TRANSPORTA UN SAKARU INSTITŪTS

NATAĻJA PETUHOVA

TEMPORĀLIE DATU MODEĻI DZELZCEĻA TRANSPORTA INFORMĀCIJAS SISTĒMĀS

PROMOCIJAS DARBS

izvirzīts inženierzinātņu doktora zinātnisko grāda iegūšanai

Inženierzinātņu nozares

apakšnozare "Transporta un loģistikas zinātne"

Zinātniskais vadītājs

Dr.habil.sc.ing., profesors

Jevgeņijs Kopitovs

RĪGA – 2010

UDK 004.6:656.2

PE 900

Transporta un sakaru institūts

Petuhova N.

PE 900 Temporālie datu modeļi dzelzceļa transporta informācijas sistēmās. Promocijas darbs. Rīga: Transporta un sakaru institūts, 2010. 164 lpp.

TRANSPORT AND TELECOMMUNICATION INSTITUTE

Natalia Petukhova

TEMPORAL DATA MODELS IN THE INFORMATION SYSTEMS ON THE RAILWAY TRANSPORT

DOCTORAL THESIS

to obtain the scientific degree Doctor of Science in Engineering

Scientific area “Transport”

Scientific research direction “Transport and Logistics”

Scientific supervisor

Dr.habil.sc.ing., professor

Eugene Kopytov

RIGA – 2010

ИНСТИТУТ ТРАНСПОРТА И СВЯЗИ

ПЕТУХОВА Наталья Юрьевна

ТЕМПОРАЛЬНЫЕ МОДЕЛИ ДАННЫХ В ИНФОРМАЦИОННЫХ СИСТЕМАХ НА ЖЕЛЕЗНОДОРОЖНОМ ТРАНСПОРТЕ

ДИССЕРТАЦИОННАЯ РАБОТА

на соискание учёной степени доктора инженерных наук

Научная область “Транспорт”

направление “Транспорт и логистика”

Научный руководитель

Dr.habil.sc.ing., профессор

Копытов Евгений Александрович

РИГА – 2010

ANOTACIJA

Nataļjas Petuhovas disertācija "Temporālie datu modeļi dzelzceļa transporta informācijas sistēmās". Darba zinātniskais vadītājs habilitētais inženierzinātņu doktors, profesors Jevgeņijs Kopitovs.

Šajā darbā ir sniegti temporālo datu izmantošanas dzelzceļa transporta informācijas sistēmās (IS) pētījumu rezultāti, kurus autors ir veicis laika posmā no 2001. līdz 2010. gadam. Kā pētījumu rezultātu pielietojuma objekts ir izvēlētas Va/s "Latvijas dzelzceļš" (LDz) izmantotās IS. Iegūtajiem rezultātiem ir universāls raksturs un tie var tikt izmantoti citos transporta uzņēmumos. Darbā ir izpētīta temporālo IS specifika, nosprausti un klasificēti galvenie IS uzdevumi, kas saistīti ar temporālo datu uzskaiti un apstrādi LDz; noteiktas problēmas, kas parādās šo sistēmu ekspluatācijā un to projektēšanā.

Tiek apskatītas temporālo objektu attēlošanas problēmas datu bāzē, kā arī to veseluma un savstarpējās mijiedarbības nodrošināšana. Tika izpētītas laika principu realizācijas īpatnības relāciju datu modeļa ietvaros un piedāvāts daudzversiju objektu modelis ar abstraktu identifikatoru. Kā arī tika izpētīta šī modeļa savietojamība ar eksistējošām IS. Darba gaitā tika ieviesta jauna temporāla forma Historical-Overlapped, kas ņem vērā temporālā objekta versiju maiņas īpašo režīmu, kas savukārt ir raksturīgs dažādiem dzelzceļa transporta procesiem. Tika izstrādāts temporālā objekta modelis, kam piemīt periodiskums un atkarība no kalendāra. Tika izpētīts temporālā objekta aktuālās versijas noteikšanas uzdevums. Šī uzdevuma atrisināšanai tika piedāvātas divas alternatīvas metodes: loģisko noteikumu metode un temporālo elementu metode. Tika izstrādāta metode piekļuves norobežošanai pie relāciju sistēmu datiem relācijas korteža līmenī, ar kuras palīdzību tiek risināta piekļuves vadības problēma pie dažāda laika periodu temporāliem datiem. Tiek piedāvāta datu transformācijas metode, ar kuru tiek atrisināta sarežģītu aprēķinu problēma veicot analītiskos pieprasījumus. Tika izpētīta temporālās komponentes loma datu bāzu noliktavu pārvaldības procesos.

Tika sniegti iegūto rezultātu pielietošanas piemēri praktisku uzdevumu risināšanā. Pētījumu rezultāti ir pielietoti Latvijas dzelzceļa informatīvajās sistēmās.

ABSTRACT

Natalia Petukhova's doctoral thesis "Temporal Data Models in the Information Systems on the Railway Transport". The scientific supervisor Dr.habil.sc.ing., professor Eugene Kopytov.

The work presents the results of the research dedicated to the problems of using the temporal data in the information systems on the railway transport. The investigation has been pursued during the period from 2001 to 2010. As an object of the research results application the complex of the information systems used in the State Joint Stock Company "Latvian Railway" has been chosen. At the same time the results have a universal character and can be applied to other transport companies. The specificity of the temporal information systems on the railway transport has been investigated in the work under consideration; the main tasks of the information systems, connected with the temporal data accounting and processing on the Latvian Railway, have been found out; the problems, originated in the development and implementation of these systems, have been determined.

The dissertation takes into consideration the problems of the temporal objects presentation in the databases, as well as the problems of provision of their integrity and the questions of interaction with them. The peculiarities of the temporal database principles implementation within the framework of the relation data model have been investigated; the temporal data model with the object abstract identifier has been offered. The opportunity of its compatibility with the existing information systems has been examined. The new temporal form Historical-Overlapped is introduced; this form takes into account the special mode of the temporal object versions switching, which is intrinsic for some processes on the railway transport. The investigation of the task of the temporal object active version determination has been done. The work presents two alternative methods of this task solution: the logical rules method and the temporal elements method. The method of the access controlling the relation system data on the level of the relation tuples has been worked out, and it tackles the issue of the access control to the different time periods temporal data. The method of the interval data transformation is presented. This method is to solve the problem of the complicated computation in the process of the analytical queries executing. The work analyzes the issues of taking into account the temporal constituent in the procedures of the data warehouses management.

There are some examples of the obtained results implementation in the practical tasks solution. The research results are applied in the information systems, operated on the Latvian Railway.

АНОТАЦИЯ

Диссертационная работа Натальи Юрьевны Петуховой “Темпоральные модели данных в информационных системах на железнодорожном транспорте”. Научный руководитель хабилитированный доктор инженерных наук, профессор Евгений Александрович Копытов.

В настоящей работе представлены результаты исследований по проблемам использования темпоральных данных в информационных системах (ИС) на железнодорожном транспорте, проведённые автором в период с 2001 по 2010 гг. В качестве объекта приложения результатов исследований выбран комплекс информационных систем, используемых в ГАО “Latvijas Dzelzceļš” (ЛЖД). Вместе с тем, полученные результаты носят универсальный характер и могут быть применены другими транспортными предприятиями. В работе исследована специфика темпоральных ИС на железнодорожном транспорте; выявлены и классифицированы основные задачи ИС, связанные с учётом и обработкой темпоральных данных на ЛЖД; определены проблемы, возникающие в процессе проектирования и эксплуатации этих систем.

Рассмотрены вопросы представления темпоральных объектов в базах данных, обеспечения их целостности и взаимодействия с ними. Исследованы особенности реализации временных принципов в рамках реляционной модели данных и предложена модель темпоральных данных с абстрактным идентификатором объекта. Исследована возможность ее совместимости с действующими ИС. Введена новая темпоральная форма Historical-Overlapped, которая учитывает особый режим смены версий темпорального объекта, свойственный некоторым процессам на железнодорожном транспорте. Разработана модель темпорального объекта, обладающего периодичностью и зависимостью от календаря. Исследована задача определения актуальной версии темпорального объекта. Для ее решения предложены два альтернативных метода: метод логических правил и метод темпоральных элементов. Разработан метод разграничения доступа к данным реляционных систем на уровне строк отношений, который решает проблему управления доступом к темпоральным данным различных временных периодов. Предложен метод трансформации интервальных данных, решающий проблему сложных вычислений при выполнении аналитических запросов. Исследованы вопросы учёта темпоральной составляющей в процессах управления хранилищами данных.

Приведены примеры применения полученных результатов при решении ряда практических задач. Результаты исследований применены в информационных системах, эксплуатируемых на Латвийской железной дороге.

Acknowledgements

The promotion work is written with the financial assistance of European Social Fund. Project Nr. 2009/0159/1DP/1.1.2.1.2/09/IPIA/VIAA/006 (The Support in Realisation of the Doctoral Programme “Telematics and Logistics” of the Transport and Telecommunication Institute).

CONTENTS

Type Codes, Terms and Abbreviations

ANSI – American National Standards Institute, www.ansi.org

AOID – Object Abstract Identifier

APFIS – Financial and Statistical Information Analytical System of the Latvian Railway Passengers' Transportation

CIA – Confidentiality, Integrity and Availability

DB – database

CIS – Commonwealth of Independent States

DWH – Data Warehouse

FK – Foreign Key

ETL – is a process in data warehousing that involves: Extracting data from outside sources, Transforming it to fit operational needs and Loading it into the end target

Express – the automatic system of booking seats and the railway passenger transportation management, operational in the area of the Baltic States and CIS (automated passenger traffic management system)

ER – Entity – Relationship

IS – Information System

ISO – International Organization for Standardization, www.iso.org

ID – Identifier

Latvian Railway – State Joint-Stock Company "Latvijas Dzelzceļš"

NF – Normal Form

OLAP – On-Line Analytical Processing

OLTP – On-Line Transaction Processing

RDB – Relation Database

RDBMS – Relation Databases Management System

RDS – Referenced Data System

DBMS – Database Management System

SAR – The reference interactive scheduling information system on the Latvian Railway

SQL – Structured Query Language

TDB – Temporal Database

TNF – Time Normal Form

TT – Transaction Time

VT – Valid Time

Introduction

Actuality of the problem. The majority of the processes on transport are connected with time. Timing is a fundamental for the tasks of planning and forecasting of the transport enterprise activity, as well as for practically all analytical and operative missions, demanding the usage of information systems data measured in different moments of time. Partly these data, called the *temporal data*, are active only on the definite time span; they are connected with the definite dates (time marks) or time slots. The accurate proper organization of temporal data in databases and the access facility to them to a large extent determine the functioning efficiency of the transport enterprise information systems.

Each transportation mode has some additional specific peculiarities, which should be taken into consideration in the process of generating of the information system using the temporal data. In particular, the railway transport is determined by inelastic schedule, which is attributable to many factors: transportation system physical limitation, the occurrence of many changes in schedule, the presence of numerous waypoints between the main points on the route, interconnection between the schedules of passenger trains and freight trains. Certainly, the design of the databases, capable of characterizing the railway transport system condition at any moment of history, is exclusively complicated and science-intensive task. As a result, despite the continuing development of the railway information systems complex, the systems users quite often undergo serious hardship in receiving the necessary information, characterizing the saved object condition in different moments of time. As a rule, the information systems databases hold only indicators' sampled values, and on frequent occasions only indicators' averaged on time basis values. In this respect, it is necessary to restore the data from different sources, to perform approximation and extrapolation; these activities bring much allowance, averages and discrepancy in the employed data set.

According to the author's view, the situation is conditioned by the fact, that the information systems continue using the technology of classical relation databases; and these databases are not directed on the storage of all changes in all databases objects. In other words, these technologies do not support the multi-version of these objects. The undertaken study of the operative and still developing information systems on the railway has shown that these systems use the temporal principles inefficiently. The following drawbacks of the information systems demonstrate this inefficiency:

- imperfection and low descriptive component of the temporal objects models;
- disk space wasteful usage;
- low efficiency of the interaction with the temporal objects;
- low databases expansibility;
- high complexity of the databases development;
- hardship with supporting the data integrity;
- difficulties with organizing the access to the information;
- problems with data providing security and privacy.

Under the condition of using the analytical information systems this list should be amplified by the following shortcomings:

- absence of the apparatus of high-performance temporal data processing, in consequence of which the enormous information potential, hidden in a bulk volume of the cumulative data, is used incompletely;

- imperfection of data preparing processes for analysis; the processes are not directed on the data temporal characteristics and there are hardships, connected with the data accessibility and integrity providing.

For many years the deterrent for the temporal databases technology development has been the immense requirements for the resources necessary for supporting multi-versions of the storage objects. But in view of the rampant development of the computing technologies this problem receded into the background. However, there are some other factors constraining the temporal databases usage in such big enterprises as the railway. Among them are the following: the absence of the unitary standard for the temporal data processing; the peculiarities of the temporal objects of the enterprise systems; the lack of necessary skills for the work with the temporal data, the majority of developers suffer of; step up requirements for the data integrity, accessibility and privacy.

The listed issues give evidence of the urgency of papers, directed on the efficiency enhancement of the temporal data usage in the railway transport information systems, which is a basis for the investigation under consideration.

Degree of the theme studies. Management of the information systems temporal data on the railway transport is a complex task, the investigation of which is carried out in the following directions:

1. Information systems development and functioning on the railway transport;
2. Temporal information systems and databases designing: standards, methods, models, approaches;
3. Temporal data operative and analytical processing;
4. Temporal data information security providing.

In the area of the *information systems designing and functioning* on the railway transport the works of Russian scientists E. Letsky, V. Pankratoff, V. Sharoff, Y. Mukhopad and some other researches can be mentioned. Their investigations cover the railway information systems issues in different countries, demonstrate the complexity and peculiarities of the railway functioning and its information systems development, and put forward some functioning models on the macro- and micro-levels [115, 116, 117]. The authors show the importance of timing factor accounting, but at the same time do not provide the peculiar solutions for the issues, having in view mostly the classical approaches.

Among the investigations implemented to the information systems on the Latvian Railway, the research work by V. Demidov [113] should be mentioned specially. The research is oriented on the increase of the efficiency of the decision making support system on the railways by the way of perfecting the structure and the functioning processes of the dataware system. The research suggests the models describing the topology of the transport network of the railway and the passengers' transportation processes. It also presents the developed models and methods of data representation and transformation in the information analysis systems under consideration, including the data virtual stochastic models. The work presents the method of the hierarchically connected views, the conceptual models of the railway transportation prognoses, etc. If to speak about investigating the problem of the temporal data employment, it is important to note the solution of the whole range of the issues connected with accumulating and saving the history of changing the objects states, as well as the tasks of providing the access facilities to them at any moment of time. The development of the fundamentals of the temporal databases designing and implementation of the temporal logics is also worth mentioning (V. Demidov and the author of the thesis under consideration cooperated in these problems solving [4, 7, 8]). At the same time a lot of temporal system problems occurring in the process of the information systems development and operating in transport system, V. Demidov did not solve or even did not consider. First of all, it is necessary to mention the events periodicity management and the special calendar employment, the issues of supporting the temporal data integrity and privacy.

The urgent problems and achievements in the temporal databases area are summarized in the cooperative work by S. Kuznetsoff and B. Kostenko [67]. There are also remarkable fundamental works, devoted to the *temporal technologies*, developed by C. J. Date, H. Darwen, N. A. Lorentzos, C. S. Jensen. Their cooperative investigation “Temporal Data and the Relational Model” [51] covers the problems of storage and interaction with the temporal data within the framework of the relation model. However, the authors mostly concentrate on the general problems of temporal databases generating and Tutorial D language usage, but do not approach the issues, connected with the periodicity consideration and special calendar consideration, which is exclusively important for the transportation information systems, primarily in the questions of traffic scheduling.

One of the most powerful investigations in the sphere of the temporal databases is the set of works by Christian S. Jensen written during the period from 1991 to 1999 in cooperation with the famous scientists from different countries. He worked together with J. Bair, R. Busatto, J. Clifford, C. E. Dyreson, F. Grandi. H.Gregersen, S. Hsu, T. Isakowitz, T. Y. Cliff Leung, L. Mark, J. Skyt, R. T. Snodgrass, M. D. Soo, A. Steiner, K. Torp. In 2000 this set of works was published in the form of thesis having the volume of more than 1300 pages [52]. The majority of the above mentioned scientists took part in a committee which developed a specification of the temporal data query language TSQL2 and SQL-3 standard. It is important to study TSQL2 and SQL-3 for developing the methods and models for temporal data operation.

The immense contribution in the temporal databases development of the international organization TimeCenter [66] is worth mentioning. This center deals with the support of temporal applications on the basis of both traditional and up-starting technologies of the database management systems (DBMS). The leading scientists in the above mentioned sphere work in this center. Among many outstanding papers P. Terenziani’s researches attract attention, as well as his colleges’ works [62, 63, 65, 64]. Their investigations give a special consideration to processing the temporal constraints for the events, reoccurring with the course of time (so called “periodicity”). They investigate the “symbolic” language of high level, used for presenting the periodicity, and user-oriented compared to the mathematical formulae language. The same work considers the temporal relation model, supporting the definite “symbolic” periodicity appointed by the user (for example, “the second Monday of every month”). Determining the temporal annex for the standard relation algebra operators, the additional new operators and functions are introduced. The temporal algebra, which is consistent supplement for classical (not temporal) one, is presented.

The implementation of *the operative and analytical temporal data processing* is found on the researches in the sphere of the business intelligence and decision-support systems, as well as the databases warehousing theory. Such scientists as B. Inmon, R. Kimball, N. Pendse, T. Pedersen made an immense contribution to the formation of the fundamentals of the multidimensional data modeling and the classical approaches to the tracing of any changes in dimensions.

The investigations belonging to the last-mentioned block are exactly connected with the *information security* problems. They cover the issues dedicated to the discretionary and mandatory security models. Several aspects of these models application in the databases systems are reflected in the works of the below mentioned authors: N. Jukic, S.V. Vrbsky, D.E. Denning, S. Jajodia, R. Sandhu, S. Osborn. The general concept of the information security is reflected in the standards “Trusted Computer System Evaluation Criteria” [102], “Information Technology

Security Evaluation Criteria" [103], "Common Criteria for Information Technology Security Evaluation" [104].

The level of the above mentioned works must be highly appreciated. However, it is necessary to remark, these papers do not provide the solution of many problems, emerging in the process of using the temporal data in the information systems on the railway. There are many researches devoted to the problems on the railway, and among them some can be mentioned specially. They are dedicated to the narrow issues of the optimization of the procedures connected with the temporal component, for example, the optimization task for the trains schedule algorithm development [124-133]. But there are no complex works directed to the organization task and the task of using the temporal data, taking into consideration the specificity of the transport processes. According to all said above, it is possible to claim that the issue, covered in the work under consideration, is formulated for solution for the first time.

The ***goal*** of the doctoral research is the enhancement of the development and functioning of the railway information systems by means of the temporal databases technologies application supported by the detailed elaboration of the problems of the temporal data models designing and employment, as well as the problems of the access to data methods.

In accordance with the set goal, the author of the thesis is to solve the following ***tasks***:

1. Investigation of the railway temporal information systems design specificity and revealing the problems requiring the solution.
2. The critical analysis of the investigations, concerning the temporal databases and the temporal information systems design field, choice and substantiation of existing approaches for the practical application in the railway information systems, and the research under consideration trends statement.
3. Temporal data models development and the methods of interaction with the temporal objects, taking into account the specificity of data processing tasks in the information systems on the railway transport.
4. Temporal data security methods development, capable of supporting the exclusive requirements the railway enterprises put forward for the data integrity, privacy and accessibility.
5. Development of the relation facility extension for manipulating the data implementing the temporal data transformation in the analytical systems.
6. Application of the investigation results for solving the practical problems in the information systems of Latvian Railway.

Methodology and methods of investigation. The dissertation research is based on the following concepts:

- results of the railway transport processes modeling;
- results of the scientific research projects, performed by the Transport and Telecommunication Institute and the Latvian Railway with the personal participation of the thesis author;
- author's experience, acquired in the process of participation in different projects, devoted to the information technologies in the field of databases and Web systems, as an expert, architect, and databases and application developer;
- materials of the scientific conferences, with the author's personal participation;
- technical documentation, scientific and technical literature and periodical publications on the topics covered by the concerned thesis.

In the process of thesis development the following scientific methods were used: the system analysis, Boolean algebra, relational algebra, predicate calculus and sets theory.

Scientific novelty of the work. The presented work is a thorough integrated study of the railway transport temporal information systems composition and functioning.

The author performs the following scientific investigation results:

1. Principles of the temporal databases design in the relation environment, customized for the railway transport information systems, are offered;
2. Unique brand new models and methods for the work with the temporal objects have been worked out:
   - temporal data model with the object abstract identifier;
   - temporal relation model with the overlapping lifespans Historical-Overlapped;
   - model of the temporal object, involving periodicity and a special calendar;
   - method of providing the user's temporal environment;
   - methods of the temporal object active version determination: the logical rules method and the temporal elements method;
   - interval data transformation method;
   - method of the access controlling the temporal data on the relation tuples level.

Work practical value and its implementation. The work *practical value* lies in the designing and employing the temporal technologies on the railway transport.

The presented results allow:

- to equip the databases of the existing railway transport information systems with the temporal features and provide the temporal upward compatibility, excluding the necessity to upgrade the existing application software;
- to improve the temporal data representation in the databases and the methods of interaction with them.
- to increase the temporal data integrity, accessibility and privacy;
- to sophisticate the processes of the temporal data transformation and analysis in the warehouses of the railway data.

The following activities were done on the basis of the obtained results and with the author's personal participation (see Appendix 2):

- interactive information referred system of the passenger trains schedule on the Latvian Railway has been designed;
- central system of the railway classifiers and codifiers has been improved; this system is employed in various information services on the Latvian Railway;
- variety of information systems of the Latvian Railway has been upgraded; this variety includes the Financial and statistical information analytical system of the passenger transportation, the Registration of the hardware and software system, etc.

Approbation of the research. The scientific results, acquired in the process of investigation, were reported on the 16 scientific and research and practice conferences in Belgium, Israel, Latvia, Lithuania, and Russia. The list includes the following items:

1. The VI International Conference "**TransBaltica 2001**", Riga, Latvia, June 7-8, 2001;

2. The Research and Practice and Educational and Methodological Conference "**Science and Technology is the Step into the Future**", Transport and Telecommunication Institute, Riga, Latvia, May 2-3, 2002;
3. VII International Conference "**TransBaltica 2002**". Riga, Latvia, June 12-14, 2002;
4. 43rd International Scientific Conference of Riga Technical University. Riga, Latvia, October 10-14, 2002;
5. The International Conference "Reliability and Statistics in Transportation and Communication" **RelStat'02**. Riga, Latvia, October 17-18, 2002;
6. The International Scientific and Technical Conference "Civil Aviation at the Present Stage of Science, Facilities and Society Development", **MSTUCA**, Moscow, April 17-18, 2003;
7. The International Conference "Modelling and Simulation of Business Systems" **MOSIBUS 2003**, Vilnius, Lithuania, May 13-14, 2003;
8. The VI International Conference "Computing Anticipatory Systems" **CASYS '03**, Liege, Belgium, August 11-16, 2003;
9. The International Conference "Reliability and Statistics in Transportation and Communication" **RelStat'03**, Riga, Latvia, October 16-17, 2003;
10. The VI International Baltic Conference "Databases and Information Systems" **DB&IS 2004**, Riga, Latvia, June 6-9, 2004;
11. The International Conference "Reliability and Statistics in Transportation and Communication" **RealStat'05**, Riga, Latvia, October 13-14, 2005;
12. The Research and Practice and Educational and Methodological Conference "**Science and Technology is the Step into the Future**", Transport and Telecommunication Institute, Riga, Latvia, December 15-16, 2006;
13. The International Conference "Reliability and Statistics in Transportation and Communication" **RealStat'07**, Riga, Latvia, October 24-27, 2007;
14. The International Conference "Modelling of Business, Industrial and Transport Systems" **MBITS'08**, Riga, Latvia, May7-10, 2008;
15. The International Symposium on Stochastic Models in Reliability Engineering, Life Science and Operations Management, **SMRLO'10,** February 8-11, 2010, Beer-Sheva, Israel.
16. The 9th Joint Conference on Knowledge-Based Software Engineering, **JCKBSE'10**, August 25-27, 2010, Kaunas, Lithuania.

The author's works in the field of the transport information systems enhancement were commended by:

- The award of the "Social and Economic Systems, Model, Simulation, Agents, Decision Support" symposium for the best scientific article "*Principles of Creating Data Warehouses in Decision Support Systems of Railway Transport*". The sixth international conference "Computing Anticipatory Systems" **CASYS '03**. August 11-16th, 2003, Liege, Belgium (see Appendix 3).
- The prize after Karlis Irbitis (doctoral scholarship) from Latvian Academy of Science, "Latvijas Gaisa Satiksme" corporation and Latvian educational fund "Education, science and culture" for the work "*Transport Information Systems Security Enhancement*", 2005 (see Appendix 4).

Publications. 27 papers have been published basing on the results of the researches [1-27]. The sources include 17 scientific papers and 10 brief outlines of the scientific reports. They consider the problems of the information systems developing on the transport enterprise and the issues connected with the temporal databases design and implementation. The special attention is paid to the methods of huge data amount processing, such as the problems of the temporal data integrity, accessibility and confidentiality providing.

Moreover with the author's direct participation there were developed two new concepts of the development of scheduling information system on the Latvian Railway [28, 32] and two technical design specifications for the systems: "Electronic Booking in the International Trains" [29] and the "Registration of the Hardware and Software System" [31], as well as the technology of the system "Electronic Booking in the International Trains" [30].

Structure of the thesis. The paper consists of Introduction, 4 chapters, Conclusions and 4 appendices. It includes 80 figures and 8 tables, constituting 164 pages in total. The references list comprises 134 sources.

Introduction considers the topic urgency of the dissertation, formulates the goal and the tasks of the research. The novelty and the practical value of the obtained results are shown in combination with the brief executive summary of the work.

Chapter 1 describes the information systems peculiarities and specificity on the railway transport, demonstrates the time and the temporal data role in these systems. The temporal data evolution and their usage in the information systems are investigated. The temporal data employment problems are analyzed. The information systems tasks, connected with the temporal data registration and processing are revealed and classified. The investigation task specification is performed.

Chapter 2 considers the basic terms used in the temporal database theory. The critical review of the software solutions implementing the temporal technologies is carried out. The choice of the platform for the temporal information systems development is substantiated. The chapter reviews the temporal objects presentation in the databases, as well as the problems of providing their data integrity and interaction with them. It also determines the models and methods, intended for the employment in the transport temporal information systems; the classification of these methods and models is adduced. The problems of the selected models and methods implementation are formulated, and the detailed elaboration of the research tasks is introduced.

Chapter 3 considers the fundamentals of the temporal databases development in the relation environment using the proposed data model with the abstract identifier and the temporal logics. A special attention is dedicated to the introduction of the temporal objects, furnished by the periodicity and correlated with a special calendar. The new temporal form – Historical-Overlapped – is defined; this form considers the peculiar temporal object version change mode. The issues of interaction with the temporal data are solved. The method of providing the user's temporal environment is offered. The temporal object active version determination problem is solved. The method of the interval data transformation for analytical queries is suggested. The issues of the temporal data security providing are solved.

Chapter 4 gives the examples of the practical employment of the research results in the Latvian Railway information systems. The chapter presents the application scheme of the generated methods and models in the Interactive Local Train Traffic Timetable on the Latvian Railway. The processes of the data warehouses management are investigated. The special attention is paid to the temporal data transformation procedures. The practical employment of the generated methods in the Financial and Statistical Information Analytical System of the Passenger Transportation is considered.

The *Conclusion* contains the main research results.

Chapter 1. ANALYSIS OF THE PROBLEM OF EMPLOYING THE TEMPORAL DATA IN THE TRANSPORT INFORMATION SYSTEMS

1.1. "Temporal Data" Concept Definition

A lot of basic data measured in time terms is employed for solving such issues as registration, planning, searching for the managerial decisions and forecast in the railways information systems. Data of this type are called temporal. Some scientific publications operate with other terms, for example, time data, historic data, chronological data, time-referenced data, time-varying data, time-dependent data. Unfortunately it emerged that the fundamental researches in the area of the temporal databases [49, 51, 52] do not give any definition to this phenomenon. For the moment there is no unified worked-out conventional standard definition of the temporal data concept. Beneath there are some definitions of the well-known ones:

- "The temporal data in a wide general sense are the arbitrary data which are connected with the definite specified dates or the time periods obviously or in implicit way" [67];
- "Temporal data - Data that explicitly refer to time" [118];
- "The term temporal data means the concept where data or say an object is defined to have some time related information associated with it, e.g. a 'built-in' timestamp" [60].

In some scientific areas, for example, Statistics and Data Intellectual Analysis (DataMining), the temporal data are defined in a more exact way, but the definition envelopes only the narrow spectrum and range of the temporal data, which fall under the investigation of this scientific area:

- "Conventionally, the temporal data is classified to either categorical event streams or numerical time series …" [101].

All the above presented definitions are correct, consistent and non-contradictory, nevertheless, no one does allow to characterize unambiguously the whole spectrum and range of the temporal data. The absence of the particular definition required the author to formulate it with specified elaboration and clarification. In the process of definition generation the author took into account the special peculiarities of the railway transport. The temporal data are assumed to be *the data, based on the object version nature, accumulated historical data and time series data.* Accordingly, the temporal data aggregate the data from three different groups. These groups are examined below in details:

1. *The data, based on the object version nature* (version-based temporal data). In the process of the railway transport operation many objects participating in the railway transport life activity and functioning arrangement are subjected to the particular changes: change in the trains traffic routes, currency exchange rates, railway fares and rates for other railway services, rates for power resources, parceling of the railway network into the lines and sectors, and so on. The previous states of these objects (versions) are important for various calculations and for analysis. To manipulate the versions, active in different moments of time, every state of the object is connected with the peculiar time range – the time of its activity. The described objects states (versions) can refer not only to the past or to the present moments of time, but to the

future moments as well, for example, the planned changes of tariffs, schedule, assigned for the next season, and so on.

2. *The accumulated historical data* (historical data) – are the various operations, documents and other homogeneous data, which come to the system persistently and continually and accumulate within this system (receipts on tickets sale, invoices, consignations, freight operations, etc.). These data characterize the functioning of the principal railway processes in time. Every instance of these data is connected inseparably with the specified date or with a particular moment of time. Precisely this time factor becomes the principal dimension in the analytical calculations.
3. *The time-series temporal data* (time-series based temporal data, streaming data) are practically the data, assembled in the process of the observation the object, under the condition that these observations were done consequently in time or with the particular time intervals, or uninterruptedly and continuously (for example, the results of the carriages movement speed measurements, the coordinates of their position in time). The observations of this type can be disposed in the chronological order.

There are also the *composite* types of the temporal data: so, the accumulated data on the sales of the passenger ticket comprise not only the facts of the tickets sales, but also the facts of their further cancellation, consequently, there is the fact of multi-version presence.

The temporal data, referring to the different groups, are closely connected in actual practice and co-exist together. The full-fledged analysis of the accumulated historical data is often impossible without the navigation in the time dimension of the accompanying them the multi-version data from the first group. The following situation can exemplify the above said: the retrospective analysis of the losses resulted from the currency corridor employment, and this analysis is based on the foundation of the accumulated documentation of the freight transportation process (invoices and consignation notes) and the currency exchange rates [14], and also the analysis of the passengers transportation, founded on the basis of accumulated data on the tickets sales and the tariff multi-version data and referenced data [1, 2, 4].

The problem of the temporal data efficient employment is not solved fully. The part of the data is scratched, or lost, in the course of time some contradictions and inconsistency, discrepancy and data conflicts are discovered. As a result, a great deal of the helpful information potential is not employed.

As it may be inferred from the classification, presented above, every type of the temporal data in the information systems in the railway transport is characterized by the specific peculiar tasks and, consequently, has the peculiar problems. The various types of the temporal data in transport and the problems of their efficient employment will be considered in further details in point 2.4.

1.2. Description of the Application Area of Research

As the object of application of the research under consideration the Latvian Railway, as the typical representative of the railway industry, was chosen. The Latvian Railway takes the 30th position in the world according to the railway network coverage density indicator among the countries with the area of more than twenty thousand square kilometers (see appendix 1).

From the point of view of the information technologies penetration, the railway schedule is supposed to be a very “complicated” object. In the first instance, the multi-variance and diversity of the transport system components are worth mentioning. These components are included in this large-scale complex and have the different levels of the information demands.

They are the thousands of the railways, covering almost all the country area, they are the hundreds of stations, equipped with the electricity supplying facilities, with the automatic systems, with the tele-mechanical and connection systems, with the freight handling equipment, with the container terminals and so on; they are the thousands of the passenger carriages and the freight carriages, they are the hundreds of the locomotive engines, they are the many-thousand-personnel, operating the principal and auxiliary procedures on the displacement of the movable equipment of the railway system and displacement of freight, they are the permanent structures and facilities, they are handling of the documents and moving them, they are the management system based on the facilities of the data transmission and processing [117]. We shall append the services, which are non-standard, non-typical for the industry, to the above mentioned items. These services can be delivered due to the well-developed infrastructure of the Latvian Railway. For instance, provision of the electronic connection comprises the data transmission, the international lines lease, the local lines lease, the phonic telephony, the equipment placement.

Thus, taking into consideration everything said above, the author educes the following features of the complexity of functioning systems and the systems under development:

1. The great number of processes and functions, data components and the tight interconnection between them;
2. The impossibility of representing the system as the combination of the independent sub-systems;
3. The factor of the ready-to-use analogues absence, restricting the possibility of usage of the standard design concepts;
4. The necessity of integration of the existing and newly-developing supplements;
5. The factor of functioning in the heterogeneous environment and on the different hardware architecture;
6. The insulation of the different groups of the developers by the level of qualification and the traditions of the information facilities employment.

The strong interrelation and interdependence of the transport components (the railways, the stations, and the railway junctions) require the corresponding organization of the information systems. Thereby, the train, which was generated on one station, falls under processing on the other stations, the carriages are loaded and unloaded on different stations, and so on. It is the reason why the efficient operation of the systems with the interacting components requires the centralized management. Moreover, there are such management issues, the solution of which is possible only on the centralized basis.

One more problem of the information interaction is worth mentioning: the different parts of the railway complex belong to the different owners. Besides the State Joint Venture “the Latvian Railway”, the market of the freight transportation is nowadays presented by two private joint ventures: “The Baltic Express” and “The Baltic Transit Service”. They transport 10% of the general freight volume. This indicator is one of the highest indicators in Europe. The local passenger transportations are done by the joint venture “The Passenger Trains” and “Gulbenes-Alucksnes Banitis” ltd. [34]. Concerning the railways equipment and track facilities, it can be mentioned that the State Joint Venture “the Latvian Railway” is not the only owner as well.

Despite the fact of having the different owners, the railway is to work as a tightly cooperating complex oriented on the life support of the whole system. It is indisputable the railway is a very significant component in the vital activity in the life of the Latvian State, and it is connected with the following very important factors: the economic factor, the financial factor, the geopolitical factor and the social factor. The graphs in Fig. 1.1 and Fig. 1.2 allow everybody to understand completely the role of the railways transport in Latvian Economy.

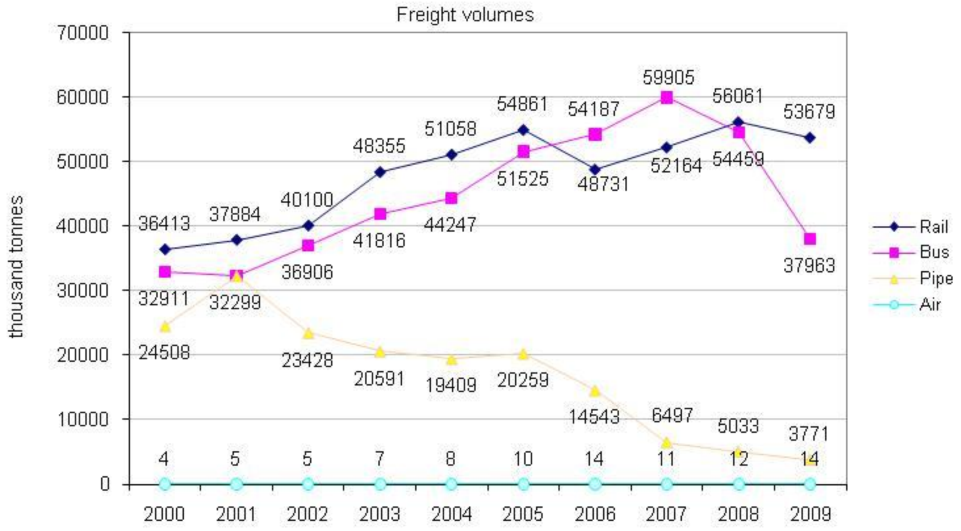


Fig. 1.1. The volume of the freight transportation divided by the principal transport means in Latvia

Fig. 1.2. The volume of the passengers transportation divided by the principal transport means in Latvia

The railway freight transportations compose about the half of the total amount of all land transportation (land transit), and the passenger transportations compose about 10% of the land transportation volume (in the suburbs this characteristics increases up to 30%). The transit transportations compose 85% of all railway transportations; mainly these transit transportations go from Russia and Belorussia to the sea ports of Latvia (Eastern-Western transit corridor). The inland domestic traffic composes about 5% of the total volume. This factor can be explained by the comparatively small distances of such transportations [34].

Turning to the peculiarities of the railway transport information systems support, the author notes that the significant part of the tasks of the railway transport management requires the solution in the real time mode, in other words, in the same rate as the external procedures passing – the trains movements, the procedures with the carriages, the procedures with the customers, and so on. This requirement results in the necessity of estimation of the temporal characteristics of the information procedures in the process of development, and taking them into consideration in the process of choosing the information processing technology, the architecture of the information systems.

The time factor takes the enormous part in the process of providing the life activity support for the railway transport. The railway is a compound complex system managing the variety of the resources changing dynamically; the majority of the railway business-processes are subordinated to the time factor and connected with the historical data accumulation, the elapsed time periods analysis and the future planning. The real time data in the numerous information systems of the transport processes life activity support become obsolete and out-of-date very quickly and they are exchanged with the new ones, but their initial measures are necessary for the analytical systems. It is important to mention that the full-scale comprehensive analysis of the railway transport system functioning is impossible without both the assessment of the current and previous enterprise activity and this activity forecasting. This factor is the object of the research under consideration.

1.3. Evolution of the Temporal Data in the Latvian Railway Information Systems

The Latvian Railway is one of the leading enterprises in the terms of the computerization not only in the transport industry but also among other Latvian enterprises of this scale. The development of the computer equipment base of the Latvian Railway has become especially intensive in three recent decades, since the middle of 80s of the previous century (Fig. 1.3). During the relatively small time period of the computer technologies employment – about 50 years – the Latvian Railway employed several generations of the information systems, which diverge considerably from each other not only with the technical and programming equipment, but with the implemented technologies as well. The high requirements for the information systems safety and efficiency have stimulated the introduction of the new methods of saving and processing the information, used on the purpose of the industry management. Every new generation is extended with the possibilities of the data employment, including the temporal data.

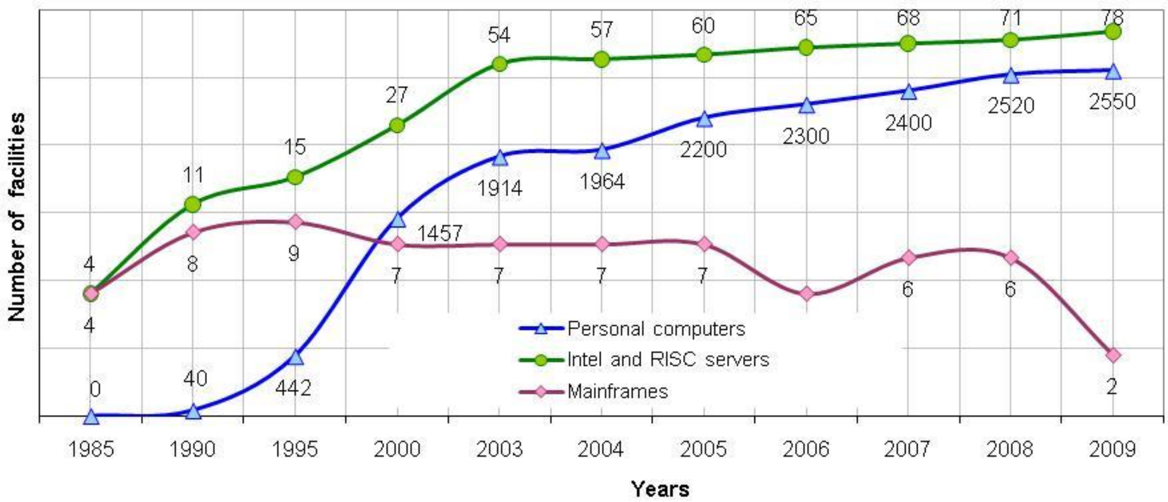


Fig. 1.3. Development of the computing equipment base in the Latvian Railway

1. Manual and mechanical information processing. Starting with the beginning of the previous century, with the beginning of the railway commercial operation and up to the 60s there is only manual and then electro-mechanical approach towards the data processing. The principal facility of the data collecting and processing were the electro-mechanical accounting machines. The data preparation process was enormously labour-consuming.

2. The local automation of the certain tasks of the railway activity. This stage is characterized with the information technologies formation: computer equipment occurrence, separating the information systems and the automation facilities in the independent field of activity. The main problem of this stage is the time lag between the advanced level of the hardware development and the software.

3. The freight and the passenger transportations management centralization. It is characterized by the occurrence of the information systems of mass-service, centralized in their nature but geographically spread (from the Baltic to the Russian Far East). The principal issues of this stage are connected with the solution of the tasks of the efficiency and safety increase. The absence of any standardization in the field of the data saving and processing should be mentioned.

4. De-centralized data processing on the basis of the personal computers. The centralized data processing is still takes place on the basis of the big data-processing machines, as it was before, but it becomes more and more important to solve the local tasks and to process the local databases on the user's working place. Standardization is employed in the sphere of data saving and data access, the new facilities of the software designing are used (the Latvian Railway gains the advantage of Clipper everywhere). These processes promoted the increase in the velocity of the software designing without prejudice to the quality and functionality of the product. Designing the local supplements on the personal computers basis allowed not only prompt solution of the problems of automation of the various functions of the railway process and their functionality extension, but also it arouse numerous issues connected with the process of the data de-centralization and the big number of the geographically separated client stations. Among them there are following: the permanently changing software versions and the time sensitive data reference on the clients' machines are kept in correspondence; the efficiency of the users' hardware and data capacity of the network are continually increased, etc.

5. Introduction of the corporate databases with the employment of the relation databases management system. This stage is characterized by the solution of the problems connected with the data de-centralization, transit to the relation databases management system IBM DB2 and adopting the new disk resources promoting the new possibilities for the big data volumes processing. The client-server architecture of the databases management system and its relation model allow the data saving and processing on the conceptually new, revolutionary, more efficient level. After switching of the range of supplements to the centralized database, the analysts and the superiors obtain the possibility to receive the integrated operation reports more often, whereas the issue of the protracted data collecting and integrating from different working places is eliminated.

6. Introduction of the Internet Technologies and the Data Warehouses Technologies. This stage is described by increasing the relevance of the systems, oriented on the analytical processing of the data and the support of the decision making; it is also described by expansion of the range of the information systems and the central database users due to the open access to the Internet. The computation resources virtualization and the virtualization of the various services of the transport information systems (the cloud computing) take place.

The results of the investigation of the temporal data evolution in the Latvian Railway information systems of the above-listed generations are presented in Table 1.

Table 1. The temporal data employment evolution in the Latvian Railway information systems

Stages of the IS development	**Hardware and software**[1]	**Temporal data and their usage**
1. Manual and mechanical information processing (till 1962)	*The electro-mechanical accounting machines*	*Temporal data* Annual reference books (tariff tables etc.), periodical statistic reports on the hardcopies. *Employment of the temporal data* In the processes of calculation the corresponding reference books are used. In view of the complexity of the urgent issue and distribution of the reference books, they fall under correction minimally, and at the same time the amount of the reference data employed in the calculations is minimal (especially compared to the contemporary methods of calculation).
2. The local automation of the certain tasks of the railway activity (1960 -1980)	*Universal Digital Computers* Peripheral input/output devices: *alphanumeric printers*, *teletypes*, *multiplexers etc*. The external memory: *punched card media*, *punched tapes*, *magnetic tapes* and *disks*	*Temporal data* Multi-version data are practically not used. The statistic information only is saved with the time notes. The historical accumulated data are present in the computer memory only at the stage of data processing. *Employment of the temporal data* Data are processed by small portions. After the processing procedure the operational data are archived on the magnetic tape, and the statistic results are kept in hardcopies and in electronic form. The analysis and the comparison of the characteristics for the different reporting timeframes as a rule take place on the basis of the hardcopy of these statistical data. In case of necessity of the repeated information processing the data are downloaded from the tapes and disks again and again. This process is connected with the solution of the problems of the partial information loss, as far as the tapes are not safe facilities of information saving at the moment. The access to them is connected with the physical movement of the disks containing data. This process requires the human interference, because the entire historical information is stored separately from the data processing machine: on the tapes or on the disks in specialized cabinets.
3. The freight and the passengers transportations management centralization (1985-1990)	*The operative system of the freight transportation management APOVS* *The passengers transportations system Express-2* *Network X.25*	*Temporal data* The occurrence of the temporal component in the operational information systems data. In their essence the multi-version objects (freights, carriages, and so on) in information systems do not possess the evident time dimension. Nevertheless, the history of changes in their states can be recreated from the accumulated procedures of their impact, but only in cases when the information system provides the accumulation of these procedures. The method of these calculations is very specific for every information system. The depth of accumulations is limited within several days. *Employment of the temporal data* The procedures of the historical data archiving and accumulating takes place in electronic form. In the process of archiving some data portion is thrown away due to the necessity of saving the space on the memory facilities.

[1] The hardware and the software, inherited from the previous stages, are not shown.

Table 1. The temporal data employment evolution in the Latvian Railway information systems (continuation 1)

Stages of the IS development	**Hardware and software**	**Temporal data and their usage**
4. Decentralized data processing on the basis of the personal computers (1990 - 1997)	*Personal Computers* *Intel-servers* *Desktop databases*, using the data format *DBF* *The Internet* *The Ethernet*	*Temporal data* *a*) the electronic reference sources for different timeframes organized as the set of dated files. *b*) the electronic archives of the accumulated historical data and time series for the past time periods. *c*) multi-version data of the trains schedule subordinated to the periodicity and the specified calendar. *Employment of the temporal data* a) The rules of navigation in the electronic reference sources are implemented in the programmes. It is necessary to synchronize the continually changing versions of the software and the reference sources data sensitive to the time, if these versions are situated on the clients' machines, and the solve the issues of the data non-coordination. *b*) The data archives are separated from the principal system, as it was before. The access to them has become easier, because they are stored on the disk systems which are more accessible and reliable, and their refreshment is not connected with the interruption of other systems operation. *c*) the time-dependant trains schedule is practically the series of replicated versions, and this fact implicates the redundancy and the limited depth of storage (see point 1.5.2).
5. Introduction of the corporate databases with the employment of the relation databases management system (1997- 1999)	Databases Management System *IBM DB2 UDB*	*Temporal data* The occurrence of the time dimension in the relation structures of the reference sources and practically unlimited depth of the storage. The function of the automatic saving of the objects previous states is implemented in the whole range of the information systems. The problems connected with the employment of the shared database of classifiers and codifiers are revealed. There is the pressing necessity to give the characteristics of the temporal database to the shared database of classifiers and codifiers; in this new database with the temporal features the "outdated" data are not scratched by the new values, as it was before, but stay in the database, but with defining the life expiry date – the period of their usage. Transfer to the relation environment and the reference sources flat structure decomposition complicates the interactivity with data[2]. Extension by the time dimension adds to the complexity of data access. The renunciation of the database temporal expansion takes place, and the statistic data are employed at the expense of the functionality of the supplements under development. *Employment of the temporal data* In the most operational information systems on the railways (*APOVS, Express-2*) the principle of the temporal data employment has not changed. Information systems, implementing the relation databases management system (new information systems and transmigrated from the desk-top databases), have the possibility to employ the whole range of the facilities for storage and processing *the accumulated historical data and the time series*. The access to the temporal multi-version data of the relation systems is provided with the assistance of complex SQL statements.

[2] In the process of transition to the relation environment, the flat structure of DBF of the reference sourse file is transformed into the set of the relation tables, containing many attributes, which, instead of comprising the data themselves, comprise the points of reference in other tables, and so on. In this connection it has become more difficult to describe the interaction with the data.

Table 1. The temporal data employment evolution in the Latvian Railway information systems (continuation 2)

Stages of the IS development	**Hardware and software**	**Temporal data and their usage**
6. Introduction of the Internet Technologies and the Data Warehouses Technologies (2000 and further)	*The optical lines of communication* *Blade-servers* Information Systems: *APOVS, APIKS, SAVS, Express, MySAP* *Data Warehouses technologies, operational analytical processing technologies, complex data analysis* *IBM WebSphere MS Share Point* Databases Management System *IBM DB2 UDB, MS SQL Server*	*Temporal data* *a*) multi-version data in the temporal relation structures. *b*) time is the principal dimension in the data warehouse and in the data mart, and it is also the parameter under consideration in the data transformation processes. *Employment of the temporal data* *a*) The environment of supporting the temporal integrity and the access to the multi-version data is developed. Nevertheless, the general universal approach to the solution of the problem of the temporal data employment is not designed. There are certain information systems created with the special temporal framework employment, for example, the WEB system of the inland passenger trains schedule. There two stages separated in the process of the system development: • Till the year 2003 – the system, providing the users with the data organized as the set of seasonal tables with the schedule corrections marks; the navigation of the data time dimension is absent; • After the year 2004 – the system, created in the relation environment on the temporal fundamentals basis and allowing choosing any time slice (date) in the interactive mode by the users. *b*) the key problem in the process of the data transformation is the navigation on the time dimension of the multi-version data (classifiers).

As it is seen from the above presented table, the temporal data role is constantly growing with the development of the information systems. The tendency of the increasing role of the data time dimension shows the traces of the technologies development impact, new standards and technologies of software creation occurrence, data organization and storage.

On the first stages of the informatization process it is always important to bring the discipline to the procedures of the data routine processing. The operational systems – the traditional systems of the data processing - are oriented on this process, that is why the advanced development of this system class is quite explainable. The similar tendency is traced on the railway transport as well. The necessity of satisfying the information demands of the principal systems of the railway placed the task of the historical data efficient usage in the middle distance. Processing of the historical data in some operational tasks (for example, in the express system) has been approaching to the maximum accessibility, safety and execution speed provision, as well as the simplicity of the implementation of the access program functions, which were excessively complicated even before this. This type of organization had the negative impact on the duration of the historical data storage in the accessible form and the efficiency of the disk space employment. The railway systems are highly demanding to the characteristics of accessibility and safety, and it is the reason why the information systems were designed with a special emphasis on the achievement of these exact characteristics even at the expense of the efficient organization and employment of the historical data, as well as the duration of their storage in the accessible mode.

The analytical systems – the systems of the second class under consideration, the systems of the decision making support, using mainly the historical data, are the secondary systems relatively to the operational systems class. The informatization of the analytical tasks, connected with the historical data processing, was complicated by the limitation of the technical resources and imperfection of the technologies for the data bulk volumes processing on the first stages of the process. But in the course of the successful implementation of the principal operational tasks, necessary for providing the railway life activity, there were created the conditions for the conceptually new tasks setting, and these tasks are directly connected with more efficient employment of the historical data. The recent years the new fruitful stage of the developing the temporal data technologies has been caused by the introduction of BI technology (the enterprise intellectual resources), the principal components of which are the data warehouses, the systems of the data analytical operational processing in the real time OLAP and the systems of the data intellectual processing Data Mining.

