch lines by a basic train control. In: Proc. IEEE Intelligent Transportation Systems Conference, Toronto, Canada, September, 2006, s17–20.