Undoubtedly, in the comparison with the 70-90 years of the previous century the contemporary level of the hardware development gives more possibilities for the efficient employment of the temporal data, giving birth to the new approaches to the organization and registration of the temporal data in the information systems on transport, considered below.

- *Multi-version temporal data.* On the Latvian Railway the series of the dated files (every file comprises the reference data referring to the specified moment in time only) were exchanged by the relation set in many information systems using various classifiers and codifiers. Every tuple in this relation is presented as a version of any indicator value, associated with the time period of its operation. The data of the objects, which did not have the temporal features before, have underwent the reorganization in the similar way. However, many of them still remain static, though their state fall under the changes in time and the process of obtaining the time dimension might increase the functionality of some information systems. *The automatic registration of the previous states of the objects for the multi-version temporal data* (saving the changes history) is implemented in a number of information systems, for example, in the system "The Registration of the Computing Peripheral Devices", employed on the Latvian Railway. Nevertheless, the existing implementations have the range of shortcomings. One of the widely-spread approaches is based on the separation of the active objects state from their history, in other words, placing them in various relation tables. The approach gives the time dimension to the data, provides the compatibility with the operating applications, working with the active objects versions only. It is important to mention the problem of the data doubling in case when the latest version of the object is contained in both tables, or, in case of doubling absence, to mention the problem of "inconvenient" search for the active object version at the specified moment, as it is necessary to look through the data of both tables. This function, as a rule, operates on the level of the user's application, and this fact makes its safety unreliable; this situation is supposed to be the drawback of these implementations. The following fact is worth mentioning: the table of the changes history is often not employed in the further processes and in its essence it serves as a log register.

- *Accumulated historical data and time series*. The majority of the issues connected with the processes of saving and processing these types of the temporal data are solved in the information systems, designed the recent years. At the same time, the inherited systems are connected with the restrictions resulting in the durability of the temporal data storage, the access to these data, and so on, as it was before. The data of the different time periods are stored in common integrated place, which is accessible for the analysis and is operable for practically unlimited time; this factor is unambiguously positive change occurring in the new systems. The data aggregation in the analytical systems necessarily takes the time dimension into consideration. The stored and accumulated aggregation result, as a rule, is arranged in the form of the temporal data, for example, the distribution of the number of the passengers-kilometers by months. Currently the historical data processing is free of the restrictions which were intrinsic for the technologies, used before, and nowadays the data processing takes place for the whole period under processing but not in portions. The data processing is usually implemented with the SQL language help, or, if the logics of computation is very complicated, with the help of the program specially designed for this. In some cases the special construction tools for the data transformation processes (ETL) are employed. Nevertheless, the special means which are able to simplify and facilitate the time dimension of the interconnected multi-version data of the reference sources have practically not been developed yet.

Although it is necessary to mention the positive developments and shifts in the area of the temporal data employment in the information systems on the Latvian Railway, the full-fledged implementation of the temporal technology is hampered for the variety of causes. To the best of author's belief, these causes are typical not only for the enterprise under consideration, but also for the majority of the transport companies and organizations. These causes comprise the following:

1. Absence of knowledge in the field of designing the systems of the temporal databases management; this factor results in the "reinvention of the wheel" in many cases, as well as in increasing the development cost, in negative impact on the information system work efficiency and in a growth of the errors occurrence risk.
2. "Pressure" of old technologies and the general methods of designing and developing the databases applications. The existing prototype of the system under development and the inherited traditions of the databases applications designing have a significant influence on the process.
3. The deliberate conscious renunciation of giving the time dimension to some objects with the view of simplifying the process of working with them, and this fact results in some definite loss of the system functionality.
4. The tough demanding time frameworks, allotted for the project implementation, do not allow paying close attention to the data time dimension.

As appears from the above said, the issues of taking into consideration the human factor impact and the processes of making the temporal technology popular among the developers also have great importance. The Latvian Railway breaks fresh ground in this direction.

1.4. Interconnection of the Latvian Railway Information Systems with the Temporal Objects

As it was marked in point 1.3, there are a lot of information systems (IS) employed on the Latvian Railway, and these systems were designed in different time periods and with different technologies implementation. All these systems utilize the temporal data but to different extent. There is a special support for some of the temporal data, but for some of them there is no such support. The temporal component occurs either in the information systems processes, interacting with the time dimension of the data in implicit or obvious way, or in processed data, even if they have no time dimension, being in their essence the temporal data.

The scheme of the interconnection of the Latvian Railway information systems and the objects with the temporal characteristics, as well as these objects belonging to one of the temporal data types, are presented in Fig. 1.4. The scheme represents 17 different systems: APIKS-2, APNIS, APFIS, APOVS, and so on. These systems are functionally divided into four applied areas: freight transportations, passenger transportations, economics and finances, and infrastructure management; additionally, the Referenced Data is detached into separate group.

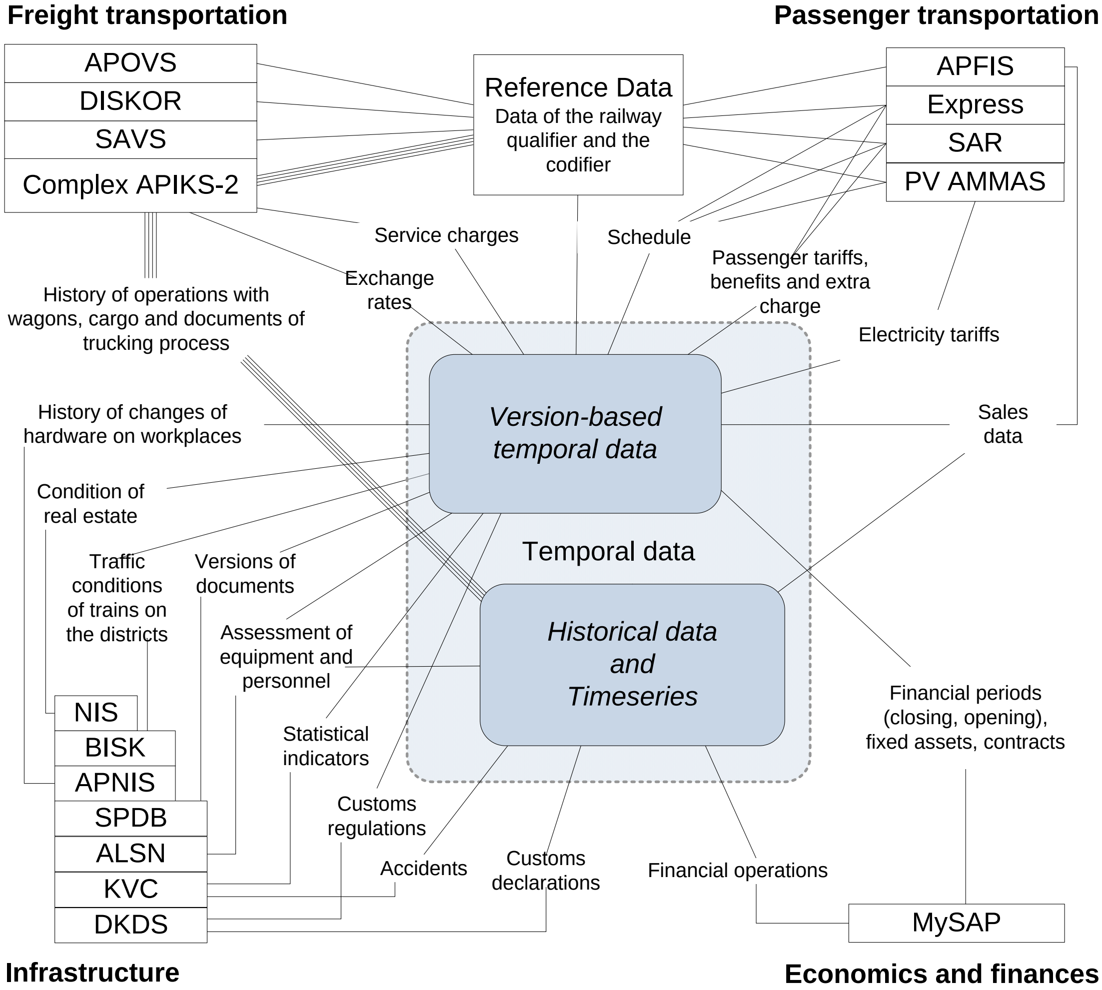


Fig. 1.4. Temporal data in the information systems of the Latvian Railway

The above presented scheme presents the temporal data divided into two groups:

1. The data, the actuality of which is connected with the determined time period. *Multi-version character, multi-variant essence* or the *periodicity* are the peculiar features of these data;

2. The data, having the exact linkage to the exact determined moment of time. This group comprises *the time series* and *the accumulated historical data*. These data have the tendency to be subjected to the various temporal aggregations, as a rule, with the view of receiving the constantly recurrent reports (the statistic indicators/ characteristics). The temporal data belonging to the first group are the results of these aggregations. For example, as a result of the procedure of the temporal aggregation of the accumulated data of the tickets sales for the travel in the passenger trains cash business transactions it is possible to receive the statistic characteristics, the values of which are active in the determined time periods: “the number of transported passengers during May month of year 2005 (in other words, in period from May 1, 2005 till May 31, 2005)”, or “in April of 2005” and so on.

The analysis of the above mentioned information systems performance has presented that the temporal component appears to the different extent practically in all processes. It is revealed either in the procedures of the information systems, interacting in the implicit or evident way with the data time dimension, or in the processed data, which are temporal in their essence even if they do not display the time dimension. Consequently, there are two principal components, giving the key role to the temporal factor, among the elements of the information systems on the railway:

1. Temporal data. Their arrangement, storage and access to them;
2. Data processing procedures, interacting with the temporal data. The capability of using the data time dimension and the capability to adjust to the necessary time period play the principal role in these procedures.

Many temporal objects, shown on the scheme of the temporal components employment in the information systems on the Latvian Railway (Fig. 1.4) take the key roles in the information systems functioning:

- the temporal data of the railway classifier and codifier (the referenced data) are employed as the fundamental of the performance of the majority information systems of the freight- and passengers transportations;
- the trains traffic schedule, dependant on time, lies in the basis of the operating system of tickets for the main line passenger trains reservation and sales (*Express*); the reference interactive scheduling information system (*SAR*); the systems of processing the route of the passenger trains engine driver and the fuel expenditure records (*PV-AMMAS*). The role of the temporal data of various tariffs (fare costs, energy resources costs) is also evident in these systems;
- the indicators of the enterprise activity are always linked to time. The operation of the economic and financial information systems is based on the temporal data fundamental: the financial periods, agreements, payments, reports, lists of staff and the employees' wages, company deposits, and so on;
- the principal object of the system of the notices delivery *BISK* – the notice has several time dimensions: the time of registration in the system, the time of sending for the agreement, the period of activity (time of coming into operation and time of going out of operation);
- the system *SPDB* in the process of the documents changing (schemes, plans and other technical documentations of the stations) needs in providing the automatic saving of the previous versions and ensuring the access to them.

Table 2 describes the temporal objects of the information systems and the tasks connected with them. There were determined five groups of the tasks, similar to the temporal data in their demands and in the type of interaction:

① Temporal data storage and access to the data of various time periods;

② Automatic saving of the previous object version;

③ Restriction of the access to the data on the basis of time, they refer to (on the basis of the data time dimension;

④ Temporal data analytic processing;

⑤ Augmenting the performance speed in the process of the querying to the temporal data.

Solving some of these tasks in a number of systems can bring to existence the issue of supporting the inherited applications or make the interaction with the database more complicated. Accordingly, the new additional task arises:

⑥ Providing the temporal upward compatibility of the database.

Table 2. Temporal objects of the information systems and the tasks, connected with them

Information Systems			Systems Temporal Objects	Tasks, connected with the temporal data processing*
Referenced Data System	The centralized railway classifier and codifier (**KLASIFIK**)		Names of the railway stations, cities, residential localities, states; railway stations codes for several information systems; the set of procedures carried out on the railway stations; freight codes and names, railway lines and sectors; rates of currency; list of internal customers and list of external customers of the railway	① Data storage and access to the data of various time periods ⑥ Support of the classifier interface for the systems, which are not able to work with the temporal data ③ Restricting the access to the data of various time ranges: history inviolability, restricted access to the information which has not come into operation.
Freight Transportations	Automatic management of the freight classification yards (**ASUS**)		Disposition of the carriages and the transported freights, running the movable equipment of the railway system, field of freight operation *(change of these objects state are available from the history of the procedures with the carriages and freights)*	⑤ Augmenting the speed of the access to the bulk volume of the historical data from the operational and analytical tasks ④ Station operation analysis: carriages and locomotives standing idle on the stations ④ Planning the pushers loading
	Information system of registration of the transportations revenue **(APIKS)**	Shipment planning (**IP**)	Regulations by the weights of the railway sectors, carriages weight-carrying capacity, the Referenced Data from KLASIFIK	① Data storage and the access to the data of the various time periods ① The Referenced Data employment
		Services sales registration (**PP)**	Referenced Data from KLASIFIK	① The Referenced Data employment
		Arrangement of the banking documents **(BD)**	Referenced Data from KLASIFIK	① The Referenced Data employment
		Invoices processing **(CP)**	Basic and special tariffs, additional charges and discounts, terms of transportation, conditions of determining the payment for transportation, calculation algorithm, invoices, payments codes and permission for their employment in various documents, Referenced Data from KLASIFIK	① The Referenced Data employment ① Data storage and the access to the data of the various time periods ⑤ Historical data registration and accumulation ④ Historical data processing
		Freight transportations analysis (**AIS**)	Freight traffic, freight transportation revenues *(changes of these objects states are reflected in the papers of the freight-transportation procedure)*, Referenced Data from KLASIFIK	① The Referenced Data employment Synchronization of the data of the operative and analytical systems (the main complexity: the cycle of the invoice state changes can be longer than several weeks) ④Data analysis: underutilization of the carriages, loss due to the currency corridor employment.
		The registration of the Referenced Data of the freight transportation process (**NII**)	Referenced Data from KLASIFIK	① The Referenced Data employment

Table 2. Temporal objects of the information systems and the tasks, connected with them (continuation)

Information Systems		Systems Temporal Objects	Tasks, connected with the temporal data processing*
Freight Transportations	The system of the registration of the railway infrastructure employment and settling with the carriers (**ADIIAS**)	Infrastructure state and belongings, employment tariffs, Referenced Data from KLASIFIK	① Data storage and the access to the data of the various time periods ① The Referenced Data employment
	System of registration of the carriages packaging arrangement (**KVK**)	Physical state, carriages packaging arrangement and description, Referenced Data from KLASIFIK	① The Referenced Data employment ① Storage of the history of the carriages packaging arrangement change (mounted wheels, bolsters, etc.), change of owner, change of the carriage type and number
Passenger transportations	Passenger trains schedule (**SAR** – inland traffic, **SAREX** – international traffic)	Schedule, fares, allowances, terms of allowances receiving, Referenced Data from KLASIFIK	① The Referenced Data employment ① Support of the schedule for different time periods. The registration of operative long-run and short-run schedule changes ⑤ Providing the speed performance in the process of query to the schedule
	The financial and statistical information analytical system of the passenger transportation (**APFIS**)	Passenger traffic, stations loading, ticket office operation *(changes of these objects state are reflected in the data on the tickets sales)*, Referenced Data from KLASIFIK	① The Referenced Data employment ④ Processing the data of the tickets sales of different time ranges in the process of analysis ③ Discrimination of the access to the data of different time periods in the process of analysis ④ Analysis of the data in the form of "what-if".
	Automatic system of booking seats and the railway passenger transportation management (**Express**)	Passenger traffic, schedule, trains routes, fares, allowances, stations (name, type, procedures)	① The Referenced Data employment ① Support of the schedule for various time periods. The registration of operative long-run and short-run schedule changes.
Infrastructure	Registration of the hardware and software (**APNIS**)	The railway staff, the state of the technical equipment and the software on the working place	② Automatic storage of the history of the staff movement between the structural departments ② Automatic storage of the history of changes of technical equipment and software on the working places (the process of logging)
	System of the stations plans database **(SPDB)**	Documents of the stations technical descriptions: plans, schemes.	② Automatic storage of the document previous version in the process of its modification or removal
	Systems of the notices delivery **(BIS-K)**	List of notices, list of the railway sectors and tracks	① Data storage and the access to the data of the various time periods ③ Restriction of the access to the notices which have not come into operation

* Symbols ①,②,③,④,⑤,⑥ mark the tasks which are similar to the temporal data according to their interaction.

Three information systems were distinguished by the author as the objects of detailed consideration:

1. The passenger trains schedule system for inland traffic *SAR*;
2. Referenced Data System *KLASIFIK* (the centralized railway classifier and codifier);
3. Financial-Statistical information system of the passenger transportations *APFIS*.

The chosen information systems are described by the fully completed set of the temporal peculiarities. The procedure of choosing has taken into consideration the principle of significance in the working process arrangement, the degree of typicality in the railway field and existence of problems in the efficient management of the temporal data.

1.5. Problems of Employing the Temporal Data in the Information Systems on the Latvian Railway

The information systems on the railway, possessing the temporal aspect, can be divided into the following classes according to their interaction with the Databases Management System:

- systems oriented on the prompt (transaction) data processing;
- systems oriented on the data analytical processing and the decision making provision.

Problems and peculiarities of the temporal data employment in the operational and analytical systems are different, and it is more reasonable and appropriate to examine them in different ways. There are only certain information systems on the Latvian Railway and their detached components, demonstrating the especially peculiar characteristics of the temporal data employment considered in details.

The issues of the temporal data employment in the operational systems are considered in the research. The operational systems lie in the basis of life activity of any transport enterprise. They are the elements responsible for carrying out the principal transport procedures: the tickets reservation and sales, arrangement of the documents for the freight transportation, the procedures of scheduling and the train timetables delivering, freight packaging arrangement on the stations, loading planning, etc.

The operational systems are often called the transaction systems or the systems of the prompt processing of the transactions (OLTP – On-Line Transaction Processing). They are characterized by the enormous number of small transactions or operations of recording – reading. The operational systems are adjusted and optimized for carrying out the maximum amount of transactions during the short time periods. As a rule, these operations do not require the great flexibility, and the fixed set of secure and safe methods of data collection and reporting is employed very often. The number of transactions, carried during the minute is supposed to be the characteristics of efficiency. Usually the detached operations are very small and they are not linked to each other. However, every data recording (reservation, ticket-office operation, data on freight loading or unloading, call to the support service, order, visiting the company Web site, and so on) can be used as a source of receiving the conceptually new information, and exactly for working out the report statements and carrying out the analysis of the transport enterprise functioning. These procedures are usually accomplished by the analytical systems.

As a result of the analysis of the temporal data employment in the operational systems on the railway transport the following problem clusters were determined:

- the efficient organization of the temporal data storage;
- the efficient access (the operation of reading) to the temporal data from the point of view of the implementation and employment simplicity, as well as from the position of accessibility and privacy criteria;
- the efficient manipulation with the temporal data (the operations of removal, changing and extension) from the point of view of the implementation and employment simplicity, as well as from the position of accessibility and privacy criteria;
- the efficient access to the accumulated historical information of the bulk data volumes.

These problems are examined in details as exemplified by demonstration of two operational systems, illustratively presenting the temporal data employment: the Referenced Data System and the reference system of the train schedule SAR.

1.5.1. Referenced Data System

The Referenced Data System is the united integrated classifier and codifier of the transport area objects the information on which is absolutely necessary for arranging and supporting the great amount of procedures of the railway enterprise. The Referenced Data System comprises the following information: the reference source of the states, railways and stations; the directory of currencies and currencies rates; the directory of extra charges; the reference source of the freight forwarders allowances and coefficients, directories of the freight classification, and so on. Practically all components of the Reference Data System are subjected to the changes in time. The data in the directories become outdated and are exchanged with the new ones; however, their previous values are necessary for the analytical systems. Some of the Reference Data System changes are projected in advance. It means that the new values are prepared for the future, and these values are to come into operation on a due time.

The majority of the operational and analytical information systems on the railway transport are connected to the centralized database of the Referenced Data System. Consequently, any structural changes of the Referenced Data System connected with the data extension with the time dimension might have negative impact on the functionality and efficiency of the existing information systems. Accordingly, the problem lies in the field of providing the temporal upward compatibility with the information systems in operation.

The Referenced Data System has been chosen as one of the principal investigation object. The versatile analysis of the issues of this system, as well as the approaches to their solution, is described in the subsequent chapters of the research.

1.5.2. Train Schedule Systems

There considered two information systems approaching to solving the task of the train schedule registration: SAR system and Express system.

The passenger train schedule system SAR is the reference system, presenting the information on the passenger trains schedule for any period of time on the web site. Its principal task is the storage of the information on train traffic in the past, in the future and at the current moment, as well as provision of the access to this information for the railway customers on the purpose of travel planning and for the analysts on the purpose of carrying out the versatile analysis of the train traffic.

The system of the passenger trains schedule possesses the following peculiarities, determining the rather significant level of the complexity of its implementation:

1. The active schedule of the trains traffic is arranged on the basis of core fundamental (seasonal) timetable and the number of other schedules, occurring as a result of such procedures as the trains removal, changes in the traffic schedule due to the accidents, pre-planned or unplanned repair operations, public events (the City Days, concerts, etc.), short-run including / excluding the stations in / from the route. Further these schedules are called as *the schedule versions*.

2. In the course of time the various versions rotate. One and the same version can come into operation repeatedly: the fundamental timetable exchanges the altered one, and vice versa.

3. One and the same train might have the different schedules on the different week days.

4. On holidays and other special days (for example, holding the NATO summit in Riga in November 2006) all the trains run in accordance with the day off schedule. There is an exception for the certain trains when the special timetable is assigned for these days. But from practical side it is known that only 1% of all trains are assigned with the special schedule.

5. The trains run on the days of week or month the schedule is assigned for (for example, on Fridays only, or twice a week on Tuesdays and on Thursdays, on the working days, on the week-ends, on the even days of the month, daily).

6. There is a break in the local commuter traffic in Latvia at night. In this connection the following peculiar features can be mentioned: the passengers and the system percept the opening and the closing of the day in different ways: the system day spans from 00:00 till 23:59, but the passengers' day can span for more than 24 hours, and for convenience are determined to span from 04:00 of one day and till 03:59 of the following day. All the trains, departing the night of the following day, the passengers refer to the current day. There is a special term – the night trains. For the passengers' convenience the system of the schedule delivery should take into account this peculiarity of understanding the day notion.

7. The system of scheduling the long-distance trains must take into consideration the time zones, changing in the course of the train movement along the route.

There are also peculiarities connected with the schedule data management (schedule registration):

8. First of all the system obtains the core fundamental schedule, then with the time running the operative changes are contributed to this schedule. Consequently, several versions of the schedule appear. Quite often the period of operation of such altered schedule is so prolonged, that it also becomes subjected to certain corrections. The operational time of these or those corrections can be contracted or prolonged.

9. The data on the train schedule version comprise the concise information about the order, in accordance with which the new schedule is introduced, the date of this schedule version coming into operation, the date of going this schedule version out of operation, the type and the number of carriages in the train, the set of stops at the stations with information

on the time of arrival or departure, the type of the transport at the stations (in case of certain repair works the train between the definite stations can be exchanged by the bus).

10. The system of schedule is utilized by the customers by means of the Internet very intensively. As the statistics states, the number of addresses to the system reaches 50 times during one minute period. The search for the corresponding train takes place under different conditions and criteria consideration: the station of departure and arrival, the date and the time range of the departure, the possibility of travel in transit, and so on. As all the queries to the system occur in the on-line mode, it becomes very important to provide the velocity of the result delivery. This task is complicated by the great volume of the processed data. In the relation database one version of the timetable for one train takes about 1 KB. The minimum of data is taken into account, without considering the reference information (such as the stations names, trains types, etc.). The integrated volume of all data of the passenger trains commuters schedule of the Latvian Railway takes more than 100 MB (about 250 thousand notes).

The system of schedule is closely connected with *the Referenced Data System***,** the objects of the railway classifier and codifier, for example, the stations. Some of the stations characteristics alter in the course of time, for example, the name, the code, the type, the set of the carried out operations, and so on. In general, for the schedule system it is necessary to have only the active values, nevertheless, there is the necessity of obtaining the information on the previous or future values. For example, the information about the name of the station for the certain time period is the information the passengers operate with, sometimes having no idea about the change of the station name.

It is reasonable to use the data of the shared referenced data system, whereas the operators of the local system of schedule are frequently not able to support this referenced data in active form. For example, the schedule of the long-distance trains (the Baltic countries and CIS) embraces more than twenty thousand stations, and every month there are some changes in the schedule, connected with the changes of the stations names, including the new stations in the list, excluding the existing stations, and so on.

The schedule system is connected not only with the railway classifier but also with *the directory of fare tariffs for passenger trains travel.* The different variants of the tariffs, as distinguished from the schedule, do not usually fall under tendency to enter to or to exit from the shadow zone repeatedly. If they were changed once they do not come into operation again. The only exception from this is the discounts coming in operation periodically, from time to time. For example, the Latvian Railway uses the discount of 25% for the definite trains group on the working days, but on weekends and holidays the principal tariff comes into operation.

The system of schedule SAR has been chosen as one of the cardinal investigation objects. The versatile analysis of the problems of this system, as well as the ways of these problems solution is described in the subsequent chapters of the research.

There is the examination of the organization of the scheduling temporal data in *the international system of the tickets e-sales and e-reservation "Express-2".* The timetable of the passenger trains was arranged in the form of the template or mould for the entire year and it was introduced to the system "Express-2" in this form. This timetable was called as "the reference timetable". Besides this schedule there also existed the operational timetable, according to which all the operations of the tickets sales and reservation took place. The operational timetable existed in the form of the 45-days unit. First time this unit was practically the copy of the reference timetable, but then it was corrected with the operative changes in the train traffic. The current day was the beginning of the functioning unit. On the expiry of the 24-hour time unit, the unit shifted forward: the first day in the beginning of the unit was eliminated, and the day from the reference timetable was copied in the end of the unit. The accumulated operative data were as well stored for the limited time period – 45 days.

Consequently, the schedule in the "Express-2" system possessed the time dimension, but with some reserve – the time length was limited (for the reference schedule – the year, and for the operational ones – only 45 days). The exact peculiar version of the schedule was stored for every specified day, even in case that this state was not changed for several days. Accordingly, the timetable data were replicated manifold. Most probably, this data arrangement was connected both with the restrictions of the data storage and processing possibilities, and with the complicated problems of the synchronization of the data with the regional centers. Employing the language of the contemporary technologies of the temporal databases, it could be said that the timetable in "Express-2" was arranged in the form of the consequence of the set of the time slices, having the specified time periodicity of one day.

The tickets and the criterion of the seats occupancy in the trains are also present the temporal objects: the seats are exposed for sale, the reservations are booked, the tickets for these seats are sold, but then they can be cancelled. In the system "Express-2", as well as in the system APOVS, the description of the information systems objects comprises only the active state of these objects, which was altering in the process of the introduction to the system such information as the reports on the operations with the tickets and seats. However, the data on the state of these objects at any moment of time could be found out on the basis of the accumulated data about these operations.

The system "Express-2" employed the schedule of the long-distance trains only, since the tickets sale and reservation with the help of this system was restricted by these trains only and sales of the season tickets for the local commuters, and the last mentioned activity does not employ the information on schedule. The automation of the tickets sales for the local passenger trains (commuters) happened only in 1998. The software for the cash registers KR1Z and PCU was developed. In the process of the ticket sales for the local commuters the schedule was not taken into account, as the ticket is active for some definite period of time – 6 hours for the travel in one direction, and 12 hours in case of a round trip. Accordingly, it was quite possible to sell the ticket for the travel which could not take place due to the real active timetable. Starting with 2005 the Latvian Railway employs the new-age cash registers with the new software implementation.

1.5.3. Provision of the Procedures of the Data Warehouses Functioning

There is investigation of the issues of the temporal data employment in *the analytical systems*.

As it was mentioned previously, the full-fledge analysis of functioning the railway transport system should comprise not only the estimation of the current and the previous enterprise activity, but also its operations forecasting. With this exact purpose the contemporary data warehouses containing the bulk volumes of information are created. The data warehouses consolidate the data which have been generated by the enterprise operative information systems from the external sources. The issues of providing the data reliability, trustworthiness and completeness, as well as the issues of taking into consideration the data temporal nature and opportunities of their flexible utilization come forward. The data warehouses have the shareable nature of destination, and this fact brings a lot of problems, connected with the provision of the consolidated data privacy, by the virtue of the fact that these data warehouses are used not only by the analysts of the internal structures, but also by the analysts of the external commercial structures. It is frequently necessary to distinguish the restricted specified analysis zone for every analyst, working with the analytical systems. The results of investigating the analytical systems, data warehouses and other issues connected with these themes are stated in the author's works [1, 2, 9, 10, 11].

At the current moment the Latvian Railway employs various systems of the transportations accountancy and registration; these systems process over 12 million transactions annually on the passenger transportations only (see Fig. 1.5).

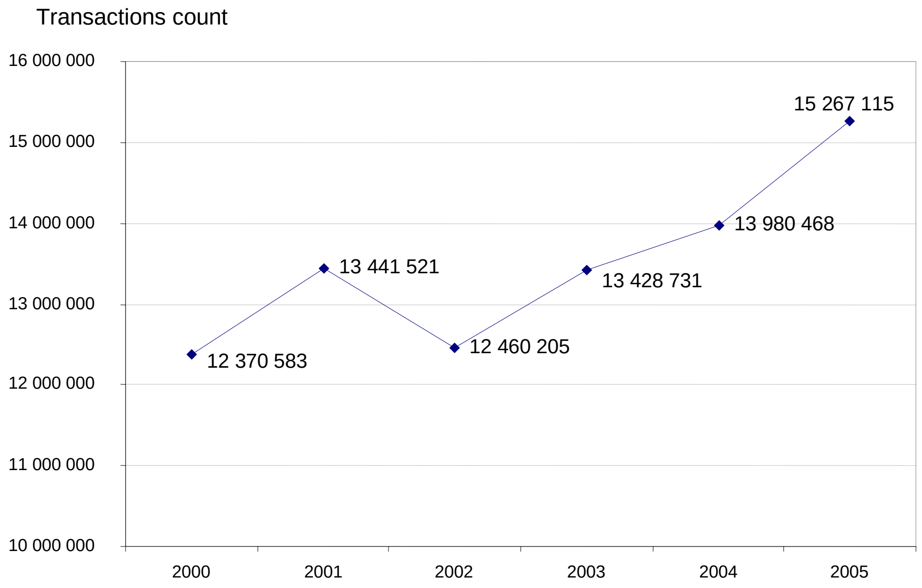


Fig. 1.5. Distribution of the Registered Transactions Number by the Passenger Transportation Systems by the Years

Some objects, which are temporal in their nature, do not possess this characteristic, when they are used in the active operational systems. This situation can be explained by the whole range of reasoning:

- there is no necessity of employment of the historical information in some operational systems, and, consequently, the data in these systems are not temporal;
- some systems have the technical restrictions occurred in the process of their development or maintenance, and they cannot endow their objects with the temporal characteristics. The range of the inherited and still functioning systems, developed in the end of the previous century, can serve the example of the above mentioned phenomenon;
- there are systems, storing the historical information, but due to the technical reasons for the limited time period only.

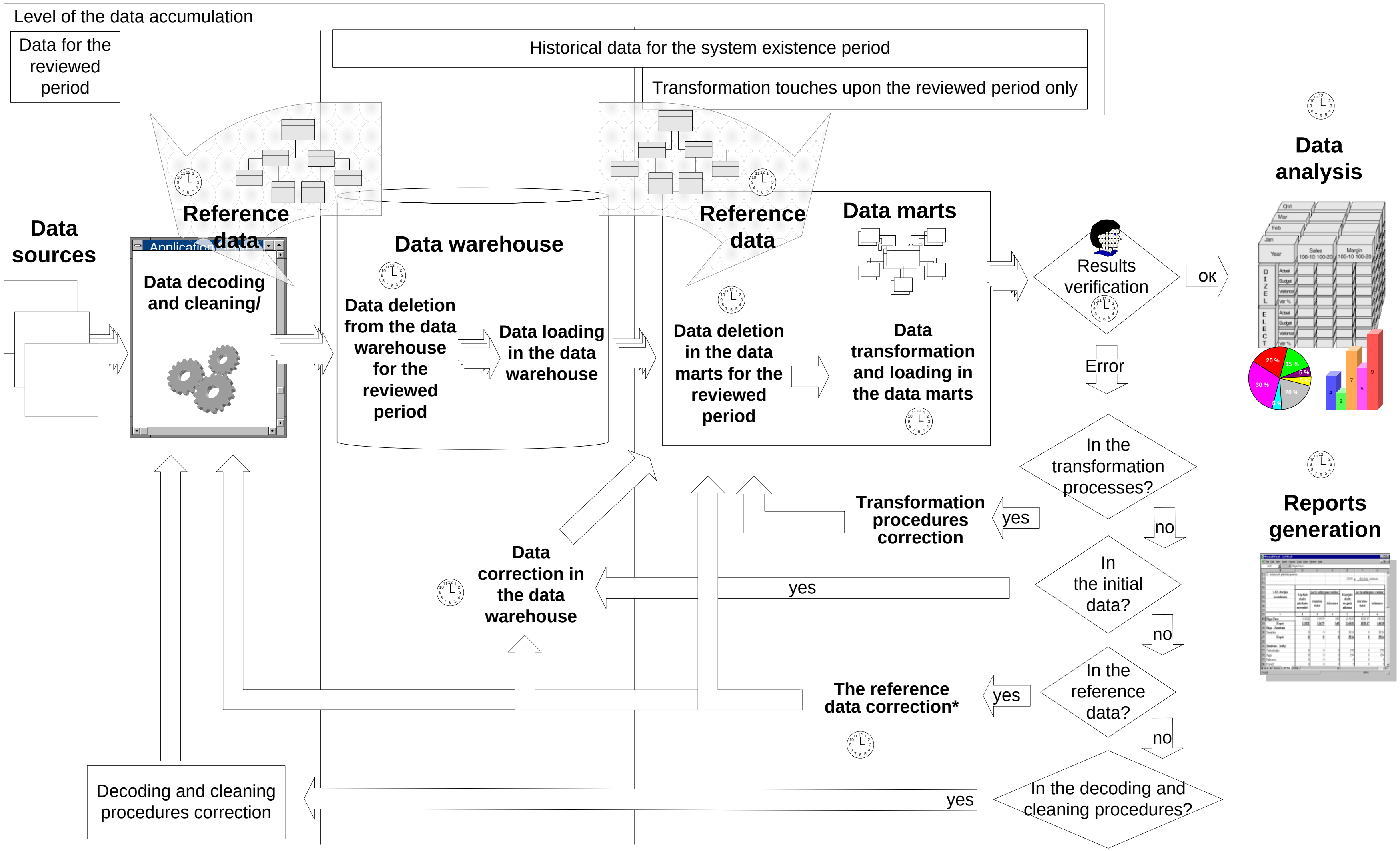
For the objects of the systems of this type it is possible to obtain the full-fledge temporal characteristics only in the data warehouse.

The data warehouse creation is a very complicated task, especially for such big enterprises as the railway. The reasons and causes of these complications were considered by the author in works [1, 2, 9, 10, 11, 14]. Among them the issue of the efficient organization of the procedures of the data loading and transformation in the data warehouses should be mentioned; the important role of the processed data factor should be taken into consideration.

The procedures of the data immersion in the warehouses and distribution them among the data marts are rather critical operations, whereas the process of the interaction with principal warehouse resources takes place, and the warehouses include the data for the whole period of the system existence. In case the errors were discovered after the ordinary data loading, the data of the warehouse are liable to the partial removal: the data for the period of time under processing are cleaned from the warehouse and from the data marts. The situation when the errors appear is not possible to eliminate, because the system is “alive” and the new conditions arise regularly, both in the framework of the original data and in the customer’s requirements. The risk lies in the field of the warehouse integrity, which can be broken in the process of the data cleaning if the data of other periods are affected. The reinstatement of these data can be excessively costly operation if to speak about time and other recourses employment. The situation can be aggravated by the delayed discovering of the troubles in the data warehouses; and in this case the probability of the infeasibility of the full data restoration is rather high. The situations with the failures of the data marts are not so critical, since the data marts are designed on the basis of the data warehouses. But in the course of time the processes of the data transformation are changed, as well as the Referenced Data System information does, and the restoration of the data marts by the way of the repeated transformation may probably not present the same result. The restoration of the warehouse or its parts from the backup copies is not a good solution as well.

There is one more problem connected with the complexity of the re-generation of the data in the data marts for any chosen period. The hardness lies in the fact that the data transformation processes can be very complicated. On the purpose of the designing simplification, the whole procedure, starting with the data loading and finishing with the preparation for the analysis, is attempted to tie to the routine order, existing in that or in this system. This approach takes place in the information-analytical system of the passenger reporting *APFIS*, operating on the Latvian Railway; the data in this system are loaded in the warehouse several times a month, but always for the previous month which is the last accountable period. This fact was fixed in the system, and it simplified significantly the system structure. In cases of deviation from the existing order of the data loading the partial cleaning of the warehouse and the data re-generation was rather complicated (in terms of implementation) and risky operation (the risk of the occasional disturbance of the data integrity).

There is the detailed consideration of the technological process of the analytical system of the passenger transportations *APFIS* and distinguishing the critical operations connected with the data integrity damage in the warehouse. The technological process of the analytical system of the passenger transportations is presented in Fig. 1.6. in the schematic form. The mark  on the diagram shows the areas of interactivity with the data time dimension, including the following: the procedure of the implementation of the time slice of the data, referring to the processed time range, and the procedure of the choice of the required version among the temporal multi-version data.



- Time slices implementation

The reference data correction*– depending on the fact what data have been corrected or replenished, the technological process recommences from the different stages. It is sometimes convenient to correct the data in the warehouse, but not to decode and to reset them again

Fig. 1.6. Diagram of replenishment of the data warehouse and preparing the data for analysis

The data from the operative sources (*Data Sources*) pass the procedures of decoding and cleaning, and then they are loaded in the warehouse (*Data Warehouse*) and distributed among the data marts (*Data marts*). Distribution is a complicated process of transformation, and the database of the referenced data system (*RDS*) participates in this process. After the data loading in the data marts the analyst can estimate the result and make a decision on the data readiness for the following analysis and the reports generation. In the process of the result estimation the analyst operates in accordance with the aggregated data from the operational systems and other sources. If the result is unsatisfactory or inadequate, the part of the technological process is repeated again and accomplished with the operations on cleaning the warehouse and the data marts. Depending on the source of the error occurrence the technological process recommences from the different points. The following errors sources can be mentioned:

- the original data;
- the procedures of decoding and cleaning;
- the database Referenced Data System;
- the data transformation processes;
- the problems occurring in the process.

In case the error has been stated in the procedures of decoding and cleaning the correction of these procedures take place and decoding runs according to the new rules. Before loading the corrected data in the warehouse it is necessary to clean the incorrect data from the warehouse which were loaded earlier. It means the necessity of distinguishing from the bulk data volume in the warehouse the part, referring to the data which will be loaded again. It is difficult to put the automation in this operation in full volume, as only the part of the data is re-loaded. That is why the operator can employ the prepared and tested facilities. However there are the situations when it is necessary to clean the warehouse manually by records or with the implementation of the SQL queries and employing the indispensable filters in them. This operation is rather critical, because the operator can occasionally remove the essential data from the warehouse in case of poor implementation of the filters.

If the errors are discovered in the original data, these data are corrected in the warehouse directly. The errors in the data are unconventional, that is why they are corrected, as a rule, manually with employing the SQL queries and implementing the necessary filters. But there is again the risk that the operator can touch the not referred data.

Occurrence of any of the above mentioned mistakes in the technological process promotes the further regeneration of the data in the data marts. The regeneration consists of cleaning the data marts which can contain the error, and further loading of the transformed data in the cleaned data marts. The data mart cleaning can be automatized, but supporting the automation facilities requires the additional efforts, connected with the testing the correctness of the data removal from the data marts.

Consequently there are at least four critical operations occurring in the technological process of the analytical system, and in the process of fulfilling these operations the probability of damaging the integrity of the warehouse information resources is inadmissibly high. These processes are following:

- the partial removal of the data in the warehouse;

- the correction of the data in the warehouse;
- the correction of the database of the Referenced Data System;
- the partial removal of the data in the data marts.

The criticality of the operations increases in cases of breaking the accepted order of loading the data in the analytical system when the current period under processing is situated further in the past, compared to the last processed period. The following problems occur in this case:

- the data transformation for the data marts is based on the data of the previous period under processing. Accordingly, the part of the technological process should be done cyclically, starting with the period under processing and till the last one, existing in the system;
- the procedures of the data marts cleaning and the data transformation is not intended for the processing the data for any time period, and work in the time range, specified by the stated order of the analytical system employment;
- the Referenced Data System (in case of non-temporal database or incorrectly done the time slice) does not correspond to the period under processing, and the result of the data processing can be incorrect.

Taking into account the above mentioned problems it is suggested the perfection of the technological process of the analytical system of the passenger transportations by the way of developing and implementing the methods, decreasing the risk of damaging the data warehouse integrity and simplifying the preparation of the data for analysis of any time period.

1.5.4. Information Security Maintenance in the Temporal Databases

The transport enterprises operate in the condition of hard competition. The competition makes from one hand protect the data from the competitors' access as safely as possible, but from the other hand the competition makes the enterprise open access to these data for the company customers. The temporal data form the basis of practically all vital and essential business-processes on transport. As a result of this, the enterprises are very sensitive to these data distortion or damage.

The temporal data often promote the necessity to restrict the user's activity in time: to forbid the information processing for the previous time period, or limit his access to the information of the future period, in other words, to restrict the user's activity by some specified time period. There is also necessity to provide the chronological access according to the type of the demanded operation: for example, the user can have possibility to analyze the information for the previous two years, but he is able to modify the data for the current month only.

Despite the decades of investigating the field of the data safety, the existence of a great number of standards and ready solutions, the issue of controlling the access to the temporal data is still urgent for the majority of the big transport companies as it was before. The urgency and importance of this issue is explained by the whole range of reasons, the most essential of which are the following ones:

- *The value and privacy of the accumulated information resources*. The accumulated historical data present the great value for the procedures of planning and forecasting the enterprise activity, for distinguishing the tendencies and decision making. The task of providing the information security of the temporal data is difficult to overestimate, whereas exactly the quality and integrity of these data determine the efficiency of the information systems functioning [1, 2].
- *The existence of the well-developed information infrastructure.* For example, the infrastructure of the Latvian Railway comprises more than one hundred and fifty thousand registered users of the computing network, several hundreds of which employ the corporate database very actively in the process of managing the general resources of the enterprise. The users' number and their roles are regularly changed. In accordance with the above said the problem of providing the information security of the multi-users temporal databases is becoming more and more urgent. The dynamics of the increased number of the users is presented in the Fig. 1.7.
- *The dynamic of transition to the new technologies of the information systems designing*. For example, the nature of provision of the data privacy has changed revolutionary due to the occurrence of the personal computers, due to the transition to the multi-link architecture and due to the data accessibility via the Internet.

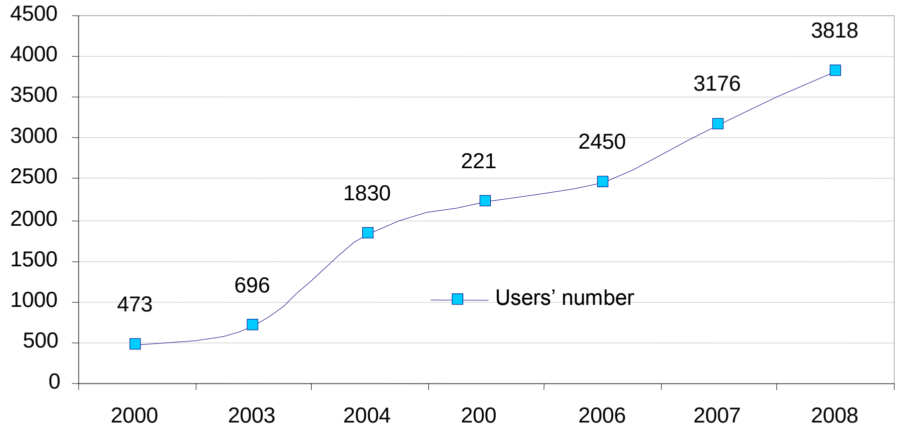


Fig. 1.7. The dynamics of growing the users' number of the Latvian Railway local network

The safe management of the shared corporate resources, part of which is presented by the temporal data, is to correspond to the enterprise security policy, bases on the numerous laws, regulations, internal orders, which, in their turn, can be operable only for some time (they are temporal). As an example of the information system falling under these requirements, it is possible to point to the following information systems of the Latvian Railway: the system of classifiers and codifiers (KLASIFIK), the system of accountancy of the real estate objects, spread over the territory of Latvia (NIS), the working place of the fare collector (PK), the computer peripheral devices accountancy (APNIS). Only in the above mentioned tasks the policy of security assumes the employment of different aspects of the access discrimination: the time frameworks, the territorial and administrative division, functional roles and the information privacy. Fig. 1.8 presents the complex representation of the task of the information security provision and numerous factors, influencing the level of access to the information object.

Despite the fact, that the issue of safety of the relation data for the current moment has been considered rather extensively, the implementation of the complex conditions of the security policy in the information systems presents the serious problem.

The complexity lies in the fact that the generally accepted approaches in the field of the data safety do not solve the key problem – these approaches do not allow employing the information, comprised in the data themselves as an inherent part of these data, as a facility, restricting the access to the information. For example, the criterion for restricting the access to the out-dated data or the previous objects versions lies in the time dimension of these data, or in these data themselves.

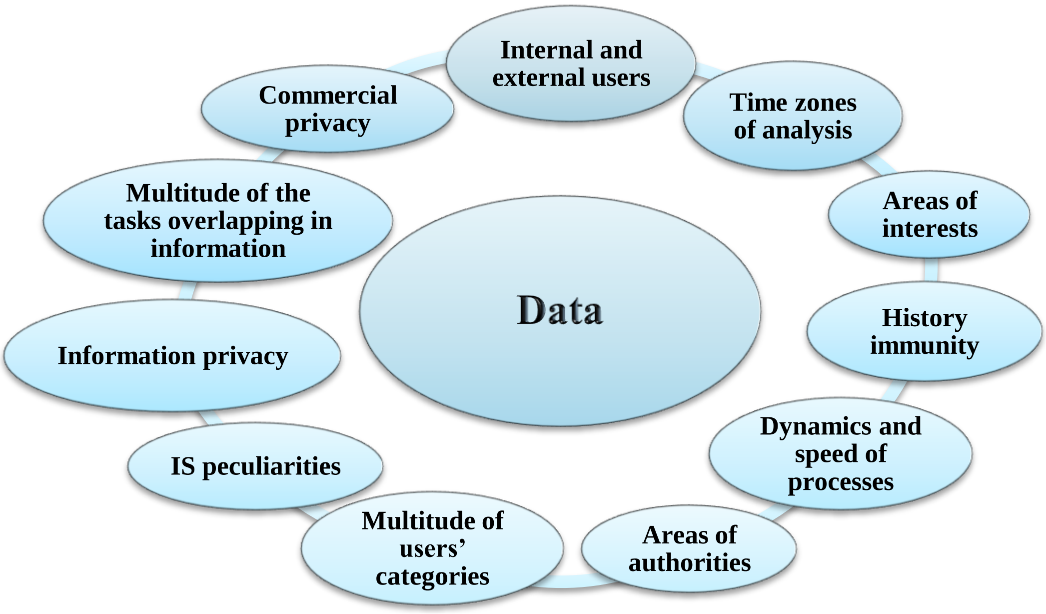


Fig. 1.8. Factors having impact on the access to the data

It is known that the principal factors, decreasing the quality of the data of the corporate information systems are not the intentional attacks of the external violators, but the malfunction of the equipment, administrators' mistakes, applications errors, the users' mistakes [112]. For the temporal databases, the semantics of which is complicated compared to the conventional relation databases, the truthfulness of this assumption is displayed especially obviously. That is why the main attention should be paid to the issue of protection of the historical data from the registered users of the system. The problems of non-authorized access, cracking, and network security are not the objects of the research under consideration and are not regarded here.

Conclusions of the First Chapter

1. The performed analysis of the information systems on the railway indicates the constant extension of the field of the temporal data employment; this fact is connected with designing the new information systems and their functionality extension. Nevertheless, this tendency is not provided with the new technologies in the temporal data processing and storage. This fact decreases the efficiency of the information systems functioning in general and assigns a lot of issues to the developers and users. Among these issues there are the following ones:

- the complexity of interaction with the temporal data, presented in the necessity of designing the bulky SQL-constructions, and the high probability of the error occurrence connected with this;
- the temporal non-extensibility of the databases of the active information systems: supplementing the time dimension to the data threatens the operability and efficiency of the active information systems;
- uneconomical employment of the disk space and various restrictions in the process of the temporal data storage;
- the absence of the facilities, supporting the information security of the temporal data.

2. The above mentioned issues require the complex investigation in the area of modeling and implementing the temporal databases in the information systems on the railway transport, directed on:

- the choice of the implementation platform of the temporal databases;
- the provision of the upward temporal compatibility;
- the choice of the representation models of the temporal data and the methods of interaction with them; they are to take into consideration the peculiarities of the transport tasks, such as the support of the periodicity, the specific calendar and the peculiar mode of the data introduction and the versions alternation;
- the choice of the methods of supporting the integrity, the safety and the accessibility of the temporal data;
- the simplification of the interactivity with the temporal data in the process of the data preparation for the analytical processing.

3. As the object of application of the research results the State Joint Stock Company "Latvijas Dzelzceļš" has been chosen. This company is a typical representative of the railway enterprises and has a usual collection of IS, functionally divided into four applied areas: the freight transportations, the passenger transportations, economics and finances, and infrastructure management. The analysis of the information systems of the Latvian Railway has promoted the diagram of the interconnection of the information systems and the temporal objects. This fact has allowed choosing the definite information systems for the further investigation. The following ones are among them:

- the information system of the passenger trains scheduling *SAR*;
- the financial-statistical information system of the passenger transportation *APFIS;*
- the information system of the Referenced Data System *KLASIFIK*.

Chapter 2. INVESTIGATION OF THE ISSUES OF THE TRANSPORT TEMPORAL INFORMATION SYSTEMS DESIGNING

2.1. Principle Fundamentals of the Temporal Database Technology

2.1.1. Evolution of Studies in the Area of the Temporal Databases

Despite a set of unsolved problems, investigations in the area of the temporal database (TDB) started more than 30 years ago. This is reflected in details in the research papers [51, 67, 71]. The most significant research results are examined in works [42, 52, 74], there are also dozens of publications in journals and conferences proceedings, which have a practical interest for modern temporal systems developers (for instance [57, 58, 62]). Analysis of famous works in the area of temporal database let the author mark out four stages in the field of research (see table 13):

1. Preliminary stage came into existence in the period of active development of relational database theory and it considers the idea of using the temporary component of the data.

2. The stage of scientific field development in the area of the Temporal Databases is characterized by the development of the theoretical foundations of the temporal databases, as well as the development of the first temporal DBMS.

3. The stage of standardization. Practical application of the temporal database theory in the developed information systems was held back by the absence of the standards and the appropriate specialists and it led to establishing the working group for developing the standards and training personnel to the fundamentals of the temporal databases.

4. The stage of the temporal database technology development. The results of the preceding stages are summarized and developed in the fundamental works, which form the temporal database technology. The present stage has not been accomplished so far and every new investigation provides new opportunities for the temporal database users. This stage is characterized by the emergence of works on practical applications in the definite areas where the present research also belongs.

Table 3. Research evolution in the area of the temporal databases

Stage	Years	The main achievements	The leading researchers and their works
1. Preliminary stage	70-ies of XX c.	The first research papers touching upon some of TDB conceptions	G.Wiederhold, J.F. Fries, S. Weyl [70], B.Schueler, S.Jones
2. The stage of the scientific field development in the area of TDB	1982	The first focused work in the field of the temporal data management	Jacob Ben-Zvi [55]
	80-ies of XX c.	Works on temporal logics, the usage of data dependent on time, its presentation inside the system and visualization for users	James F. Allen [69]
		Data modeling issues research, external memory data management, query languages, etc.	James Clifford, Gad Ariav, Richard T. Snodgrass, Ilsoo Ahn, Jan Chomicki
	1986	Postgres – the first concrete implementation proposal for Rollback relations	M. Stonebraker [47], [48]
	1987	Temporal aspects in information systems – the first conference devoted to the temporal databases	Gad Ariav, James Clifford, Arie Segev, Nikos A. Lorentzos, Shamkant B. Navathe, Rafi Ahmed

Table 3. Research evolution in the area of the temporal databases (continuation)

Stage	Years	The main achievements	The leading researchers and their works
3. Standardization stage	1990	The temporal databases appeared in two undergraduate database textbooks	Chris Date, Ramez Elmasri, S. B. Navathe [41]
	1992–1993	The emergence of the idea of a possible temporal extensions of the standard query language for the relational databases SQL-92	Richard T. Snodgrass [42], [65], Ilsoo Ahn, Gad Ariav, Don Batory, James Clifford [49], Curtis E. Dyreson, Christian S. Jensen [52], Ramez Elmasri [72], Fabio Grandi, Wolfgang Kaefer, Nick Kline, Krishna Kulkarni, Ting Y. Cliff Leung, Nikos Lorentzos, John F. Roddick, Arie Segev [58], Michael D. Soo [61], Surynarayana M. Sripada, Abdullah Uz Tansel, Shashi Gadia [49], S. B. Navathe, R.Ahmed [44], Andreas Steiner [56], Michael H. Böhlen [57], Claudio Bettini, Alexander Tuzhilin, David Toman, Jan Chomicki
		"ARPA/NSF International Workshop on an Infrastructure for Temporal Databases" showed scientific community interest in the creation and usage of the temporal extension of databases query language standard [58]	
		The establishment of the committee on the creation of the temporal query language specification. The leading scientists in the sphere of databases joined the committee	
	1994	The issue of query language specification TSQL2 and comments on it [59]. Parts of TSQL2 were included in a new substandard of SQL3, ISO/IEC 9075-7, called SQL/Temporal. However, the ISO project responsible for temporal support was canceled near the end of 2001	
4. The stage of temporal database technology development	2000 up to now	Fundamental works and the further development of the temporal database technology. "*Developing Time-Oriented Database Applications in SQL*"– the first book dedicated entirely to temporal databases has appeared (2000) [42]	Richard T. Snodgrass, Christian S. Jensen, Chris Date, Hugh Darwen, Manolis Koubarakis, Sushil Jajodia, Arie Shoshani, Kristian Torp, Claudio Bettini, Michael H. Böhlen, Jan Chomicki, Carlo Combi, Curtis E. Dyreson, Per F. V. Hasle, Johann Gamper, Dengfeng Gao, Like Gao, Fabio Grandi, Sushil Jajodia, James B. D. Joshi, Vijay Khatri, David Lomet, Nikos A. Lorentzos, Nikos Mamoulis, Angelo Montanari, Mirella M. Moro, Peter Øhrstrøm, Peter Revesz, John F. Roddick, Arie Shoshani, V. S. Subrahmanian, Abdullah Uz Tansel, Paolo Terenziani [62],[63], David Toman, Kristian Torp, Vassilis J. Tsotras, X. Sean Wang, Jef Wijsen, Yue Zhang

The results of the research analysis in the field of the temporal database have allowed the author to outline the range of both solved problems and problems requiring further study, as shown later in the thesis.

2.1.2. Notion of the Temporal System and the Temporal Database

Under the term ***temporal information system***, we will understand *information system, which operates a set of real world objects changing in time*. Changing the state of the object in such systems is characterized by the appearance of its new version. All ever-created versions of the object remain in the system during the time of its existence (conditionally) and are used later for different purposes. The foundation for the temporal information system is the temporal databases.

The main principles of the temporal databases are the storage of all states of the object since its creation until destruction and providing an access to any of the states at any time. In other words, the main thesis of the temporal systems is the following: for every data object, created at time t_1 and destroyed at time t_2, all its states in time interval $[t_1, t_2)$ are kept and available to users in the database.

The temporal databases do not present a radical new concept. Basically this concept reflects a change from the current state of data into historical and store them in a form accessible to the user. What is radical in it is the way this information is managed.

At present, there is no an unambiguous conventional definition of the temporal databases. Until now, in the scientific works authors prefer to give theoretical foundations showing what is understood by the temporal database in the given research. In their definitions, the main emphasis is on structural or functional component.

In the course of the given research, several different definitions of the **temporal databases** were found. The first definition belongs to Richard Snodgrass, which he offers as official: "*a database that supports some aspect of time, not counting user-defined time*" [43]. "User-defined" time is understood as such an aspect of time, which is not recognized by the DBMS as a particular type of data. " *'User-defined' time is defined as 'an uninterpreted attribute domain of date and time... parallel to domains such as 'money' and integer'* ". An example of 'user-defined' time is data such as birthdates. A classical database essentially treats data such as birthdates as text strings. This treatment of date data does not allow for much manipulation of the data when performing ad hoc queries on the database.

Another more formalized and structural definition is made by Shamkant Navathe and Rafi Ahmed. According to them, a temporal database is defined as "*a union of two sets of relations Rs and R1, where Rs is the set of all static relations and R1 is the set of all time-varying relations*" [44]. There are also less strict definitions, such as: "*a temporal database is a database that records time-varying information*" [54]. "*Temporal databases are the databases that store temporal data. However, these databases and the data they contain can be regarded as temporal only if the rules of interpretation of timestamps and intervals for a specific database management system are known"* [67].

A universal and unambiguous definition of the temporal database cannot be given, as besides various kinds of temporal data, there are several forms of implementation of temporal databases: Historical, Rollback and Bitemporal (see point 2.1.4). In this research a ***temporal database***, in general, is understood *as the database, which in some way supports the time dimension of data*.

At the initial stage in order to develop the temporal database for the transport information system, it is necessary to solve the following issues:

- designing a representational model of the temporal objects data in the database;

- selection of the necessary time dimensions and forms of the temporal relations;
- identification of the modification operations implementation logics and data select (active version of the object select).

In order to solve the above mentioned issues, we will consider a number of conceptual notions of the temporal databases technology, many of which are formulated taking into account the specifics of their application in the IS in transport.

2.1.3. Time Dimensions

The main notion used in the temporal databases is the notion of the *timeline* or so-called *time dimension*. There are two basic dimensions: valid and transaction. We will consider the purpose of these dimensions in the temporal systems in more detail.

In case the data, presented in the database, are considered as a certain reflection of the current state of reality for the world under modeling, each record can be interpreted as a certain fact, real at the definite moment or in the definite period of time. In the process of transition to the temporal database for each of these facts it is possible to specify the exact time period when it is valid. Such representation of the time when the data are connected by the time span of their relevance in the simulated world is called *valid time* [67]. Note that the values of this type of time can be moments referring to both the past and the future (planning).

There is an illustration of the arrangement of the facts on the axis of real time. Fig. 2.1 is an example of the sphere of railway passenger transportation: the dynamics of fare travel change along the axis of valid time.

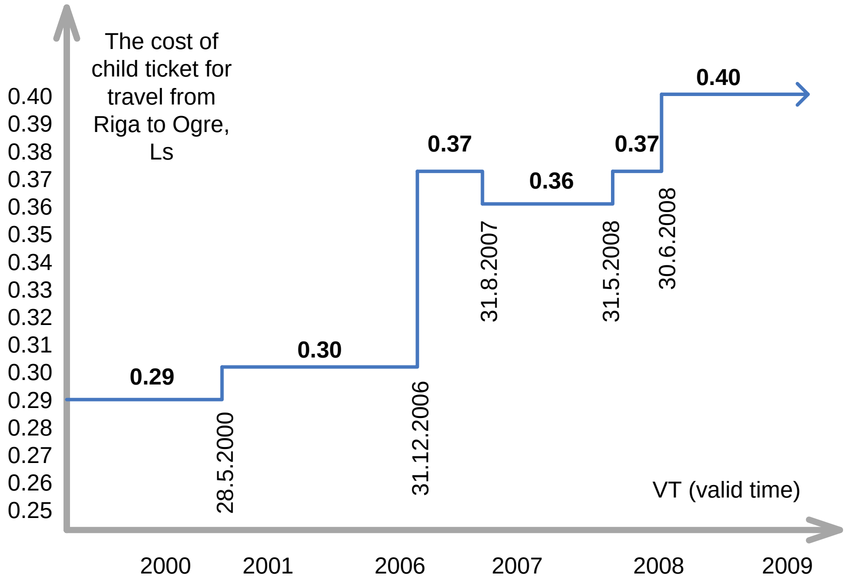


Fig. 2.1. The graph of changing the electric train fare for children from Riga to Ogre from 1995 to 2008

The intervals on the VT axis indicate when a certain rate functioned in the Latvian Railway enterprise. For example, the cost of child ticket for travel from Riga to Ogre in the period from 01.09.2007 to 31.05.2008 was 0.36 Ls.

Another line of time, which is regarded by researchers of the temporal databases, is a *transaction time*. Each table entry can be compared with that period of time when it was represented in the database, i.e. period of time between adding a record and deleting it from the database. Whereas, it should be mentioned that the data modification operations implies the preservation of the previous version of the record, and simultaneously places it in the “shadow zone”, i.e. performs its removal at the logical level. Obviously, the values of transaction time cannot be related to the future. DBMS transaction time is mainly used for the work with the locking mechanism and log-files for the system recovery. In some systems administrators can use special SQL language expansions, which allow access to the transaction time and change history records in the database (see point 2.2.2).

Thus, the notion of valid time is applicable to the facts of the simulated time, and transaction time – to the records in the database. In this paper, we define a valid and transaction times as follows:

Valid Time (VT) is real-world time of a fact when this fact is true in the modeled reality (TDB Glossary, Christian S. Jensen [53]). In other words, VT is the period of the object existence, which is characterized by the time of its beginning and its end. The values of VT, as a rule, are supported at the user level.

Transaction Time (TT) is the period of the records in the database, which is usually characterized by the time adding records to the database and the time of its logical deletion. The values of transaction time are recorded automatically. The beginning of the period TT is the time when the record gets to database and the end is the time of its deletion from the system.

TT timestamps provide information about the time change of data or error correction, and VT timestamps store information about the change of some parameters of the modeled reality. Thus, VT and TT are orthogonal to each other. VT reflects our understanding of the history, which may change, in contrast, the TT reflects the history as it is, but the history remains unchanged. TT is managed by the system, but VT – by the user. So the values of VT can be changed but TT – no.

Researchers of the temporal databases generally use one of the above-mentioned dimensions of time or both simultaneously. Some research offers other dimensions, which storage of values might be of interest for the user, but they can all be reduced to one of the above. The object with such additional time dimensions is taken as an example. And this object is a "warning" in the railway information system of issuing warnings (BIS-K). Warning has several time dimensions:

- date and time of warning registration in the system (transaction time);
- date and time of warning validation (valid time);
- the period of validity warning, characterized by the start time and end time (valid time);
- the period of the train driver notification, which depends on the departure time of a certain train. The list of urgent warnings can be given to the train driver not earlier than 15 minutes before departure time (valid time).

Another notion, which will be used further in the temporal database models studies, is the object’s **lifespan**. It defines the period of time associated with the existence of an object in a particular state. This period of time is formed by either transaction time or valid time depending on the place of observation of the object in time: in databases or in a real world, in other words, on chosen temporal form (see 2.1.4).

Lifespans may be unbounded. In some temporal systems for undefined range limits value NULL, the maximal value of data type (for example, 31.12.9999) or any other special symbols are used. The chosen way of limiting the time interval has a significant impact on the query algebra to temporal data. In the given research paper the value NULL will be used for undefined range limits.

In determining the temporal model for objects with temporal characteristics, it is necessary to define the **granularity**, which shows to what extent close moments on the time axis, will still be distinguishable from each other [37]. Granularity can be a day, a month, a second etc. The minimal segment is called **chronon**. Granularity influences the type of the lifespan boundary data, which can be presented only by a date or date and time.

Granularity of the temporal relations can be different. As a rule, in most issues of transport information systems, granularity of TT dimensions is much more accurate than VT dimensions, as it is shown in Fig. 2.2. Usually chronon of TT dimension is the smallest unit of time, which DBMS is able to fix (for example, millisecond), since transaction time, as its name implies, reflects the time of the transaction.

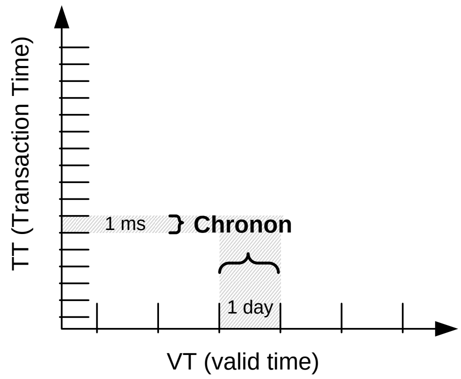


Fig. 2.2. Example of different granularity of the time dimensions

However, the transport specificity allows to use TT a little differently. In some Information Systems in transport it is assumed that the modification of one and the same record within some short interval (for example, a day) is the error correction. Such short-term versions of the record are of no value either from a functional position, or in terms of safety (audit). The peculiarities of such features implementations are set out in point 3.6.4 of the dissertation, as well as in works [4, 8].

In most operational transportation tasks, chronon VT is equal to one day. In analytic issues, this parameter is longer, for example: a week, a month, a quarter or year.

2.1.4. Forms of the Relations in the Temporal Databases

Fig. 2.3 represents graphically the temporal model of a certain object O_i in three-dimensional Bi-temporal form, where:

- the first axis *O* represents objects of IS (freight, wagons, stations);
- the second axis *TT* shows the history of data changes in the database;
- the third axis *VT* reflects the history of objects changes relative to VT;

- each point in such a coordinate system (O, TT, VT) describes the state of the object O_i in the moment of the valid time t_m^{VT} in the form it was described in the database in the moment of the transaction time t_n^{TT}.

There are three planes in the bitemporal models:

- plane **α** is the state of the data in the databases in the moment of the transaction time t_n^{TT};
- plane **β** is the reality modeled in the database, active in the moment of the valid time t_m^{VT} and including in itself the history of its modeling (changes of the records);
- plane **γ** is a range of object O_i states that represent the history of their changes in the database and in reality.

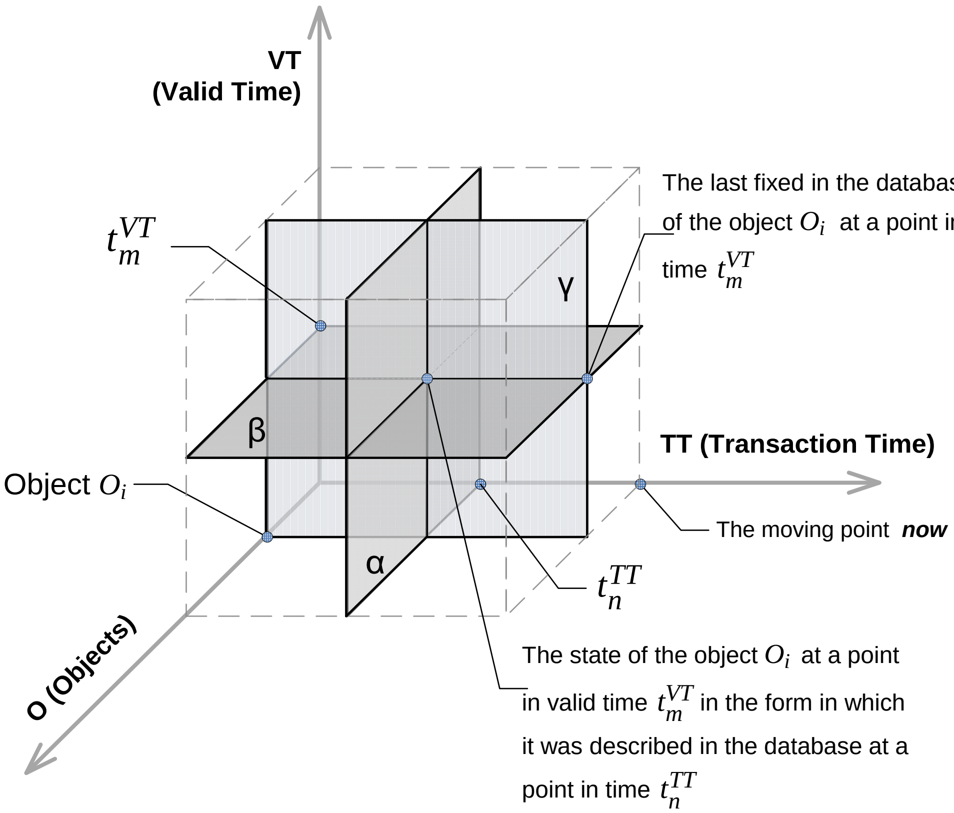


Fig. 2.3. Dimensions of the temporal database

There is a transition from the graphical presentation of temporal objects to the ways of its presentation in the database.

The main issue in temporal database design is the choice of temporal forms for relations. The form of temporal relation defines its temporal functionality, influences its performance while working with it, influences the data access complexity and the volume of disk space. Basically, temporal forms are divided according to the degree of the main time dimensions inclusion: valid time and transaction time. Thus, there is a possibility of four **forms of relations**: *Snapshot, Historical, Rollback* and *Bitemporal* [51, 52, 53, 56] (see Fig. 2.4).

The relations of Bitemporal, Historical and Rollback forms are the most widespread and described in literature (for example, [37, 51, 52]). We will further call them **classical forms of**

relations. On the basis of these forms, the majority of temporal objects on transport can be described. Railways classifiers and codifiers, guidebooks can be taken as an example.

As it has been mentioned before, all temporal forms have their own functionality and purpose. The brief description of each of them is presented, and their capabilities are shown.

The form **Snapshot** is a usual relation, does not support any time dimensions. It is called by analogy with database Snapshot. These are traditional databases, which store the snapshot of objects of subject area model, which is, as a rule, model's last known state. Thus, any change of the object in the database makes the previous state of the object inaccessible. This form of relations is also used in the temporal system database. Objects and static information where only current values are of interest can be kept in these relations.

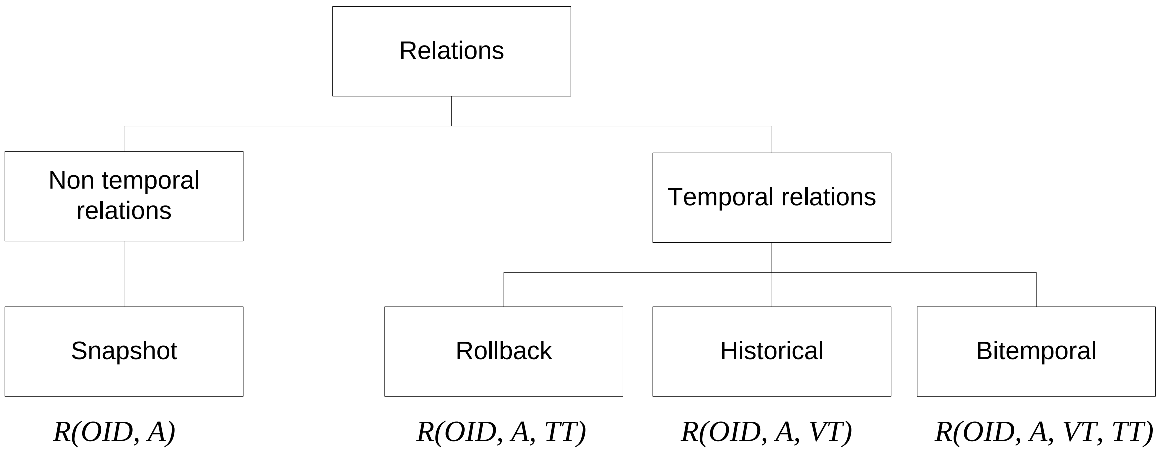


Fig. 2.4. Classical forms of the relations in the temporal databases

The **Rollback** form sustains only transaction dimension. Version nature of the records is maintained automatically. Due to such a character of the time dimension support the form may contain data related only to the past or present. The main purpose:

- data recovery at the level of relations at a specified point in time without having to resort to a backup database that allows you to recover data without database administrator;
- action audit with records;
- the possibility of obtaining knowledge about what records and at what time were presented in database;
- the possibility to get deleted ("old") data.

The **Historical** form sustains only valid time. It allows to correlate certain facts (object state) to specific time periods of their relevance. Thus, it is possible to store the data of the last, present and future.

The **Bitemporal** form sustains two time dimensions: valid time and transaction time, thus combining the properties of the forms Rollback and Historical: the same as in Historical, it stores the state of the object at all periods of time, and as in Rollback relations all changes of records are registered. Apart from the degree of time dimensions inclusion, temporal relations also characterize the **restrictions imposed on lifespan**:

1. For Historical and Rollback forms, the example of restriction is a condition of existence for only one tuple, which characterizes the state of the object at any given time. It

means that the object lifespans cannot intersect. In other words, the validity periods of the various state of the object is strictly ordered in time and never overlaps;

2. The relation of Bitemporal form allows to keep both real and erroneously input data, so each tuple in Bitemporal relation presents a record of the object version as well as the version itself. Thus

   a. the same as in Rollback relation, the periods of records relevance which describe the object version changes in the database never intersect on the transaction time TT axis;

   b. the periods of the object version along the valid time VT axis can overlap due to the fact that the relation contains array of records of its changes. However, at any time t_n^{TT} of database existence only one record of the object version can be considered as active, which becomes possible by the restriction on the intersection of vital segments of records in transaction time TT (see point ***a***).

The listed restrictions of the temporal integrity are defined as follows:

- for the relation of the form Historical of the scheme $R(OID, A, VT)$ the following restriction is active:

$$\forall r, p \in R(r.OID = p.OID) \wedge (r \neq p) \Rightarrow (r.VT \cap p.VT = \varnothing); \tag{1}$$

- for the relation of the Rollback form of the scheme $R(OID, A, TT)$ the following restriction is active:

$$\forall r, p \in R(r.OID = p.OID) \wedge (r \neq p) \Rightarrow (r.TT \cap p.TT = \varnothing); \tag{2}$$

- for the relation of the Bitemporal form of the scheme $R(OID, A, VT, TT)$ there are two restrictions:

$$\forall r, p \in R(r.OID = p.OID) \wedge (r \neq p) \Rightarrow (r.TT \cap p.TT = \varnothing) \tag{3}$$

  and

$$\forall r, p \in R(r.OID = p.OID) \wedge (r \neq p) \wedge (t_n^{TT} \in r.TT) \wedge (t_n^{TT} \in p.TT) \Rightarrow (r.VT \cap p.VT = \varnothing)); \tag{4}$$

  if we consider the integrity of Bitemporal relations only for active tuples (for which the time of logical deletion from database is not defined, i.e. $t_e^{TT} is\, null$), then restrictions of the integrity (3) and (4) turn into the following rule:

$$\forall r, p \in R(r.OID = p.OID) \wedge (r \neq p) \wedge (r.t_e^{TT} is\, null) \wedge (p.t_e^{TT} is\, null) \Rightarrow (r.VT \cap p.VT = \varnothing). \tag{5}$$

In (1)-(5) *R* is relation of the given form; *r* and *p* are tuples of relation *R*; *A* are attributes of *R*; *OID* is the unique identifier of the object; *VT* and *TT* are attributes of valid time and transaction time accordingly, defined as $VT = [t_s^{VT}, t_e^{VT}]$ and $TT = [t_s^{TT}, t_e^{TT}]$; operation of the time intervals intersection $r.VT \cap p.VT$ (or $r.TT \cap p.TT$) can return either a new interval (if the area of intervals intersection exists) or vacuous interval $\varnothing$.

The above given restrictions define the logic of supporting temporal relations integrity; affect the mechanism of data operation implementation, in the first place, operations of change and addition records and also active version choice operation.

In order to demonstrate how data in different temporal forms can be stored and how valid time and transaction time correlate and also how with the temporal functionality expansion the stored set of data becomes more complex, consider the example of employees' salary data organization in the database. The employees' salary original table, which does not have any time dimensions, is shown in Fig. 2.5.

Employee	**Salary**
Alksnis	1100
Osis	1400

Fig. 2.5. Data in the Snapshot form: The current employees' salary

With the help of valid time dimension, it is possible to define an employee's salary at different points in time and whether there has been any salary at all. Thus, the salary data is the sequence of time varying values (see Fig. 2.6).

Employee	**Salary**	**Valid Time**	
		t_s^{VT}	t_e^{VT}
Osis	1400	01.08.2005	
Alksnis	1000	01.01.2006	31.03.2006
Alksnis	1100	01.04.2006	31.10.2006

Fig. 2.6. Data in the Historical form: Employees' salary and periods of its relevance

With the help of transaction time dimension, it is possible to find out when the changes are introduced into the table. However, this form does not give information when an employee received a salary (see Fig. 2.7).

Employee	**Salary**	**Transaction Time**	
		t_s^{TT}	t_e^{TT}
Osis	1400	11.05.2005 11:50:00	
Alksnis	1000	03.01.2006 13:10:00	02.03.2006 12:30:00
Alksnis	1100	02.03.2006 12:30:01	

Fig. 2.7. Data in the Rollback form: Employees' current salary
and the history of the records changes

There is a consideration of the case when both valid and transaction time are sustained for the table (see Fig. 2.8). Now, if incorrectly entered data are subsequently corrected, it will be possible to say exactly when it was done. There might be a situation when information about such changes is needed, as incorrect data might have been already used in some calculations. When you update the values in the table (even in the case of error correction in data), the interval of transaction time is also updated, so we can see a list of changes in the database.

Record number	Employee	Salary	Valid Time		Transaction Time	
			t_s^{VT}	t_e^{VT}	t_s^{TT}	t_e^{TT}
1	Osis	1400	01.08.2005		11.05.2005 11:50:00	
2	Alksnis	1000	01.01.2006	31.03.2006	03.01.2006 13:10:00	
3	Alksnis	1100	01.04.2006	31.10.2006	02.03.2006 12:30:01	
4	Alksnis	1100	01.11.2006	31.12.2007	02.09.2006 16:45:00	02.11.2006 10:00:00
5	Alksnis	1300	01.01.2007	31.12.2007	01.11.2006 10:00:01	02.01.2007 15:30:00
6	Alksnis	1100	01.11.2006	31.12.2006	02.11.2006 10:00:01	
7	Alksnis	1400	01.01.2007	30.04.2007	02.01.2007 15:30:01	
8	Alksnis	1550	01.05.2007	31.07.2007	02.01.2007 15:30:01	
9	Alksnis	1700	01.08.2007	31.12.2007	02.01.2007 15:30:01	

Fig. 2.8. Data in the Bitemporal form: Employees' salary in different period of time and the records change history

In the Bitemporal table in Fig. 2.8 the first three records correspond to the contents forming scenario of the tables Historical (Fig. 2.6) and Rollback (Fig. 2.7). And the subsequent records 4-8 appeared as a result of operations which took place after 02.09.2006. Indeterminate value of the transaction time final period deserves our attention. In the table Rollback with the support of transaction time for the second record (Alksnis, 1000Ls, 01.01.2006 – 01.03.2006) the termination of the term was determined 01.03.2006. But in the Bitemporal table salary records during the period from 01.01.2006 to 01.03.2006 continue to be valid. The closing period of transaction time in the bitemporal table means the logical deletion of the record and per se it marks the record as erroneous and its data as never existing in reality. The fact is that in respect of supporting only transaction time (Rollback) such a concept, as the version of the object does not exist. There is a concept the version of the record, the data are written "over", that is, the previous values logically cease to exist and the date of completion is set automatically for them.

The table content is explained with the employment of the following procedure. The data of the Bitemporal table (Fig. 2.8) were formed as a result of six operations, executed by the personnel manager:

1. On 11.05.2005, the information that the employee Osis salary will be Ls1400 starting from August 1, 2005 was put into the data (**record 1**).
2. 03.01.2006 they put into the data the information that on January 1, 2006 the employee Alksnis was hired for a trial period of 3 months with a salary Ls 1000 (**record 2**). Since the termination of the cooperation is known beforehand, the validity period of these conditions, March 31, 2006, is closed immediately.
3. Two months later (02.03.2006) the company manager agreed on the employee Alksnis' suitability for the position. In this connection, another contract with a salary of Ls 1100 came into force since April until December (it is effective for 6 months). Thus, a new version is formed (**record 3**). The date of the contract termination is also known, this is recorded in the database (October 30, 2006).
4. The employee Alksnis is hired into the permanent staff with the same salary (Ls 1100) till the end of fiscal period (two years). Thus, only the term of cooperation was extended (**record 4**).

5. The employee Alksnis showed a good performance and his salary is going to be increased by Ls 300 (Ls 1300) next year, which is put in the database beforehand on 01.11.2006 (**record 5**). However, the time range of this decision and the corresponding range in the **record 4** partially overlap. So at the time of new data input (01.11.2006), **record 4** is marked as invalid and the date of termination of its existence is set. **Record 6** presents the valid remainder of this period (November 1, 2006 – December 31, 2006).
6. On 02.01.2007 the increase in the employee Alksnis' salary for 2007 is being planned. It is going to be in three stages. The new salaries and the stages of their introduction are represented by **records 7, 8** and **9.** The operation time of these salaries overlaps completely **record 5**, as a result of which the data of logical deletion is set. (01.01.2007). Thus Alksnis will never receive the salary of Ls1300. However, the data that such salary has been once planned will be kept in the database (**record 5**).

In Bitemporal relation, the change of any value can be presented in the form of a surface. An example of such a surface was shown in Fig. 2.9. The axes are time dimensions and the observed parameter is the employee's Alksnis salary. Every point on the surface is the employee's salary at the moment of VT as it was shown in the database in the moment of TT.

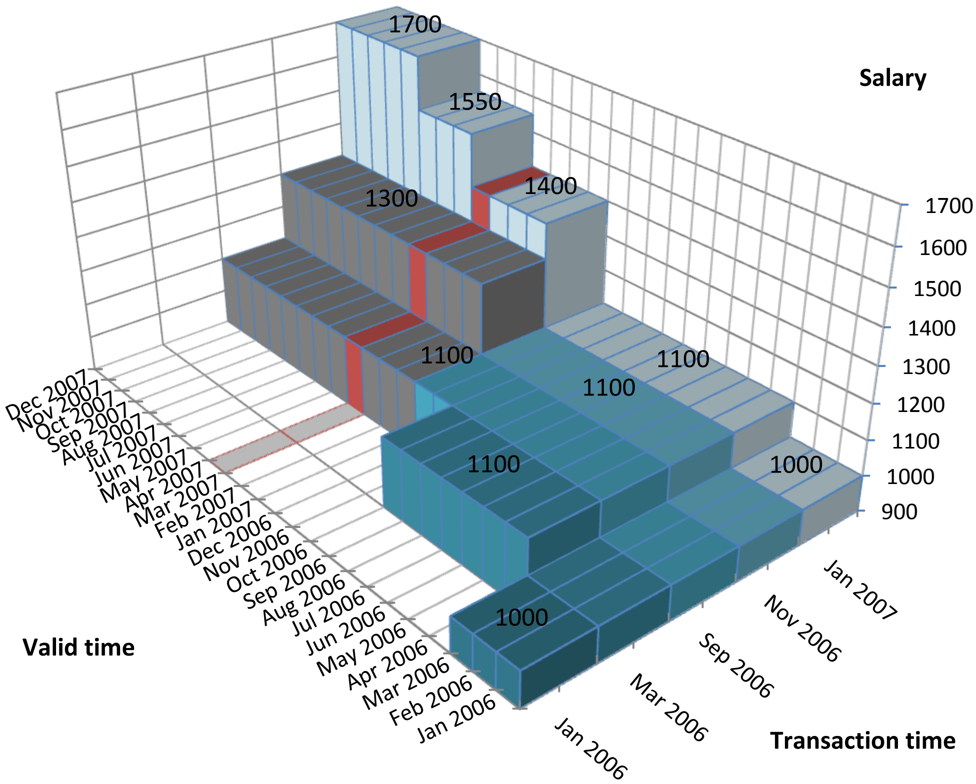


Fig. 2.9. Chart of the employee's salary change along all the Valid and the Transaction time

Let us consider how to orientate on the surface shown in the chart, which reflects the salary at different period of time.

Making a slice along the axis of valid time dimension in the moment of time April 2007, it is possible to determine what salary the employee Alksnis had in April 2007 (it is pointed in the chart with the red line). As a result of this slice, we get several values, among which there are blank values and three marked (1100, 1300, and 1400). This result includes the previous values of April salary, which has been changed three times. If we are not interested in the previous (erroneous) values of the salary, then we also have to make a slice on the axis of the transaction time by the time of the query. On the chart, this moment corresponds to the last series of the columns along the transaction time January 2007, as the data has not been changed since then. Thus, we receive the result: in April 2007 the employee Alksnis had a salary of Ls 1400.

The query might be also different: "Which data were recorded in the database on 20 November 2006 regarding the employee Alksnis' salary on April 2007?". On this occasion, the slice on transaction time is done for the period of time November 20, 2006. We get another result: a record from the "shadow zone" – Ls 1300.

2.1.5. Integrity of the Temporal Relations

The observation of the Time Normal Form or TNF is an important aspect for maintaining the integrity of TDB. TNF is defined in the following way: "the relation is in TNF if and only if it is in the third normal form of Boyce-Codd and there are no time dependencies between non-key attributes" [41].

The temporal dependence in relation $R(O, A_1, A_2, \ldots, A_m, t_s, t_e)$ between two non-keys attributes A_i and A_j exists if for the observed object *O*, there will be at least two tuples r_1 and r_2, where the following conditions were fulfilled:

1. $r_1.O = r_2.O$;
2. $(r_1.t_e = r_2.t_s - 1) \vee (r_2.t_e = r_1.t_s - 1)$;
3. $(r_1.A_i = r_2.A_i) \vee (r_1.A_j = r_2.A_j)$.

To illustrate the discrepancy to the definition of the Time Normal Form, we show the temporal relation of the Stations Classifier, where *O* is a constant object identifier. It is showm in Fig. 2.10.

O	***Code***	***Name***	t_s	t_e
1245	2009912	Oshkalne	1.12.1977	1.10.1999
1245	2009912	Zemitani	2.10.1999	*Now*

Fig. 2.10. Temporal relation of the Stations Classifier

In this relation there is a temporal dependency between attributes *Code* and *Name.* Due to this temporal dependency, during the creation of a new version caused by attribute *Name* change, one and the same attribute *Code* value will be present both in old and new versions. To illuminate this undesirable dependency, it is necessary to carry out the decomposition of the source relation splitting it into two relations: *R1* (*O, Code,* t_s, t_e) and *R2*(*O, Name,* t_s, t_e).

A list of research is devoted to the description and solving of the problems of *redundancy*, *wordiness* and *discrepancy* [51, 52]. However, the results presented there, as a rule, are applicable to the relations of the classical temporal forms (the relations with non-overlapping lifespans), while in practice in transport area it is not always possible to be restricted to the relations of this kind. There are tasks where it is necessary to operate with the lifespans of the objects, which overlap [8, 15], shown in point 2.4.3. Questions of integrity in relations with overlapping lifespans are studied in point 3.1.2.

2.1.6. Operations with the Temporal Data

There are four types of operations with the records in the database: SELECT, INSERT, UPDATE and DELETE. In the temporal databases, the performing of each of these operations has some peculiarities depending on the form of temporal relations. Consider the most significant of them:

1. *Insert* – an ordinary record insert operation is complemented by forced checks on compliance with data integrity. Basically, the presence of previous versions of the record or object (depending on temporal forms of relations) is checked. If it is detected, then a series of actions on lifespan change of the previous versions is performed. In Rollback relation the moment of insertion is fixed in the attribute of transaction time for a new record;

2. *Update* – an ordinary operation of record update in Rollback relation is complemented by the conservation of the previous version and fixation of the values in the attributes of transaction time for the old and new versions. In Historical relation, the operation is supplemented be the forced checks on compliance with temporal integrity of the data. If we change the values of valid time attribute, the presence of other versions of the objects is checked. The operation time of these versions intersects in the valid time with a new version. If such intersection is detected, additional actions are performed to eliminate overlapping of lifespans.

3. *Deletion* – in Rollback relation deletion is substituted by operation of closing lifespan through a transaction time – a period of the record presentation in the database. Thus, the data are not deleted but move to the “shadow zone”. In Historical relation the version of the object is deleted, as a rule, without additional actions.

4. *Select* – is the most difficult and powerful operation. It is connected mainly with the fact that for each requested point in time it is necessary to calculate the current version of the data. The queries for temporal data select are connected with the usage of array of temporal operations: temporal slices, aggregations, the functions of lifespan comparison, the function of valid version of the object definition at the given point in time.

The operations of stream and historical data modification (see point 1.1) do not possess any peculiarities as for their storage ordinary relations are used and they do not have temporal functionality. All of the above mentioned expansions of the modification operations are working for the multi-version data, which accordingly are located in the temporal relations.

The research and the realization of the mechanism of select, insert, update and deletion in the temporal relations are presented in points 2.4.5 and 3.2.

2.2. Choice of the Implementation Platform for the Temporal Information System

2.2.1. Problems of the Implementation Platform Choice

Despite advantages and topicality of the temporal technology, the existence of separate scientific direction of the research, and array of elaborations in the field of the temporal databases, the availability of a great amount of research papers and a great number of completed research and great reserve in the field of temporal variants of the language SQL [59], there are still problems of industrial implementation of temporal databases. To a certain extent, these problems are connected with the choice of platform for temporal databases. The choice of platform depends not only on requirements of efficiency and functionality of temporal mechanisms of database, but a number of other factors:

- high reliability and performance;
- integrity control;
- adherence to standards (ANSI SQL-92 and SQL-99);
- tolerance of the developed temporal information systems;
- support of the manufacturer;
- integration with other enterprise information systems;
- compatibility with application development tools.

It should be noted that, despite widespread recognition of the usefulness of temporal databases, in the leading relational DBMS, until recently there was no support for temporal features [40]. The available implementation of the temporal technology is considered below, as well as the possibility of its employment for the industrial use in transport.

2.2.2. Critical Review of the Software Products that Implementing the Technology of the Temporal Databases

Nowadays, there is no full-featured commercial implementation of DBMS, but many manufacturers offer a variety of extensions and frameworks that allow working with the temporal data. Below there are brief characteristics of these developments.

According to Time Center [66] and other sources [57, 67], generalizing the well-known developments in the field of temporal databases, nowadays the following implementations are known:

- Temporal framework *TIMEDB* offers a temporal query language interface to valid-time, transaction-time, and bitemporal tables. *TIMEDB* translates temporal statements into SQL statements, which are then fed to an underlying standard DBMS. The temporal query language is temporally upward compatible with SQL [66]. *TIMEDB* prototype was established in 1997 by Andreas Steiner during the preparation of his dissertation. In the prototype both valid and transaction time are supported, but it was tied to a particular relational DBMS, since it used a call-level interface of Oracle. Later *TIMEDB* 2.0 version was released implemented in Java language. An interface JDBC was used to work with DBMS, so there was no bound to a particular manufacturer. *TIMEDB* 2.2 version can be freely downloaded from the company Time Consult for research purposes.

- *Teradata Database 13.10* provides a temporal version of SQL, including enhancements to the data definition language (DDL), constraint specifications and their enforcements, data types, data manipulation language (DML), and query language for temporal tables. Teradata supports two built-in time dimensions that can be used to create temporal tables, transaction time and valid time [98]. The Teradata Database 13.10 is the first time-aware database. It is the first to deliver intelligent temporal analytics making it easier for customers to create a historical picture of enterprise business transactions and activities [99].

- *Oracle Flashback* allows to examine the state of database at any time in the past without making any structural changes in a database. Flashback Query operation can be performed without administrator, and it allows developers to add the functions of data recovery to their applications. *Flashback Version Query* allows you to view all versions of rows of a given table in a given interval of time, to find transactions that changed the given rows; *Flashback Transaction Query* provides an opportunity to see the changes specified by a given transaction. And means of Flashback Database, Flashback Query and Flashback Drop recovery allow to return to a previous state in constant time, which does not depend on databases volume. In Oracle 11g a tool *Flashback Data Archive*, which allows to track and store changes of all the data automatically, appeared. These data can be stored for an indefinite period of time and can be requested by Flashback SQL.

Oracle positions its tools of category Flashback, primarily, as a means to restore the databases after user errors. However, it is clear that they can be used to provide temporal properties of databases.

- *Oracle Workspace Manager* provides multi-version of the data and support states snapshots. It allows to manage the current, prospective and historical data values in the same database. It uses workspaces as a virtual environment to isolate the set of data changes, keep history of data changes and create multiple scenarios to analyze possible future. Created workspaces can be inherited from other workspaces or the standard "current" workspaces LIVE (by default). Here by the current state of the parent's workspace is fixed. In a specific workspace we can also use save points that allow you to roll back changes to a specific point in the past. It also provides operations to merge a child's workspace with a parent's workspace. An option of complete tracking of all data changes, including the operations of insert, deletions and update is of a special interest. In fact, this is the support of transaction time. From the perspective of working with temporal data, the possibility to fix requests for the specified time is worth noting.

From a technical point of view, workspace is a set of representations of the database with handler triggers INSTEAD OF. When the user wants to add version control support for the table, workspaces manager renames the table into <name of table>_LT, adding four work-related columns to it, then creates a view with the name of the initial table <name of table>, for which the INSTEAD OF triggers developers make the necessary actions to change the data in the initial tables. If we simplify a bit, then this solution is a solution of the intermediate layer, only implemented immediately inside the DBMS.

Considered tools *Flashback* and *Workspace Manager* suggest that the company Oracle is moving towards providing a full range of temporal extensions. However, so far many of the ideas of temporal database technology in Oracle DBMS are not implemented.

- *IBM Informix TimeSeries DataBlade.* Module TimeSeries extends the functionality of a database Informix, adding processing and analysis of processes based on time series data. It contains the definition of new types of data – time series and calendar, and provides more than forty functions for processing data including timestamps. This module provides a great opportunity to analyze the dynamics of individual process, and also to store the large volumes of time series data efficiently and provide an access to them.

- *Immortal DB*. The aim of the project Immortal DB, implemented by a research division of Microsoft, has been providing support of transaction time, built-in SQL server, but not based on any proxy-level [68]. An attempt was made to demonstrate that in DBMS, except for support of images, can be implemented full support of transaction-time, and with good performance. With the implementation of support a standard mechanism of snapshots was extended, which allows to get a snapshot of the database at the time of the past.

- DBMS *Postgres*. The main features of the system memory management in Postgres are, firstly, the fact that it does not have the usual journalizing of database changes and it instantly provides the correct database state after restarting the system with the loss of the status of RAM, and secondly, the memory management system supports the historical data. Queries can contain temporal characteristics of objects of interest. Realizably these two aspects are related. The main solution is that whenever you update the record a new copy is created, and the previous version continues to exist. Even after removing, all the accumulated variants are stored in the database. Any variant of the record can be extracted from the database if you specify the moment or the period of time when this variant was valid. The reviewed functions Postgres DBMS have been further commercialized in DBMS Informix.

- *Temporal PostgreSQL* – tools for maintaining and querying time data in DBMS PostgreSQL [45, 46]. It includes temporal data types, functions, operators, opclasses (for indexing), and triggers.

- *LogExplorer* from Lumigent provides an analysis tool for Microsoft SQLServer logs, to allow one to view how rows change over time (a nonsequenced transaction-time query) and then to selectively back out and replay changes, on both relational data and the schema (it effectively treats the schema as a transaction-versioned schema).

- *Quest Log Reader for SQL Server* – product from the company Quest similar to the previous.

- *aTempo's Time Navigator* is a data replication tool for DB2, Oracle, Microsoft SQL Server and Sybase that extracts information from a database to build a slice repository, thereby enabling image-based restoration of a past slice; these are transaction time-slice queries.

- *IBM's DataPropagator* can use data replication of a DB2 log to create both before and after images of every row modification to create a transaction-time database that can be later queried.

The review of the considered software products shows that there are various tools which let the developer work with the temporal data. These tools differ in software product category, completeness of implemented temporal functionality, possibility of integration with operating platforms, control difficulties, power, reliability, manufacturer support, costs etc. Analysis of these existing IS tools implementation has proved the conclusion that there is no unified opinion what software products are more preferable to use for temporal data support. In order to assess the possibilities of these various tools of temporal data support in transport companies IS, the author has performed their classification and has singled out the following four groups:

1. Temporal DBMS;

2. Temporal frameworks;
3. Relational DBMS with transaction time support;
4. Software products which allow to interact with the previous versions of the data.

In point 2.2.3 the advantages and disadvantages of each group is considered.

2.2.3. Justification of the Platform Choice for Designing the Temporal Information Systems in Transport

There is an analysis of the possibilities of applicability of temporal DBMS, existing temporal frameworks, relational DBMS with transaction time support and separate software products, considered above, as well as classic relational DBMS as a platform for designing temporal IS in transport.

The major *advantage of temporal DBMS* such as Postgres, is support of the temporal databases at kernel-level system and the presence of temporal query language. It should be noted that this advantage does not compensate for all the problems associated with the support of temporal data. The main *disadvantages of temporal DBMS* that prevent their use by large transport companies today are:

- lack of uniform standard;
- intolerance of challenges developed by means of temporal DBMS;
- lack of a complete set of tools to support temporal data in transport (periodicity support, a special calendar, a specific mode of data input (see point 2.4.3));
- noticeable lag in power and reliability of industrial relational systems;
- poor support of the manufacturer;
- the complexity of integrating with other IS, applicable in the enterprise.

It also should be noted that the classical temporal functionality, implemented in temporal DBMS, does not fully satisfy the needs of the IS in transport [19, 20, 22]. For example, the use of classical temporal relations is not always the best solution for information systems in transport, and the support of the periodicity and the special calendar is not a part of mandatory functionality of temporal DBMS at all.

The considered *temporal frameworks* (TimeDB, Temporal PostgreSQL), developed for relational DBMS (Oracle, Sybase, PostgreSQL), allow interact with different versions of the data with the help of SQL-like queries and support both transaction and valid time dimension. Frameworks are designed in the form of intermediate between application and database. This intermediate link controls temporal integrity, converts temporal queries in SQL-92 queries, etc. Such design allows interaction with database without temporal framework and thus, allows to break the integrity of the data. For such databases as RDS database, which as a rule, are used by several IS, developed in different time and interacted with database “directly”, it is important to have temporal upward compatibility support. Considered temporal framework do not provide such function.

Concerning the *software products, allowing interaction with the previous versions of the data* (LogExplorer, Quest Log Reader for SQL Server, aTempo's Time Navigator, IBM's Data Propagator), it is worth noticing that this functionality is not the principal objective of these products; this fact complicates the IS support and does not ensure "convenient" interaction with the temporal data.

Relational DBMS with transaction time support (Oracle) are suitable for the assignments which require only transaction time, which is evident from the name of this software product group. As the analysis of Latvian Railway IS, presented in Chapter 1, shows, in the IS on transport both time dimensions are of demand, as well as periodicity and a special calendar.

From a technology perspective the temporal database the most functionally completed are Teradata, TimeDB and PostgreSQL, which includes support for both transaction and valid time dimensions. However, they also do not possess the periodicity and the special calendar. Using a complex of temporal frameworks and considered software products to achieve complete functionality to work with temporal data cannot be possible or effective for obvious reasons: the incompatibility of decisions, the complexity, etc.

The advantages and disadvantages of using the relational DBMS as a platform to build the temporal IS in transport are considered below.

The benefit of relational DBMS the following facts support:

- they are tested and proven over several decades systems;
- DBMS manufacturers are constantly developing their products;
- support for commonly accepted standards;
- relational environment is still the best for storage and data integrity provision;
- most businesses have successful experience working with relational systems;
- a set of existing application systems are developed on relational technology;
- many new IS continue to be implemented in a relational environment.

Speaking about the *problems of using a relational DBMS* when working with temporal data first of all, we note the following:

- relational DBMS do not have specialized tools for the development of the temporal database, while the temporal databases are created specifically for this task;
- query languages of the relational DBMS are not designed to work with a time dimension;
- relational DBMS do not possess any knowledge about the semantic of time dimension of temporal relations and cannot control the correctness of its value. Consequently, the problem is to ensure the integrity and confidentiality of data;
- there is no ready-made full-function framework to the relational DBMS for the building of temporal IS on transport.

Weighing all the pros and cons of known approaches to temporal systems implementation, as well as taking into account transport companies' investments in relational systems and the accumulated experience of their use, the author considers it expedient to sacrifice some functionality of the existing temporally-oriented products in favor of the reliability of relational platform and development prospects. Despite the practical lack of standard tools for building temporal systems in the relational DBMS, their implementation on the basis of most relational systems are possible and appropriate, however, requires fundamental design study data models, knowledge of temporal technology, and peculiarities selected for implementation of the DBMS. Successful experience in temporal frameworks building for DBMS Oracle and PostgreSQL supports this choice (see point 2.2.2).

2.3. Choice of the Models and Methods Subjected to the Usage in the Temporal Information Systems on the Railway Transport

The special models and methods are required for the temporal data support in the railway enterprises information systems. These methods and models partially existed formerly, some of them required some approbation and verification in the process of functioning, and several models and methods were developed in the course of the thesis writing. The author has analyzed the Latvian Railway IS for the possibility of using temporal models and methods. In the process of analysis, passenger and freight transport operative and analytical systems, a set of information systems of railway infrastructure and financial and economical sectors were considered. The result of this research is the structural diversity of models and methods, urgent in the area of temporal data control in the IS on transport. The total set of models and methods are presented in Fig. 2.11. For the better integrity of the picture, both famous models and methods and those developed in the given paper and considered in the further chapters are shown on the scheme.

The analysis of the information systems employed on the railway has demonstrated that the components maintaining the temporal data, are located on different levels: the system level, the logical level and the application layer. Therefore the level is the first feature of structuring. The system level includes databases management system facilities; the logical one comprises the components which allow describing the temporal model; and the application layer incorporates the models and time-oriented (applied systems, the operation of which is closely connected with the registration of the data temporal properties, oriented on the temporal data processing, as well as the time-dependent systems) application systems methods, taking into account the data temporal characteristics. The double line on a scheme shows the model groups and methods under investigation in this research. The scheme demonstrates the details of the groups requiring the development. A part of the methods (it is especially typical for the system level) are general-purpose methods, but some of them are the temporal data-oriented. The known models are indicated with the white colour, and the models developed by the author are shown in green. In its entirety the presented scheme reflects the plan of the research done within the frameworks of the promotion work. At the same time the main attention is dedicated to the logical level and in particular to the representation of the information systems temporal objects in the database and the methods of the interaction with these temporal objects. It is an important point to be said that the presented scheme has a generality and it can be used for pursuing the similar researches in other application fields with some specifications and additions.

Every of the above-described and presented levels is observed in details.

2.3.1. System Level

Almost none of the modern industrial systems are developed from scratch, although it was naturally an intrinsic practice for the 60-70s. Most of them use the existing data control platforms. These platforms take into account all the necessary methods for physical data arrangement. However, the information about the low-level methods of the data access of the chosen DBMS is necessary for the effective temporal data control.

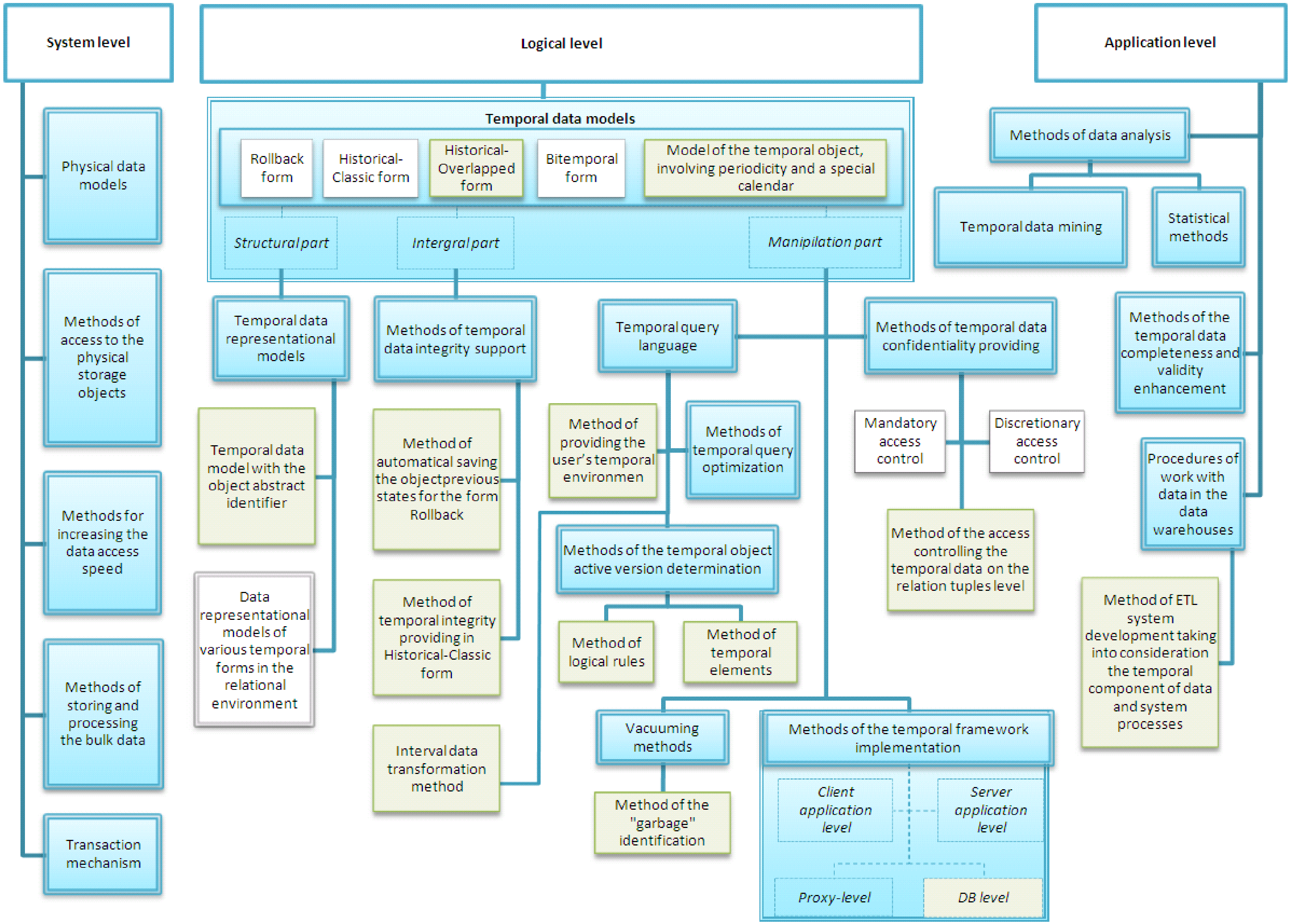


Fig. 2.11. Models and methods aggregate subjected to the usage in the temporal information systems on the railway transport

There are five groups of the models and methods suggested for employing on the system level. Such models and methods as the *data storing models*, *the methods of access to the physical storage objects*, and *the methods for increasing the data access speed* are connected with the table data arrangement and organization within the files on disk systems, application of various indexing methods (i.e. including data clustering), parallel data access organization. The similar tasks are solved by the *methods of storing and processing the bulk data*, but these methods are large-data-amount operation-oriented. The above mentioned circumstance is especially significant for the temporal systems; consequently this group of methods is shifted into the separate block. *The transactions* play an important part in the temporal model integrity implementation. The issues of the system level model and method employment are considered in point 2.4.8 of this work. The practical application of them in the information system on transport is presented in point 4.1.

2.3.2. Logical Level

The choice of the *temporal data model* (that is a temporal form) is a key point on the level of logical design of the database. The temporal model is considered similarly to the relational one, consisting of three parts: the structural part, the manipulation part and the integral part. The differences between the temporal models in the structural part are expressed in the set of temporal attributes; in the integral one – in the temporal integrity limitations, and in the manipulation part – in the algebra of queries to the temporal data. Accordingly, the methods presented on the scheme have their own peculiar implementation for every data temporal model.

The structural part suggests the following pattern: *the temporal data model with the object abstract identifier* serves as a universal model of the temporal data representation in the relational environment (see point 3.1.1 and [4, 8]). It allows presenting the temporal object in the form of relation set.

Relations, describing the temporal object characteristics, may have various forms: Rollback, Historical-Classic, Bitemporal, Historical-Overlapped. The first-named three forms are widely known [51, 52] and considered before in point 2.1.4, but the fourth form, Historical-Overlapped, is developed by the thesis author and employed in the information systems on the railway transport (see point 3.1.2 and [18, 19]). The developed form takes into consideration the specific mode of the temporal object version change. This mode is inherent for some processes on the railways. Moreover, there is a variety of objects on the transport, possessing the property of periodicity; these objects are connected with the specific (alternated) calendar. The *model of the temporal object, involving the periodicity and a special calendar* is suggested by author for the similar object description [19, 20, 21, 22, 24].

The manipulation part comprises the methods, describing the correct interaction with the temporal data as well as a temporal framework, providing this interaction and taking into account the integral part. They provide the automatic saving of the object previous version and its transfer to the shadow zone and guarantee the existence of the only active version of the object at the moment of the query, and so on.

Whereas the specialized temporal DBMS have not been employed in the industrial field for a number of reasons (see point 2.1 and [4, 7, 18]), the developers have to generate the logic of the temporal objects manipulation by their own efforts and resources. The relational DBMS do not present at all or present partially the facilities for the temporal data management. In this respect the manipulation and the integral parts of the temporal model in the relational environment require the expansion of the corresponding relational apparatus by the temporal logic. This logic can be implemented on the different levels of the system architecture: on the *level of database*, on the independent transitional *proxy-level*, on the *level of server* or *client (user) application*. Therefore different *methods of the temporal framework implementation* can be used. The paper [18] considers the issues of the level of the temporal framework implementation. Further in view of the database level preferability, the author takes into consideration this level only and suggests the temporal framework methods, considering the peculiarities of the temporal data support in the relational environment in the structural, integral and manipulation aspects. The paper presents the method of the temporal framework implementation as the fundamentals of the temporal databases development in the relational environment [4]. It constitutes the complex approach to the temporal databases in the environment of the relational databases management systems, which can be cooperated with not only from the position of the temporal applications. The temporal framework of the non-temporal database are organized by the aggregate employment shown on the scheme of the logical level methods.

The employment of the *temporal query language* (the special language constructions for the queries to the temporal data), as well as the *methods of the temporal queries optimization* and special techniques of referring to the temporal data allows avoiding the ponderous standard query language offers and increases the efficiency of the queries execution. The solution for many of the above mentioned problems is found in the *method of providing the user's temporal environment*, developed in the research under consideration. This method allows setting the user at the definite point in time and executes the non-temporal queries to the temporal data relatively to this time. Besides this method is the facility of the temporal queries structural optimization.

Every temporal form requires the peculiar solution for the task of the determination of the temporal object version, active for specified moment of time. The complexity of the efficient solution of this problem increases simultaneously with the increase of the number of the temporal peculiarities of the temporal form. The paper suggests two alternative methods of the temporal object active version determination for the problem solution: *the logical rules method* and *the temporal elements method* [20, 21, 22, 26].

The majority of the multi-version data are stored in the databases of the information systems in the interval form. The interval form is improper for many computations from its inception, but their complexity multifold grows if the following aspects occur: the periodicity, the special calendar and the necessity of using the data from several interconnected tables for doing computation. The employment of method, suggested by the author, of the *interval data transformation* can solve the problem of "clumsy" data presentation for computation by converting the interval fact into the time sequence [25].

The *methods of the temporal data confidentiality providing* comprise the methods which protect the temporal data from unapproved access on reading and changing procedures. The problem of confidentiality providing is observed in point 3.6 within the framework of the solution of the problem of the *temporal data access control*. The known approaches to the access management – the *discretional access control* and the *mandatory access control* – do not cover all the requirements of the delimitative temporal data access strategy. In their traditional version they are not capable of considering the time dimension, containing the necessary delimitative information. One of the tasks, belonging to this problem cluster, is observed by the author in her work [15]. This work suggests the way of controlling the users' access to the different time periods data with the *method of the access control to the temporal data on the relation tuples level* [6, 13, 15].

The access to the temporal data is considerably more complicated and more demanding for the computational resources applied. The increase in the temporal data access efficiency is achieved by the way of employing the *method of databases cleaning up* of the out-of-dated unnecessary records (vacuuming). Every temporal form has the corresponding *method of the "garbage" identification* [18].

The integral part is represented by the *integrity providing methods*, comprising the methods of supporting the temporal normal form, the values integrity of the temporal attributes and other integrity delimitations. The integrity part also incorporates the support of the temporal relation functionality of different forms, exempli gratia, *the method of the automatic storage of the object previous version for the form Rollback* [8].

Despite the fact that the temporal relations are relational at the same time, the known facilities for the relational structures integrity providing are insufficient, because the objects, incorporated in these systems, are the multi-version objects. In the classical approach, the relation model does not cover this case, but the temporally-oriented methods of the integrity provision are necessary. The issues of the integrity and accessibility providing are considered in the course of analysis and development of the temporal data models in Chapter 3 of the dissertation and in [15].

2.3.3. Application Level

Temporal methods of applied assignments in IS are specific methods and techniques which do not fit the general theory, for example, the problem of train schedule optimization.

The methods of the temporal data completeness and validity enhancement are in demand in both the analytic and the transaction systems. This rank of methods does not need any additional adversaries; the only thing to be highlighted is the fact that the algorithms of the methods take into account the temporal data components. The following cases serve as an example of these methods practical employment: the omitted values regeneration, noise disposal from the data observed.

The temporal data analysis by virtue of any *analysis methods*, including the *statistical methods* and *data mining methods,* contrary to the trivial methods, is extended by the skill of "understanding" the data temporal structure.

The systems of the bulk data processing, such as the data warehouses, are inevitably connected with the temporal data. Any models and methods of developing the systems of the bulk data processing (for example, the virtual data model in the predicting systems [10, 11, 14], the Financial and statistical information analytical system of the passenger transportation [1, 2]) are to take it into account. *The processes of the data operating in the data warehouses* are connected with the temporal aggregates computation and with the time sequences activities. *The method of ETL*[3] *system design, taking the temporal data component and system procedures into account*, allows to simplify the data processing implementation and increases the temporal data security.

[3] ETL (*Extract, Transform, Load*)

2.4. Problems of the Chosen Models and Methods Implementation in the Relational Environment

2.4.1. Choice of the Indicators of the Temporal Data Models

The implementation of temporal models in the certain IS assignments depends on the variety of factors. The author offers to assess temporal models according to the two groups of indicators: informational and exploitation.

The first group comprises such factors as:

1. Automatic save of the earlier versions of the stored objects;
2. Identification of erroneous input data (“garbage”);
3. Support of logical deletion of the data;
4. Storing of the data related to the past, present, future;
5. Support of the periodicity;
6. Support of a special calendar.

The principles of CIA model (Confidentiality, Integrity, Availability; ISO 17799, [110]) of information security were offered in order to form the indices of the second group. The group of performance indicators includes the following properties:

7. Ensuring the confidentiality of stored data;
8. Sampling rate to determine the current version of the object;
9. The speed of data modification (adding a new version of the object);
10. The integrity maintenance during data changes operation;
11. Economical use of disc space.

The purpose of providing of the information security is granting to the authorized users the access to the necessary information and maintenance of confidence that this information is correct and that the system is accessible. Information security is based on three basic indicators of databases: confidentiality, integrity, availability. All three indicators harmoniously supplement each other, and excessive strengthening of one indicator can negatively affect others (see Fig. 2.12). I.e. confidentiality strengthening (an indicator 7) can negatively affect availability indicators (indicators 8 and 9), and an increase of the latter one can affect integrity (an indicator 10).

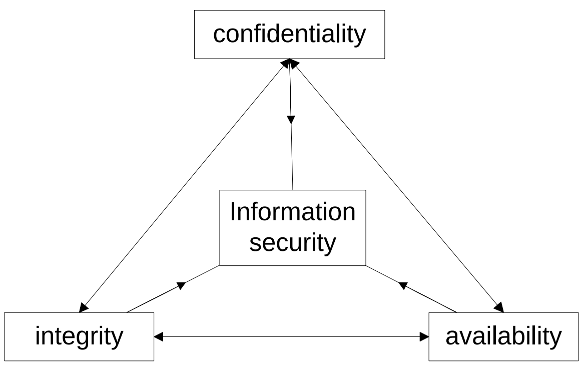


Fig. 2.12. Interrelation of the information security and the components of the CIA model

Concepts of confidentiality, integrity and availability are stated in the popular literature [112, 110], however at work with the temporal systems the author has faced necessity to expand the given definitions in a context of temporal systems:

Confidentiality is the security in time of the temporal data from unapproved access on acquaintance and change operations. On Fig. 2.13 the change in time of values of a sign of temporal tuple confidentiality concerning the temporal databases users is illustrated. One and the same tuple during the two different moments of time is accessible to different users, which is shown in figure by two time slices in the moments t_1 and t_2. At the moment t_1 the tuple is accessible to *User 1*, *User 3*, and during the moment t_2 – to *User 1*, *User 2*, *User 4*. The tuple in this case can display a lifespan of the object or its property.

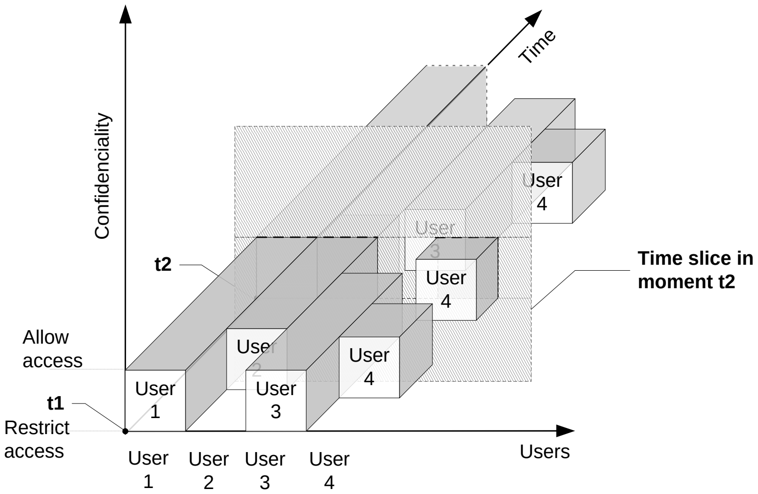


Fig. 2.13. Change in time of the temporal tuple confidentiality sign concerning DB users

Availability is the possibility to employ the information in the temporal databases when it is necessary. In other words, it is the state of the data, when they are in the form, the user needs, in the place, the user needs and at the moment of time, the user needs. *Availability means* provide an unimpeded access to the temporal data for carrying out the authorized operations on acquaintance, change and deletion. As the temporal database is on an order more difficult than usual relational (non-temporal), the problem of availability of the information becomes number one.

Integrity is the security of historical information from unauthorised change, maintenance of its consistency and completeness. *The integral part* includes set of rules, which provide the temporal database maintenance in a consistent state, and protect it from losses during operation.

2.4.2. Representation of the Temporal Data

In classical database used in OLTP, data represent a snapshot of the information model of enterprise vitality and any modification of an object of this model 'erases' its previous state, thereby eliminating the possibility of its chronological analysis[4]. In this connection, the version of such classical database does not fit some above-mentioned tasks implemented at Latvian

[4] It should be noted that the majority of commertial relational DBMS still preserve the previous state of the object in the change register but provide with virtually no access to it.

Railway. The solving of problem of temporal data use in the IS the author sees in the application of the principles of temporal databases, which complement the stored data with the property of time [37, 39].

Scientific publications on the principles of design of temporal systems in the relational environment appeared in the mid-80s (see the evolution in point 2.1.1). Most of them devoted to investigations of temporal relations and extensions of relational algebra to manipulate with the temporal data. In the course of development of relational database management technologies and standards new research appears, but in the field of active database usage, which almost all modern and widely used relational DBMS possess, there are not so many investigations. None of them had a comprehensive solution for implementing the functions of the temporal databases into the relational system. In addition, IS on transport, which, as was noted in Chapter 1, have a number of specific features, and the construction of temporal features for them is connected with notice of increasing demands for performance, implementation simplicity, integrity, availability and confidentiality of data.

Researchers of the temporal databases have always had in their disposal several ways of creating the temporal DBMS. Initially, the creation of the possibility of the temporal database DBMS from scratch, i.e. independent realization of some temporal model was considered. However, relational DBMS were progressing rapidly, their functionality was expanding, so the creation of “scrimp” temporal DBMS was considered unwise, but the resources and possibilities of all means of relational DBMS re-implementing (during the creation of temporal DBMS) were not available. Accordingly, there was no users’ demand, either. Therefore, researchers began to consider various ways to complement and expand ordinary relational DBMS by the temporal data model support. Almost all these ways resolved into the creation of a functional block – the temporal framework over the relational database. The main difference was the level of “interference” in the relational DBMS, and the degree of intellectuality of the given block. The comparison of different variants of the intermediate layer implementation is presented in [97].

Until the early 90s, relational DBMS (RDBMS) did not support active databases, stored procedures and functions. Therefore, until the advent of this support, the platform RDBMS was significantly inferior to a conventional application development tools as a software platform for the implementation of temporal logics. Apparently, it can be explained by the fact that the vast majority of research and implementations chose the way of addition and expansion of relational DBMS at the intermediate proxy-level or application level. Most studies in the field of temporal database design traditionally do not use RDBMS tools even now assigning them only basic functions of storage and query processing. However, the author believes that temporal database design with the use of modern RDBMS is more preferable than with the use of an outer shell for several reasons: more reliable integrity insurance and security of the temporal data, the concentration of logics in one place, the simplification of information system architecture. Due to the changes in the level of implementation, a number of unsolved issues regarding the organization of temporal data and manipulating with them are revealed.

There is a consideration of the problem of designing the temporal object in a relational environment. Objects on transport, as well as many other objects have multidimensionality, i.e. they have many properties, characteristics, parameters. Each of the properties may be a scalar or multiple, between each other the properties can be dependent and independent, changing value of a property can be strictly related to the change in another property and vice versa, such connection might not exist, different object properties change with different frequency.

Much of the research in the field of temporal data representation is focused on the temporal relations and on issues of representation of time dimensions, rather than on the temporal object as the object of the simulated world, which may constitute a scheme of interrelated temporal relations. It seems that the problem of temporal object modeling in a relational database is related to the ordinary problem of designing a database. In this case, this problem, which, virtually, is a typical problem, every temporal database developer, has to solve 'from scratch', making traditional errors that entail integrity support issues: the appearance of *redundancy*, *wordiness*, and *data inconsistencies*. Empowering the time dimensions of multidimensional objects requires a special approach to modeling in a relational DBMS.

Solution to the problem of adding time support in the existing relations is often considered on examples, but not in a formal way. Moreover, the relations shown in examples do not contain enough attributes to show the principle of relation decomposition to conform to TNF. It should be noted that questions of keys definition, referential integrity and TNF are often solved partially or separately, for example in [51].

The problem of determining the primary key in temporal relation is often solved by including in the key one or more temporal attributes. Thus, the *integrity of relationships* maintenance is carried out. This design in the futured does not allow to use the relational mechanism or the *referential integrity* (foreign key) and its advantages such as simplicity, reliability and productivity during modifications. At the level of database referential integrity for such relations can be implemented only by using triggers, which complicates the logic of the database work, slows down the integrity check during data modification operations, and most importantly, does not guarantee the referential integrity of the data itself. Relevant question here is “child” relations, when a temporal object acts as a “parent”. Moreover, it is necessary to consider the temporal dependence of the object on other objects represented in the relational database, which can also be “objects of public use” that is used in many IS of the enterprise.

2.4.3. Use of the Classical Relation Temporal Forms in the Transport Information System

There is a consideration of the possibility of using classical temporal relations in the Latvian Railway information systems chosen for the research.

Classical forms of temporal relations, namely the Historical form, cover virtually all requirements for data modeling in the Reference Data System (*KLASIFIK*). It allows you to specify the period of relevance to the time-varying names and codes of stations, they belonging to the regions and areas. The possibility of using this form is also evident in the systems *IP*, *KVK* and *ADIIAS*.

In the system of passenger trains schedules *SAR* based on Historical form it is possible to organize temporal data of tariffs. However, this form does not describe the periodicity, which is being increasingly used in the ridership tariff policy of Latvian Railway (for example, discounts for travelling on weekdays). The same drawback can be also noted for the invoice processing system, which carries out customers' payment (*CP*).

Rollback form is suitable for the implementation of automatical saving the hardware and software history change in the workplaces in the system *APNIS* and forced saving the previous versions of technical documentation of stations in the IS of railway infrastructure *SPDB*.

Nevertheless, the practical application of the classical temporal forms in a relational environment raises issues related to maintaining the integrity, automation of the rending functions into the “shadow zone”. The problem of “garbage” accumulated in the system due to operators’ errors should be also mentioned.

The investigation of the feasibility of using the classical temporal forms for organizing the data schedule in the IS *SAR* is presented below. Rollback form will not be considered, since it by definition cannot contain data of the future, i.e. will not allow to plan a schedule. The possibility to register Bitemporal form records changes is not so important within the framework of this task, that is why the consideration of the Historical form (Bitemporal form without the functions of records changes registration turns into a Historical form) is shown.

As it was already mentioned in the Chapter 1 (point 1.5.2), the schedule of passenger trains has multi-version related to seasonal schedules, cancellations of trains, repairs, etc. Historical form supports multi-version object and allows you to plan schedules shift. However, its use is difficult or ineffective for the following reasons:

1. In the course of time, different versions of the train schedule interchange. One and the same version can go into effect many times: a basic schedule replaces the modified and vice versa. In the Historical form all versions of the object SUCCESSIVELY substitute each other, one and the same version once went out of power, does not come into it again. Of course, we can replicate (duplicate) versions and work with multiple copies, placing them in the correct order. But this approach has no future, involves a large data redundancy and wasteful consumption of disk space.

2. For one and the same train on different days of the week a schedule might be different (eg, a schedule for weekdays and weekends). In addition, trains run on those days of the week and month for which the scheduled timetable (for example, only on Fridays, twice a week on Tuesdays and Thursdays, on weekdays, on weekends, even days of the month, daily), that is, in all other days of the train do not ply. Historical form does not sustain periodicity. Option replication of versions theoretically acceptable, but in this case, it complicates the organization of data so much data, that makes schedule management virtually impossible. For example, the change in the train schedule on Saturday needs to be done in each of the many “copies”, which will cause a lot of changes. The function of copy versions and putting them in the appropriate range is a separate problem. In addition, this approach limits the planning schedule to the last day of the calculation (in this case, the last revealed Saturday).

3. On holidays and other special days, (such as, holding the NATO summit in Riga in November 2006) all the trains run on the day off schedule. As an exception, for some trains a special schedule is made for those days. Historical form does not support the periodicity and ignores the "modified" calendar. But the option of replicating versions is quite acceptable.

4. Train schedule consists of multiple versions, which appearance is due to the seasonal cycles of the train schedule change, days of weeks, planned and unplanned maintenance work, moving holidays and working days and other factors. The change of the current schedule is a new schedule of certain duration. In the case of Historical form when such a schedule occurs, it will be necessary to make a lot of actions with those versions of the schedule which affected the new, i.e. some of them will have to be deleted, some will require their effective range change. Moreover, in this case, the information about which schedule is the main and which appeared because of operational changes is lost.

2.4.4. Maintenance of the Temporal Data Integrity

When designing the TDB, an important condition is certainly compliance with the concept of TNF, as described in point 2.1.5. However, even if we follow TNF but there are not specific temporal mechanisms to ensure integrity, in the database the threat to the integrity of the given temporal relations remains.

Maintenance of the temporal forms integrity also includes crossing of the time intervals restrictions defined in point 2.1.4. If these restrictions are violated, then there is ambiguity, and the active version of the object will be defined incorrectly. Thus, the data availability will be harmed. Standard relational mechanisms of integrity insurance do not provide such functionality. However, using triggers, it is possible to achieve integrity support for example, to implement a ban on the appearance of more than one version of the object in the relation at any point on the valid or transaction time axis.

2.4.5. Temporal Logic and the Temporal Query Language

One of the most important reasons to have a database which supports the time dimension is the ability to perform ad hoc queries on the data. The current standard for conventional (relational) databases is Structured Query Language or SQL. SQL has become the industry standard for RDBMS because of its ease of use due to its English-like syntax. SQL is both feasible and user-friendly. But the addition of the time dimension, however, greatly increases the complexity of the queries on temporal data. In point 2.1.4 it is shown how complex the data set becomes after expanding it with the time dimensions. With the additional element of time, in its current form SQL is no longer able to process ad hoc queries as it did on the (relational) classical database. A new query language or extension to SQL is necessary.

There are many studies in the field of query language to temporal data. A standard of the language TSQL2 [59], has been developed, but unfortunately, has not yet been implemented in any commercial DBMS. Thus, this standard can only be gone by in developing your own temporal data interaction mechanism. However, this standard does not describe the periodicity of events. Research in this area is discussed in point 2.4.6.

The performance of one of the basic principles of temporal technology - ensuring the correct access to previous versions of the object and guarantee the existence of only one version at any time, is seen as a very urgent task in the temporal transport systems. The condition under which this principle is implemented requires not only the correct data, but also correct and robust methods to an access to them and their processing.

A certain problem moment is the change of the existing data structures and support for older programs that are designed to a specific data structure. Therefore, another requirement for the temporal database enquiry mechanism processing is the *Temporal upward compatibility* provision; it is necessary for all legacy constructions to work as they used to after adding of the temporal support to the system [42, 76, 77]. For example, a query to determine train fare between stations A and B should consider the current database state only. Constraints and assertions also work exactly as before: they are applied to the current state and checked on database modification. This implies that no existing applications should feel the transition from regular relational database to the temporal one. That is, addition of the time dimension to any particular table should not affect the correctness of queries to it. On the other hand, such addition should allow the use of temporal queries in new programs, notably facilitating the work of programmers and system administrators.

2.4.6. Support of the Special Calendar and the Periodicity

Calendars and periodicity play a fundamental role in many applications. For example, transport systems are characterized by using periodicity connected with the days of the week, parity of days, working shifts, etc.

Recently, some commercial databases started to support user-defined periodicity in queries in order to provide “a human-friendly way of time handling” (see, e.g., TimeSeries in Oracle 8). On the other hand, only few relational data models support user-defined periodicity in the data, mostly using “mathematical” expressions to represent periodicity. In this area Paolo Terenziani and his followers' research is worth attention [62, 63, 64]. They offer a high-level “symbolic” language for representing user-defined periodicity, which is more human-oriented than mathematical ones. In the above-mentioned works they consider temporal relational model which supports user-defined “symbolic” periodicity (e.g., to express “on the second Monday of each month”) in the validity time of tuples and cover time ranges (e.g., “from 01.01.2009 to 28.02.2009”). Temporal counterpart of the standard operators of the relational algebra is defined, new temporal operators and functions are introduced. Temporal algebra, which is consistent extension of the classical (atemporal) one, is offered.

Periodicity means that the continuity of the object version’s lifespan is broken cyclically and is valid only at the moments, which match the specified periodicity. The periodicity of events is mathematically represents by the pair:

$$\langle [t_s, t_e], p \rangle,$$

where $[t_s, t_e]$ is time interval, time of beginning and end of which is presented by t_s and t_e accordingly, *p* is periodicity. The examples of *p* are following: “on Tuesday and Wednesday”, “on even days”, “at weekends”, “every first Monday of the month”.

Periodicity *p* can be represented in different ways, ranging from binary or numeric representation and ending with a high-level symbolic language like Terenziani’s. The first way is not visual and is limited to a specific set of recognizable types of periodicity. That is, when the periodicity is a very complex expression, then you may encounter with a limitation or ambiguity. For example, the periodicity of "the second Monday of every even month, as well as the first working day of every odd month except February," is a complex periodicity based on the basis of more simple types of periodicity connected by the certain rules.

Terenziani’s approach allows to take into account such a complex periodicity. To do this, it computes the set of time moments of activity of each simple component of the periodicity and performs the subsequent operations with the computed item sets. Among these operations, there are all operations that can be performed with item sets: merge, intersection, etc. However, you can look at the periodicity from another angle. It also represents a logical expression built on the basis of simpler types of periodicity, connected by the logical connectives in a certain sequence. Such a view gives quite different possibilities for computation. However, this approach has never been investigated so far.

A separate task is to use the periodicity in multi-version temporal relations. For example, in dealing with overlapping time range (see point 3.1.2) when determining the active version of the object, misinterpretation of the order of two conditions is likely probable: the periodicity and priority of versions, which affects the reliability of the data negatively. Representation of the temporal data structure in the form of ER-model cannot show such specificity, which designates the order of filters usage. Let’s consider this problem on the example of passenger train schedule task.

Example. An ambiguous understanding of the ER-model of passenger trains schedule. In this task by the word “schedule” we understand the version of the train timetable *Train_ID*, characterized by an identifier *Sch_ID* and a set of stop times at the stations, described in a child relation *Schedule*. The train runs only on certain days of the week. ER-model of this problem is shown in Fig. 2.14. ER-model is designed according to the data model with an abstract object identifier [7].

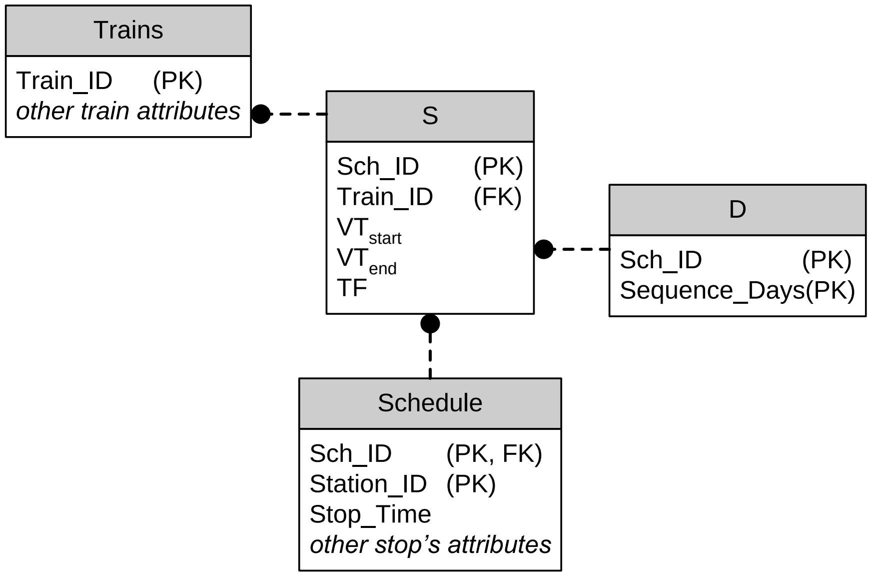


Fig. 2.14. Elements of the ER-model of the train schedule task

The relation *Trains* is non temporal relation that contains train’s abstract identifier *Train_ID*. The relation *S* is a temporal relation, that describes versions of train schedule, and *Sch_ID* is a version of the schedule. Days when the train runs according to the schedule *Sch_ID* are described in the relation *D*. A detailed description of the schedule with the stop time at the stations is described in *Schedule* relation.

Fig. 2.15 and Fig. 2.16 show a diagram of the schedule time interval intersection and time slice. The relations of the database, which are needed to clarify the example, are shown in Fig. 2.17 and Fig. 2.18.

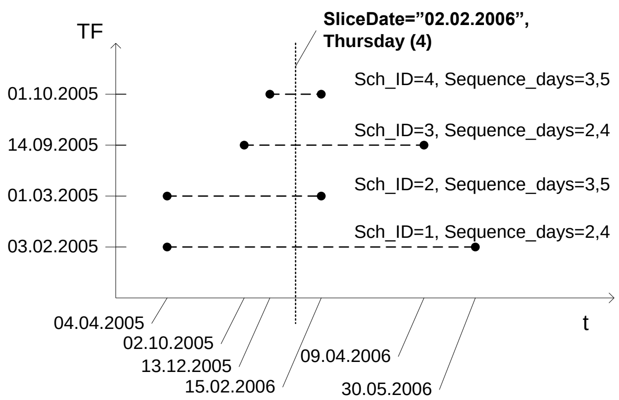


Fig. 2.15. Diagram of the time interval intersection of the schedules and the slice at the moment of time SliceDate

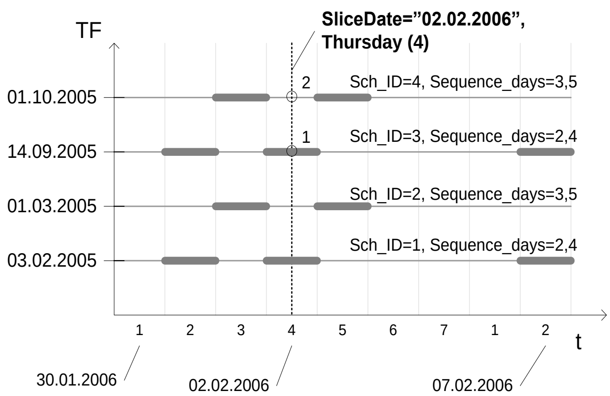


Fig. 2.16. Periodicity of the train running according to the schedule of the different versions in the vicinity of the time slice SliceDate

Sch_ID	Train_ID	VT_{start}	VT_{end}	TF
1	14B	04.04.2005	30.05.2006	03.02.2005
2	14B	04.04.2005	15.02.2006	01.03.2005
3	14B	02.10.2005	09.04.2006	14.09.2005
4	14B	13.12.2005	14.02.2006	01.10.2005

Fig. 2.17. Relation S

Sch_ID	Sequence_Days
1	2
1	4
2	3
2	5
3	2
3	4
4	3
4	5

Fig. 2.18. Relation D

In the description of the database with the use of the available ER-model designation, there is an ambiguity in the interpretation of the valid model of train running. Possible interpretations of the model and the corresponding scenario of finding a valid schedule are given below.

Interpretation 1. The train runs several times a week according to different schedules. In one period of time the train can have several schedules, for each day of the week (see Fig. 2.19).

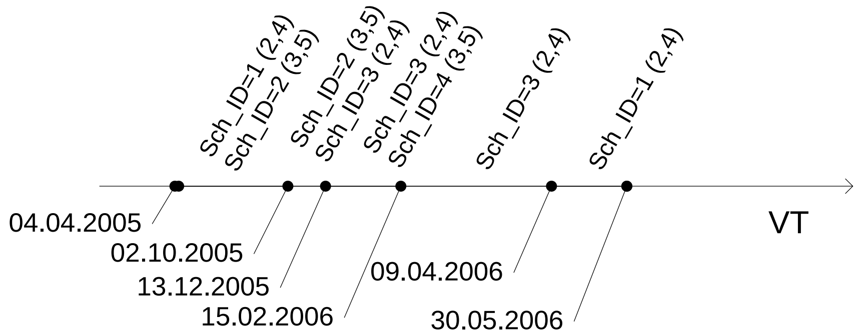


Fig. 2.19. Overlapping the different versions of the schedule on the axis of valid time according to the interpretation 1

As it is seen from relation *D* train 14B plies on Tuesdays, Wednesdays, Thursdays and Fridays, but on Wednesdays and Fridays according to one schedule, and on Tuesdays and Thursdays according to another one. The search scenario of the valid schedule in this interpretation is next: define the valid schedule for the time slice, taking into account the days train running.

The search function can be written in algebraic form:

$$Sch\_ID = \pi_{Sch\_ID}(\sigma_{t(SliceDate), fd(SliceDate)=Sequence\_days}(S \rhd\lhd D),$$

where *t*(*SliceDate*) is the temporal function of the lifespan search valid by the time of slice *SliceDate*, *fd*(*SliceDate*) is the function of converting the date of time slice into a attribute format of *Sequence_day* periodicity.

In this case, at the time slice 02.02.2006 (Thursday) the schedule *Sch_ID=4* will be found, and on 03.02.2006 (Friday) – the schedule *Sch_ID=3*.

Interpretation 2. The train runs several times a week according to the same schedule. At one period of time the train can have only one schedule. The train runs on the days of the week, which are assigned to this schedule.

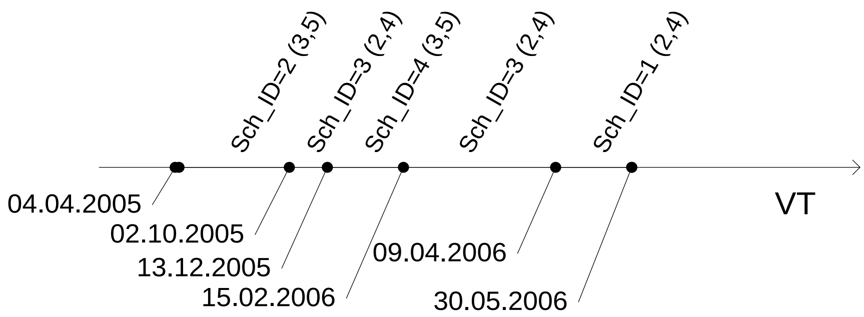


Fig. 2.20. Overlapping the different versions of the schedule on the axis of valid time according to the interpretation 2.

Train 14B runs only twice a week: from 04.04.2005 to 02.10.2005 on Wednesdays and Fridays, according to the schedule *Sch_ID=2*, in the period from 03.10.2005 to 13.12.2005 on Tuesday and Thursday according to the schedule *Sch_ID=3*, in the period from 14.12.2005 to 15.02.2006 on Wednesdays and Fridays, according to the schedule *Sch_ID=4*, from 16.02.2006 to 09.04.2006 on Tuesdays and Thursdays, on schedule *Sch_ID=3* again, etc. (see Fig. 2.20).

The search scenario of the valid schedule in this interpretation: choosing the valid version of the schedule *S*, and then check the conformity of the periodicity of the requested date attributes to the attributes of the periodicity of the selected version. Then the search function can be written in an algebraic form:

$$Sch\_ID = \pi_{Sch\_ID}(\sigma_{fd(SliceDate)=Sequence\_days}(D \rhd\lhd \sigma_{t(SliceDate)}(S)) .$$

In this case, the valid one for the time slice 02.02.2006 (Thursday) will be the schedule *Sch_ID=4*, and for the 03.02.2006 (Friday). It will be determined that the train does not run.

As can be seen from the above example, the standard designations for ER-models do not allow to interpret the valid model of the object unambiguously. The chosen treatment may dramatically change the result of determination of the temporal object version. This situation affects negatively the integrity and availability of the data.

This problem occurs not only in dealing with overlapping time intervals, but also in other forms of temporal relations, implementing periodicity.

2.4.7. Analytical Processing of the Temporal Data

All the considered types of temporal data are subject to analytical processing: multi-version data, collected historical data and time series. However, each species has different classes of problems and solutions, as well as the associated problems and outstanding issues.

The peculiarities of the *time series* data processing are reviewed.

In the systems in railway transport time series are: the speed of the train, the coordinates of wagon location, the capacity of transport routes, the intensity of selling tickets for trains, the situation on the railroad tracks and many other indicators that change over time.

Typically, *time series*, and the *accumulated historical data* represent a considerable amount of information stored in the information systems, and is widely used in analysis and forecasting. It is worth being noted that the accumulated historical data can be easily transformed into time lines with the help of simple aggregation operations. A wide range of time series analysis systems offer a variety of methods to extract information about the past and the future.

As a model of time series data representation the relation containing, except the observed indicators, the observation time can be taken. It might be an ordered in time sequence, characterized by the number of observations. In this case usually the periodicity of observations is known, on the basis of which it is possible to calculate the time of particular observation. Thus, the time dimension is represented either by the observation time or is calculated basing on the number of observation.

Time series processing is a complex operation. The complexity is connected with the large volumes of data, for which it is not easy to ensure the effective manipulation. There are various other processing problems:

- differing data formats;
- different frequency of data collection (Differing sampling rates);
- noise, lost value, and so on.

Here are the classes of the problems which are solved while working with time series: prediction, correlation, regression, benchmarking, periodic pattern mining, temporal association finding, causality analysis, sequential event patterns, threshold selection, frequency analysis, anomaly detection, clustering and classification.

However, there is a problem of accumulated knowledge and developed techniques usage in the area of time series processing. Considered in point 2.1 the model of *multiversion data* representation store the versions in the interval form. This form of data representation, complicated by the relational dependencies and periodicity and special calendar which are typical for transport problems, is not suitable for analytical calculations. The majority of used methods of data analysis, and, accordingly, analytical systems, does not operate with interval forms of data but with the time series. Therefore, one of the most important aspects that should the temporal system possess is the tools for converting temporal data to a suitable for analysis form, in particular, to the point form. In the point form, a fact is not associated with the interval, but with a set of moments of time of the selected granularity (on transport the most common level of granulation is one day). In addition, operations on such sets are useful for theoretical calculations and formalization.

Here is an example of this problem. There is an assumption that in the table (see Fig. 2.21) the schedule of a several trains is presented in the interval form. The schedule changes in time, the periodicity of train running change too. As it is shown, the train can run either on weekdays or at weekends, or every day.

Train	**Time**	$\mathbf{t_s}$	$\mathbf{t_e}$	**Periodicity**
Train1	17:00	01.01.2009	30.04.2009	Work days
Train1	17:25	01.01.2009	30.04.2009	Days-off
Train1	16:59	01.05.2009	–	Mon, Fri
Train3	23:45	01.01.1999	–	Sunday

Fig. 2.21. Railway schedule (Train stops at the station X)

Train1 leaves the point of departure at 17:00 on weekdays and at 17:25 at weekends. This schedule is valid until April 30. Then, the train runs only on Mondays and Fridays with a different time of departure - 16:59. For the particular day only one version of the train is valid. However, the calculation of the valid version may be complicated by the existence of a special calendar where the signs of periodicity have been changed for the specific days and where the working day may be announced as a day off. There is one more fact to it: Monday, November 19 is declared a day off and all the trains should run according to the day off schedule.

Under these conditions, the usual query of an analyst, who wants to build a chart of the number of trains per day, (see Fig. 2.22), becomes a difficult task.

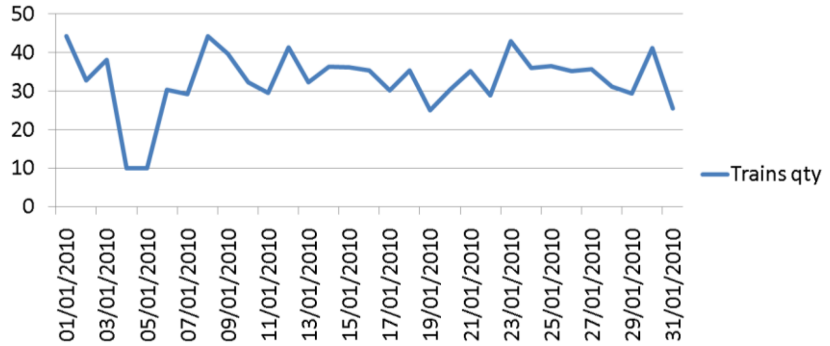


Fig. 2.22. Analyst typical request: the number of trains per day on a particular route section calculated from the schedule's data

The above facts require the development of a task of the data transformation, which consists of the transition from the interval form of data to the point one. Solution to the problem should allow to build the simple sentences in SQL language. The method of transformation of interval data, developed by the author, is presented in point 3.5 and in [26].

Next, there is an examination of the problems of processing the *stored historical data* in the systems of generating data for analysis on the transport associated with the need to interact with multiversion data of the temporal objects. Preparing data for analysis and processing of the primary source of data – *stored historical data* – often requires interaction with other sources, for example, with different directories, codifier and classifiers, many of which are temporal. In this case, the current version of connected by the key data, it is necessary to calculate on the basis of the time dimension of the primary source of the data. An example of this is the system of preparation the data for the passenger traffic analysis (APFIS) on the Latvian Railway, described in Chapter 1 and in [1, 10, 14], where the names of stations, the partition of the rail network on the line and sites, fares and passenger transportation, etc. are presented as the temporal multilateral version data. The search of a valid version, along with other calculations (aggregations, relationships between tables, etc.) complicated the whole process of preparing the required data set incredibly. Such complexity in the data processing exists in other systems on railway (for example, many freight systems). Using a common approach to writing SQL queries to perform similar calculations is rather time-consuming. Traditionally, this problem can be solved in several ways:

- refusal to navigate through the time dimension associated multilateral objects (using static values);
- construction of complex systems in which the usage of SQL queries is combined with additional software processing.

The solution of the problem is seen in the development of:

- simple and convenient means of calculating the valid version of temporal objects that could be used in SQL queries, along with other operators (see the realization of the author in point 3.4 and in [7, 25]);

- mechanism for setting the system to the desired time slices and thus provision the ability to perform "non-temporal" SQL queries, but with temporal functionality as a result (see the realization of the author in point 3.2).

2.4.8. Organization of the Low-Level Access to the Temporal Data

Nowadays, information systems on transport are being implemented on the ready data management platform rather than develop methods of access independently. An example of this trend is the Express system, which for various reasons (see Chapter 1) in its new version of the Express-3 moves away from using their own data management and moves to the RDBMS IBM DB2 UDB. However, the knowledge of low-level methods for accessing data in the selected DBMS, how physically the data are stored in external memory, the presentation of the parameters of this storage and appropriate access methods is necessary for the effective management of temporal data and the databases design with a given performance.

A set of operative applications in the transport area generates a large amounts of data both in the form of temporal streams, and in the form of historical data (see Chapter 1, the system *APFIS*). Temporal aggregation is the dominant operation in the analysis of these types of temporal data. However, it is also a very costly operation in terms of consumption of computing resources. A number of research [83–89] is devoted to the topic of effective temporal aggregation. They study the problem of computing temporal aggregates along the data streams. A set of proposed approaches is divided into two groups:

- approaches that compute a temporal aggregate at a time when it was requested (usually moving through the related data);
- approaches that support specialized aggregate index [88, 89].

In later studies, an approach when aggregates are pre-calculated and stored in a specialized index dynamically. This leads to more efficient use of disk space, since aggregating index takes less space than the actual data, and faster query execution, as the ad-hoc unit is calculated by simple moving to a branch in the index. All these approaches are used at the system level.

As an example of the special requirements of operational challenges to the methods of low-level access, we give the IS schedule of passenger trains SAR, which is very heavily used via the Internet. Statistics show that the number of hits to the system in one minute is 100. The situation is complicated by a large amount of data being processed. As queries to the system occur in the on-line mode, a quick delivery of results is extremely important. Without the special organization of schedule's temporal data is impossible to ensure a rapid response of the system (<1seconds).

In the area of storing and organizing of an access to relational system at the physical level, there is a lot of research and successful implementation for the specific DBMS. They are: various receptions of the optimization data storage in external memory (fragmentation, clusterization, pages table dimensions management, extents and databases files) different methods of indexing (B-tree, hashing, bitmap), data compression etc. However, these methods are general in nature and do not take into account the organization peculiarities of temporal data.

In the literature, there are works devoted to research in the organization access to temporal data on the physical level (eg, [92]). Different methods of indexing and placement of temporal data are investigated and optimized. The main criteria for comparison are the performance, disk space consumption and processing of data updates (time and other data modification costs). Such criteria as page numbering of the index, the ability to cluster related data together, the ability to separate effectively old from current data are of great importance too.

However, despite the well-conceived matter no clear guidance on placement of temporal data on the physical level by means of commercial DBMS have been worked out so far. The approach used by the author is demonstrated in the examples in point 4.1.

2.4.9. Ensuring the Security of the Temporal Data

The issue of security of data in relational systems has found its reflexion in many research works and is being successfully solved in the existing IS. A number of commercial implementations and long experience of their use in known databases is the proof of it. Such roles as the roles of an administrator of the database server, database administrator, privileged database users, as well as the capabilities and tools to control the rights to perform routine operations, such as altering the structure of database objects, select, update, delete and insert records, are defined and standardized.

However, the temporal IS require the use of more complex patterns of data protection. Security feature of temporal data is that data from different time intervals are logically located in one place – in a single relational table. Differentiation of user access to data of different periods, i.e. to a subset of tuples of a relation is not included in a set of standard tools of limiting access of the RDBMS. Such opportunities are not in the standard language for relational databases, SQL. It is worth being noted that in the specialized temporal database this issue also receives little attention. The above mentioned facts explain the relevance of research on information security insurance in the temporal database.

The author investigated the applicability for the protection of temporal data from two classical approaches to access control: discretionary and mandatory. The results of these investigations are presented in papers [3, 5, 6].

Discretionary access control allows to restrict an access between named subjects and named subjects [107, 112]. It is included in the most commercial relational products and meet the most information systems requirements, but it has its limitations in the flexibility of management.

An example of a discretionary approach is the SQL command control privileges GRANT. The syntax of the GRANT command is as follows:

Privileges to the object access appointment operator	**GRANT**	
Access privileges	**(INSERT, DELETE, SELECT, UPDATE) (*column-name*)**	*Access for select and modification can be limited by several columns*
Type of the object	**ON TABLE**	
Name of the object	***table-name view-name***	
	TO (USER, GROUP)	
Username or group names	***authorization-name***	
Privileges delegation	**WITH GRANT OPTION**	

The command syntax shows that the access privileges for users (*authorization-name*) are associated with the named information structures (*table-name, column-name, view-name*), which contain the data, and the connection with the data is not available. Security system is not aware of their semantics, as well as a time dimension. Privileges exist separately from the data itself, which, from a security standpoint, are a kind of impersonal.

Mandatory access control marks an access of subjects to data objects based on the label of the confidentiality of information contained in objects, and the official authorization of subjects to access information of such a level of confidentiality [107, 112]. For mandatory approach rigid adherence to the recommendations of the "Orange Book" of U.S. Department of Defense and data security model Bell La Padule is remarkable [108]. Mandatory access control involves compulsory and rather specific security data administration. Such an approach is useful for military and other problems associated with top-secret data. And it is specially designed for them. Famous implementations are designed in the form of closed systems, which impose severe restrictions on portability. This explains the limited application of the mandate approach in the real IS on railway transport, whose security policy does not fit into its model. In addition, for most applications the need for such explicit data administration is a necessary condition.

Fig. 2.23 shows the key difference between these two approaches: the discretionary protection provides an access to the data up to table column, and mandatory - to the record (in special cases – up to a table cell). Between columns and records there is an important difference: the column is a part of the table structure, static by its nature, and the record is a piece of data, dynamic by its nature. Thus, in the first case, protection is on the structure, while the second – on the measures present in the data. In the mandatory approach, as well as discretionary, the security system is a separate element of the application task. Mandatory access control in this respect is preferable to discretionary: as was mentioned earlier, it allows you to restrict an access to specific objects (records) rather than to the structures in which objects are stored, as it happens when a discretionary approach is used. Still in the mandatory approach there is no connection with the temporal data dimension, and consequently, the security system is not integrated with the application task.

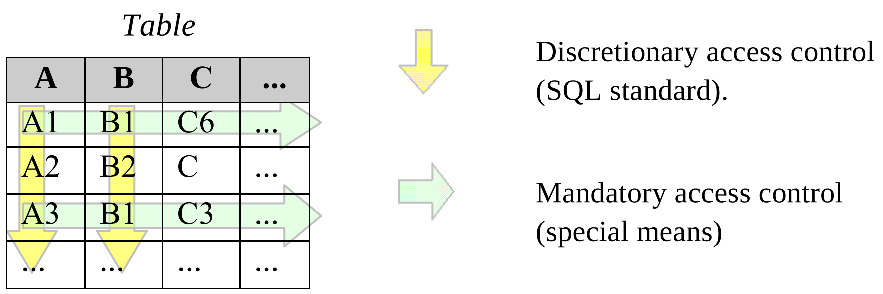


Fig. 2.23. Types of data access control

Thus application of classical approaches to the access control does not solve the problem and it requires finding new ways to solve the considered task better meeting the security demands of IS on transport.

The simplest implementations of the by-record access to select are described in a number of sources, such as [112]. This access is based on the use of views. The problem of protecting the record from the changes using standard relational DBMS has not been solved so far. However, there are few implementations of this access control, discussed below.

In DBMS Oracle, in the version 8i a concept of *Virtual Private Database* has appeared. *A Virtual Private Database* offers *Fine-Grained Access Control (FGAC)* for secure separation of data. This ensures that users only have access to data that pertains to them. Using this option, one could even store multiple companies' data within the same schema, without them knowing about it. That is, line-by-line access is implemented, but the criteria for admission to the record is the owner of record, not the time dimension, as it is necessary for the temporal database.

Another mechanism *Oracle Label Security* (formerly called *Trusted Oracle MLS RDBMS*) uses the *Virtual Private Database* feature of Oracle 8i to implement row level security. Access to rows is restricted according to a user's security sensitivity tag or label. *Oracle Label Security* is configured, controlled and managed from the *Policy Manager*, an *Enterprise Manager-based GUI* utility [105].

Unfortunately, both implementations mentioned above are specific to DBMS Oracle and the author is not aware of their applications for the temporal database

The above facts explain the relevance of research on database control access insurance in a relational environment. The development of technology of access restriction to the record in temporal database suggests a number of issues, we present some of them:

- basing on what factors an access to the record should be granted;
- what a logical organization of the data protection should be like;
- various options for the data protection implementation;
- preservation of database productivity in the condition of implementation of the temporal data security.

In point 1.5.4 it is shown that a lot of factors influence an access to temporal data, so the security system that protects data with regard to their time dimension should be integrated with the operating system of database security as time dimension is not the only criteria of data which limits an access to the data of transport IS. Time dimension along with other attributes of the data contains information for access control and therefore the permission to an access should depend on the values of a set of attributes. For example, two users are involved in the scheduling of trains; one is planning the schedule for next season, while the other is adjusting the current schedule. The first user has no right to create versions of the schedule affecting the current quarter and the second, on the contrary, has the right to manipulate the schedule of this period only. The other two users have similar rights, but their range of authority is limited to diesel trains and different directions. Thus, it is necessary to create such a method of data protection that allows you to restrict an access based on ANY properties or dimensions present there: temporal, spatial or geographical, administrative and other. Application of this method along with the access control system used in relational DBMS, constitute an integrated security system.

The development of requirements for an integrated security system is what the work of the author devoted to [13]. We present the main ones:

- the level of access to data is determined by the data itself and depends on their semantics, i.e. the data themselves are involved in the process of limiting access to them. In other words, the access control system can operate as an integrated part of IS and follow the real rules of conduct of subjects and objects, which means that the functioning of security system without direct administrative intervention is possible;

- the system should allow the realization of any security policy, any of its rules in any combination;
- the user can have different access to the same set of data, depending on the requested operation, That is, a data set of the same information object (relational tables) accessible for select, deletion and modification can vary;
- the system must have the possibility of logging of user actions;
- mechanisms of the access limitation must be implemented at the database level and not the application;
- application of the developed method of access control should be possible on most modern relational systems.

We should note that the implementation of basic temporal principles in IS in some way increases its information security. For example, when using the axis of the transaction time (temporal data of Rollback or Bitemporal form) the temporal database automatically saves all previous states of information system's objects and provides access to them at any time. Thus, there is similarity of data backup.

The task of developing a method of restricting access is complicated by the fact that in the temporal systems there are several forms of temporal relations, and methods of ensuring information security of each form can be different.

Despite the importance of issues related to information security in temporal systems in transport, existing investigations have not paid enough attention to their solutions.

In some studies the problem of ensuring security and temporal database technology is considered from different angle, namely the use of the latter to implement or improve security. For example, the research [78] investigates the applicability of the parametric model for temporal data to query multilevel security data. In a multilevel security database there are multiple beliefs about a given real world object. The ability of a database model to accommodate multiple beliefs is termed polyinstantiation in the multilevel security literature. Authors use the fact that in an abstract sense polyinstantiation is a priori present in all models for temporal and spatial databases. Other research [80] presents a discretionary access control model in which authorizations contain temporal intervals of validity. An authorization is automatically revoked when the associated temporal interval expires. The usage of the presented model allows to implement the user's access to a relation only for a day or a week.

Conclusions of the Second Chapter

1. The analysis of the known approaches to the implementation of the DBMS has shown that there is still no commercial sale of the temporal DBMS that allows solving the entire complex of the problems associated with the employment of the temporal data on the railway transport. The existing temporal DBMS and settings are not flexible enough for substantialization of the methods totality. Taking into account the above mentioned facts, as well as the enterprises investments in the relational systems and the accumulated experience of their employment, it is offered to carry out the development of the temporal systems in the relation environment.

2. Despite the considerable amount of published research in the field of the temporal systems, the procedures on their standardization have not been completed yet. There is no unified approach to the temporal data modelling, arrangement and processing. Correspondingly the research under consideration devotes a special attention to the theoretical issues elaboration.

3. The analysis of the temporal objects employment on the railway has revealed the set of the models and methods, subjected to the application in the temporal systems, and to the determination of their elaboration degree. The variety of the exposed models and methods has been structured and classified; there is the determination of the research principal tasks, and the determination of the specific peculiarities of the transport facilities and information systems, including:

- development of the temporal objects data representation in the relational environment;
- supporting the specific mode of the data introduction and the versions alteration;
- designing the modification operations logics and data reading (the choice of the valid version of the object);
- implementation of supporting the integrity of the temporal relations of different forms;
- support of the periodicity and the special calendar;
- transformation of the temporal data from the interval form into the point form;
- simplification of the interaction with the temporal data in the process of the data preparation for analytical processing, employing the procedure of setting for a specified time slice
- ensuring the information security of the temporal data.

Chapter 3. TEMPORAL MODELS AND METHODS DEVELOPMENT AND IMPLEMENTATION

3.1. Temporal Data Modeling in the Relation Environment

Taking into consideration both the huge investments in the relation databases, which are the basis for functioning of the majority information systems on the railway and the accumulated practical experience, the author suggests employing the relation environment for the temporal databases implementation with developing the corresponding temporal framework; the issue of transition to the specialized temporal databases management systems is beyond the tasks of the dissertation in review (see point 2.2.3). Solving the problem of employing the relation environment proceeds in two directions: the design of the temporal data models in the relation environment; and development of the temporal logic with the relation facilities application.

In the context of the first direction there suggested the following models designed by the author: *the temporal data model with the object abstract identifier, the Historical-Overlapped model, the model of the temporal object, involving periodicity and a special calendar.*

In the context of the second direction the author suggests the various components of the temporal frameworks, extending the databases management with the temporal opportunities and implementing the temporal logics within the relation model limit.

3.1.1. Representational Model of the Temporal Data with the Object Abstract Identifier

In research [4] the author considers the various problems connected with the incorrect temporal data modeling, such as:

- the "uncertainty" of the dependent tuples of the child relations from the parent relation, containing the multi-version objects;
- in case of the object new version appearance, the uncertainty referred to the dependent tuples arises. The certain tuples have to change the existing reference to the new version, but some of tuples have to refer to the previous reference;
- the complicated logic of implementing the dependences alteration;
- in the course of operations DELETE or UPDATE in the parent table it is rather difficult to suppress these operations without issuing the error to the client, producing these operations;
- the operation DELETE cannot be produced due to the probable existence of the references to this tuple.

The open model with the object abstract identifier (AOID model) [4], providing the object multi-version nature, is suggested as an optimal model of the temporal data in the relation environment not furnished with the mentioned shortcomings. In general case the suggested model has the view presented in Fig. 3.1. This model describes the life cycle of the modeling object via the life cycles of the properties of this object, and these properties are determined in various relations. This approach slightly exaggerates the data structure, but makes this system open for alterations and reduces the requirements to the disk space for storage of the set of the objects versions. Every of these relations can be both temporal and non-temporal, and this fact depends on the object property, represented by this relation. As well, depending on the task condition, the temporal relations can correspond to the various temporal forms: Rollback, Historical, Bitemporal, with the support of the periodicity, and so on.

There is an examination of the principal model peculiarities.

The first parent relation characterizes the abstract non-changeable identifier of the object OID – unique for all its versions.

The object properties (dynamic and static) are described in the separate dependent relations that are not, as a rule, parent relations for any of the relations. One parent relation with OID can possess not more than child relation, characterizing the properties of this object. These properties are divided by different relations in dependence with the frequency and the time of their alteration. The properties, changing without any dependence from each other, are stored in the separate relations (the requirement of TNF), one relation can associate only the properties which are changing simultaneously.

The interaction with the model data and the integrity support are provided by the relational tools, as well as the temporal framework, obtaining different implementation, depending on the employed temporal forms. The common ones are the following:

1. The rule of OID formation: in case of the new object appearance the system automatically generates the new abstract identifier of the object, which is unique in the frameworks of this relation. OID cannot be introduced by force and it cannot be removed by force, it is arranged only in the moment of producing the operation of inserting (INSERT);

2. The guaranteed existence of a single active version of the object at any moment of time.

The other properties are inherited from the temporal forms, employed in the model. For example, the active version of the object property for the Rollback form is the tuple with the non-determined time of ending of the operation of the version (the value NULL). In all other tuples, characterizing the chronology of this object states, this attribute is to possess the value, corresponding to the moment of time in the past. But the active version for the Historical form is the tuple, the operation time of which embraces the current moment as well.

The point 3.2 is devoted to the issues of implementing the temporal frameworks, providing the interaction with the temporal data of the model AOID, as well as providing its data integrity and safety.

The mathematical description of the model AOID is presented further.

3.1.1.1. The Mathematical Description of the Model

It is assumed that the object *O* of the temporal system is characterized by the unique identifier *OID* and the set of properties *A* and can be represented in the terms of:

$$O = (OID, A) \ .$$

The set of the object properties $A = \{a_1, a_2, \dots, a_m\}$ is divided into two subsets: the set of the static properties A^s, which are not subjected to the time changes, and the set of the dynamic properties A^d, changing in time, moreover:

$$A = A^s \bigcup A^d ;\quad A^s \bigcap A^d = \varnothing ;\quad A^s = \{a_1^s, a_2^s, \ldots, a_k^s\} ;\ A^d = \{a_1^d, a_2^d, \ldots, a_n^d\},$$

where a_i^s and a_i^d are the static and dynamic properties respectively; *k* and *n* are the amount of the static and dynamic properties respectively; *m=n+k* are the general amount of the object *O* properties.

As the attribute values in the relation databases have to be atomic, it is not enough to introduce only one relation for presenting the numerous static and dynamic properties. The object *O* is presented as the interconnected relation scheme:

$$O = (R^{OID}, R_1^s, R_2^s, \ldots, R_l^s, R_1^d, R_2^d, \ldots, R_q^d),$$

where $R^{OID} = R^{OID}(OID, A_0^s)$ is the parent relation, describing the object abstract identifier *OID* and comprising the aggregate of the object static atomic attributes $A_0^s, A_0^s \subseteq A^s$; *OID* is the primary key of the relation R^{OID};

$R_1^s, R_2^s, \ldots, R_l^s$ are the child relations, describing the object A^s static properties ($l \le k$) and presented as the scheme:

$R_i^s = R_i^s(OID, A_i^s), i = 1,2,...,l$, where A_i^s is the subset of the object static properties, logically aggregated in one relation (on the principle of storage optimization, the necessity of the correspondence to the first normal form, information and functional separating and other criteria which are taken into consideration by the database developer), $A_i^s \subseteq A^s$. In case the relation R_i^s describes the object atomic property, the *OID* is the key, and if there are the multi-valued properties, the key comprises the additional attributes;

$R_1^d, R_2^d \ldots, R_q^d$ are the child relations, describing the discrete ranging in time dynamic attributes A^d ($q \le n$) and presented in terms:

$$R_i^d = R_i^d(OID, A_i^d, T_i), i = 1,2,...,q,$$

where A_i^d is the subset of the discrete dynamic attributes, logically and on the basis of the simultaneous values changing aggregated in one relation, $A_i^d \subseteq A^d$. By employing such attributes separating for *q* relations the temporal normal form is achieved; T_i is the time attributes vector, which can present valid time (VT) or transaction time (TT), or both of them simultaneously, depending on the relation temporal form: $T_i \subseteq T$, where $T = \{t_s^{VT}, t_e^{VT}, t_s^{TT}, t_e^{TT}\}$ is the set of time dimensions attributes. In case the relation R_i^d describes the object atomic properties the relation key can be in the form of (OID, t_s^{TT}) or (OID, t_s^{VT}), depending on the temporal form. In the case of the multi-variant properties the key comprises the additional attributes.

The general view of the temporal data model with the abstract identifier, presented in the form of ER scheme, is shown on Fig. 3.1.

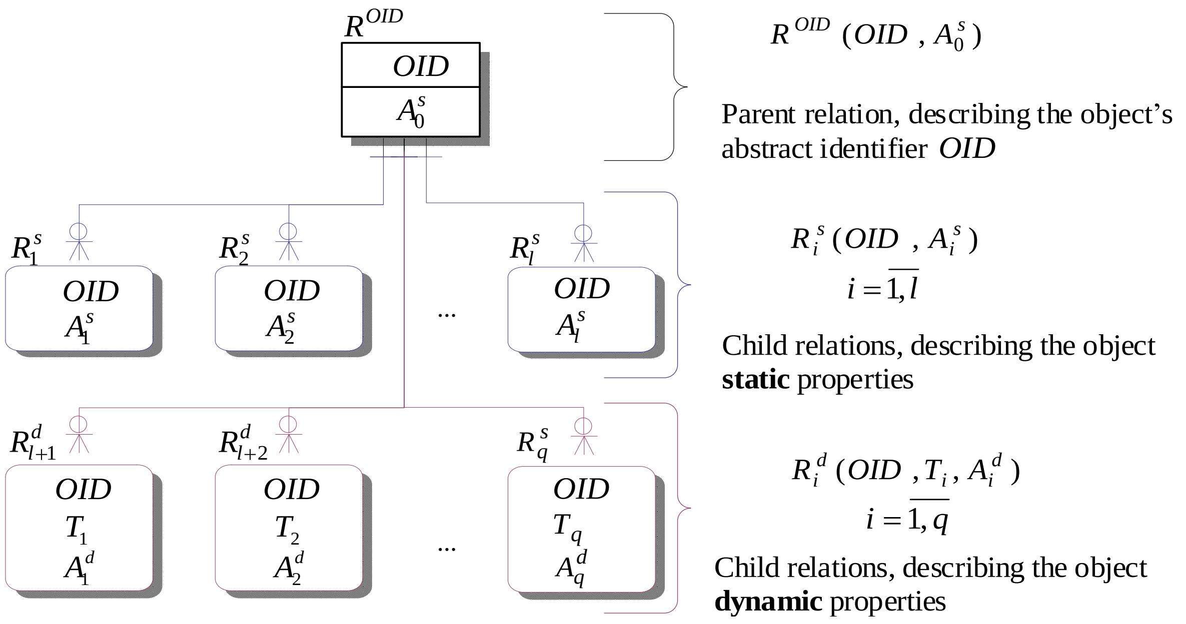


Fig. 3.1. Data model with the Object Abstract Identifier in the form of ER scheme

The implementation of the database of the Referenced Data System can serve as an example for illustrating the model operating (see point 3.1.1.2).

3.1.1.2. Example of Implementing the Database of the Referenced Data System

The database of the Referenced Data System (RDS, Reference Data System) plays the fundamental part for the numerous information systems on the transport, that is why the great attention is paid to the designing the database of the Referenced Data System in the process of the contemporary information systems development [38]. In the course of developing the model of the centralized Referenced Data System on the Latvian Railway, the procedure of which was described in the research [4], the author of this thesis and the co-authors of this publication faced the necessity to endue this model with the properties of the temporal database. The Referenced Data System on the Latvian Railway is presented in the form of the railway classifiers and codifiers system (for example, the freight codes, the stations codes and names, and so on); these classifiers and codifiers are subjected to the changes from time to time. The history of their changes is necessary for the analysis of indicators, belonging to the different periods of time.

There is the examination of the segment of ER-model of the Referenced Data System, presenting the structure of the database of the stations classifier [1], employed on the Latvian Railway. This model describes the object lifespan via the lifespans of all its properties, determined in different relations and possessing the temporal attributes DATE_START (the moment of the lifespan beginning) and DATE_STOP (the moment of the lifespan ending).

The first parent relation RDS.K_STATION describes the abstract object "station", which is characterized by the identifier OID, unique in the frameworks of the relation – the uniformed one for all versions. It is arranged in the course of the object creation and then it is preserved invariable during all its "life" and is available in the system after its logical removal. The physical removal of the object is not allowed. In the relation RDS.K_STATION only the operation INSERT is permitted. The removal of the object on the logical level is degenerated into the alteration of the object properties responsible for the object activity.

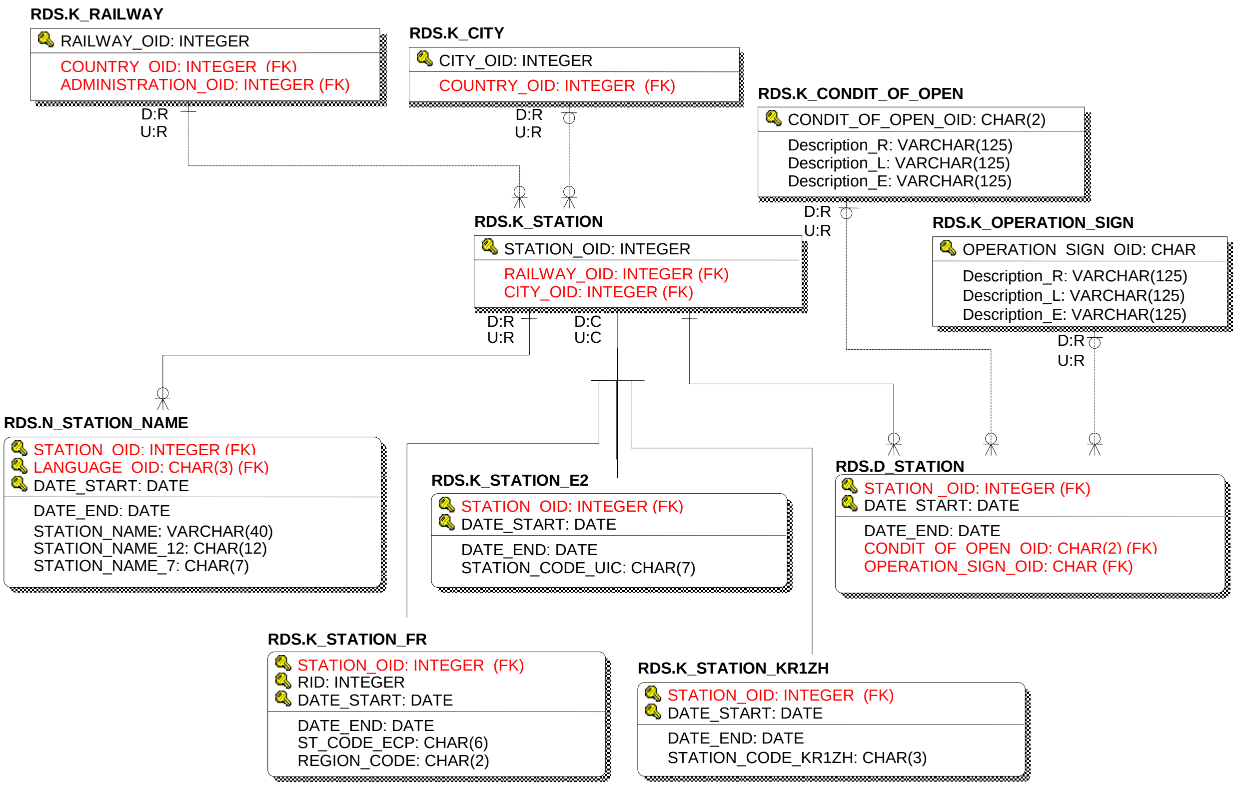


Fig. 3.2. The segment of the data model RDS – the Stations Classifier

The object properties are described in the separate dependent relations. Any of the parent relations with OID possess more than one child relation, characterizing the object properties. These properties are distributed among the various relations in accordance with the maxims of correspondence TNF.

The presented example describes all the temporal relations corresponding to the forms Rollback or Historical-Classic, in which the periods of life of the objects do not overlap. The attribute DATE_START is introduced in the primary key of the dependent relations, characterizing the object properties; the attribute DATE_START describes the time of beginning the object properties actualization and in cooperation with the foreign key (FK) identify unambiguously the version of the object properties. Every version of the object state is characterized with two temporal attributes DATE_START and DATE_END, and these attributes in cooperation make the period of the version actuality. It is worth noting that the attribute DATE_END is cable of obtaining the non-determined value (NULL), but the temporal frameworks controls the fact that object has to possess only one tuple with the non-determined values in the relation characterizing its properties. The example of the implementation of the above described temporal logic in the Referenced Data System is submitted for consideration in the point 3.2.2.

3.1.2. Relation Temporal Form "Historical-Overlapped"

The certain tasks on transport are characterized by the peculiar requirements for the life cycle of the temporal objects. It is important feature for them that the lifespan is capable of being interrupted (discrete) and come into operation repeatedly. The system of registration of the passenger trains schedule can serve as an example of the task obtaining such types of requirements [7] (see point 1.5.2). The temporal object of this system is the train schedule itself. One and the same version of the schedule can be active on the several lifespans, in other words, during the definite period of time it is capable of falling out of the active state for several times, and abandon to the “shadow zone” and then to return back. For example, the principal schedule of the railway traffic might be exchanged by another schedule, adjusted to the remedial operations, for several days, and after the end of the repair works the original schedule comes into operation again. The well-known sources [51, 52] take into consideration only the cases when every version of the object exists only during one continuous period, specified for this version and not interrupted.

The interruption of the lifespan can also appear in the tasks where the management of the temporal object is connected with the calendar and the periodicity (the holidays, the days of week, the evenness of the day, the working shifts and so on). These tasks are considered further in the point 3.3.

As it has been demonstrated in the point 2.4.3, the usage of such known temporal relations forms as Snapshot, Historical and Rollback is not efficient for the description of the temporal objects with the mentioned peculiarities; it is seen not only through the wasteful disk space employment but also through the excessive risk of breaking integrity of the temporal data in case of their changing. Under the condition of the steady continuous increase of the data volumes and the advanced requirements for the data integrity and availability in the transport systems the listed shortcomings become crucial.

The author suggests the special temporal form without the above mentioned drawbacks. It is the relation form with the overlapping lifespans Historical-Overlapped (Fig. 3.3, the dotted line), considered in the author’s works [7, 19, 20, 21, 22]. It comprises the principal properties of the classic temporal relations forms, exactly the possibility of existence of the several versions of the same object, becoming the active one consecutively. At the same time the investigated form permits the object versions to become active more than one time.

The peculiarity of this approach consists in the fact that the object state in the process of alteration is capable of not losing its actuality, it refers to the past and to the future, and only for a short period of time the object state is exchanged with other version of the state. In this case the concept of the number of the object version as the facility of the determiner of the actuality of the object state loses its meaning considering the fact that at one and the same time there can exist more than one tuple, describing the various properties of one and the same object with the open period of the version operation, that is not permitted in the forms Rollback and Historical.

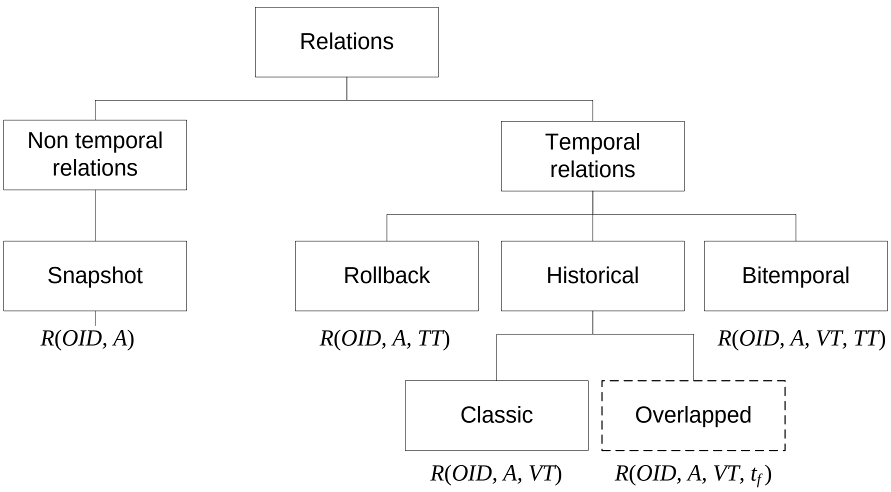


Fig. 3.3. Form Historical-Overlapped in the general classification of the relation forms of the temporal databases

The temporal relation scheme with the overlapping lifespans Historical-Overlapped is performed in the terms:

$$R(OID, A, VT, t_f),$$

where t_f is the time of the transaction fixation (corresponds to the attribute t_s^{TT} considered above, see point 2.1.4). The transaction fixation time is generated in the moment of entering or changing the record in the database. The active version of the object *OID* is computed on the basis of this attribute values.

The existence of two and bigger number of versions with the overlapping time ranges does not contradict the fact of the relation Historical-Overlapped integrity. It means that the relation *R* can comprise two tuples *r* and *p* with the following conditions:

$$(r.OID = p.OID) \wedge (r.VT \cap p.VT \neq \varnothing). \tag{6}$$

The model Historical-Overlapped stipulates the operations over the objects the period of life of which is discovered not only in the present or in the past, but also in the future. In case the query on the schedule of the train traffic, which will be for example in a month time, enters the database, the system will deliver in response the time slice active for the required moment of time.

The logic of interaction with the relation of the Historical-Overlapped form is explained on the example of the train traffic schedule, described by the diagram presented in Fig. 3.4.

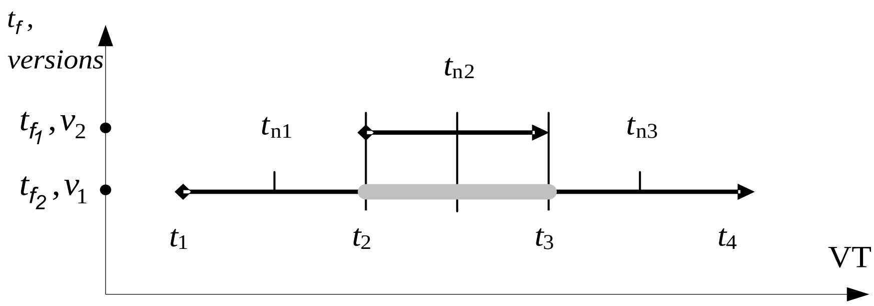


Fig. 3.4. Overlapping the life cycles of the object

It is assumed that t_n means the time of observation. The train schedule (the version v_1) is assigned from t_1 to t_4. Later for the period of time from t_2 to t_3 the schedule alteration (the version v_2) is introduced. In case of the query on the train traffic at the moment of time t_{n1}, there will be delivered the version of the schedule v_1, and at the moment t_{n2} – the schedule version v_2, and at the moment t_{n3} – again the version of the schedule v_1. It is important that only two tuples exist in this relation: tuples with the schedule versions v_1 and v_2. The active version of the schedule is calculated on the basis of t_f. The version that has been introduced the latest has the priority in case there is a choice comprising several versions.

The logic controlling the delivery of the active state of the object for the required time interval can be produced by the temporal framework. Furthermore, the user (the developer) does not have any necessity to understand the data structure, supporting the multi-version nature of the object and arrange the complicated query, calculating the active version. The temporal framework, responsible for the active version determination, is considered in the point 3.4.

The efficiency of implementing the suggested temporal form Historical-Overlapped is examined from the position of providing the integrity and accessibility of the data (the indicators, suggested in the point 2.4.1).

The task of the schedule storing of the trains departure from the station is considered. The schedule of the traffic of the majority trains possesses the prolonged nature but as a result of the various situations there are possible pre-planned and unplanned changes of the departure time. The task is the following one: at any moment of time on any day in the past or in the future to determine the active for this day departure time of the specified train from the station under observation.

Fig. 3.5 presents the example of the scenario of altering the train schedule. This example is considered in the details. The core fundamental seasonal schedule *(Base)* is assigned for the train number 123P for the summer period from 01.05 to 30.08. Afterwards in the connection with the necessity of conducting the repair works for the period from 03.07 to 15.07 the train schedule was changed (*Repair works*). And on the City Anniversary 06.07 the special timetable was assigned (*City anniversary*).

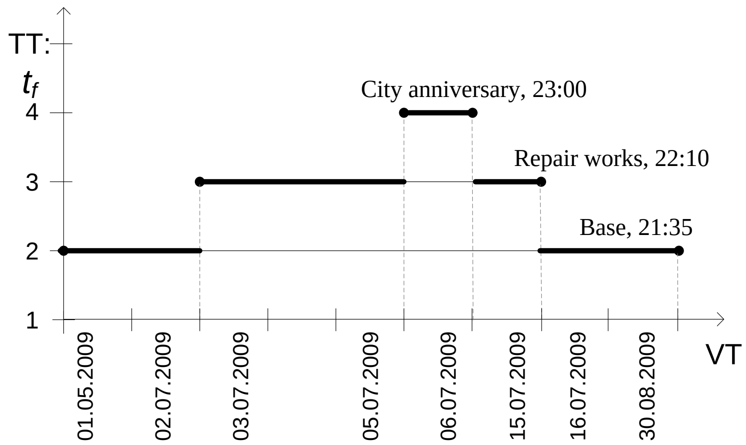


Fig. 3.5. Example of overlapping of object lifespans in the train schedule

Fig. 3.6 demonstrates two ways of performing the information of the above example: in classical relation Historical-Classic with non-overlapping lifespans and in the relation Historical-Overlapped with the overlapping lifespans. The attributes of the valid time t_s^{VT} and t_e^{VT} define respectively the beginning and the end of the period of the schedule validity; the attribute t_f shows the transaction time.

Historical-Classic

OID Train	Departure time	t_s^{VT}	t_e^{VT}
123P	21:35	01.05.2009	02.07.2009
123P	22:10	03.07.2009	05.07.2009
123P	23:00	06.07.2009	06.07.2009
123P	22:10	07.07.2009	15.07.2009
123P	21:35	16.07.2009	30.08.2009

Historical-Overlapped

OID Train	Departure time	t_s^{VT}	t_e^{VT}	t_f
123P	21:35	01.05.2009	30.08.2009	1
123P	22:10	03.07.2009	15.07.2009	2
123P	23:00	06.07.2009	06.07.2009	3

Fig. 3.6. Relation "Schedule of the train departure from the station" in the forms Historical-Classic and Historical-Overlapped

The advantages and the shortcomings of employing the temporal form of the relation Historical-Overlapped are considered relatively to the Historical-Classic form.

In the considered example the relation Historical-Overlapped *is more economical* from the point of view of the amount of tuples. It is not difficult to define that, in case the number of tuples (versions) in relation Historical-Overlapped equals g, the minimal number of tuples in the relation of the Historical-Classic form, reflecting the same information, will be in the diapason from g to $2g$-1 depending on the situation. The greater amount of the overlapped factors and consequently the greater number of cases of coming into operation of one and the same version, the greater is the benefit of employment the Historical-Overlapped form.

It also should be mentioned that the relation Historical-Overlapped expresses the semantics of the task without introduction of the additional attributes. The benefit gained in the process of the semantics reporting is achieved due to the fact that the tuple in the relation Historical-Overlapped represents additionally the factor of the exchange appearance, but not only the order of the schedules alteration as it happens in the relation Historical-Classic.

The concept of the uniqueness of the object version in the Historical-Classic form is absent. Every tuple represents the period when the version is active, corresponding to the time of operation VT, and there can be a lot of such periods of every version. In case there two tuples in the Historical-Classic form, and these tuples contain the equal values of the parameter under consideration (in the examined instance they are the schedule) it is impossible to state if both of these values belong to the same version without a special registration.

The concept of the object version in the Historical-Overlapped form is identical to the concept of the relation tuple. In other words, every version of the object represents the tuple, and vice versa. However the time of operation VT, comprised in the tuple, does not mean the time when the version is active. By the virtue of the fact that the versions are overlapping, it is necessary to calculate the time when the versions are active.

There is an assumption that $V=\{v_1, v_2, \dots , v_g\}$ is the set of the unique versions of the object, and g is the number of these versions, $K=\{k_1, k_2, \dots , k_h\}$ are the periods of time when they are active, and h is the number of such periods, in this case $g \leq h$. Fig. 3.7 demonstrates the interconnection of the objects versions and the periods when they are active. There presented two versions v_1 and v_2 overlapping in the time of their operation, and there also presented the periods when they are active: k_1 and k_3 are the periods when the version v_1 is active, and k_2 is the period when the version v_2 is active. The components of the set V correspond to the tuples in the relation of the Historical-Overlapped form, and the components of the set K – to the tuples in the relation of the Historical-Classic form.

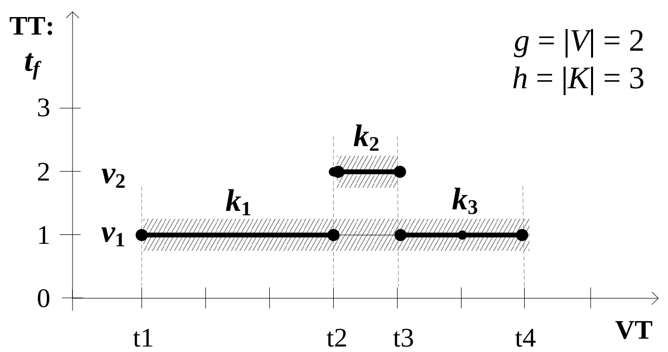


Fig. 3.7. Versions of the objects and the periods of their activity

There considered the support of *the integrity in the process of the data correction*. The principle of managing the versions of the objects in the relations Historical-Overlapped differs from the management in the relation Historical-Classic. The distinctive peculiarity of the Historical-Overlapped form lies in the fact that the procedure of processing the values of the temporal attributes of the adjacent (the previous and / or the following) object versions do not precede the fact of appearing the new object version or the fact of altering the existing object version. On the contrary, the creation of the new object version or the modification of the existing one in the relation Historical-Classic are connected with the processing the boarders of the lifespans of other versions, in case the lifespans of theses versions are overlapped fully or partially as a result of the modification (in the majority of cases it refers to the previous and to the following versions of the object).

There is the comparison of the procedures of adding the new version of the object for these two relation forms. Three situations can appear in the process of the versions adding:

1. The new version does not overlap any of the existing ones in valid time (see Fig. 3.8);
2. The lifespan of the new version validity is completely within the lifespan of another version (see Fig. 3.9);
3. The lifespan of the new version covers the lifespans of other versions (see Fig. 3.10);

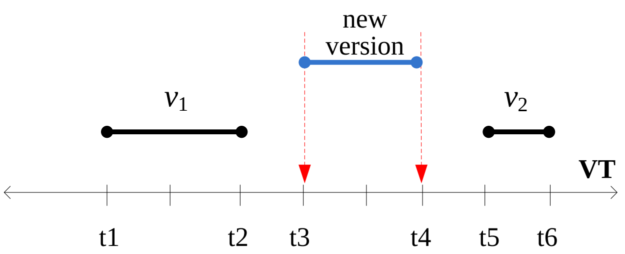


Fig. 3.8. New version does not overlap any of the existing ones in valid time

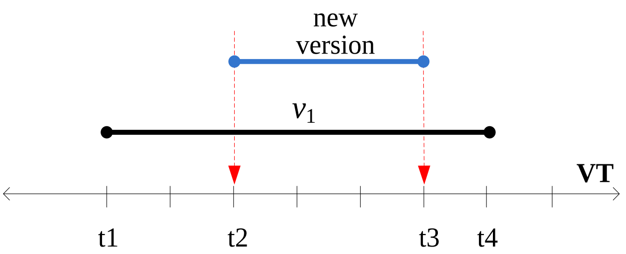


Fig. 3.9. Lifespan of the new version is completely within the lifespan of another version

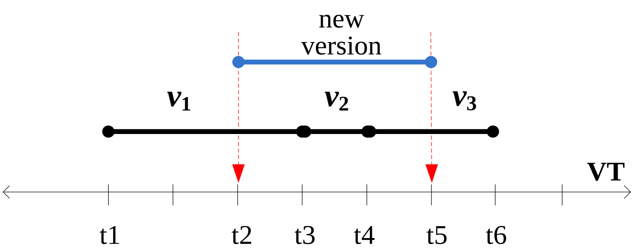


Fig. 3.10. Lifespan of the new version covers the lifespans of other versions

The results of the comparison of the procedures providing the integrity after adding the new version of the object for the relations Historical-Classic and Historical-Overlapped are presented in Table 4. They are the activities that are to happen obligatory after inserting the new tuple in the temporal relation of the above mentioned forms.

Table 4. Additional operations after adding the new version of the object

Situation \ Temporal Form	Historical-Classic	Historical-Overlapped
1. The new version does not overlap any of existing ones in the time of activity	absent	absent
2. The diapason of the operation of the new version is completely within the lifespan of another version	1 tuple is changed and 1 new tuple is added	absent
3. The diapason of the operation of the new version covers the lifespans of other versions	1-2 tuples are changed, *d* tuples are removed (*d* is the amount of absorption)	absent

There is the comparison of the *availability* factors occurring in the course of the data sampling from the relations of the forms Historical-Classic and Historical-Overlapped, active on the specified day *t*.

As it has been already marked, the periods of the versions activity are not stored in the Historical-Overlapped form in an explicit form, it is necessary to calculate them on the basis of the fixation time and the periods of operation of other versions. Accordingly, the task of the identification of the object version, active at the moment *t* in the Historical-Overlapped form is connected with more complicated access to the data, as far as, besides the checking the fact of occurrence *t* in the range of the operation of the object version $[t_s^{VT}, t_e^{VT}]$, it is necessary to choose “the youngest” version of the object, applying the attribute t_f. But the period of activity in the Historical-Classic form corresponds to the presented time of operation and no additional calculations are to be performed.

For comparison the number of the tuples *g*(*t*), active for the specified day, are taken as the fundamental, as far as they are exactly the objects of additional calculation under the condition of employing the Historical-Overlapped form. In this case, if the Historical-Classic form is applied, the value *g*(*t*) will be either 1 or 0, if the object does not exist on the peculiar specified day. Under the condition of employing the Historical-Overlapped form the value of *g*(*t*) will be in the range from 0 to *g*(*t*).

Therefore it can be mentioned that the more complicated access to the data becomes the shortcoming of the Historical-Overlapped form employment, because, besides the checking the fact of occurrence in the range of the operation of the object version, it is necessary to choose “the youngest” version of the object.

The summarized results of the comparison of the forms of relations Historical-Classic and Historical-Overlapped are presented in Table 5. The procedure of comparison took into consideration two possible situations of real modeling: when one and the same version of the object becomes active not more than one time only and when one and the same version of the object is active on several lifespans.

Table 5. The results of comparison of the relation forms Historical-Classic and Historical-Overlapped

Situation / Indicator	The same object version does not become active on several lifespans: $g = h$		The same object version is active on several time lifespans: $g < h$	
	Historical-Classic	**Historical-Overlapped**	**Historical-Classic**	**Historical-Overlapped**
The size of the temporal relation (the number of tuples)	g	g	$[g + 1; 2g - 1]$	g
Integrity in case of the data alteration (the number of tuple operations when the new object version occurs)	0	0	$[0; u + d + i]$	0
Accessibility in the process of identification of the active object version at the moment t (the number of processed tuples)	$\{0, 1\}$	$\{0, 1\}$	$\{0, 1\}$	$[0; g(t)]$

The Table 5 employs the following terms:

g is the number of the object versions (see Fig. 3.7);

h is the number the version active periods, $g \leq h$ (see Fig. 3.7);

$u \in \{0, 1, 2\}$ is the number of the tuples, corresponding to the partially overlapped lifespans; these u tuples have to be changed;

$d \in \{0, 1, \ldots, h\}$ is the number of the tuples, corresponding to the entirely overlapped lifespans; these d tuples have to be eliminated or marked as non-functioning;

$i \in \{0, 1\}$ is the number of the tuples, corresponding to the life sections which overlap the new version lifespan; these i tuples have to be added, moreover

$$(i = 1) \Rightarrow (d = 0); (d > 0) \Rightarrow (i = 0);$$

$g(t)$ is the number of the object versions, which are valid at the moment t.

Consequently, the analysis of the relations of the Historical-Overlapped form, performed with accounting the criteria of integrity, privacy and accessibility, has demonstrated that they are the safest if speaking about breaking the rules of integrity under the condition of the data alteration, as well as the most economical if refer to the memory resources.

3.2. Temporal Logic Implementation within the Framework of the Relation Model

The second direction of the task of designing the temporal framework is the development of the temporal logic within the framework of the relation model. For its solution the author has suggested the enhancement of the manipulative and integral parts for the relation model. The principal tasks of this enhancement are the following:

- support of the time-based multi-version and temporal integrity providing;
- executing such commands as a removal operation DELETE, the renewal operation UPDATE and the operation of the expanded selection SELECT taking into account the temporal properties of the object;
- support of the temporal upward compatibility with the working non-temporal applications;
- providing the access to all previous and all following object versions;

- assurance of the existence of the single active version of the object at any moment of time.

There are principal peculiarities of the temporal framework implementation.

For preservation the logic of the legacy applications it is necessary to allow the removal operation DELETE and the operation of the object properties renewal UPDATE. Nevertheless in the process of these operations execution no of the object states must disappear. In this case the following technique of executing the operations of the object modification is offered. At the time moment t_o of the i-th change in the object state the element of the temporal framework turns the non-active version of the object in the shadow zone with the automatic fixation of the time of the life cycle period ending $t_e^{(i)} = t_o$. Thereby the process is completed for the operation DELETE, but in case the operation UPDATE takes place, the temporal framework generates the new relation tuple (the new record) – the active $(i+1)$-th object version with the time of life cycle beginning $t_s^{(i+1)} = t_o$ (see Fig. 3.11). This temporal logic corresponds to the Rollback temporal form.

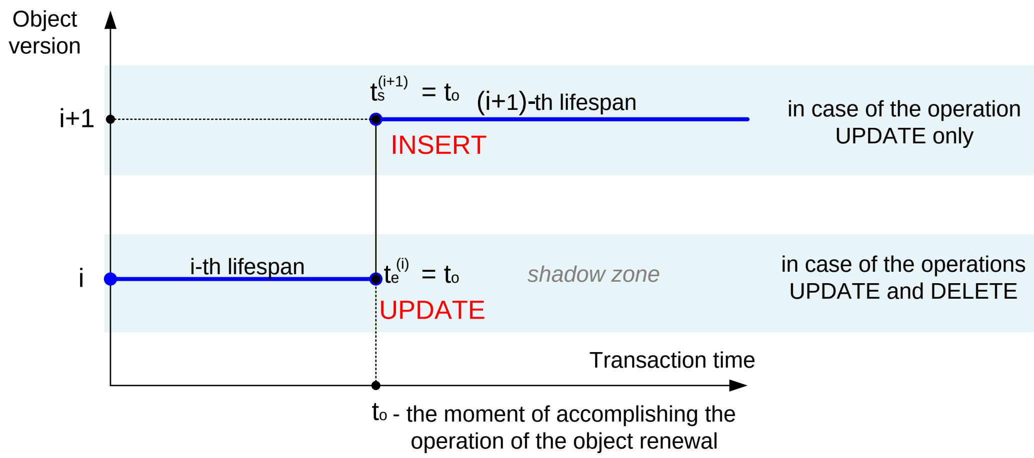


Fig. 3.11. Diagram of changing the object state in the temporal form Rollback

The Rollback form supposes that the users interact with the database via the non-temporal queries. And what is important, first of all it concerns the operations of modifications (UPDATE, DELETE). The procedure of supporting the Rollback form functionality automatically stores the previous versions of the table records. Speaking about the operations of reading, the logic of the Rollback form allows performing both temporal and non-temporal queries. In other words if the query enters without evident time specification the system has to deliver the active value for the moment of time of the query.

For simplification of the operations with the objects, possessing the temporal nature, the following procedure is offered: in the applications, processing the data that are active for the moment of interaction with them, it is suggested operating not with the objects themselves but with the data-VIEW-filtering the data of the "shadow zone" basing on the moment of time when the query enters, and this moment is calculated automatically. Accordingly, these applications will always operate with the data under question without complicating the logic of the software.

The author has suggested the scheme of interacting with the database, implementing the above described logic in the process of the operation UPDATE executing, and this scheme is presented in Fig. 3.12. The standard facilities of the active databases are used for this implementation: SQL, views and triggers.

In the process of adding the temporal properties to the information systems objects, the serious structural changes occur in the databases, starting with the primary key repertory and to the extent of the relation decomposition. These changes are bound to affect the logic of applications functioning. The support of the existing logic for the functioning applications is guaranteed by holding the interface of their cooperation with the databases.

For solving the task of saving the "old" interface in the databases extended by the temporal properties, it is offered to organize the view of the data, possessing the structure and the components identifiers of the original database before the structural changes. In this case all the components of the database that were subjected to the structural alterations have to be given the new names. Fig. 3.12 demonstrates the example, showing the process of transformation under the condition of the temporal extension of the table *Table1* into the set of tables projected in accordance with the model AOID: *Table1_T0, Table1_T1…* and so on depending on the number of the dynamic attributes. And with the usage of its name *Table1* the representation, reflecting the stored data in their original form, has been created. The representation *Table1* accomplishes the composition of the model AOID data and the time slice of all temporal tables, comprising all active values of the data at the moment of query. The *Legacy Application* interacts with the *Table1* by the means of queries, employed before the temporal extension of the table, and receives the results, corresponding to its original structure. The query for modification UPDATE, generated by the application, carries no data on the time range for the new value. The temporal framework intercepts this query automatically and by the means of the trigger *Trigger(U)* enlarges it with the set of activities oriented on keeping the previous version and setting the temporal attributes values for the new one.

Fig. 3.12 also presents the processing scheme for queries sent by the application, accounting the temporal data properties (*Temporal DB Application*). The data modification in this case takes place by means of inserting the new version with all the time marks pointed. The temporal framework here plays the different part – employing the trigger *Trigger(I),* it actualizes the support of the data temporal integrity, performing the renewal of the time ranges of the neighboring versions and, if it is necessary, dividing the range of the previous version into two pieces.

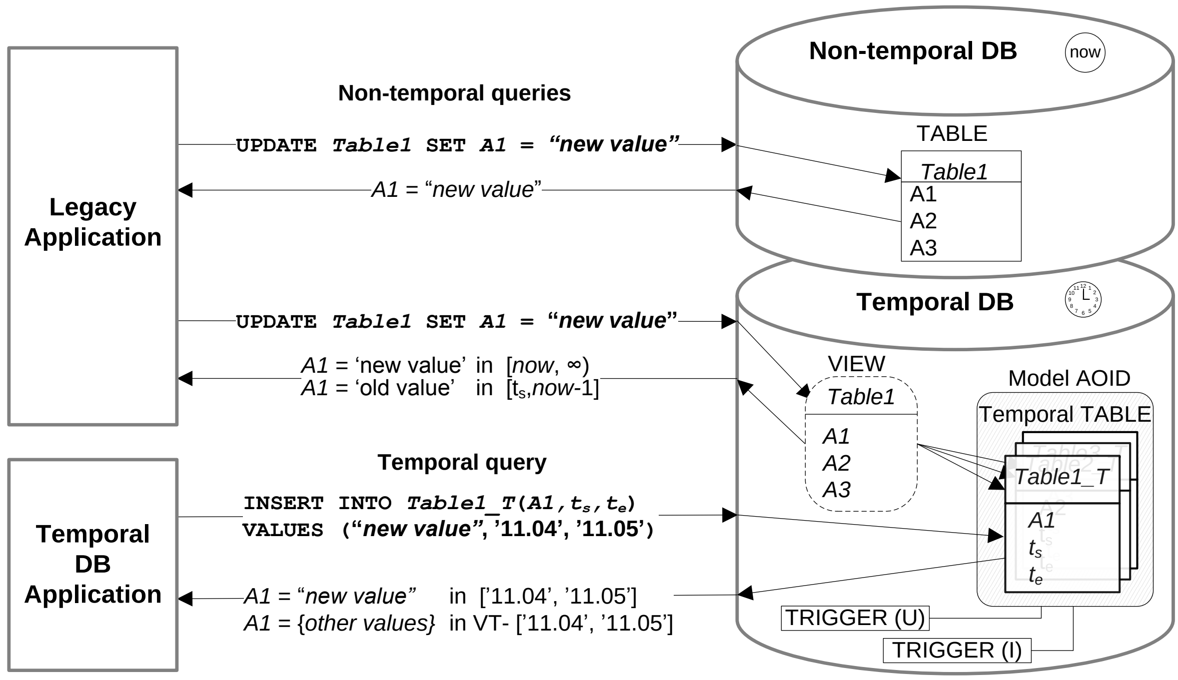


Fig. 3.12. Support of the queries of the temporal data update

The above described principle of interacting is also employed for the queries of the temporal data reading. Fig. 3.13 demonstrates this principle.

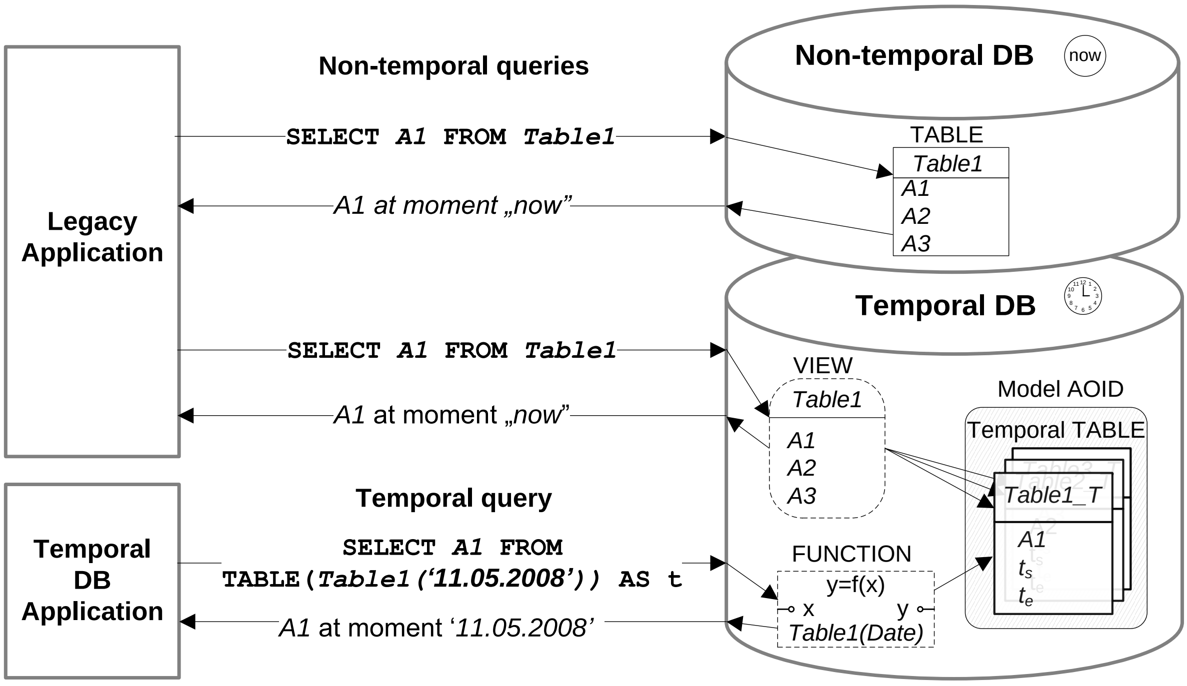


Fig. 3.13. Support of the queries for the temporal data reading

For simplifying the temporal queries code it has been suggested the employment of the TEMPORAL SLICE FUNCTION, and this fact is reflected with the help of the corresponding block FUNCTION in Fig. 3.13. The advantages of its employment are demonstrated by the example of accomplishing the query on the fare of travel on 01.01.2008. The tariffs data are comprised in the temporal table K_PRICE of the Historical-Classic form. The query has the following form with the SQL standard constructions implementation:

```
SELECT * FROM K_PRICE AS t
      WHERE (t.DATE_START<='01.01.2008') AND
            (t.DATE_END>='01.01.2008' Or t.DATE_END Is Null);
```

There is the creation of the time slice function K_PRICE() for receiving the tariffs on the specified date:

```
CREATE FUNCTION K_PRICE (d DATE)
  RETURNS TABLE ([fieldlist])
  RETURN SELECT [fieldlist]
  FROM K_PRICE  WHERE d BETWEEN  start_date AND VALUE(end_date,d);
```

It is obvious that the query is significantly simplified with this function implementation:

```
SELECT * FROM table(K_PRICE('01.01.2008')) AS t;
```

Consequently the suggested method allows providing the database with two different interfaces simultaneously: the relation classical (non-temporal) interface and the temporal one. The first one provides the temporal upward compatibility with working non-temporal applications, giving the data actual for the moment of interaction, the second shows the temporal data in the original form. At the same time non-temporal interface supports all the operations of modification, and the temporal framework, invisible for the application, supports the temporal data integrity automatically. This method is considered in the works [4, 8] and is employed in practice in the information systems Referenced Data System on the Latvian Railway.

Nevertheless, for the whole range of the transport information systems it is not enough to interact with the values, active for the query moment only. First of all it refers to the systems, processing the accumulated historical data. Every entry of these data (payment, agreement, transport operation, etc.) is connected with the definite time moment, and consequently with the reference information active for this entry. However the queries sent to the reference data database by the working applications do not take into account the temporal data dimension, and therefore they are not able to get customized to the necessary time slice. The method of providing the temporal environment for the user, considered below in the point 3.2.1, is suggested for finding the solution of the issue of the non-temporal queries transition to the specified point in time.

There are two examples of the temporal databases implementation, referring to the different types of the tasks, and illustrating the temporal logic in points 3.2.2 and 3.2.3. Both examples demonstrate the objects, possessing the altering periods of life, but in the first example these periods cannot overlap, whereas in the second one it is acceptable.

3.2.1. Method of Providing the User's Temporal Environment

The fundamental of this method is the displacement of the current time point *now* and placement of the user or application into the definite moment in time. The execution of all his/its queries for the data occurs relatively the displaced point. This time point is specified in the user's deliberately defined temporal environment and it is called *environment variable*. Hence the method allows performing the non-temporal queries to the multi-version data in various time slices. The principal concept of its functioning is shown in the performed example below.

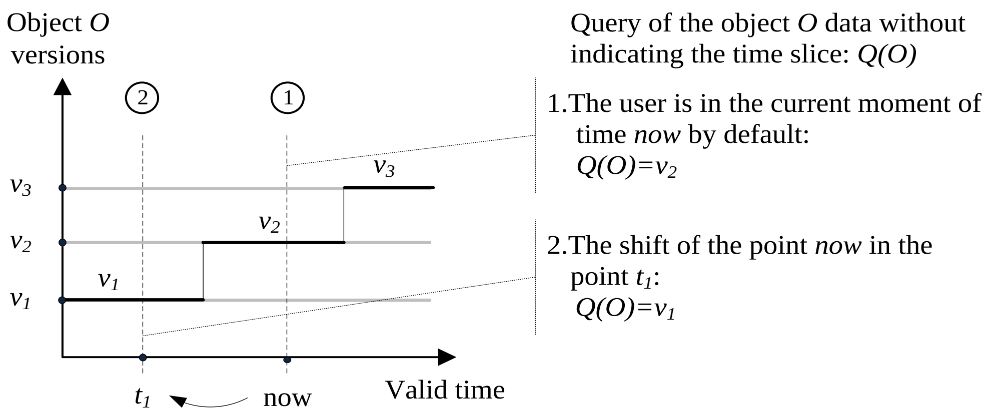


Fig. 3.14. Point "now" displacement for the user's queries

Assuming, there is a form of the non-temporal query $Q(O)$ to the database, returning the object O data. After the process of allotting the database with the time dimension it comprises not one version of the object O state, but three of them: v_1, v_2 and v_3, (as it is presented in Fig. 3.14).There are two variants of the query functioning under the observation: under the condition of the non-defined variable of the user's environment and when this variable equals t_1 – a definite point of time in the past. In the first case the temporal framework executes the query relatively the current time *(now)* and the result of query *Q(O)* is the object version v_2, active for the moment of the query executing. In the second case the query is executed relatively to the determined time point in the time environment, equal to t_1 and in the result the version v_1, active for the moment t_1 will be selected.

Fig. 3.15 demonstrates the transition from the non-temporal database to the temporal one for the existing application. It is shown in the scheme that the application generates the same non-temporal queries, however, besides the non-temporal interface holding, the new temporal database is capable of processing the queries in different time slices, which a necessary condition for more accurate application functioning. The example considers processing the queries of two independent users with different determined environment variables. It is obvious the users receive the different values as the responses to the same query, and these values are the active ones for the time moment, pointedly specified in the user's environment variable.

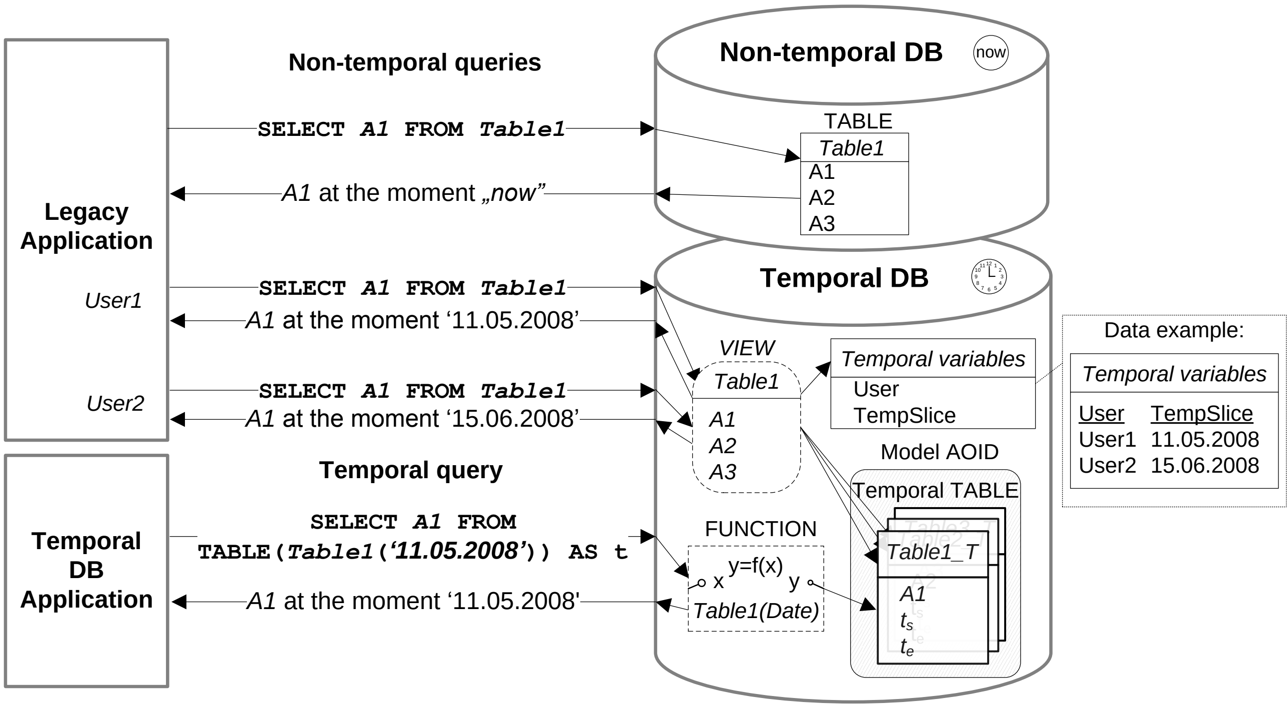


Fig. 3.15. Supporting the interface of the non-temporal operation of the data reading employing the user's environment variable

With the help of this method it is possible to manipulate efficiently the time slice, where the user has to work, without changing the application code and without employing the additional conditions in the queries. Under the condition of the environment variable alternation the queries start functioning in the different time layer automatically and in a flash. It is important, that using this method all users have possibility to work independently as every user has his temporal environment determined.

The suggested method employment is advisable in the case of the new information systems development as well. Its application permits to simplify considerably the queries code for the multi-version data, as it delivers from the necessity to specify the additional conditions for getting into the determined time layer. This fact decreases the probability of the errors emergency in the queries code, and consequently it increases the reliability of the received results.

The user's temporal environment is also suggested for using for the computation of the temporal aggregates, which are the necessary component of any analytical processing. Executing the temporal aggregation procedures is connected with the determination of the time domain (the processed period with the determined dates of the beginning and the end), for which the aggregates are computed. The basis of this method is the time range determination within which the user operates, and its further forced employment in the computation. The complicated calculations, such as the preparation of the reports on the railway passenger transportation [1, 2, 10, 114], use the series of the interconnected aggregates computations, and excluding the option of the time range specifying from the query will both simplify the query code and decrease significantly the probability of the errors emergency.

The user's temporal environment is suggested for applying also as the facility of the temporal data security increasing. The special environment variable allows limiting the user's acceptable field of the user's operations by the data set, related to the determined time range. The application of this temporal environment variable is described in the author's researches [15, 18].

3.2.2. Example of the Temporal Logic Implementation in the Referenced Data System

There is the segment of the ER-model of the database of the Referenced Data System (see Fig. 3.16), presenting the structure of the freight directory before its temporal extension. The object "freight" is described with the unique identifier GNG_ID in the relation GNG and possesses four principle properties: the code GCODE, name in the Latvian language is NAME_L, name in the Russian language is NAME_R and annotation PIEZ. It is worth mentioning, the directory active in the Latvian Railway contains considerably greater amount of the properties, but for the simplification of the exposition their number in the given instance has been diminished.

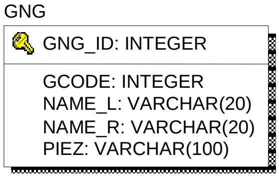


Fig. 3.16. Non-temporal table "The Freight Directory"

After the extension of the directory with the time dimension of the active time according to the model AOID, the structure of this reference source has been transformed in the set of tables, demonstrated in Fig. 3.17. For the task under consideration, as for the majority of the directories on the railway, the time dimension chronon equals one day period. The properties NAME_L and NAME_R are altered simultaneously, that is why they have been placed in one and the same relation. The history of changing the schedule PIEZ holds no value, only the current value is important, and it is the reason why this property has been left in the parent relation K_GNG, describing the freight abstract identifier GNG_ID. The change of the freight number does not implicate the change of its name, and it is the reason why the property GCODE has been detached in the separate table.

The primary key of the dependent relations, describing the object characteristics, comprises the attribute DATE_START, specifying the time of the object properties activation, and jointly with the foreign key (FK) unambiguously identifies the version of the object state.

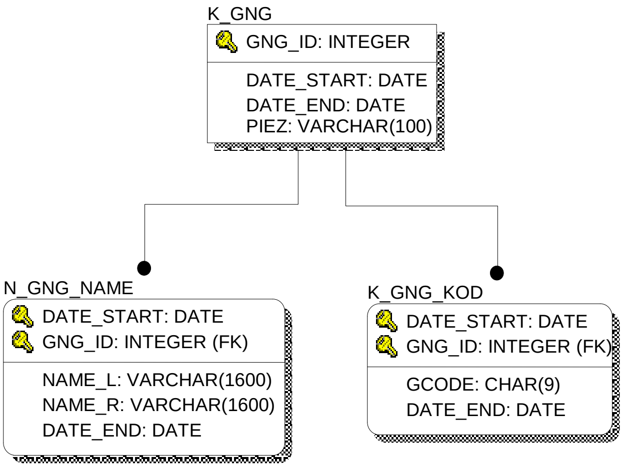


Fig. 3.17. ER-model of the temporal database "The Freight Directory"

For providing the "old" interface it has been created the data view GNG, reveling in the full volume the original table GNG and presenting the data active for the moment of the query from the temporal tables:

```
CREATE VIEW GNG (GNG_ID, GCODE, NAME_L, NAME_R, PIEZ) AS
SELECT K_GNG.GNG_ID, PIEZ, GCODE,   NAME_L, NAME_R
 FROM K_GNG
 LEFT JOIN (
       SELECT *  FROM K_GNG_KOD
       WHERE DATE_END IS NULL OR DATE_END >= CURRENT_DATE) K_GNG_KOD
 ON K_GNG_KOD.GNG_ID = K_GNG.GNG_ID
 LEFT JOIN (
       SELECT *  FROM N_GNG_NAME
       WHERE DATE_END  IS NULL OR DATE_END >= CURRENT_DATE) N_GNG_NAME
 ON K_GNG.GNG_ID = N_GNG_NAME.GNG_ID
 WHERE K_GNG.DATE_END  IS NULL OR K_GNG.DATE_END >= CURRENT_DATE
```

The data view GNG permits not only to browse the data (employing the instruction SELECT), but also to produce the operations of the data manipulation: INSERT, UPDATE and DELETE. Moreover, all the previous versions are saved and transit into "the shadow zone". There is the examination of the suggested procedure of the temporal logic provision exemplified by the triggers, implementing the operation of the records extension, removal and renewal, using the GNG view.

The new records appending via the GNG view is accomplished with the employment of the trigger TINSTEADOF_INSERT_GNG. This trigger intercepts the operation of the data extension INSERT and exchanges it with the series of the following activities:

1. Initialization of the object identifier GNG_ID in case if it is not specified;

2. Creation of the new version for every of the described properties (GCODE, NAME_L and NAME_R) in the corresponding temporal tables.

The DDL-instruction for creating the trigger TINSTEADOF_INSERT_GNG has the following form:

```
CREATE TRIGGER TINSTEADOF_INSERT_GNG
  INSTEAD OF INSERT
  ON GNG
  REFERENCING  NEW AS n
  FOR EACH ROW
BEGIN ATOMIC
 DECLARE G_ID INT;

 -- initialization of the object OID
 IF n.GNG_ID IS NULL THEN
       SET G_ID = (NEXT VALUE FOR seq_k_gng);
 ELSE
       SET G_ID=n.GNG_ID;
 END IF;

 -- Creation of the object
 INSERT INTO K_GNG (GNG_ID, PIEZ, DATE_START)
 VALUES (G_ID, PIEZ, CURRENT_DATE);

 -- Creation of the new version for every property
 IF n.GCODE  is not null  THEN
       INSERT INTO K_GNG_KOD (GNG_ID, GCODE, DATE_START)
       VALUES (G_ID, n.GCODE, CURRENT_DATE);
 END IF;

 IF (n.NAME_L is not null  or n.NAME_R  is not null)  THEN
       INSERT INTO N_GNG_NAME (GNG_ID, NAME_L, NAME_R, DATE_START)
       VALUES (G_ID, n.NAME_L, n.NAME_R, CURRENT_DATE );
 END IF;
END;
```

The data changing via the GNG view is implemented with the trigger TINSTEADOF_UPDATE_GNG employment. This trigger exchanges the operation of the data alteration UPDATE via the GNG view and for every altered property (GCODE, NAME_L and NAME_R) produces closing of the existed version and the new version creation in the corresponding temporal table. Besides this the trigger recognizes the "erroneous" versions, which have appeared as a result of the operators' errors. Such errors are corrected as a rule on the day of their appearance. In this case the trigger does not save the previous erroneous version. The DDL-instruction for the trigger TINSTEADOF_UPDATE_GNG creation has the following form:

```
CREATE TRIGGER TINSTEADOF_UPDATE_GNG
  INSTEAD OF UPDATE
  ON GNG
  REFERENCING OLD AS o  NEW AS n
  FOR EACH ROW
BEGIN ATOMIC
 IF (o.GCODE<>n.GCODE) AND (n.GNG_ID=o.GNG_ID) THEN

       IF EXISTS(SELECT * FROM K_GNG_KOD a
                   WHERE a.GNG_ID=n.GNG_ID AND a.DATE_END is null
                         AND a.DATE_START=CURRENT_DATE) THEN

             -- in case of the repeated data corrections on the same day
             UPDATE K_GNG_KOD A
                   SET A.GCODE = N.GCODE
             WHERE A.GNG_ID = N.GNG_ID     AND A.DATE_END IS NULL;

       ELSE
```

```
                -- closing of the previous version
                UPDATE K_GNG_KOD A
                      SET A.DATE_END = CURRENT_DATE - 1 DAY
                WHERE A.GNG_ID = N.GNG_ID AND A.DATE_END IS NULL;

                -- the new version creation
                INSERT INTO K_GNG_KOD (GNG_ID, GCODE, DATE_START)
                VALUES (n.GNG_ID, n.GCODE, CURRENT_DATE);

          END IF;
    END IF;

    IF (o.NAME_L<>n.NAME_L or o.NAME_R<>n.NAME_R) AND (n.GNG_ID=o.GNG_ID)
   THEN

          IF EXISTS(SELECT * from N_GNG_NAME a
                      WHERE a.GNG_ID=n.GNG_ID AND a.DATE_END IS NULL
                            AND a.DATE_START=CURRENT_DATE) THEN

                -- in case of the repeated data corrections on the same day
                      UPDATE N_GNG_NAME a
                 SET a.NAME_L=n.NAME_L, a.NAME_R=n.NAME_R
                WHERE a.GNG_ID=n.GNG_ID and a.DATE_END is null;

          ELSE

                -- closing of the previous version
                UPDATE N_GNG_NAME A
                 SET A.DATE_END = CURRENT_DATE - 1 DAY
                WHERE A.GNG_ID = N.GNG_ID AND A.DATE_END IS NULL;

                -- the new version creation
                INSERT INTO N_GNG_NAME (GNG_ID, NAME_L, NAME_R, DATE_START)
                VALUES (n.GNG_ID, n.NAME_L, n.NAME_R, CURRENT_DATE);
          END IF;
    END IF;

    IF (o.PIEZ<>n.PIEZ or (n.PIEZ<>'' and o.PIEZ is NULL))
          AND (n.GNG_ID=o.GNG_ID) THEN
          UPDATE K_GNG a
           SET a.PIEZ = n.PIEZ
          WHERE a.GNG_ID = N.GNG_ID;
    END IF;
   END;
```

The records removal via the GNG view is implemented with the trigger TINSTEADOF_DELETE_GNG employment. This trigger intercepts the operation of the data removal DELETE and exchanges it with the series of activities directed on closing the lifespans in all relations. The DDL-instruction for the trigger TINSTEADOF_DELETE _GNG creation has the following form:

```
CREATE TRIGGER TINSTEADOF_DELETE_GNG
  INSTEAD OF DELETE
  ON GNG
  REFERENCING
    OLD AS o
  FOR EACH ROW
BEGIN ATOMIC
       UPDATE K_GNG_KOD a
       SET a.DATE_END=CURRENT_DATE-1 day
```

```
        WHERE a.GNG_ID=o.GNG_ID and a.DATE_END is null;

        UPDATE N_GNG_NAME a
        SET a.DATE_END=CURRENT_DATE-1 day
        WHERE a.GNG_ID=o.GNG_ID and a.DATE_END is null;

        UPDATE K_GNG a
        SET a.DATE_END=CURRENT_DATE-1 day
        WHERE a.GNG_ID=o.GNG_ID;
END;
```

Here is the demonstration of the presented procedure of providing the temporal logic exemplified by the operations sequence accomplishment with the freight data classifier. There is the chronology of these operations.

1. 01.02.2010 - Via the GNG view the new record was appended (the new referenced value of the freight):

```
INSERT INTO GNG (GCODE, NAME_L, NAME_R)
VALUES ('690410020', 'Keramiskie celtniecības ķieģeļi,
          ugunsneizturīgie',
         'Кирпич строит.керамич., неогнеупорный');
```

2. 01.02.2010 - Via the GNG view one more new record was appended, moreover, without indication of the value for the field GCODE in the entered instruction:

```
INSERT INTO GNG (NAME_L, NAME_R)
VALUES (' Izstrādājumi no ģipša', ' Изделия из гипса');
```

After accomplishing these two operations the GNG view demonstrates the data, shown in Fig. 3.18.

GNG_ID	PIEZ	GCODE	NAME_L	NAME_R
10		690410020	Keramiskie celtniecības ķieģeļi, ugunsneizturīgie	Кирпич строит.керамич., неогнеупорный
11			Izstrādājumi no ģipša	Изделия из гипса

Fig. 3.18. GNG view data after the second operation accomplishment

3. 01.02.2010. The mistake in the name of the freight GNG_ID=11 was discovered. Via the GNG view the values of the fields NAME_L and NAME_R were changed repeatedly, implementing the SQL-instruction:

```
UPDATE GNG
 SET   NAME_L = 'Izstrādājumi no ģipša vai no maisījumiem uz ģipša
                 bāzes',
       NAME_R = 'Изделия из гипса или смесей на его основе'
 WHERE GNG_ID = 11;
```

By virtue of the fact that the operation 3 was accomplished on the same day when the operation 1 took place, the previous erroneous version was not saved. Hence, the protection from the existence of the needless “erroneous” versions in the temporal relations was fulfilled.

4. 15.02.2010. Via the GNG view the new values for the fields GCODE and PIEZ were determined for the freight with the identifier GNG_ID=11 employing the following instruction:

```
UPDATE GNG
 SET  GCODE = '690430222', PIEZ='VBP:11'
 WHERE GNG_ID = 11;
```

5. 28.07.2010. Via the GNG view the values of the fields GCODE were changed for the freight with the identifier GNG_ID = 10 employing the following instruction:

```
UPDATE GNG
 SET  GCODE = '690410000'
 WHERE GNG_ID = 10;
```

As a result of the presented manipulations, the GNG view demonstrates the data, shown in Fig. 3.19, and the temporal tables K_GNG, N_GNG_NAME, K_GNG_KOD contain the records presented in Fig. 3.20.

```
SELECT * FROM GNG;
```

GNG_ID	PIEZ	GCODE	NAME_L	NAME_R
10		690410000	Keramiskie celtniecības ķieģeļi, ugunsneizturīgie	Кирпич строит.керамич., неогнеупорный
11	VBP:11	690430222	Izstrādājumi no ģipša vai no maisījumiem uz ģipša bāzes	Изделия из гипса или смесей на его основе

Fig. 3.19. GNG view data after accomplishing the fifth operation

```
SELECT * FROM K_GNG;
```

GNG_ID	PIEZ	DATE_START	DATE_END
10		01.02.2010	
11	VBP:11	01.02.2010	

```
SELECT * FROM K_GNG_KOD;
```

GNG_ID	GCODE	DATE_START	DATE_END
10	690410020	01.02.2010	27.07.2010
10	690410000	28.07.2010	
11	690430222	15.02.2010	

```
SELECT * FROM N_GNG_NAME;
```

GNG_ID	NAME_L	NAME_R	DATE_START	DATE_END
10	Keramiskie celtniecības ķieģeļi, ugunsneizturīgie	Кирпич строит.керамич., неогнеупорный	01.02.2010	
11	Izstrādājumi no ģipša vai no maisījumiem uz ģipša bāzes	Изделия из гипса или смесей на его основе	01.02.2010	

Fig. 3.20. Data of the tables K_GNG, N_GNG_NAME, K_GNG_KOD after accomplishing the fifth operation

6. 29.08.2010. Via the GNG view the freight with the identifier GNG_ID=10 was removed employing the following instruction:

```
DELETE FROM GNG WHERE GNG_ID = 10;
```

As a result of this operation implementation, the object GNG_ID=10 itself is logically removed (the table K_GNG sets the value of the attribute DATE_END), and all active versions of this object in the corresponding temporal tables N_GNG_NAME, K_GNG_KOD are closed. The GNG view does not deliver any data about the freight with the identifier GNG_ID=10.

For providing the interaction with the freight classifier data, active on the specified date, *the function of the time slice* is created:

```
CREATE FUNCTION GNG (d DATE)
 RETURNS TABLE (   GNG_ID int, PIEZ varchar(100), GCODE char(9),
                   NAME_L varchar(1500), NAME_R varchar(1500))

 RETURN SELECT K_GNG.GNG_ID, PIEZ, GCODE, NAME_L, NAME_R
 FROM K_GNG
 LEFT JOIN (
       SELECT *  FROM K_GNG_KOD
       WHERE d between DATE_START and VALUE(DATE_END,d)) K_GNG_KOD
 ON K_GNG_KOD.GNG_ID = K_GNG.GNG_ID
 LEFT JOIN (
       SELECT *  FROM N_GNG_NAME
       WHERE d between DATE_START and VALUE(DATE_END,d)) N_GNG_NAME
 ON K_GNG.GNG_ID = N_GNG_NAME.GNG_ID
 WHERE d between K_GNG.DATE_START and VALUE(K_GNG.DATE_END,d);
```

The example of employing the function of the time slice GNG () and its operation results are demonstrated in Fig. 3.21:

```
SELECT * FROM table(GNG(date('01.03.2010'))) AS t1
```

GNG_ID	PIEZ	GCODE	NAME_L	NAME_R
10		690410020	Keramiskie celtniecības ķieģeļi, ugunsneizturīgie	Кирпич строит.керамич., неогнеупорный
11	VBP:11	690430222	Izstrādājumi no ģipša vai no maisījumiem uz ģipša bāzes	Изделия из гипса или смесей на его основе

Fig. 3.21. The example of using the function of the time slice GNG()

3.2.3. Example of the Temporal Logic Implementation in the Train Schedule System

The instance considered below is interesting by the fact that the different periods of life of the object are overlapping. The schedule temporal nature is intrinsic for it, and the temporal database superimposes one more additional time dimension on it. The active schedule is practically overlapping these two dimensions. The peculiarity of this approach is enclosed in the fact that the object state in the process of alteration might not lose its activity in the past and in the future – it is exchanged by another version of the state only for a certain period of time. At the same moment there can more than one tuple exist, characterizing the different properties of one and the same object, and this is inadmissible in the instance examined in the point 3.2.2. The most suitable form of the temporal relation for arranging these types of the data is the Historical-Overlapped form, designed in the thesis (point 3.1.2).

There is the train traffic schedule, characterized by the diagram presented in Fig. 3.22 [14].

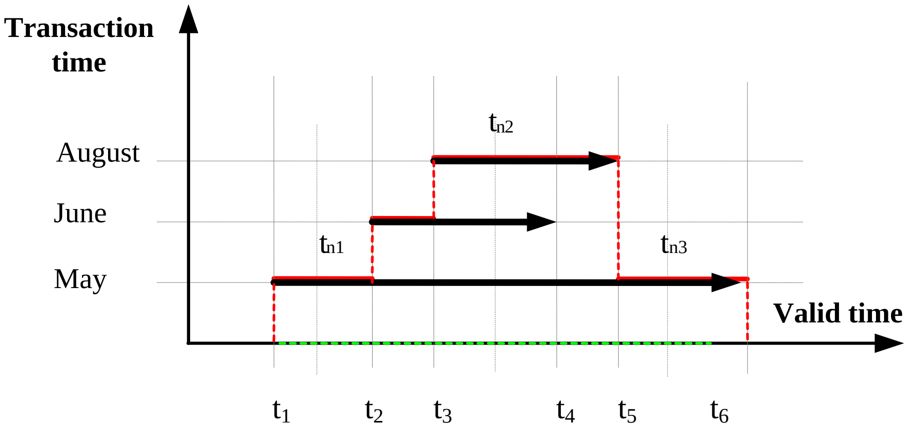


Fig. 3.22. Overlapping the object lifespans

It is assumed that t_n designates the time of observation. The train schedule is assigned for the period of time from t_1 to t_6. Later, for the period from t_2 to t_4 the schedule change is introduced and after that for the period of time from t_3 to t_5 one more alteration of the schedule is entered. The query on the train traffic at the moment of time t_{n1}, is responded by the way of delivering the schedule t_1-t_6, and at the moment t_{n2} – the schedule t_3-t_5, and at the moment t_{n3} – again the schedule t_1-t_6. All the logic, controlling the delivery of the active train schedule for the specified time moment is accomplished by the temporal frameworks, implemented in the form of the time slice function TRAIN_SCHEDULE().

```
CREATE FUNCTION TRAIN_SCHEDULE (D DATE)
  RETURNS TABLE (TRAIN_OID INT, SCH_ID INT, DATE_START DATE,
      DATE_STOP DATE, SCH_NAME VARCHAR(100))
      RETURN SELECT TRAIN_OID, SCH_ID, DATE_START, DATE_STOP, SCH_NAME
      FROM TRAIN_SCHEDULE AS T
      WHERE (T.DATE_START <= d AND
             (T.DATE_STOP >= d OR T.DATE_STOP IS NULL)) AND
             T.TIME_FIX=(SELECT MAX(TIME_FIX) FROM TRAIN_SCHEDULE A
                               WHERE A.TRAIN_OID=T.TRAIN_OID
                               AND DATE_STOP <= d
                         AND (DATE_STOP >= d OR DATE_STOP IS NULL)).
```

Employing the above demonstrated function, the user does not need to understand the data structure, supporting the object multi-version nature, and to form the complicated query, calculating the active version.

The method, implementing this logic, operates successfully with the objects, the period of life of which lies not only in the past or in the present, but also in the future. In case the database receives the query on the train traffic schedule, which will be active in a month, this query should contain them obvious indication of the necessary date, and as an answer the system delivers the active schedule for the demanded moment of time.

3.3. Periodicity Model in the Train Scheduling Tasks

The principal issues of the train traffic schedule storage in the databases are connected with the existence of the multitude of its versions conditioned by the seasonal cycles of schedule changes, the days of week, by the systematic and unplanned repair operations, the transfers of the working and festive days, and other factors (see point 1.5.2). The task of supporting the schedule can be solved by the method of designing the calendar of the every train running on each day of the year. However the number of the stored records, describing such facts as the day, the time and the stop stations for every train will be about the product of the number of the trains multiplied by the average number of stations and by the number of the days, for which the timetable is stored. Taking into consideration the above described approach, about two million records are needed to store the schedule of running the trains of the inland traffic on the Latvian Railway in the relation database for a year time period, and for the bigger railway companies the volume of the corresponding database will increase by times. Moreover, in case of storing the schedule for several years it is possible to speak about the hundreds of millions of records. Together with the technique development, the issue of the bulk data is taking the middle ground, and the problem, involving the complexities of the data reliability provision and the promptitude of the changes introduction to the schedule, is becoming more and more actual. Under the condition of the frequent alternations introducing these tasks becomes extremely urgent.

The *temporal object model with the periodicity and special calendar* and *the methods of the object active version determination: the logical rules method* and *the method of the temporal elements* (see point 3.4 and works [20, 22, 24]) were developed within the frameworks of this investigation for solving the above mentioned issues. The received results are applied in the process of creating the information system of the inland train traffic schedule on the Latvian Railway.

3.3.1. Employing the Periodicity Principle in the Railway Schedule

As a matter of practice the train can have several active schedules simultaneously during the long period of time (see point 1.5.2). Furthermore every train schedule is organized inclusive of days of running with the different *periodicity properties*, for example: on working days, on weekends, on the definite days of week, the first working day of week, the last day off of the week (notably it can be even Monday), on even or odd days etc.

As an example the multi-variant train schedule is considered (Fig. 3.23). The figure performs that the train has two basic schedules – one for the working days *wd*, and one for the weekends *rd*. Both timetables are valid on the time interval $[t_1, t_4]$, but on different week days. Then on Tuesdays and Thursdays (*Tue,Thu*) the amendment of this train schedule in the time period from $[t_1, t_4]$ is introduced for the systematic repair operations providing. Therefore in the time interval $[t_2, t_3]$ there are three timely schedule versions, activated on the days corresponding to the characteristics of *wd*, *rd* and (*Tue,Thu*), moreover, only one schedule version *wd*, *rd* or (*Tue*,*Thu*), can be active one day.

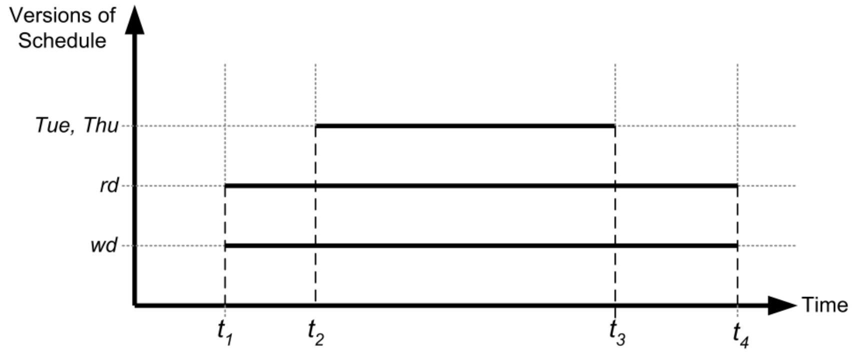


Fig. 3.23. Multitude of the valid versions of the train schedule: wd, rd and (Tue,Thu)

This assertion is illustrated with help of the diagram in Fig. 3.24. The axis "Day of Week" shows the days of week, respectively 1 is Monday, 2 is Tuesday and so on. Time in this system is of the discrete nature, the level of the schedule granularity equals one day period. As the mentioned figure presents, one of three versions is the active train timetable in different days of week, and the other two versions are in the "shadow" zone at this moment. Moreover, the train schedule on Tuesdays and Thursdays (characteristic *Tue,Thu*) overlaps the working day schedule with the characteristic *wd*.

However besides the recurring versions of the train schedule there are also one-time changes in the timetable, connected with the special (altered) calendar. The following changes can be listed in this category: the festive days, the additionally appointed days-off, or the canceled day-offs. There is also such special occasion schedule alteration when the calendar working day exchanges the places with the calendar day-off. For example, in Latvia in 2007 the 30th of April, Monday (a working day) was transferred on the 14th of April, Saturday (a day-off). As a result on the 30th of April the trains ran according to the schedule of the weekends, and on the 14th of April the trains ran according to the working days timetable – instead of the 30th of April.

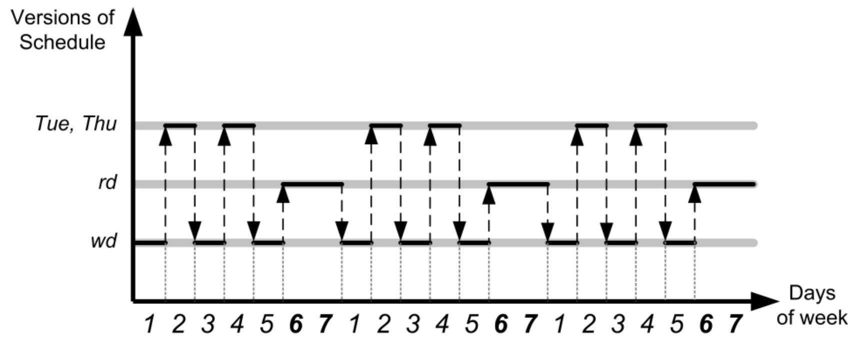


Fig. 3.24. Periodicity in the train schedule

Such types of changes in the timetable or assignations provoke the anomalies (paradoxes). For example, the 19th of November, Monday, in 2007 was announced as an additional day-off, and all the trains moved this day according to the schedule of the weekends and days-off (like on the 18th of November), and this event arouse the malfunction of the periodicity, as it is shown in Fig. 3.25. It is important to add that several trains with periodicity (Sundays) were canceled on the 18th of November, because the trivial Sunday is not only the day-off, but also the last day of weekend, and the additional Sundays trains are considered for the passengers transportation from the traditional locations of the resting facilities on the days-off.

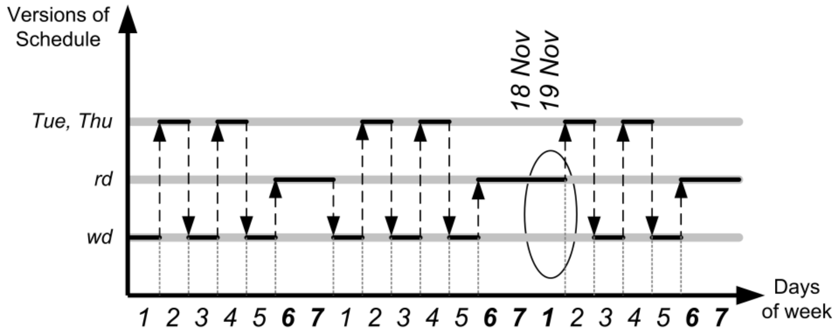


Fig. 3.25. Periodicity malfunction in the train schedule

For correct accurate functioning of the scheduling system, which is to take into account the above considered non-trivial rules, the well-defined formalization of the task assignment and the description of the scheduling temporal model are demanded. For this purpose the group of sets, characterizing the rules of arranging the trains running periodicity, the railway objects, and the entire system of scheduling itself are introduced.

3.3.2. Mathematical Description of the Train Schedule System

Two basic sets are introduced for the objects and rules description:

$S = \{s_1, s_2, \ ..., s_k\}$ is the set of all stations of the railways, $|S| = k$;

$N = \{n_1, n_2, \ ..., n_m\}$ is the set of the trains, $|N| = m$.

The schedule for the train with the number $n \in N$ will be determined by the following vector type:

$$v = \{\langle s_1^{(n)}, T_1 \rangle, \langle s_2^{(n)}, T_2 \rangle, ..., \langle s_\alpha^{(n)}, T_\alpha \rangle\}, \tag{7}$$

where the pair $\langle s_j^{(n)}, T_j \rangle$ determines the *j*-th train stop, the station $s_j^{(n)} \in S$ exactly, and the train departure time T_j ; α is the number of the stops of the train running on the timetable *v*.

Several schedule versions, characterized by the operation period, the assignation time (fixation) and the periodicity, can be hold for the train number *n*. Consequently the following scheduling parameters are introduced:

t_s and t_e are the moments of time of the beginning and ending of the operational period of the schedule version in question respectively;

t_f is the time of the fixation of the schedule version in the databases;

C is the property of the periodicity employed for the determination of the days of the schedule in question functioning.

The property *C* presents the logical expression consisting of one or several elementary characteristics of the periodicity connected by the logical operations signs $\vee, \wedge$ and $\neg$. Every elementary parameter of the periodicity p_i correlates with the predicate $p_i(x)$, that defines the day belonging to the exact group, for example, “the day is the even day of the week” and it can be *true* or *false*. The elementary parameters of the periodicity make up the set of properties which serves as the basis for the logical expressions generation, and these expressions designate any more complicated periodicity:

$$C(x) = f(p_i(x),\ i = \overline{1,\lambda})\ , \tag{8}$$

where $p_i(x) \in \mathrm{B} \equiv \{true;\ false\}$, λ is an arity of statement *C*.

The dissertation considers the different variants of the periodicity parameter, introduced by the set $P = \{p_1, p_2,\ ..., p_\pi\}$, where π is the set cardinality; the set comprises the following proposition as its elements: *ed* is "any day of the week" (it is used in case the train runs daily); *wd* is "a working day" (the train runs the working days); *rd* is "the day-off"; *Mon, Tue, Wed, Thu, Fri, Sat, Sun"* is the corresponding day of the week", *pd* is "the even days of the month"; *nd* is "the odd days of the month"; *lh* is "the last day-off of the week"; *fw* is "the first working day of the week, etc.

The predicate $p_i(x)$ determines the property of the day *x* to be the day with the property p_i. The set of the predicate truth $p_i(x)$ on the definition area $x \in [t_s, t_e]$ constitutes the aggregate of all days with p_i property in the range $[t_s, t_e]$. For example, the determination of the set of the predicate truth $Mon(x)$ for $x \in$ [01.05.2007, 31.05.2007] appears in the following form:

$\{\ x \mid x \in$ [01.05.2007, 31.05.2007], $Mon(x)\ \}=$ {07.05.2007, 14.05.2007,
21.05.2007, 28.05.2007}.

In the similar vein the truth set of the expression *C* for the specified version presents the aggregate of the days in the range $[t_s, t_e]$; this version is operative on these days.

The way of describing the periodicity *C* in the form of statements, suggested by the author, differs from the before known facility examined by Paolo Terenziani in his paper [62]; he describes the periodicity in the form of the formula of the operations with the sets. The approach, put forward in this research, is also "symbolic", easily implemented and extended, but at the same time it is more formalized and capable of increasing the calculations efficiency (see point 3.4.1).

For the specific schedule version identification for the train with number $n \in N$ the type of tuple $\langle n, C, t_f, t_s, t_e \rangle$ is employed.

Then *i*-th schedule version $v_i^{(n)}$ for the train with number $n \in N$ will be determined by the tuple type:

$$v_i^{(n)} = (\langle n, C, t_f, t_s, t_e \rangle, \langle s_1^{(n)}, T_1 \rangle, \langle s_2^{(n)}, T_2 \rangle, ..., \langle s_\alpha^{(n)}, T_\alpha \rangle)\,. \tag{9}$$

Assuming $V^{(n)} = \{v_1^{(n)}, v_2^{(n)},\ ..., v_g^{(n)}\}$ as the set of versions of the schedule of the train with number $n \in N$, where *g* is the set capacity and it equals the number of the versions of train *n* schedule, then $V = \{V^{(n_1)}, V^{(n_2)}, ..., V^{(n_m)}\}$ is the set of versions of all trains timetables.

Further the sets describing the periodicity exclusions have been assigned prescriptively:

$EX(C) = \{ex_1^{(C)}, ex_2^{(C)}, \ldots, ex_\theta^{(C)}\}$ is the set of the days-exclusions $ex_j^{(C)}$, which have been assigned with the property *C*, where θ is the capacity of the set *EX*(*C*). Two collections can serve as the typical examples of the set *EX*(*C*): the set $EX(wd) = \{ex_1^{(wd)}, ex_2^{(wd)}, ..., ex_w^{(wd)}\}$, giving the periodicity attribute $C = wd$ to the additionally assigned working days, and the set

$EX(rd) = \{ex_1^{(rd)}, ex_2^{(rd)}, \ ..., ex_q^{(rd)}\}$ the additionally assigned festive days (days-off) with the periodicity attribute $C = rd$;

$EX = \{\langle C_1, EX(C_1)\rangle, \langle C_2, EX(C_2)\rangle, \ldots, \langle C_l, EX(C_l)\rangle\}$ is the set of all exclusions for the different periodicity properties, where *l* is the capacity of the set *EX*; the set *EX* components re-determine the periodicity attributes for the dates, appointed in *EX*(*C*);

$Z = \{z_1, z_2, \ ..., z_\mu\}$ the set of the transferred days of the trains schedule, where z_i, $i = \overline{1, \mu}$ is the pair of days $\langle d_1^{\{i\}}, d_2^{\{i\}}\rangle$, determining, that the timetable assigned on the date $d_1^{\{i\}}$, is assigned on the date $d_2^{\{i\}}$ as well; $|Z| = \mu$.

Due to the complexity of managing the different periodicity schedule versions set, continuously appearing operative schedule changes, taking into consideration the non-trivial calendar, subjected to the changes, the development of the efficient methods of the active version determination is highly needed. In the investigation process two alternative methods have been suggested by the thesis author: *the logical rules method* and *the temporal elements method*, which are considered in this research and in the works [20, 22, 24].

3.4. Methods of Defining the Active Version of the Temporal Object

3.4.1. Temporal Elements Method

The peculiarity of the suggested method of determining the active version of the temporal object lies in the preliminary calculation of the dates of activity of every version of the schedule and their storage for the subsequent employment.

As a container for this calculable set of dates the author has adapted the temporal element Gadia's (TE, temporal element) [79]. In the research under consideration the temporal element is employed for determining the time moments, when the exact version of the train schedule on the extent of its life cycle becomes active. The temporal element comprises all information on the periodicity of the timetable, taking into consideration all operative changes several versions overlapping and special calendar. (The temporal element is formed in respect with the data about the activity range and the periodicity of the schedule version, the operative changes data, several versions and exchanges overlapping data).

The timetable version $v_i^{(n)}$ of the train with number $n \in N$, assigned with the expression (9), can be determined with the temporal element

$$v_i^{(n)} = (\langle n, TE_i^{(n)}\rangle, \langle s_1^{(n)}, T_1\rangle, \langle s_2^{(n)}, T_2\rangle, ..., \langle s_\alpha^{(n)}, T_\alpha\rangle), \qquad (10)$$

where the temporal element $TE_i^{(n)}$ determines all the dates when the version $v_i^{(n)}$ is active.

For calculating the moments when the periodical events are active, there was accomplished the function of the limited extension (the extension function Bounded Extension) of $BExt([t_s, t_e], C)$ type, considered in the paper [62]. Its original variant employs the apparatus of the set theory for its calculations; this procedure is followed by the considerable memory resources consumption, especially in case of processing the complex periodicities, typical for the railway schedule. Under these circumstances the author had to revise the principle of the function $BExt([t_s, t_e], C)$ calculation, and to use the predicated calculations instead of operating with the sets. Also there has been changed the format of the function result, which returns the set of the time intervals in the original variant, and in the one, suggested by the author, the set of the time points (dates). It permits to use the standard relation algebra in the process of the further result processing, while the interval representation requires the temporal extension of the relation algebra, and this extension is not the component of the relation databases management system, employed by the transport enterprises.

The process of calculating the temporal element of the schedule version demonstrated by the tuple of relation in the Historical-Overlapped form, comprises two steps: 1) calculation of the basic value of the temporal element; 2) calculation of the actual value of the temporal element. There is the examination of these steps.

The first step produces the calculation of the basic value of the temporal element, marked TE^*. The calculations take into account only the period when the version is active, and its periodicity. The times when the periodical events are active in the time specified range $[t_s, t_e]$ is performed as a pair $\langle [t_s, t_e], C \rangle$ and the set of the TE^* values in the time specified range $[t_s, t_e]$ is performed as the extension function Bounded Extension $BExt([t_s, t_e], C)$. Applying the introduced designations and functions, the temporal element for the C periodicity in the time range $[t_s, t_e]$ is determined in the following way:

$$TE^* = BExt([t_s, t_e], C) \cup EX([t_s, t_e], C) - EX([t_s, t_e], \neg C), \quad (11)$$

where $EX([t_s, t_e], C) \subseteq EX(C)$ is the set of the days-exclusions with the obtained periodicity *C*, intervening in the period of schedule version functioning, in other word, $EX([t_s, t_e], C) = \{ex^{(C)} \mid ex^{(C)} \in [t_s, t_e]\}$; $EX([t_s, t_e], \neg C)$ is the set of the days-exclusions without the property *C*.

There is the example of the calculation of the basic value TE^*.

Example. The certain train runs on the even dates of the month, but not on Tuesdays. It is needed to determine the dates of the train running in the diapason $[18.11.2007; 20.11.2007]$. The specified property of periodicity has the following form: $C = pd \wedge \neg Tue$. The calculations of the temporal element for the specified diapason can be presented in the following form:

$$TE^* = BExt([18.11.2007; 20.11.2007], pd \wedge \neg Tue) - EX([18.11.2007; 20.11.2007], nd \vee Tue). \quad (12)$$

The predicate $C(x) = (x \text{ is } pd) \wedge (x \text{ is } \neg Tue)$ is applied to every element of the determined range:

$$C('18.11.2007') = [('18.11.2007' \text{ is } pd) \wedge ('18.11.2007' \text{ is } \neg Tue)] = 1 \wedge 1 = True;$$
$$C('19.11.2007') = [('19.11.2007' \text{ is } pd) \wedge ('19.11.2007' \text{ is } \neg Tue)] = 0 \wedge 1 = False;$$
$$C('20.11.2007') = [('20.11.2007' \text{ is } pd) \wedge ('20.11.2007' \text{ is } \neg Tue)] = 1 \wedge 0 = False.$$

There are no exceptions for this period: $EX([18.11.2007; 20.11.2007], nd \vee Tue) = \varnothing$. Consequently, there is only one date satisfying the determined property in the specified time range $x = '18.11.2007'$: $TE^* = \{'18.11.2007'\}$.

The function $BExt([t_s, t_e], C)$ determines the points on the time axis (dates) within the frameworks of the time range $[t_s, t_e]$, responding the periodicity sign *C*. The verification of every point $x \in [t_s, t_e]$ for correspondence to the logical expression *C* is the basis of this function. The result $C(x)=true$ means that the day x satisfies the schedule sign C, and $C(x)=false$ means that the version of the schedule is not active on x day. Then the set of points on the time axis within the time range $[t_s,t_e]$, responding to the periodicity sign C, is generated by the following way:

$$(\forall x \in [t_s,t_e])C(x) \Rightarrow x \in BExt([t_s,t_e],C), \tag{13}$$

As an example of calculation in accordance with the formula (13) the date definition in the time range $[t_s,t_e]$ is considered. These dates are the working days and the even days of the month. The task solution can be presented in the following way.

$$(\forall x \in [t_s,t_e])[pd(x)=true \wedge wd(x)=true] \Rightarrow x \in BExt([t_s,t_e], pd \wedge wd). \tag{14}$$

In general case in the process of computation of the predicate $C(x)$, it is necessary first to calculate all the predicates $p(x)$ comprised in it. Then the logical operations $\vee, \wedge$ and $\neg$, formulating the expression C, are applied to the results.

In the process of computation of the majority of $p \in P$ it is enough to know the checking point coordinates on the time axis. For example, under the condition of knowing the date it is possible to compute the day of week, and accordingly check it on the correspondence to the periodicity characteristics *Mon, Tue, Wed, Thu, Fri, Sat* and *Sun*, and employing the rule of correlation of the days of week to the determination of the days-off, it is possible to check it on a correspondence to the properties *wd* and *rd*. However there are such periodicity characteristics, for checking which it is not enough to have the coordinates of the point under examination only, but it is necessary to know about the periodicity properties of other days. The very first and the very last days of the weekend can serve as examples of such “complicated” periodicity. It is necessary to know about the periodicity characteristics of the neighboring days for their determination: the day before the date under examination for the first day of the weekend and the day after the date under examination for the last day of the weekend. The complexity is highlighted by the fact, that for these connected days some exchanges or transfers might be assigned, and it is necessary to take into consideration the periodicity characteristics with the account of all these changes.

The second step is implementation of the calculation of the actual value of the temporal elements, employing the basic value TE^*. The actual value of the temporal element $TE_i^{(n)}$ of the schedule version, besides the properties of the periodicity and the facts of the dates transfer, takes into consideration the cases of the timetables overlapping. If such conflicts are detected, the identical dates are canceled in the temporal element of the schedule version with the earlier fixation time t_f corresponds to. Then the calculation of the element $TE_i^{(n)}$ presents the recursion function. The recursiveness is shown in the fact that the calculation of the temporal element of one version comprises the temporal elements of other subsequent versions, which, in their turn, are determined in the same way:

$$TE_i^{(n)} = (TE_i^* - \bigcup_{v_j^{(n)} \in V(v_i^{(n)})} TE_j^{(n)}) \cup z_i^{(-)} - z_i^{(+)}, \tag{15}$$

where TE_i^* is the TE^* for i-th version of the schedule; $V(v_i^{(n)})$ is the set of versions of the schedule of the train with number n, functioning of which occupies the part of the period of the version $v_i^{(n)}$ operation and the time of their fixation is subsequent compared to the version,

$$V(v_i^{(n)}) = \{v_j^{(n)} \mid v_j^{(n)} \in V^{(n)}; ([v_i^{(n)}.t_s, v_i^{(n)}.t_e] \cap [v_j^{(n)}.t_s, v_j^{(n)}.t_e] \neq \varnothing) \wedge (v_j^{(n)}.t_f > v_i^{(n)}.t_f)\}; \qquad (16)$$

$\bigcup_{v_j \in V(v_i^{(n)})} TE_j^{(n)}$ is the set of the dates, when the versions of the schedule are active $v_j^{(n)} \in V(v_i^{(n)})$, overlapping the schedule $v_i^{(n)}$ and having the higher priority compared to it;

$z_i^{(-)}, z_i^{(+)}$ are the set Z elements, adjusting the temporal element of the version with the account of changes of type:

$z_i^{(-)} = \{d_1 \mid z = \langle d_1, d_2 \rangle, z \in Z, d_2 \in TE_i^{(n)}\}$ is the set of the dates, which are to be excluded from the temporal element of the version $v_i^{(n)}$, because there is exchange assigned for them;

$z_i^{(+)} = \{d_1 \mid z = \langle d_1, d_2 \rangle, z \in Z, d_1 \in TE_i^{(n)}\}$ is the set of the dates, which are to be comprised in the temporal element of the version $v_i^{(n)}$, because its temporal element $TE_i^{(n)}$ contains the date, the timetable of which is the standard sample of the date transfer.

The condition of exiting the recursion is the achievement of such schedule version $v_l^{(n)} \in V^{(n)}$ for the train number *n*, for which there is no other schedule version $v_j^{(n)} \in V^{(n)}$ for this train, satisfying the following condition:

$$(j \neq l) \wedge ([l.t_s, l.t_e] \cap [j.t_s, j.t_e] \neq \varnothing) \wedge (j.t_f > l.t_f). \qquad (17)$$

Then for the version $v_l^{(n)}$ from (16) there is $V(v_l^{(n)}) = \varnothing$.

It should be noted that after the adjustment basic temporal element in accordance with the formula (15) the temporal elements of different versions of train *n* schedule do not intersect, in other word,

$$TE_i^{(n)} \cap TE_j^{(n)} = \varnothing \text{ for } i \neq j; \quad i, j = \overline{1, g}. \qquad (18)$$

Fig. 3.26 illustrates the transfer of the schedule from the 13th of April, Friday to the 14th of April, Saturday, comprising the exclusion of the 14th of April from the temporal element of schedule #2 for the days-off (*rd*) and inclusion it in the temporal element of schedule #1 for the working days (*wd*).

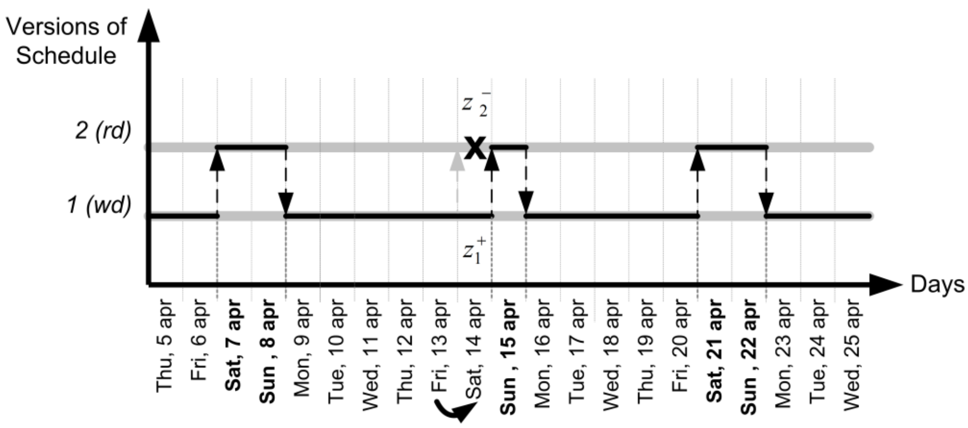


Fig. 3.26. Schedule transfer from one day to another

It is possible to implement this method in two ways: with or without the employment of the recursion.

The procedure of the successive calculation of the temporal element consists of four steps. It is accomplished successively for all versions of the train schedule starting with the latest one (descending grading according to the attribute t_f). Accordingly the algorithm embraces and calculates the temporal elements of all the versions of the train.

Step 0. The selection of all the versions of the train *n* schedule and their classification in the descending order according to the attribute t_f.

Step 1. The calculation of the basic temporal element TE^* considering the periodicity *C* and exceptions *EX* according to the formula (11).

Step 2. Correction of the temporal element according to the formula (15): considering the versions overlapping and alterations in the calendar (the set *Z*). The temporal elements of the more foreground versions (registered in the database later) are subtracted from the temporal element, calculated in the previous step. In other words, the dates, overlapped by the other versions, are excluded from the ranges of the version activity. The temporal elements of these versions have to be calculated earlier on the previous iterations.

Step 3. Transition to the next version. The steps 2 and 3 are repeated for the next version, and so on up to the moment when the last version has been processed.

The procedure of the recursion calculation of the temporal elements can start from any version, and embraces only the versions, introduced to the system after it.

Step 1. Calculation of the basic temporal element TE^*, considering the periodicity *C* and exceptions *EX* according to the formula (11).

Step 2. Correction of the temporal element according to the formula (15): considering the versions overlapping and alterations in the calendar (the set *Z*). The temporal elements of the more foreground versions (registered in the database later) are subtracted from the temporal element, calculated in the previous step. In other words, the dates, overlapped by the other versions, are excluded from the ranges of the version activity. In case the temporal element of the certain version has not been calculated before, the activation of the procedure for this version takes place.

Fig. 3.27 represents the general scheme for the recursive calculation of the temporal element for the version of the train schedule.

The existence of the temporal elements allows considerably simplifying the numerous temporal functions, such as the determination of the active object version, the determination of the points on the axis of the effective time when the exact object version appears from the shadow zone. For *the active version of the schedule of the train with number n determination* it is suggested to employ the function of $ActVer()$ type:

$$ActVer(x, V^{(n)}) = \begin{cases} v_i^{(n)}, & \text{if } x \in TE_i^{(n)}; \\ null, & \text{otherwise} \end{cases}, \tag{19}$$

where *x* is the date for which it is necessary to know the schedule.

The result of the function $ActVer(x, V^{(n)})$ is the element of the set $v_i^{(n)} \in V^{(n)}$, presenting the required active version. If the train does not run on the pointed date *x*, the function returns the *null*[5] value. The condition of non-overlapping of the temporal elements of all train schedule versions guarantees the existence of only one active version.

[5] *Null* is a special marker used in Structured Query Language

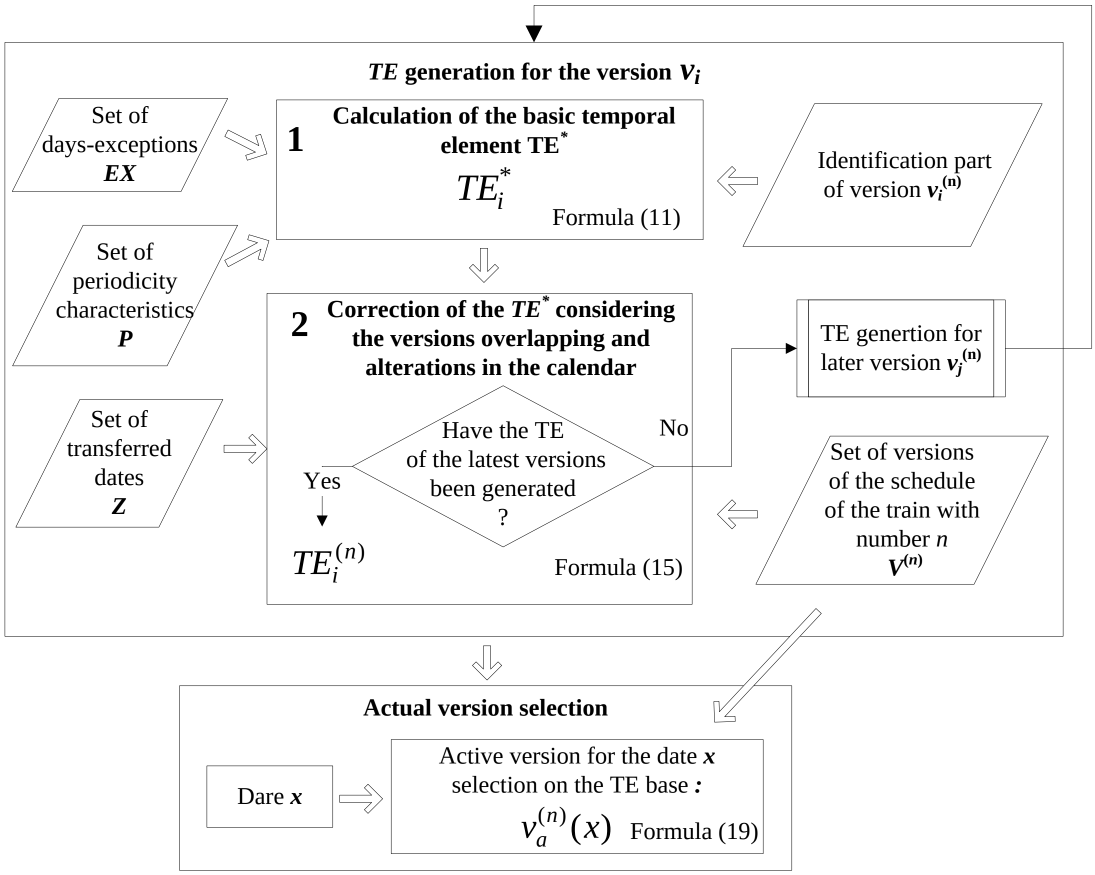


Fig. 3.27. Scheme of the temporal elements generation and the active version calculation

There is *an example of the temporal elements calculation* for the version of the schedule of the train with number *n*. The set of the schedule versions with different periodicity was presented before in Fig. 3.24 and in Fig. 3.25 (see page 104).

Fig. 3.24 obviously presents, that the train has three versions of the schedule $\{v_1, v_2, v_3\}$. It is assumed, that two versions of the timetable v_1 and v_2 in correspondence with the characteristics *wd* and *rd* were assigned for the period from 01.11.2007 to 30.11.2007; the time of these versions fixation was the 10th of October, 2007 correspondingly at 12:00 and 14:00, and the third version v_3 with the characteristic *Tue,Thu* was assigned on the 2nd of November, 2007 for the period from 05.11.2007 to 25.11.2007, then $t_1 =$'01.11.07', $t_2 =$'05.11.07', $t_3 =$'25.11.07' and $t_4 =$'30.11.07'.

Fig. 3.25 demonstrates the time period from 05.11.2007 to 25.11.2007, when three schedules operated; here the day with number 1 on the axis "Day of week" corresponds to the Monday 05.11.07, with number 2 corresponds to Tuesday 06.11.07 and so on.

In accordance with the formula (9) there are the descriptions of the versions of each schedule:

$$v_1=\{\langle n, wd, \text{'10.10.07 12:00', '01.11.07', '30.11.07'}\rangle, \langle s_1, \text{'11:05'}\rangle, \ldots, \langle s_{10}, \text{'12:15'}\rangle\};$$

$$v_2=\{\langle n, rd, \text{'10.10.07 14:00', '01.11.07', '30.11.07'}\rangle, \langle s_1, \text{'11:00'}\rangle, \ldots, \langle s_{10}, \text{'12:12'}\rangle\};$$

$$v_3=\{\langle n, (Tue \vee Thu), \text{'02.11.07 14:00', '05.11.07', '25.11.07'}\rangle, \langle s_1, \text{'11:00'}\rangle, \ldots, \langle s_{10}, \text{'12:21}\rangle\}.$$

It is worth noting that for convenience and simplification of the record only the first stop station of the train and the last one are shown here.

The day 19.11.2007 was announced as a day off in Latvia, and the trains that day ran according to the schedule of the day off. Consequently, in this case there exists the non-vacuous set of exceptions for the days off, or in other words there is: $EX(rd) = \{19.11\}$. It should be mentioned that the year 2007 has not been written in this case and further for the record simplification.

The calculation of the temporal element for the schedule version v_3, determined later than the previous two versions, is produced rather simply according to the (11), as far as the values of the temporal elements of the other versions are not required here:

$$\begin{aligned}
&TE_3 = TE_3^* = BExt([3.t_s, 3.t_e], C_3) \bigcup EX([3.t_s, 3.t_e], C_3) - EX([3.t_s, 3.t_e], \neg C_3) = \\
&BExt([5.11, 29.11], Tue \vee Thu) \bigcup EX([5.11, 29.11], Tue \vee Thu) - \\
&EX([5.11, 29.11], Mon \vee Wed \vee Fri \vee Sat \vee Sun \vee rd) = \\
&\{6.11, 8.11, 13.11, 15.11, 20.11, 22.11\} \cup \varnothing - \varnothing = \{6.11, 8.11, 13.11, 15.11, 20.11, 22.11\}
\end{aligned}$$

For computing the temporal element TE_2 of the schedule version v_2 it is necessary to know the value of the temporal element TE_3 of the version v_3, the only schedule version, fixed later:

$$\begin{aligned}
&TE_2 = TE_2^* - TE_3 = \\
&BExt([2.t_s, 2.t_e], C_2) \bigcup EX([2.t_s, 2.t_e], C_2) - EX([2.t_s, 2.t_e], \neg C_2) - TE_3^* = \\
&BExt([01.11, 30.11], rd) \bigcup EX([01.11, 30.11], rd) - EX([01.11, 30.11], wd) - TE_3^* = \\
&\{3.11, 4.11, 10.11, 11.11, 17.11, 18.11, 24.11, 25.11\} \bigcup \{19.11\} - \varnothing - \\
&\{6.11, 8.11, 13.11, 15.11, 20.11, 22.11\} = \\
&\{3.11, 4.11, 10.11, 11.11, 17.11, 18.11, 19.11, 24.11, 25.11\}
\end{aligned}$$

The schedule version v_1 is the earliest version and for calculating its temporal element TE_1 the temporary elements of the schedule versions v_2 and v_3 are necessary:

$$\begin{aligned}
&TE_1 = TE_1^* - (TE_2 \bigcup TE_3) = \\
&BExt([1.t_s, 1.t_e]], C_1) \bigcup EX([1.t_s, 1.t_e], C_1) - EX([1.t_s, 1.t_e], \neg C_1) - (TE_2 \cup TE_3^*) = \\
&BExt([01.11, 30.11], wd) \bigcup EX([01.11, 30.11], wd) - EX([01.11, 30.11], rd) - (TE_2 \cup TE_3^*) = \\
&\\
&\{01.11, 2.11,\ 5.11\text{-}9.11,\ 12.11\text{-}16.11,\ 19.11\text{-}23.11, 26.11\text{-}30.11\} \bigcup \varnothing - \{19.11\} - \\
&(\{3.11, 4.11, 10.11, 11.11, 17.11, 18.11, 19.11, 24.11, 25.11\} \bigcup \\
&\{6.11, 8.11, 13.11, 15.11, 20.11, 22.11, 27.11, 29.11\}) = \\
&\{01.11, 2.11, 5.11, 7.11, 9.11, 12.11, 14.11, 16.11, 19.11, 21.11, 23.11, 26.11, 28.11, 30.11\}
\end{aligned}$$

One of the most serious problems of the temporal elements employment is the *maintenance of the temporal elements* in the consistent condition, because in the case of the schedule changes appearing the versions temporal elements might become out-of-dated and they are to be re-calculated. However some trains continue running for several years, and have the numerous versions of their timetable, the part of which reckon only towards the past, and the introduced changes, affecting the present and the future, in no way are reflected on the out-of-dated temporal elements. In other words, the range, affected by any changes, can be not overlapped by the life cycle of the version or be overlapped only partially. In this case it is unreasonable to upgrade the temporal elements of all train schedule versions, or to re-calculate them entirely, in general. The author has proposed *the method of the partial recalculation of the temporal elements*.

The diapason, affected by the alterations, is further called *the area of the temporal elements recalculation*. Fig. 3.28 illustrates the example of overlapping the area of *the* temporal elements recalculation on the life cycles of the versions of the train schedule. It is seen how the area of recalculation only partially affects the versions of the train schedule v_1, v_2 and v_4, and the version v_3 is not affected at all.

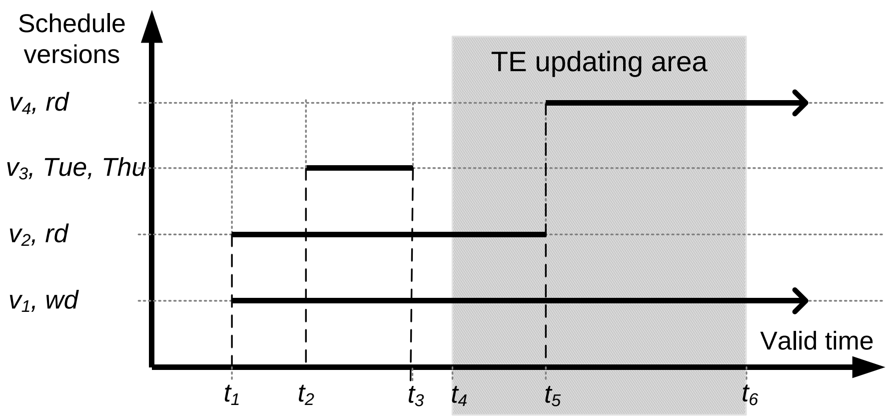


Fig. 3.28. Area of the temporal elements recalculation

There exist different changes, and the minimum area of recalculation for them is different:

- *Changes in the calendar* (alterations, removals) require the partial recalculation of the temporal elements of all versions of all trains, the time of activity of which falls on the dates with the changed properties of periodicity.
- *Including the new version of the train schedule*, *changing the diapason of activity* or *the periodicity characteristics* for the version of the train schedule requires the full or the partial recalculation of the temporal elements of the versions of this train that occur in the diapason, affected by the changes.

When all the temporal elements have already been formed, it is necessary to maintain them in the consistent state. The following procedures are suggested for it:

Formulae (20) and (21) describe the necessary exposure on the temporal elements of all versions of the set V *in case of appearing the changes in the calendar*, in other words in case of the new element $z = \langle d_1, d_2 \rangle$ appearance in the set *Z*.

The date d_2 is added to the temporal elements of all the train versions, comprising the date d_1 (the date which is assigned for transition of the schedule from the date d_2):

$$TE_i^{(n)} = (TE_i^{(n)} \cup d_2 \mid d_1 \in TE_i^{(n)}) \text{ for all } v_i^{(n)} \in V^{(n)}, \quad (20)$$

where $TE_i^{(n)}$ is the temporal elements of the version $v_i^{(n)}$.

The date d_2 is excluded from the temporal elements of all the train versions, comprising the date d_2 (the date, from which the transition of the schedule is assigned):

$$TE_i^{(n)} = (TE_i^{(n)} - d_2 \mid d_2 \in TE_i^{(n)}) \text{ for all } v_i^{(n)} \in V^{(n)}. \tag{21}$$

The formulae (22) and (23) describe the necessary exposure on the temporal elements of all versions of the set $V^{(n)}$, connected with the operations over the train *n* versions (the change of the diapason of activity, of the periodicity property, removal, addition of the new version). In case of these operations occurrence all the train *n* versions, affected by the area of recalculation of the temporal elements. The area of recalculation is computed in the following way:

$$[\tau_s, \tau_e] = [k^{(old)}.t_s, k^{(old)}.t_e] \cup [k^{(new)}.t_s, k^{(new)}.t_e], \tag{22}$$

where τ_s and τ_e are correspondingly the dates of the beginning and the end of the diapason under renewing, the marks *old* and *new* mean the old and the new values of the diapason operation of the version $k \in V^{(n)}$, starting with which the operation has been produced. In case of adding the new version the diapason $[k^{(old)}.t_s, k^{(old)}.t_e]$ is empty, and vice versa in case of removal the old version the diapason $[k^{(new)}.t_s, k^{(new)}.t_e]$ is empty.

Nota Bene. The area of recalculation of the temporal elements has to be extended up to the maximum amount of the dependent days in the elementary attributes of the periodicity (for example, the attribute of the last day of weekend *lh* depends on the previous and the next days, and the number of the dependent days is three).

Substitution of the part of the temporal elements for all versions $v_i^{(n)} \in V^{(n)}$, if $[t_s, t_e] \cap [\tau_s, \tau_e] \neq \varnothing$ is accomplished in the following way:

$$TE_i^{(n)} = (TE_i^{(n)} - [\tau_s, \tau_e]) \cup TE_i^{(n)}([\tau_s, \tau_e] \cap [t_s, t_e]), \tag{23}$$

where $TE_i^{(n)}([\tau_s, \tau_e] \cap [t_s, t_e])$ is the formula of the temporal elements calculation, analogous to the formula (11), but with substitution of the diapason of the version activity: from $[t_s, t_e]$ to $[\tau_s, \tau_e] \cap [t_s, t_e]$ (the renewing diapason in the temporal element of the version $v_i^{(n)}$).

3.4.2. Logical Rules Method

The essence of this method is in the computation of the active version of the timetable on the basis of the logical rules, taking into consideration all the peculiarities of the schedule: multi-version, provoked by the functioning changes, and polyinstantiation in case of the periodical schedule, exchanges and transfers. The method functions with the data of the schedule operative temporal database and begins taking into account all the changes in the timetable at the moment when they are introduced to the system (i.e. on-line).

The method calculations apply the set of the schedule versions $V^{(n)}$ for the train with number *n*, the set of exclusions *EX* and the set of exchanges *Z*. The train schedule version is performed according to the formula (9), and only its identification part $\langle n, C, t_s, t_e, t_f \rangle$ is used in the calculations.

The method central parameter, relatively to which all the logical rules, built into the algorithm, are employed, is the day *x* – the date for which it is necessary to determine the active version of the train schedule. In the connection with the specific calendar existence it is essential to check the date x for the purpose of re-determination of the periodicity properties and exchanges. In case they have been discovered the date x is to be re-determined into the date $\hat{x}$ according to the formula (24) and the further computations are to be done relatively $\hat{x}$:

$$\hat{x} = \begin{cases} d_1^{\{i\}}, & \text{if} \quad \exists z_i \in Z \mid z_i = \langle d_1^{\{i\}}, d_2^{\{i\}} \rangle, d_2^{\{i\}} = x; \\ x, & \text{otherwise.} \end{cases} \tag{24}$$

Only the following schedule versions can be valid for the date $\hat{x}$: first, the versions, obtaining the range of functioning capable of covering the date under the question, and, second, the periodicity property complying with this date. Consequently, it is essential to single out of the set of the train n schedule versions only the versions corresponding to the functioning time and to the periodicity, in other word performing on the day $\hat{x}$ and conforming to the property C. There is the version complying the periodicity of the day $\hat{x}$; it is the version for which the set of its predicate $C(\hat{x})$ verity on the functioning field of this version $[t_s, t_e]$ comprises the date $\hat{x}$, in other word, $\hat{x} \in \{y \mid y \in [t_s, t_e], C(y)\}$. However it is not necessary to compute the truth set, it is quite enough to apply the statement C itself to the day $\hat{x}$. In case the result is the veritable one, the version satisfies the condition of the day $\hat{x}$ periodicity. Thus, the set of all versions corresponding to the functioning time and the periodicity of the date x is sorted out of the set of the train schedule versions $V^{(n)}$ in accordance with the following formula

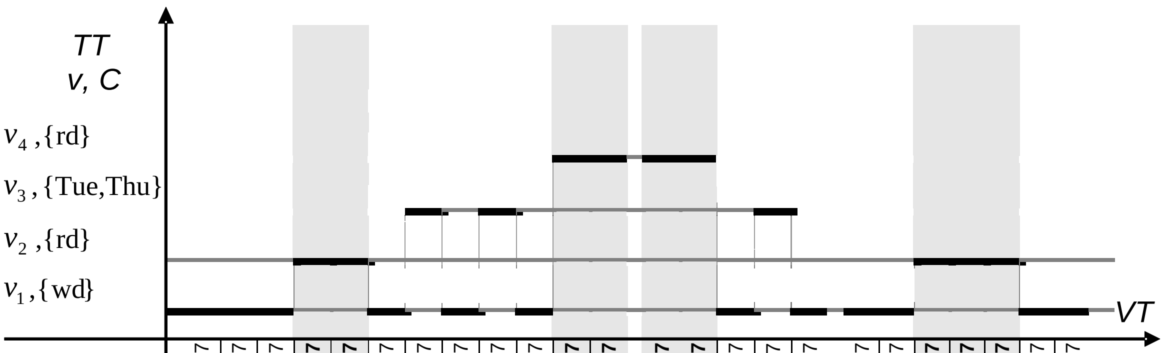
$$V_{\hat{x}}^{(n)} = \{v \mid v \in V^{(n)}, \hat{x} \in [t_s, t_e], C(\hat{x})\}. \tag{25}$$

For the taking into account of the periodicity exclusions for the proposition C the predicates $p_i(\hat{x})$ are exchanged by the $\hat{p}_i(\hat{x})$. The predicate $\hat{p}_i(\hat{x})$ in its essence corresponds to the predicate $p_i(\hat{x})$ described in the point 3.4.1, but before executing the computations, it performs the preceded additional inspection oriented on the detection of the exclusions existence. In case of discovering such date $\hat{x}$ the characteristic p_i is given to it, and, consequently, the predicate $\hat{p}_i(\hat{x})$ will be truth. The formula below performs it:

$$\hat{p}_i(\hat{x}) = \begin{cases} True, & \text{if} \quad \hat{x} \in EX(p_i), EX(p_i) \subseteq EX; \\ p_i(\hat{x}), & \text{otherwise.} \end{cases} \tag{26}$$

Nota Bene. In case the scheduling system employs the periodicity properties, dependent on the periodicity properties of the neighboring dates, such as *lh* is “the last day-off of the week” and *fw* is “the first working day of the week”, the predicate formulae $p_i(\hat{x})$ and $\hat{p}_i(\hat{x})$, comprised in C, have to be extended to $p_i(x, \hat{x})$ and $\hat{p}_i(x, \hat{x})$ respectively. It is impossible to detect accurately the periodicity properties of the neighboring dates, obtaining the information on $\hat{x}$ only.

Fig. 3.29 gives the example of overlapping the different characteristics of the periodicity P on the axis of valid time, changing its detached points (x and $\hat{x}$). The figure demonstrates the shift of the property *lh* on one day 19.11.2007.



x	…	02.01.2007	03.01.2007	04.01.2007	**05.01.2007**	**06.01.2007**	07.01.2007	08.01.2007	09.01.2007	10.01.2007	11.01.2007	**12.01.2007**	**13.01.2007**	..	**18.08.2007**	**19.08.2007**	20.08.2007	21.08.2007	22.08.2007	…	15.11.2007	16.11.2007	**17.11.2007**	**18.11.2007**	**19.11.2007**	20.11.2007	21.11.2007	…
$\hat{x}$	…	02.01.2007	03.01.2007	04.01.2007	**05.01.2007**	**06.01.2007**	07.01.2007	08.01.2007	09.01.2007	10.01.2007	11.01.2007	**12.01.2007**	**13.01.2007**	..	**18.08.2007**	**19.08.2007**	20.08.2007	21.08.2007	22.08.2007	…	15.11.2007	16.11.2007	**17.11.2007**	**18.11.2007**	**18.11.2007**	20.11.2007	21.11.2007	…
p_1	…	3	4	5	**6**	**7**	1	2	3	4	5	**6**	**7**	..	**6**	**7**	1	2	3	…	4	5	**6**	**7**	**7**	2	3	…
p_2	…	wd	wd	wd	**rd**	**rd**	wd	wd	wd	wd	wd	**rd**	**rd**	..	**rd**	**rd**	wd	wd	wd	…	wd	wd	**rd**	**rd**	**rd**	wd	wd	…
p_3	…	PD	ND	PD	**ND**	**PD**	ND	PD	ND	PD	ND	**PD**	**ND**	..	**PD**	**ND**	PD	ND	PD	…	ND	PD	**ND**	**PD**	**PD**	PD	ND	…
p_4	…					**lh**							**lh**	..		**lh**				…					**lh**			…
p_5	…	*	*	*	*	*	*	*	*	*	*	*	*	..	*	*	*	*	*	…	*	*	*	*	*	*	*	…
$v_a^{(n)}(x)$	…	v_1	v_1	v_1	v_2	v_2	v_1	v_3	v_1	v_3	v_1	v_4	v_4	..	v_2	v_2	v_1	v_3	v_1	…	v_1	v_1	v_2	v_2	v_2	v_1	v_1	…

Fig. 3.29. Example of overlapping the different characteristics of the periodicity P on the axis of valid time

In case of the schedule overlapping versions existence (the Historical-Overlapped) the set $V_{\hat{x}}^{(n)}$ might contain more than one element, and this situation is not sufficient for the requirement of only one active version existence. Consequently it is essential to single out the sole element – the active version – from the $V_{\hat{x}}^{(n)}$. In the course of calculation the priority is given to the version which has been introduced to the system after all the others:

$$v_a^{(n)}(x) = (v \mid v \in V_{\hat{x}}^{(n)},\ t_f = \max_{V_{\hat{x}}^{(n)}}\{t_f\}), \tag{27}$$

where $\max_{V_{\hat{x}}^{(n)}}\{t_f\}$ is the function of determining the maximum value of t_f in the members of the set $V_{\hat{x}}^{(n)}$.

The algorithm of the search of the active version is schematically shown in Fig. 3.30. There is a description of the main steps of this algorithm:

Step 1. The date correction. The examination of the alteration assignment on the specified date x. In case of discovering it, the following calculations are done regarding this the date of alteration. The date which has been corrected or has been simply checked according to the formula (24) is named as $\hat{x}$.

Step 2. Procedure of forming the set $V_{\hat{x}}^{(n)}$. The selection of the versions of the train n schedule, suitable according to the time of validity and periodicity to the date $\hat{x}$ by formula (25).

Step 3. Choice of the active version. The set $V_{\hat{x}}^{(n)}$, obtained before, can comprise more than one version, because the operative alterations of the schedule could have been introduced. The version, which has been introduced to the system the latest, and, consequently, is the active one, is chosen among all the versions, included in the set $V_{\hat{x}}^{(n)}$ according to the formula (27). The version of the train *n* schedule active on the date x is $v_a^{(n)}(x)$.

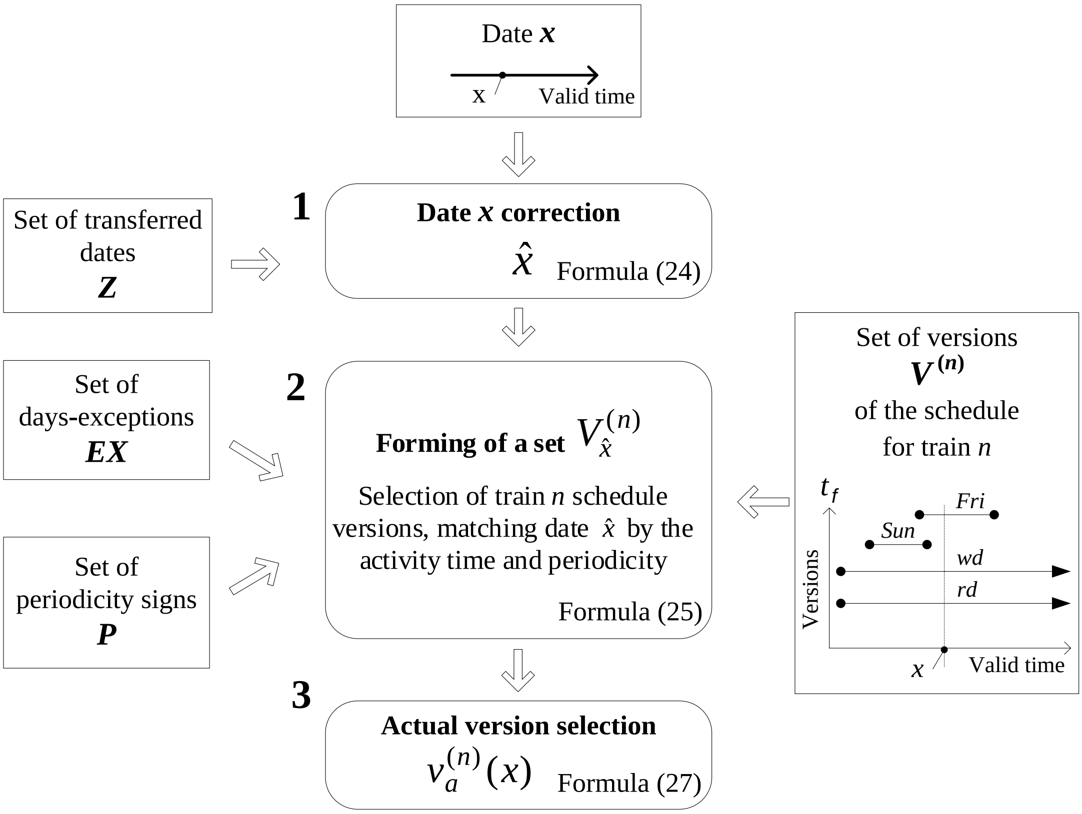


Fig. 3.30. Scheme of searching for the active version of the train schedule employing the Logical Rules method

The merit of this approach is the fact that all the computations take place on the basis of the operative data and all the changes are taken into account by the queries to the schedule at once after their introducing to the system. Concerning the shortcomings of the method, it should be noted that every time the above described searching algorithm have to be executed all over again, in case it is necessary to execute any query to the system, requiring the knowing of the active version of the train schedule.

3.4.3. Comparison of the Developed Methods in the Temporal System of the Railway Schedule

The research performs the approbation of the suggested methods of the determination of the object active version in the information systems of the trains traffic schedule on the Latvian Railway and the work defines the efficiency of their application for the solution of different tasks, emerging in the practical operation. The procedure of the investigation in question comprises three steps:

1. Defining the information systems functional tasks, determining the object active versions, and choosing the tasks awaiting for solution;

2. Organizing the system of the method efficiency indicators for the object active version determination and the choice of their estimation method;
3. The methods approbation in the process of the information systems chosen tasks solution, their efficiency estimation and the elaboration of the guideline for their practical application.

These stages are considered in details.

The analysis of the queries taking place in the various systems of the railway schedule has permitted to formulate *the following list of the principal tasks*, referring to the procedures of manipulating the versions of the railway schedule, and it is suggested employing the developed methods:

- the process of searching the train timetable active version for the specified day;
- generating the sequence of the versions changing for the specified range (for example, the schedule for a month);
- the process of searching the intervals of the determined version operation
- the process of searching the schedule immediate changes;
- the revelation of the time points (the dates) of the schedule changes in the specified range.

For the assessment of the efficiency of employing the methods of determining the active version of the object the following *indicators* have been chosen:

- the complexity of the active version searching;
- the performance in the process of the active version searching;
- the volume of the data processed;
- the volume of the stored data;
- the necessity of the preliminary data preparation for employing in the queries after the schedule changes;
- the necessity to execute the restrictions on the time area for queries;
- the operational efficiency of the changes introduction;
- providing the data integrity in the process of the schedule changing.

There is the brief examination of these indicators and the accomplishment of their evaluation for the methods under consideration.

The complexity of the process of searching of the active version in general case can be estimated by the number of executed operations. The logical rules method requires the employment of the suite of the logical rules requiring at least seven logical verifications, and their number depends on the versions number and their *C*. The average number of operations can be calculated according to the following formula: $7+|\overline{V^{(n)}}|\cdot\overline{C}$, where $|\overline{V^{(n)}}|$ is the average number of the single train schedule versions (the set cardinality is $\overline{V^{(n)}}$), $\overline{C}$ is the expression central area *C* (the average number of the components in *C*). For the schedule of the Latvian Railway this parameter equals 15 (in 2004 the value of $|\overline{V^{(n)}}|$ was 6 versions, in 2005 it were 4, in 2006 – 3, and in 2007 – 4 versions, whereas $\overline{C}$ = 1.8). The process of searching in the temporal elements method performs only one elementary verification operation for examining entry in the temporal elements.

The time of searching for the active version depends on the peculiarities of the implementation, the databases management system, and so on. This indicator value inevitably correlates with the number of logical verifications. Therefore, the temporal elements method unquestionably gains the advantage in this position.

The volume of the processed data is determined by the data set, requiring the system processing to conclude about the version activity. If the logical rules method is employed, it is necessary to operate with the periodicity characteristics, the periods of activity of all train versions. In case the method of the temporal elements is implemented, it is necessary to operate with the temporal elements of all the train versions. The volume of the processed data can vary depending on the situation but in the majority of cases the temporal elements create the bigger volume.

The volume of the stored data is compared according to the identification part of the schedule, in other words, without taking into account the data about the stops at the stations, as far as the volume of the data for the mentioned methods implementation is equal. In the described situation the comparison is performed on the basis of the actual real timetable of the passenger trains on the "Latvian Railway' and the hypothetical schedule which models the maximum lengths of the temporal elements versions with the employment of the periodicity characteristics *pd*/*nd*/*wd*/*rd* (see Table 6). The maximum size of the temporal elements for the versions is limited by the length of the calculated diapason. Consequently, the amount of the components (dates) in the temporal elements, determined for one year, does not exceeds 366. The size of one component (the date) takes from 4 to 8 bits depending on the system of storage (coding). Therefore the maximum volume of one version of the temporal elements for one year takes about 2-4 KB.

Table 6. Data volume of the identification part of the schedule for one year

Schedule	Method LR	Method TE
The real actual schedule on the Latvian Railway: Year 2007: 476 trains, 1928 schedule versions; Year 2004: 389 trains, 2031schedule versions.	~27KB	~250KB
Hypothetical schedule: 400 trains with the characteristics of periodicity *pd*/*nd* or *wd*/*rd*, running for one year.	~27KB	~2000KB

The necessity of the preliminary preparation of the data to the employment in queries after the changes in the calendar, the process of carrying out these alterations, appearance of new version of the timetable. In case of using the method of logical rules the prerequisite data processing is not demanded, because the rules automatically take into consideration all the alterations. If there an application of the temporal elements method takes place, the preliminary data preparation is obligatory. This fact creates the negative impact on the designing production cost, as well as implicates the necessity to support the integrity of the temporal elements. Moreover, the operation of calculation of the temporal elements takes significant time (see below: the changes introduction promptitude).

The necessity of maintaining the restrictions of the temporal area for queries. The certain schedule versions have no time limit of their operation, consequently, it is impossible for them to form the completed temporal element. That is why any queries to the schedule system require establishing the permissible temporal area with the length, for example, of a month or a year, in the frameworks of which the information on the schedule will be provided. Accordingly, exactly in the frameworks of this temporal area it is required to provide the temporal elements formation. The method of logical rules does not require such limitations for the queries.

Promptitude of the changes introduction and provision of the data integrity in the process of changing or delay. The method of temporal elements requires the data preliminary preparation for making the changes accessible. Just after the changes introduction the part of the temporal elements transit into non-coordinated state.

The results of the comparative efficiency estimation and the possibilities of the employment of the methods of the object active version determination in the information systems of the trains traffic schedule is performed in Table 7. Every of the indicators possesses the value of "+" or "-": it takes the value "+" if the method preponderates over its rival in this characteristics, and the value "-" otherwise.

Table 7. Estimation of the comparative efficiency of the methods of determination of the object active version

Efficiency Indicators	Method LR	Method TE
The complexity of the active version searching (the number of operation in the process of searching)	– (> 7)	+ (1)
The performance in the process of the active version searching	–	+
The volume of the data processed in the process of the active version searching	–	+
The volume of the stored data	+ (~27KB)	– (~2000KB)
The necessity of the preliminary data preparation for the application to the queries after the changes introduction to the schedule	+	–
The necessity to execute the restrictions on the time area for queries	+	–
The operational efficiency of the changes introduction	+	–
Providing the data integrity in the process of the schedule changing	+	–

On the final stage of the research under consideration there was accomplished the estimation of the efficiency of implementing the suggested methods for solving the practical tasks, existing in the system of the railway scheduling and have been stated above. As a result there were received the certain recommendations on their practical implementation, and these recommendations are represented in the Table 8, where the symbol "+" signalizes the recommended method.

Table 8. Evaluation of the possibility of implementing the methods of determination of the object active version

Applied tasks of the scheduling system	Method LR	Method TE
The search of the active version of the train schedule on the specified day	+	+
Arranging the consequence of the versions changes for the specified diapason	– (for the broad ranges) + (for the narrow ranges)	+
The search of the intervals of the exact version activity	–	+
The search of the nearest change in the schedule	–	+
The identification of the dates (the time coordinates) of schedules changes in the specified diapason	–	+

It is worth noting that all the tasks, but the very first one, are connected with operating the active versions only. Nevertheless the times of the version activity in the method of logical rules are unknown beforehand, and they are to be calculated for every day. Therefore there needed computing of the active version for every point in the specified diapason. For example, for forming the consequence of the schedule versions alteration for a week time it is necessary to accomplish the algorithm of calculating the active version for 7 times. The time of computation is directly proportional to the diapason length (see Fig. 3.31). This solution is quite accessible for small diapasons. The method of the temporal elements assumes rather simple calculation as all the necessary data have already been computed in the versions temporal elements. All the data in the specified diapason are already associated with the schedule versions and the only necessary thing is sorting the versions in the needed order. The length of the time diapason does not have any significant impact on the time of the result delivery.

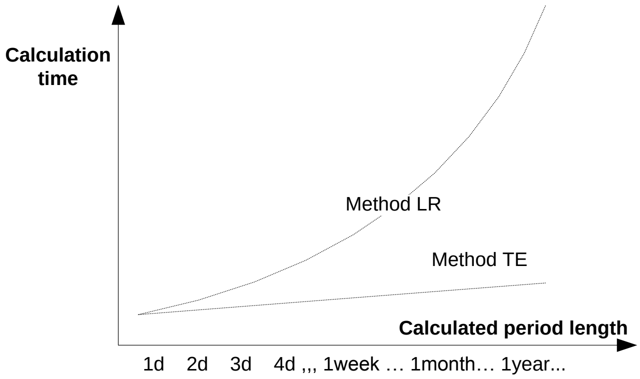


Fig. 3.31. Dependence of the method efficiency on the amount of processed days in the query

Depending on the peculiarities and the purpose of the scheduling system the various practical tasks have different weight, and the certain tasks can be highly sought but some of them cannot be employed at all. Preferring this or that method depends on the fact what position the above mentioned tasks take in the system.

3.5. Method of the Interval Data Transformation

The models of the multi-version data representation, employed in the transport information systems, usually save data in the interval form which is inconvenient for the analytical calculations carrying out. The majority of the employed data analysis methods and correspondingly the majority of the analytical systems function with the point forms of the data representation (for example, with the time sequences), but not with the interval forms. In the point form the fact is associated with the set of the time moments of the previously chosen granularity (for example, with the days, months, or years). The procedures over the described sets are very convenient for the fundamental computations or for the formal characterization. However, the mathematical apparatus of the relation databases management systems does not provide the ready-to-use facilities for the transformation of the interval form into the point one, and this fact required the elaboration of the appropriate procedures to be performed by the investigation under consideration.

Within the framework of the set task, the aggregate scheme of the interval temporal data transformation was designed. This scheme is demonstrated in Fig. 3.32.

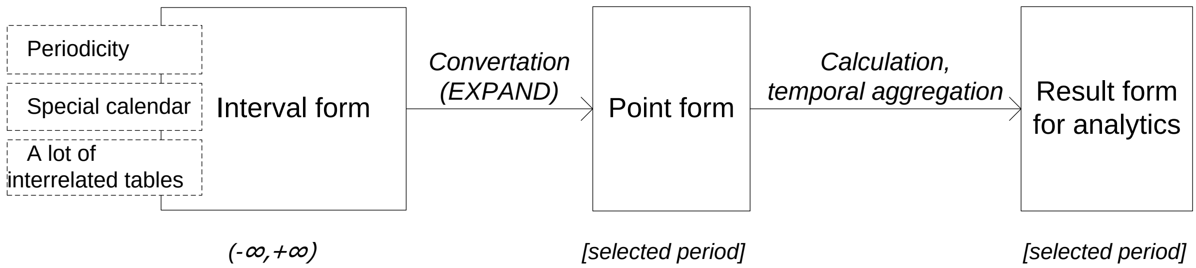


Fig. 3.32. Scheme of the temporal data transformation

There is the explanation of the presented scheme.

The classical interval form (*Interval form*) is initially inconvenient for the range of the certain calculations, but their complexity increases multifold when the following aspects exist: the periodicity, the special calendar, the necessity to use the several interconnected tables in the calculations[6].

The application of the procedure EXPAND to the interval form temporal data, suggested by the author, solves the problem of the inconvenient data presentation for the computations, by the way of turning the interval facts into the time sequences, in which every value gives characteristics of this fact existence in the definite moment of time (*Point form*). Then the data undergo the further processing, often connected with the temporal aggregation procedures. This stage might require even more serious computations connected with the temporal concepts representation. The required data form (*Result form for analytic*) is the final result of the processing procedure.

The initial sets for every stage embrace the different time periods. The set of the interval form temporal data (the first block in Fig. 3.32) is capable of covering the unlimited time range (-∞,+∞). With the transfer to the point form (the second and the third blocks in Fig. 3.32) the time area of the temporal data is narrowed to the field of analysis (selected period), and this fact considerably contracts the data bulk in the computations and substantially saves the computing resources, and solves the problem of versions existence with the indefinite values of the functioning periods beginning and ending.

There is the observation of the suggested method of the interval data transformation in the point form. There are two relations:

R1 is the temporal relation of the interval form, describing the set of the certain object versions and having the scheme $R1(A, t_s, t_e, C)$, where $A = \{a_1, a_2 \ldots a_k\}$ is the set of the attributes, describing the object, k is the number of attributes. Every version has the functioning period from the moment t_s to the moment t_e and possesses the property of periodicity C. Besides, the activity moments of this or that versions depend on the specific calendar Z, describing one day properties transfer to the others (the additional days-off or additional working days, etc.);

[6] The information systems objects in the databases, according to the relation theory are represented in the form of the interconnected tables. That is why receiving the demanded result, as a rule, requires processing procedure for several tables.

R2 is the point form temporal relation with the scheme *R2*(*A*,*d*), where *d* is the time moment when fact *A* is active.

It is necessary to reorganize $R1(A,t_s,t_e,C)$ into *R2*(*A*,*d*), and the data set *R2* has to be restricted by the fixed time period (specified period for analysis) $T^{(q)}=[t_s^{(q)},t_e^{(q)}]$, where $\forall d \in T^{(q)}$.

The transformation formula has the following view:

$$R2 = \sigma_{d\in T^{(q)}}(R1 \times \tau(t_s,t_e,C)) , \tag{28}$$

where $\tau(t_s,t_e,C)$ is the function distributing the range $[t_s,t_e]$ on the dates *d*, set, responding the requirement of the specified periodicity *C*, in other words

$$\tau(t_s,t_e,C) = \{d \mid d \in [t_s,t_e], p(d,C) = true\} ; \tag{29}$$

$p(d,C)$ is the periodicity predicate, which is true in the case if the date *d* has the property *C*. The predicate takes into consideration the specific calendar *Z* (for example, the transfer of the weekends schedule on Friday and on Monday in the connection with the national holidays in Latvia on the 1st and on the 2nd of May in year 2009);

$R1 \times \tau(t_s,t_e,C)$ is the construction, implementing the procedure of expansion EXPAND; this procedure is the central operation of the entire transformation and provides the transfer from the interval form to the point one. The selection $\sigma_{d\in T^{(q)}}(\ldots)$ restricts the result by the analysis area $T^{(q)}$. The cardinality of the resulting set can reach $| R1 | \cdot | \tau(T^{(q)}) |$.

After executing the expansion procedure the following reorganizations (grouping, the natural combination, the projection) become trivial. The employment of the proposed function $\tau(t_s,t_e,C)$ permits to simplify the computations ultimately. For its implementation the recursion[7] procedure is used and with the help of the language SQL the operation of forming the dates set $D = \{d\}$ of the specified periodicity *C* between the time moments *dStart* and *dEnd* can be represented in the following way:

```
WITH D (d) AS
 ( VALUES dStart
 UNION
 SELECT d + 1 day as d   FROM D WHERE d < dEnd )
    SELECT d FROM D WHERE p(d,C).
```

Then the instruction of the function creating $\tau(t_s,t_e,C)$ has the following view:

```
CREATE FUNCTION TAU (dStart DATE, dEnd DATE, c CHAR(20))
RETURNS TABLE (d DATE)
WITH D (d) AS
 ( VALUES dStart
 UNION
 SELECT d + 1 day as d    FROM D WHERE d < dEnd  )
SELECT d FROM D WHERE p(d,c);
```

[7]The SQL:1999 standard includes support for recursive queries

Example. There is examined the task of procurement of the time series, describing the changes of the trains number on the railway line Rīga – Daugavpils (in both directions) for May 2009.

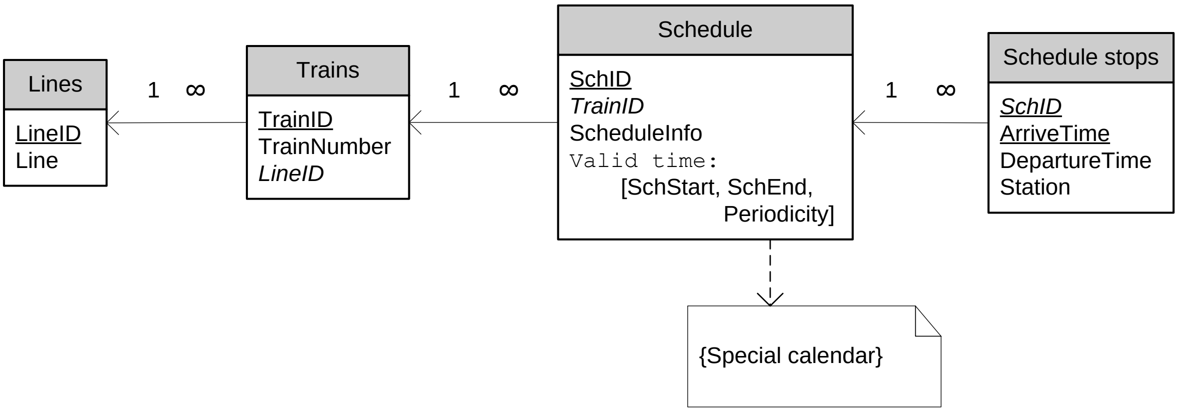


Fig. 3.33. Segment of the database of the passenger trains schedule

The segment of the database comprising the data, necessary for calculations, is presented in Fig. 3.33. The temporal table *Schedule* contains a lot of the trains schedule versions, each of which has its own period of operation from the moment *SchStart* to the moment *SchEnd* and possesses the periodicity characteristics *Periodicity*; this fact is reflected in the corresponding temporal attributes of the valid time *Valid Time*. The moments when this or that version is active depend on the peculiar calendar *Special calendar*, describing the transition of the properties from one days to the other ones (the additional days off or the working days, and so on) and affecting all the schedule versions. The detailed description of the schedule versions (time of the train stops at the stations) is introduced in the table *Schedule Stops*. The tables *Trains* and *Lines* are not the temporal ones and they contain the information on the trains, as well as their belonging to the railway lines.

The structure of the time series under examination is assumed in the form $(Date, Line, TrainQty)$, its temporal area in the form [01.05.2009; 31.05.2009] accomplishing $T^{(q)}$, the period of the schedule activity $[\,schStart,\ schEnd\,]$ accomplishing $T^{(S)}$; the periodicity attribute *Periodicity* accomplishing *C*.

Then the formula of transformation obtains the following form:

$$R(Date, Line, TrainQty) = \pi_{d, Line, COUNT(*)}(Lines \rhd\lhd Trains \rhd\lhd \sigma_{d \in T^{(q)}}(Schedule \times \tau(T^{(S)}, C))),$$

where the construction $Schedule \times \tau(T^{(S)}, C)$ executes the expansion operation. As a result of its accomplishing, 80 original tuples of the interval form of the relation *Schedule* (trains schedule on the line Rīga – Daugavpils) are transformed into 2170 tuples of the point form. Without taking into account other transformation this construction has the following view in the SQL language:

```
SELECT Schedule.*, d
FROM Schedule, TAU(schStart, schEnd, periodicity);
```

construction $\pi_{d, Line, COUNT(*)}(Lines \rhd\lhd Trains \rhd\lhd ...)$ solves the standard task of grouping and the natural joining the connected relations.

The completed record of the SQL-statement for constructing the time series $R(Date, Line, TrainQty)$ has the following view:

```
SELECT d as date, line, count(train) as TrainQty
FROM Schedule, TAU(schStart, schEnd, periodicity), Lines, Trains
WHERE Schedule.trainID = Train.trainID
 AND Trains.LinesID = Lines.LineID
 AND d between '01.05.2009' AND '31.05.2009'
GROUP BY line, d;
```

3.6. Solution of the Issues of Providing the Information Security in the Temporal Databases

3.6.1. Choosing the Level of Implementation of the Information Security of the Temporal Databases

The simplified geography of the information flow from the data storage system to the user obviously demonstrates that the location of implementing the information security can be placed on different levels of this chain (see Fig. 3.34).

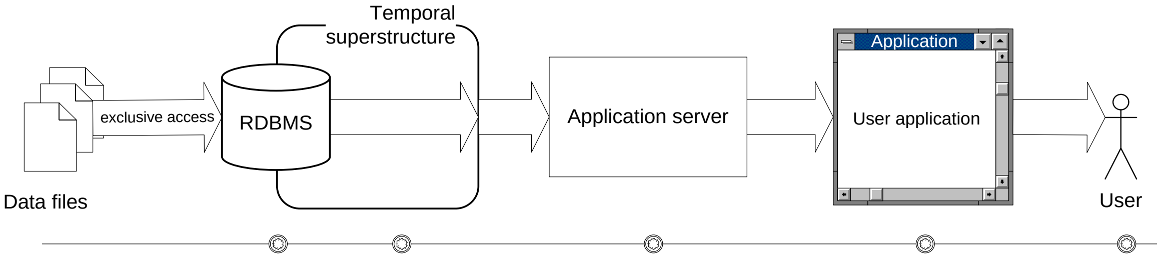


Fig. 3.34. Chain of the information flow in the temporal IS

As practice shows, it frequently happens that the level of the security system implementation, falling beyond the scope of the standard opportunities of the databases management system, is practically *the level of the customer application*. In this case the security system has the obvious gaps, because the customer can gain the access to the data escaping the considered application, and the other program, operating with the same data set, probably has different concept of provision of the data information security or have no concept of it.

The application server level can also provide a good solution of the problem, as soon as it is possible to describe simply and flexibly all the nuances of the safety system, implementing the objective languages. The necessity of the prohibition of accessing the data via another server application is the drawback of the implementation of the safety system on the level of the applications server. To provide this type of the monopolistic exclusive access to the data is not always possible under the circumstances of industrial operation. But implementation and constant support of the unified common interpretation of the safety system in two different applications is a rather complicated and hopeless task.

On *the level of the databases management systems* the opportunities of the flexible description of the safety system can be significantly lower, whereas we have to embrace the limitation of the existing language facilities of the databases management system and the relation data concept. Switching over to the issue of the information security of the temporal data, it is worth mentioning that according to the author's opinion just this level and *the level of the temporal frameworks designed within the bounds of the databases management system* are the most preferable ones, possessing the following advantages:

- The databases management system is already "acquainted" with the data temporal semantics and the safety system employs this acquaintance;
- No user's program or the access to the data by means of the SQL-instructions can avoid this protection.

The method, presented below in the point 3.6.2 is implemented on the level of the databases management system employing the facilities of the active databases; this method complements the temporal frameworks function by provision of the safety policy of the temporal data.

3.6.2. Method of Controlling Access to the Temporal Data on the Level of the Relation Tuple

As it was demonstrated in point 2.4.9, the peculiarity of the temporal data protection lies in the fact that the data of the different time periods are logically disposed in the same place – in the integrated relation table. The delimitation of the users' access to the different time periods, in other word to the subset of the relation tuples, is not comprised in the suite of the standard facilities of the security management of the relation DBMS. The method, suggested by the author, is based on the access limitation to the data of the table exact records for the operation of reading, modification and removal. Employing this method, it is possible to restrict the access to the temporal data in the way allowing the user's access to one record of the table (version) and no access to another one. Only the standard procedures of SQL-2 are accomplished in this method implementation: triggers and presentations, as well as the mathematical apparatus of the system – the relational algebra. The author's works [5, 6, 13, 15] are devoted to developing and practical employment of these methods.

There is the formulation of the key concepts used in the process of designing the method of the access limitation:

Safety policy. The safety policy can be of any type and it depends on the rules, this or that corporation keeps to. The rules can require the classical model of the capability – based safety (the models Bell-LaPadule) or any other model, specific for this exact corporation and based on the complex interconnections between the system objects. For organizing the integrated safety system it should be taken into consideration beforehand, that the safety policy can be quite versatile rest upon the various factors, for example, on the hierarchy of the users, geographical factors, and so on.

The criterion of the access to the record. The protection is organized regarding the tuple, therefore, the tuple has to have a certain characteristics – a label or several labels, according to the values of which the access to the record is admitted or restricted. These labels can appear in the form of the values of either the specified temporal attributes, or other attributes, presented in relation.

The principle of the access restriction. The user has the right to operate with the record if the package of the safety rules, specified for this relation, does not contradict it.

For the task of the access control formal characterization of the following sets is determined:

$\Re = \{R_1, R_2, \ldots, R_n\}$ is the set of database relations, $|\Re| = n$, where R_i, $i = \overline{1,n}$ gains the form of the scheme $R_i(A, L)$;

r is the tuple of the relation R_i, the protection object, $r \in R_i$;

$A = \{a_1, a_2, \ldots, a_k\}$ is the subset of the R_i attributes, having no impact on the access to its tuples, $|A| = k$;

$L = \{l_1, l_2, \ldots, l_m\}$ is the subset of the R_i attributes, determining the access to its tuples, $|L| = m$. There can be the temporal attributes, the specific security attributes or some other attributes or their scalar derivatives among them;

$U = \{u_1, u_2, \ldots, u_z\}$ is the set of the system registered users, $|U| = z$;

$\Theta = \{\theta_1, \theta_2, \theta_3, \theta_4\}$ is the set of the access operations to the tuple, $|\Theta| = 4$,

$\Theta = \{$SELECT, INSERT, UPDATE, DELETE$\}$;

$\boldsymbol{S} = \{S_1, S_2, \ldots, S_h\}$ is the set of the security policy rules suit, regulating the users' access to the table data from $\Re$, $|\boldsymbol{S}| = h$. The rules suits $S_1, S_2, \ldots, S_h$ can have the different structure, depending on the specific exact rule of the safety policy. The suit $S_j \in \boldsymbol{S}$, $j = \overline{1,h}$, contains the samples of the rules $s_1^{(j)}, s_2^{(j)}, \ldots, s_f^{(j)}$ of the same structure, $|S_j| = f$. In the process of the method implementation this rules arrangement allows placing them in the unified relation as the set of tuples manipulation with which permit to control the users' access to the data. For example, to limit the access to the temporal data of the relation R_i depending on their location on the time axis, the rule $s_y^{(j)} \in S_j$ has the following form:

$$s_y^{(j)} = \langle R_i, u_c, t_s, t_e, \theta_q, g_d \rangle,\ j = \overline{1,h}. \tag{30}$$

The rule $s_y^{(j)}$ prohibits (g_d=0) or permits (g_d=1) the user $u_c \in U$ to execute the operation $\theta_q \in \Theta$ on the object version, represented by the tuple r of the relation $R_i \in \Re$, and actual in the time range [t_s, t_e].

The calculation of the fact of the tuple r accessibility is determined in the moment of the query to it with the help of the predicate $P_b \in \Pi$, where $\Pi = \{P_1, P_2, \ldots, P_h\}$ – is the set of the predicates of the information security system, employing the set $\boldsymbol{S}$ for computation; $|\Pi| = h$, $b = \overline{1,h}$. Every predicate has the following structure in the general case:

$$P_b(R_i,\ u_c,\ \theta_q,\ L, t), \tag{31}$$

where R_i is the relation, the tuple r attributed to; u_c is the user, executing the operation θ_q with the tuple r; t is the time of query (of the operation); L is the value of the tuple of relation R_i attributes, influencing the access permission. For the above mentioned example of the access delimitation according to the time range L consist of t_s and t_e, i.e. L =(t_s, t_e). In case if the P_b true the access to the tuple $r = (A, L)$ is allowed or, otherwise, is restricted.

The forced application of the predicates from Π is implemented in the data view, employed for the reading queries accomplishment and in the triggers, controlling the queries for the data modification.

The shared employment of the suggested method and the discretionary access augments the information security of the information systems and provides the flexibility of the safety control in the databases. The access policy can be constructed in accordance with any rules of the corporation and take into consideration the complex interconnections between the objects and the subjects of the system. For example, the author's work [15] demonstrates the implementation of the described method for the restriction of the access to the data by dimension possessing the hierarchy nature – according to the official hierarchy of the users.

In the process of the development of method of access restriction it was taken into account that the data security system in respect with their time dimension is to be integrated with the operative databases security system if necessary, whereas the time dimension is not a sole parameter due to which the access to the transport information system data is restricted. That is why the data protecting method, proposed by the author, permits to differentiate the access not only regarding the temporal data dimension, but regarding any dimension or property appearing in the data: territorial, geographical, administrative, etc.

The method does not complicate the databases structure significantly and does not require the money-consuming costly support. The access rules management is available not only to the safety administrators, but also to the other users within their competency frameworks. The method has been tested in the environment of the databases managements system IBM DB2 UDB. The acquired results of testing have demonstrated that the suggested method is quite satisfactory for the requirements of the promptitude, as the response time for the query under the condition of existing 1000 rules of access has increased only by 1.1 times, and of 15000 rules of access – by 1.5 times.

The method has been approved and then implemented in the information systems APFIS, APNIS and the database of the Referenced Data System, functioning on the Latvian Railway, and this fact has permitted the solution of the whole range of issues connected with the data safety; this method has proven its reliability, implementation and employment flexibility. The method can also be in demand in other tasks solution, if it requires the discrimination of the access rights between the one-type objects, including: multi-users' documents control, discriminated staff accountancy and accountancy of other corporate resources. In the process of solving the issues of this type, every operator has to be responsible only for the part of information, referring to the sphere of his authority and privilege, and he has to obtain the access to it for the procedures of modification and supplementing, but the same type data beyond his sphere of authority has to be unavailable for him. The authority can be defined according to the operator's position in the official hierarchy and according to the regional belongings or any other criteria of statistical or calculated nature.

3.6.3. Example of Providing the Privacy of the Temporal Data of the Railway Tariffs

The active tariffs for the freight transportation are the information of the general availability. Nevertheless, the tariffs planned for the future, refer to the confidential information accessible only to the manager of the freight transportations. In the course of coming into operation the confidential tariffs gain the nature of general availability.

The example under consideration assumes the information on the tariffs planned for the day ahead and further to be the confidential information. Accordingly, the tariffs, used before, or the current tariffs, or the tariffs for the following day are the tariffs of general availability.

There is the formalization of the task of discriminating access to the tariffs and its solution examination. The following symbols, characterizing the systems, are introduced:

$R_{tariff}(OID, Tariff, t_s, t_e)$ is a scheme of relation, containing the temporal data of tariffs, where OID is an abstract identifier of the tariff type; $Tariff$ is the value of the tariff; $[t_s, t_e]$ is the tariff activity diapason.

There is the definition of the logical function F^{limit}, determining the accessibility of the tuple $r \in R_{tariff}$ for the current user *u* at the moment of the query *now*:

$$F^{limit} = (\neg(now < (r.t_s - 1))) \vee (u \in G),$$

where G is an authorized group of managers of the freight transportations, $G \subseteq U$;

U is the set of all users, $u \in U$.

Then the subset R_{tariff}^{limit} of the relation R_{tariff}, accessible for the user for the acquaintance, is calculated in the following way:

$$R_{tariff}^{limit} = \sigma_{F^{limit}}(R_{tariff}).$$

Depending on the user and the time of query R_{tariff}^{limit} comprises a different amount of tuples. The forced implementation of the function F^{limit} in case of querying the relation R_{tariff} does not permit the users, unless they are the managers of the freight transportations, to get into the “secret” time slices, and in this way the required confidentiality of the tariffs planned for the future is achieved.

3.6.4. Solution of the Issue of Vacuuming the Tables with the Temporal Support

The constantly increasing volume of the stored data might probably become the reason of the efficiency decrease in the tables with the temporal support. First of all it refers to the tables with the temporal support of the transaction time, as far as the process of saving the previous versions of the records takes place automatically. In these tables the data are not removed physically from the system, but they are only appended to the existing ones (including the cases of the records modification). From the other hand it happens after a certain period of time that the part of the data has got out of date, and these data in their essence have become a so called “garbage”, which is advisable to remove from the system physically, because it not only take a lot of space but also decreases the efficiency in the selection procedures. The problem of existing “garbage” in the temporal data of the information systems is on the verge of two tasks: provision of the integrity and accessibility of the databases. The “garbage” existence in the database makes the access to the active data more complicated, and this fact affects their accessibility, and the “garbage” itself contradicts, as it is known, the integrity concept. That is why a very important requirement to the designed database is the possibility of physical removal of the data, resulting in the operation of vacuuming of the temporal tables [100]. However, the physical data removal exposes the general integrity of the stored data to the risk. The problems resulted from this removal and the paradoxes emerged in the process of vacuuming the tables with the support of the transaction time are illustrated in the research [67]. The described problems are connected with the subsequent incorrect operation of the queries, especially the ones, employing the aggregate functions.

There are also other reasons of the “garbage” appearance. There are information systems on the transport which possess no special facilities for navigation in the time dimension and processing the previous versions of the data, despite the fact, that it is necessary to save the history of the data changes.

There are following information systems among them:

- the inherited non-temporal information systems;
- the information systems, employing the data of the external information systems by means of synchronization;
- the information system in which the data time dimension is deliberately hidden or intentionally done inaccessible for processing.

The temporal form Rollback, automatically saving the data previous versions, suits for modeling the temporal data in such information systems. It is worth mentioning that in this class of the information systems there the problem of “garbage” exists, which is accumulated in the system as a result of the operators’ errors. These “erroneous” versions are not seen by the operators and, consequently, they do not remove these versions. The classical logic of the Rollback form does not recognize the “garbage” and saves all the appearing versions. At the same time, it is possible to formulate the definite rules of the garbage identification (for example, the versions appearing as a result of the repeated alternation of the data on the same day) and not save the versions, meeting these definitions. However, this type of the functionality is not well investigated, and this fact has become the basis for developing the additional procedures in the temporal logic of the Rollback form (see implementation in point 3.2.2) in this thesis.

The issue of “vacuuming” is first of all urgent for the tables possessing the support of the transaction time, but it also exists in the tables possessing the support of the active time. Thereby, if it is referred to the storage of the certain intermediate versions of some document, it might happen that only the very last version, for example, for January of the previous year, is of an interest, and all the intermediate versions can be deleted. In this case it is also suggested employing the procedure of vacuuming, and the range of the accessible methods is considerably wider, for example, it is possible to retain every third document, or not more than one document in a month, and what is more important to employ this vacuuming to the documents of the previous year.

The method of vacuuming the temporal tables, represented in the thesis research, assumes the process of solving three tasks:

1. Formalizing the “garbage” in the definite temporal table of the information systems.
2. Identifying the “garbage”.
3. Removing the “garbage”.

For solving the task of identification and removing the “garbage” in the temporal databases it is necessary to formulate accurately the concept of the “garbage” itself. In common case the term “garbage” in the databases defines *the tuples of the temporal relation which will* *never be employed in the system*. However, for every independent task and for every type of the temporal relation the concept of the "garbage" can differ. In other words, if $\Re = \{R_1, R_2, \ldots, R_n\}$ is the set of the temporal relations of the databases, $|\Re| = n$, then for every R_i, $i = \overline{1, n}$ there can be determined its own function of the "garbage" identification $F_i^{(g)}$. For example, regarding the Historical-Overlapped form the following object lifespan can be supposed to be the "garbage": the lifespan without any special semantic message and according to the meaning of the attributes of the activity time (the time of beginning and ending the lifespan) is completely comprised in another lifespan, but with a later time of transaction. Here is the definition of the logical function for this formula for determining the tuples referring to the "garbage".

It is an assumption that $R(OID, A, t_s, t_e, t_f)$ – is the temporal relation of the Historical-Overlapped form, possessing the attributes *A*, describing the object properties, and the temporal attributes t_s, t_e and t_f. Then the tuple $r \in R$ is a "garbage", if in the relation R there is another tuple $p \in R$, describing the lifespan of the same object that is described by r, but with a later time of transaction, and moreover, the lifespan of the tuple r is completely overlapped by the lifespan of the tuple p. Respectively, for checking this property there used the logical function $F^{(g)}(r)$ having the following form:

$$F^{(g)}(r) = \begin{cases} true, \quad if\ \exists p \in R \left| \begin{array}{l} (r \neq p) \wedge (r.OID = p.OID) \wedge \\ ([r.t_s : r.t_e] \subseteq [p.t_s : p.t_e]) \wedge (p.t_f > r.t_f); \end{array} \right. \\ false, \quad otherwise. \end{cases} \tag{32}$$

Thus, it is necessary to remove the set of tuples $R^{(g)} \subset R$ from the temporal relation, and this set of tuples is determined by the following way:

$$R^{(g)} = \{r \mid r \in R, F^{(g)}(r)\}. \tag{33}$$

Nevertheless, the execution of the operation of the data removing from the temporal tables is connected with the risk of emerging the various conflicts with the procedures of the temporal frameworks and the safety system, implemented with the triggers usage. Depending on the functions of these procedures this type of removal can result in the data integrity damage. That is why the following limitations for the vacuuming operation are suggested for introduction:

1. The operation of vacuuming is accomplished by the databases administrators only.
2. The monopolistic (exclusive) access is provided at the moment of the operation accomplishing.
3. All the triggers, connected with this table, are disabled for the time when the operation is accomplished.

The above considered method has been implemented by the thesis author in the information system *APNIS*, functioning on the Latvian Railway for vacuuming the history of the employees' transferring between the enterprise departments and history of changes of the hard- and software on the working places.

Conclusions of the Third Chapter

1. For implementation of the temporal databases in the relation databases management system there has been designed the model of the temporal data with the abstract identifier and the apparatus of the temporal logics granting the user and the developer with the opportunities of the simplified interaction with the temporal databases, exactly:

- to operate in any time slice, employing the ordinary non-temporal SQL queries using the method of the temporal environment;
- to interact with both the temporal interface of the databases and the classical relational one – non-temporal;
- to support automatically the integrity of the temporal objects, for example, to store the previous objects versions, solve the problems of overlapping with the other versions, determine and remove the inaccurate invalid objects versions.

2. Investigation of the issues of the trains schedule arrangement possessing the properties of periodicity, overlapping, specified calendar and other peculiarities has required designing the new temporal form of relation with the overlapping lifespans Historical-Overlapped, and developing the algorithms of determining the object active version and the method of transformation of the interval form of the temporal data into the point form.

3. The suggested method of transformation of the interval form of the temporal data into the point form on the fundamental of the standard means of the relation database taking into account such complicated temporal conditions as the periodicity and the special calendar existence, can be employed in the process of the temporal data transformation in various applied tasks.

4. The developed method of restricting access to the temporal databases allows limiting the access to the table data in the way that the user can obtain the access to one record of the table, but cannot obtain the access to another one. The suggested method of the access control can be in demand in other information systems as well, because the condition of adaptation to any rules of the corporation lies in its basis.

5. The offered system of the schedule support can serve not only as a basis of the information systems of the railway transportations, but also serve as a part of the management system, permitting the trains schedule modeling, generating automatically the offers for the analyst for transiting the trains on another day of the week when the new day off is assigned, and as a part of the analytical system allocated for the analysis and forecasting of the passenger traffic flow.

Chapter 4. PRACTICAL APPLICATION OF THE RESEARCH RESULTS IN THE INFORMATION SYSTEMS ON THE LATVIAN RAILWAY

4.1. Investigation Results Implementation in the Interactive Local Train Traffic Timetable

The system of the passenger trains interactive scheduling SAR is one of the principle objects of the research under consideration. The object is interesting by the fact of operating with the temporal data obtaining the maximum amount of the temporal peculiarities in comparison with other information systems of the Latvian Railway. The following peculiarities can be emphasized: the multi-version, the repeated cases when one and the same version becomes operative, the periodicity, and the peculiar calendar. The large data bulk and the requirements for the high speed of performance of the queries to these data increase the interest to the object.

The system of the passenger trains interactive scheduling has the WEB interface (http://sirius.ldz.lv/sari), applying which all the Internet users can request for some information on the passenger trains schedule for any day and in any direction in the interactive mode. The system permits to take into account any prompt changes in the existing schedule and to project the timetable for the long run.

The system manipulates with the data thematically, referential to three fields: the schedule, the fares, and the classifiers (reference data). The users, interacting with this system data, can be divided into three categories according to the nature of their queries: the Internet users, the operators of the scheduling and fares systems introduction, and the operators of some other systems, including the inherited systems, not capable of operating with the temporal data.

The diagram of the developed methods and models employment in the system of the interactive passenger trains scheduling in the process of different users' categories interaction with the scheduling system data is performed in Fig. 4.1.

The methods and the models are divided into 4 groups in accordance with the fulfilled functions: the temporal data representational model, the temporal data integrity, the access to the temporal data and the data physical arrangement.

At the current moment the SAR system of the passenger trains interactive scheduling, implemented with the usage of the developed methods and models, is successfully employed on the Latvian Railway. The developed methods and models employment in the system of the passenger trains interactive scheduling SAR has allowed:

1. Considerable simplifying of the system development;
2. Scheduling and fares data arrangement in the most space saving way, taking into consideration all the above mentioned data temporal peculiarities (the model with OID, the Historical-Classic form, the Historical-Overlapped form, the model of the periodicity and the special calendar);
3. Transformation of the segments of the classifiers central databases into the temporal ones, and at the same time provision of the temporal up-warding compatibility with the other previously developed applications (the Rollback form, the Historical-Classic form, the method of automatic saving of the object previous states, the method of providing the user's time environment);
4. Provision of the well-formed proper access to the data active versions (the logical rules method);

5. Provision of the data temporal integrity (the method of the temporal integrity provision for the Historical-Classic form, the method of automatic saving of the object previous states);
6. Provision of the inviolability of the schedule historical data, the fares and the classifier (the method of the access delimitation of the relation system on the relation tuples level);
7. Reaching the required speed performance with the employment of the different indexing methods, including clustering.

The temporal data organization is considered in details. Fig. 4.2 presents the ER-model of the database of the system SAR. The model is divided into three areas:

1. *RDS* (KLASIFIK) – the referenced information from the centralised database of the classifiers and codifiers, containing the data about stations, cities, states and the railways. This area is marked with the blue colour on the scheme;
2. *Schedule* (SAR) – the package of the tables, describing the trains schedule. This area is marked with the orange colour on the scheme;
3. *Tariff* – the package of the tables, describing the fares tariffs. This area is marked with the yellow colour on the scheme;

The tables, marked with the green colour in the picture are the temporal tables:

- the tables N_PILS_NOS, N_VALST_NOS, N_DZ_STAC_NOS, N_DOR_NOS of the schemes KLASIFIK, SAR.TRAIN_VAGON and TARIFF.K_PRICE correspond to the temporal form *Historical-Classic*;
- the table SAR.SCH_INSTR corresponds to the form *Historical-Overlapped*. It describes the version nature of the timetable.

The tables, modeling the periodicity, are marked with the violet colour:

- the table SAR.TR_ATR_SCH describes the periodicity of the trains schedule versions;
- the table TARIFF.TRAIN_VID_TARIF describes the periodicity of the fare discounts implementation for the determined specified trains. At the current moment the seasonal discounts for the travel by certain trains on the working days in the day time are characterized on the basis of this table.
- the table SAR.SPEC_DAYS represents the set *Z* (see point 3.3.2) and describes the calendar of the peculiar days (state holidays, and so on).

The presented database is projected in accordance with *the temporal data model with the object abstract identifier*. For the temporal table SAR.SCH_INSTR the attribute PZD_ID, defined in the table SAR.K_TRAIN, is the object abstract identifier (OID). The temporal table of the fares TARIFF.K_PRICE has its identifier, consisting of the set of attributes VID_DOC, TIP1_BIL, TAR_CAT_ID, VAGON_TYPE_ID, AP_TARIF_ID, TRAIN_TYPE_ID and DIST, defined in the tables TARIFF.N_VID_DOC, TARIFF.N_TIP1_BIL, TARIFF.K_TARIF_CAT, SAR.K_VAGON_TYPE, TARIFF.K_ATL_PIEMAKS_TARIF and SAR.K_TRAIN_TYPE correspondingly (excluding the attribute DIST, showing the number of kilometers and does not require the additional definition). This identifier introduces the invariable unalterable version of the travel (ticket), for example: *«<u>ticket for a child</u> for the round trip in <u>the common carriage</u> of <u>the electric train</u> <u>toward the resort direction</u> for the distance <u>of 15 kilometers</u> <u>without special discounts</u>"*. The values of the listed characteristics are marked by underlining. In the temporal table SAR.TRAIN_VAGON, comprising the data about the existence of the certain determined types of carriages in the trains, the object abstract identifier is also the complex one: the attributes PZD_ID and VAGON_TYPE_ID.

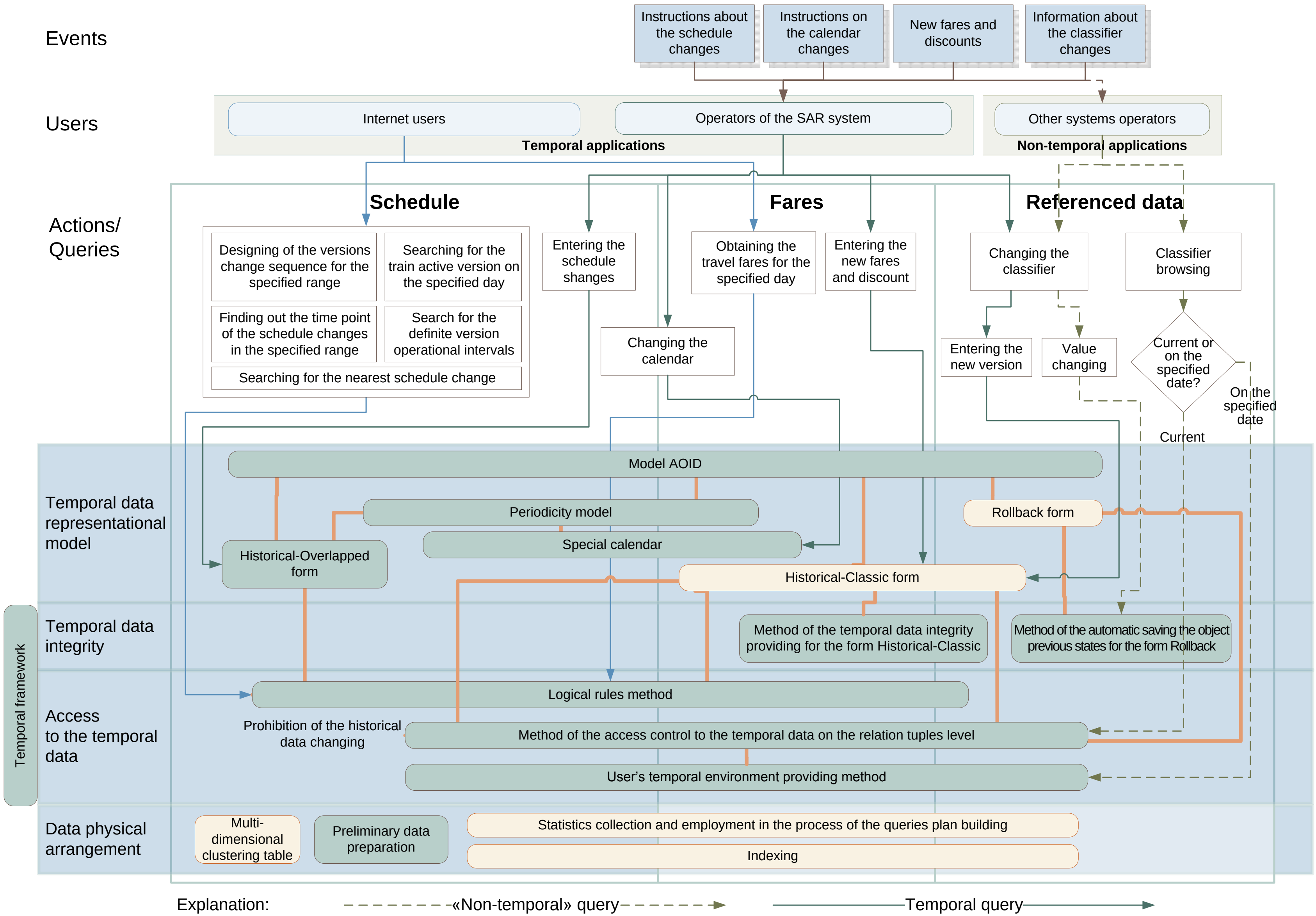


Fig. 4.1. The diagram of the developed methods and models employment in the system of the interactive passenger trains scheduling

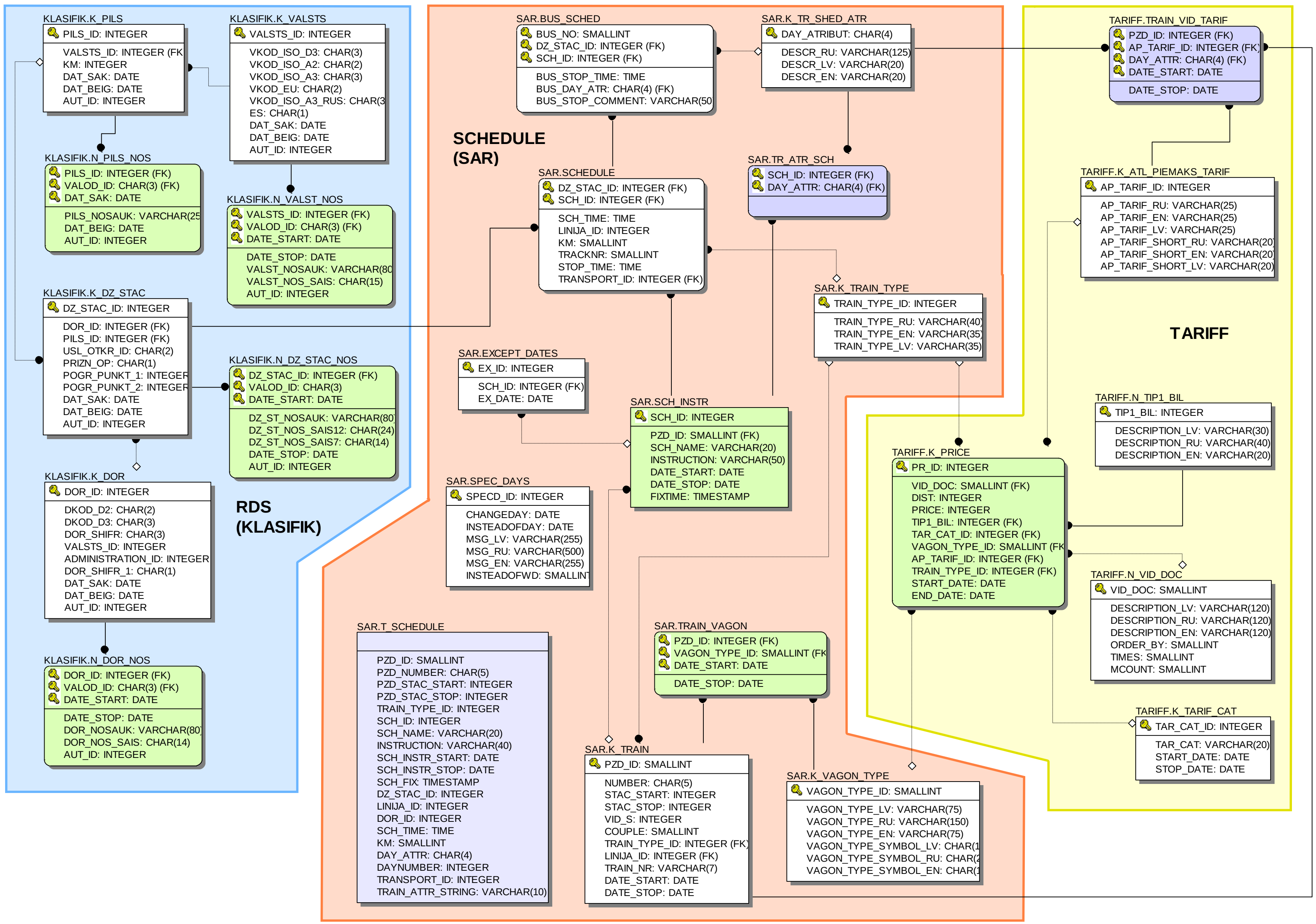


Fig. 4.2. ER-model of the schedule system of the inland passenger trains (SAR)

The table SAR.T_SCHEDULE presents the materialized view of the data from KLASIFIK and SAR schemas. The system is organized in the special way, and the majority of queries of the Internet users of the search for the suitable schedule are directed to this table exactly. The representation materialization permits not waste time for connecting the tables and some other calculations in the process of querying. The table physical arrangement is based on the fundamental of the *multi-dimensional clustering*[8], providing the maximum rapid response for the query from the system. The data in the table are renewed automatically with the specified periodicity. The SQL instruction of this table constructing has the following view:

```
CREATE TABLE "SAR"."T_SCHEDULE" (
  "PZD_NUMBER"      CHARACTER(5)      NOT NULL,
  "PZD_STAC_START"  INTEGER     NOT NULL,
  "PZD_STAC_STOP"   INTEGER     NOT NULL,
  "OWNER_ID"  INTEGER,
  "TRAIN_TYPE_ID"   INTEGER,
  "PZD_ID"    SMALLINT    NOT NULL,
  "SCH_ID"    INTEGER     NOT NULL,
  "SCH_INSTR_START" DATE  NOT NULL,
  "SCH_INSTR_STOP"  DATE,
  "SCH_FIX"   TIMESTAMP   NOT NULL,
  "SCH_NAME"  VARCHAR(20),
  "INSTRUCTION"     VARCHAR(40),
  "DZ_STAC_ID"      INTEGER     NOT NULL,
  "LINIJA_ID" INTEGER,
  "DOR_ID"    INTEGER,
  "SCH_TIME"  TIME,
  "KM"  SMALLINT,
  "DAY_ATTR"  CHARACTER(4),
  "SCH_IZM_DATE"    DATE,
  "DAYNUMBER" INTEGER,
  "TRANSPORT_ID"    INTEGER,
  "TRAIN_ATTR_STRING"     VARCHAR(10)
)
  IN "SARSPACE"
  INDEX IN "SARINDX"
  ORGANIZE BY DIMENSIONS (
    "DZ_STAC_ID",
    "PZD_ID"
    );
```

where four last records describe the dimension, according to the values of which the table data will be clustered and placed on the databases pages.

Besides the procedure of the table clustering, the speed of the query responses is increased with the employment of the regular indexes. For example, increasing the speed of the schedule active version choice takes place at the expense of the temporal attributes *indexation* (the boarders of the schedule version range – the attributes SCH_START, SCH_STOP and the indicator of the record time fixation SCH_FIX), as well as the attribute of the schedule version identifier SCH_ID. There are SQL-instructions for these indices construction below:

```
CREATE INDEX "SAR"."XIE_FIXTIME"
  ON "SAR"."T_SCHEDULE"
    ( "SCH_FIX" ASC );
```

[8] Multi-dimensional clustering was first introduced in DB2 UDB V8.1 (2002 year)

```
CREATE INDEX "SAR"."XIE_SCH_ID"
  ON "SAR"."T_SCHEDULE"
    ( "SCH_ID" ASC );

CREATE INDEX "SAR"."XIE_SCH_START"
  ON "SAR"."T_SCHEDULE"
    ( "SCH_START" ASC );

CREATE INDEX "SAR"."XIE_SCH_STOP"
  ON "SAR"."T_SCHEDULE"
    ( "SCH_STOP" ASC );
```

The regular *statistics collection* provides the optimal design of the query fulfilment plan. Instruction for the statistics collection for the table SAR.T_SCHEDULE has the following view:

```
RUNSTATS ON TABLE "SAR"."T_SCHEDULE" AND DETAILED INDEXES ALL;
```

With the help of the described physical organization, indexation and optimized queries accomplishment plan the speed of response to the table SAR.T_SCHEDULE was increased by 120 times. There is the demonstration of the interaction of WEB module of the information systems with the temporal database under consideration. Fig. 4.3 shows the WEB module cardinal page of the information system SAR, implementing which the users are able to request the information on the train schedule between two stations. For accomplishing this operation the users have to specify the station of departure, the station of destination, the date and the time range of the train departure.

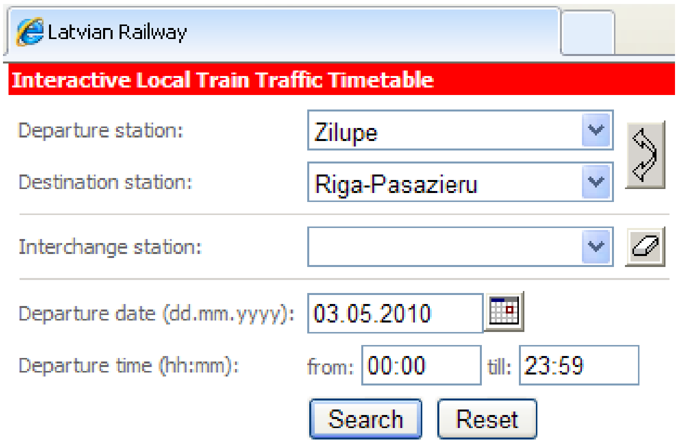


Fig. 4.3. WEB interface of the information systems SAR

The search for the active schedule for this date is accomplished with employing *the logical rules method*, taking into account the activities ranges and the versions overlapping, periodicity and the specified calendar.

There is the schedule for the train 607P (Pzd_ID=297), described by four versions:

(Sch_ID=11087) – the core fundamental schedule for the daily running from 31.05.2009 to 29.05.2010;

(Sch_ID=12543) – the change for the core fundamental schedule from 12.04.2010 to 28.05.2010, referring to the working days only;

(Sch_ID=13001) and (Sch_ID=13002) – two core fundamental schedules, activated starting from 31.05.2010, one for the days off, and the second one for the working days.

The versions operation time and their overlapping are schematically presented in Fig. 4.4. The dotted line shows the periods of the versions alteration on the working days and on the days off.

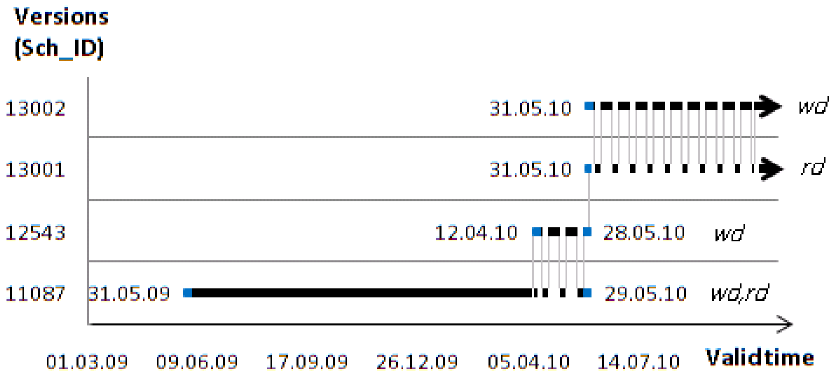


Fig. 4.4. Overlapping the set of the schedule versions of the train 607P on the axis of valid time

The data of the tables describing the version nature of the timetable and its periodicity are presented in Fig. 4.5.

SAR.SCH_INSTR					
Sch_ID	**Pzd_ID**	**Instruction**	**Date_Start**	**Date_Stop**	**FixTime**
11087	297	N1	31.05.09	29.05.10	29.01.2009 14:56
12543	297	659p	12.04.10	28.05.10	22.02.2010 11:39
13001	297	N1,657p	31.05.10		24.05.2010 15:35
13002	297	N1,657p	31.05.10		24.05.2010 15:36

SAR.TR_ATR_SCH	
Day_Attr	**Sch_ID**
everyday	11087
wd	12543
rd	13001
wd	13002

Fig. 4.5. Data of the train 607P schedule from the tables SAR.SCH_INSTR and SAR.TR_ATR_SCH

Accordingly, in the process of querying the timetable between the stations Zilupe and Riga-Pasazieru on 27.05.2010 the system delivers one schedule (see Fig. 4.6), and on 04.06.2010 – another one (see Fig. 4.7). It should be marked that the version 12543 is active on 27.05.2010 and the version 13002 is active on 04.06.2010 for the train 607P.

Route of the journey: **Zilupe → Riga-Pasazieru**
Departure date and time: **27.05.2010 (from 00:00 till 23:59)**

Train number	Train plies on[1]	Transport type	Carriage type[2]	Departure	Arrival	Train route
615P	1234567	Diesel-engine train	O	03:35	08:30	Zilupe → Riga-Pasazieru
607P	1234567	Diesel-engine train	O	17:21	22:17	Zilupe → Riga-Pasazieru
				Night trains		
615P	1234567	Diesel-engine train	O	03:35	08:30	Zilupe → Riga-Pasazieru

Notes:

[1] 1 2 3 4 5 6 7 - the yellow background indicates that operative changes have been made to the timetable for this and the following week;

[2] Carriage types:

O - a carriage with unnumbered seats;

Fig. 4.6. Schedule between the stations Zilupe and Riga-Pasazieru on 27.05.2010

Route of the journey: **Zilupe → Riga-Pasazieru**
Departure date and time: **04.06.2010 (from 00:00 till 23:59)**

Train number	Train plies on[1]	Transport type	Carriage type[2]	Departure	Arrival	Train route
615P	1234567	Diesel-engine train	O	03:36	08:30	Zilupe → Riga-Pasazieru
607P	1234567	Diesel-engine train	O	17:23	22:30	Zilupe → Riga-Pasazieru
			Night trains			
615P	1234567	Diesel-engine train	O	03:36	08:30	Zilupe → Riga-Pasazieru

Fig. 4.7. Schedule between the stations Zilupe and Riga-Pasazieru on 04.06.2010

There is the examination of the timetable of the same train for the period from 03.05.10 to 09.05.2010 (see Fig. 4.8). The schedule for this period is described by two versions of schedule – for the working days and for the days off (13001 and 13002). However, as it is seen in Fig. 4.8, the order of the schedule versions alteration has been broken: on May 3, Monday and on May4, Tuesday the schedule version for the days off is employed, and on Saturday, vice versa, the version for the working days is implemented. It is explained by the fact that May 4 is the day off (Declaration of Independence Day). It was also decided by many of the employers in Latvia that May 3, Monday would be the day off for the convenience of the employees, but this day should be compensated by work on next Saturday.

Route of the journey: the list of operative changes to the train

Route of the journey: **Zilupe → Riga-Pasazieru**
Date: **03.05.2010 (03.05.2010-09.05.2010)**

Train route: **Zilupe → Riga-Pasazieru**
Train number: **607P**
Train plies on: **1234567**
Transport type: **Diesel-engine train**
Carriage type: **A carriage with unnumbered seats (O)**

Station	km	03.05.2010 Monday	04.05.2010 Tuesday	05.05.2010 Wednesday	06.05.2010 Thursday	07.05.2010 Friday	08.05.2010 Sunday	09.05.2010 Saturday
Zilupe	**0**	**15:50**	**15:50**	**17:21**	**17:21**	**17:21**	**17:21**	**15:50**
Brigi	5	15:57	15:57	17:28	17:28	17:28	17:28	15:57
Nerza	11	16:05	16:05	17:36	17:36	17:36	17:36	16:05
Istalsna	22	16:16	16:16	17:47	17:47	17:47	17:47	16:16
Ludza	31	16:26	16:26	17:57	17:57	17:57	17:57	16:26
Cirma	43	16:38	16:38	18:09	18:09	18:09	18:09	16:38
Taudejani	48	16:44	16:44	18:15	18:15	18:15	18:15	16:44
Rezekne 2	55	16:55	16:55	18:26	18:26	18:26	18:26	16:55
Sakstagals	67	17:11	17:11	18:42	18:42	18:42	18:42	17:11
Vilani	81	17:27	17:27	19:03	19:03	19:03	19:03	17:27
Varaklani	91	17:38	17:38	19:14	19:14	19:14	19:14	17:38
Stirniene	99	17:47	17:47	19:23	19:23	19:23	19:23	17:47
Atasiene	115	18:00	18:00	-	-	-	-	18:00
Mezare	126	18:10	18:10	19:44	19:44	19:44	19:44	18:10
Kukas	137	18:27	18:27	19:55	19:55	19:55	19:55	18:27
Krustpils	150	18:40	18:40	20:08	20:08	20:08	20:08	18:40
Plavinas	167	18:55	18:55	20:24	20:24	20:24	20:24	18:55
Koknese	185	19:13	19:13	20:46	20:46	20:46	20:46	19:13
Aizkraukle	197	19:26	19:26	20:58	20:58	20:58	20:58	19:26
Ogre	245	20:07	20:07	21:37	21:37	21:37	21:37	20:07
Janavarti	275	20:32	20:32	22:09	22:09	22:09	22:09	20:32
Riga-Pasazieru	**279**	**20:40**	**20:40**	**22:17**	**22:17**	**22:17**	**22:17**	**20:40**

Notes:
17:40 - The changes to timetable regarding the previous day are marked in colour.

Fig. 4.8. Alteration of the schedule versions for the train 607P from 03.05.2010 to 09.05.2010

This situation is modeled by *the peculiar calendar Z*, presented in the table SAR.SPEC_DAYS. Fig. 4.9 demonstrates the corresponding segment of the data of this table.

The records of this table show that on Monday 03.05.10 and on Tuesday 04.05.10 the active version is the schedule assigned on Saturday 15.05.10, and on Saturday 08.05.10 – the active version is the schedule on Monday 10.05.10. It should be noted that these settings have impact on the schedules for all trains of inland traffic in Latvia.

SAR.SPEC_DAYS		
SPECD_ID	**CHANGEDAY**	**INSTEADOFDAY**
281	03.05.10	15.05.10
282	04.05.10	15.05.10
284	08.05.10	10.05.10

Fig. 4.9. Table SAR.SPEC_DAYS data

There is the demonstration of the interaction with the temporal data of the table TARIFF.TRAIN_VID_TARIF. As it is obvious from the queries results on 22.01.2010 and 12.07.2010 (see Fig. 4.10 and Fig. 4.11), the system delivers not only different versions of the trains schedule, but also the conditions of providing the discounts (in summer time there are no discounts in the resort directions).

Route of the journey: **Jelgava → Riga-Pasazieru**
Departure date and time: **22.01.2010 (from 08:00 till 12:59)**

Train number	Train plies on[1]	Transport type	Carriage type[2]	Departure	Arrival	Train route
6716	1234567	Electric train	O	08:06	08:55	Jelgava → Riga-Pasazieru
870P	123456-	Diesel-engine train "Kurzeme"	N	08:34	09:10	Liepaja → Riga-Pasazieru
6718	1234567	Electric train	O	08:50	09:39	Jelgava → Riga-Pasazieru
6720	1234567	Electric train	O	09:30	10:19	Jelgava → Riga-Pasazieru
6722	1234567	Electric train	O	10:36	11:25	Jelgava → Riga-Pasazieru
6724	1234567	Electric train	O	11:45	12:33	Jelgava → Riga-Pasazieru
6726	1234567	Electric train	O	12:39	13:27	Jelgava → Riga-Pasazieru

Notes:

[1] 1 2 3 4 5 6 7 - days when the ticket price is reduced by 25% are indicated in green;

[2] Carriage types:

O - a carriage with unnumbered seats;

N - a carriage with numbered seats;

Fig. 4.10. Schedule on 22.01.2010 between the stations Jelgava and Riga-Pasazieru with the discount fare indication in certain trains

Route of the journey: **Jelgava → Riga-Pasazieru**
Departure date and time: **12.07.2010 (from 08:00 till 12:59)**

Train number	Train plies on[1]	Transport type	Carriage type[2]	Departure	Arrival	Train route
6714	1234567	Electric train	O	08:08	08:57	Jelgava → Riga-Pasazieru
870P	1----6-	Diesel-engine train "Kurzeme"	N	08:34	09:10	Liepaja → Riga-Pasazieru
6716	1234567	Electric train	O	08:50	09:39	Jelgava → Riga-Pasazieru
6718	1234567	Electric train	O	09:30	10:19	Jelgava → Riga-Pasazieru
6720	1234567	Electric train	O	10:45	11:34	Jelgava → Riga-Pasazieru
6722	1234567	Electric train	O	12:10	12:58	Jelgava → Riga-Pasazieru

Notes:

[1] 1 2 3 4 5 6 7 - days when the ticket price is reduced by 25% are indicated in green;

[2] Carriage types:

O - a carriage with unnumbered seats;

N - a carriage with numbered seats;

Fig. 4.11. Schedule on 12.07.2010 between the stations Jelgava and Riga-Pasazieru

There is the illustration of fare alteration described by the table TARIFF.K_PRICE. As it comes clear from the queries results on 15.08.2008 and 10.06.2009 (see Fig. 4.12), the system delivers different versions of the trains schedule and fares.

Route of the journey: **Jelgava → Riga-Pasazieru**
Departure date and time: **15.08.2008 (from 00:00 till 05:59)**

Train number	Train plies on[1]	Transport type	Carriage type[2]	Departure	Arrival	Train route
6702	1234567	Electric train	O	05:20	06:09	Jelgava → Riga-Pasazieru
6704	1234567	Electric train	O	05:58	06:47	Jelgava → Riga-Pasazieru

Price
0,95
1,85
0,47
0,91
0,47
0,91
0,29
0,56
0,29
0,56

Price
3,80
7,13
9,98
30,88
15,44
14,63
27,17
44,14
51,44
56,34
64,14
36,61
42,41
45,81
51,91

Route of the journey: **Jelgava → Riga-Pasazieru**
Departure date and time: **10.06.2009 (from 00:00 till 05:59)**

Train number	Train plies on[1]	Transport type	Carriage type[2]	Departure	Arrival	Train route
6702	1234567	Electric train	O	05:19	06:08	Jelgava → Riga-Pasazieru
6704	1234567	Electric train	O	05:58	06:47	Jelgava → Riga-Pasazieru

Have a safe journey!

Ticket prices (LVL)

Ticket type	Transport type	Carriage type[2]	Price
Full (one way)	Electric train	O	1,35
Full (two ways)	Electric train	O	2,63
Child (one way)	Electric train	O	0,67
Child (two ways)	Electric train	O	1,29
Ticket with 50% off the price (one way)	Electric train	O	0,67
Ticket with 50% off the price (two ways)	Electric train	O	1,29
Luggage (one way)	Electric train	O	0,41
Luggage (two ways)	Electric train	O	0,79
Small pets ticket (one way)	Electric train	O	0,41
Small pets ticket (two ways)	Electric train	O	0,79

Train Pass Prices (LVL)

Ticket type	Transport type	Price
5 day (one way)	Electric train	5,40
5 day (two ways)	Electric train	10,13
WeeklY (two ways)	Electric train	14,18
Monthly (two ways)	Electric train	43,88
Montly child ticket (two ways)	Electric train	21,94
Working day ticket (one way)	Electric train	20,79
Working day ticket (two ways)	Electric train	38,61
Unified monthly ticket of trains and 1 public transport mode in Riga in 1 route (two ways)	Electric train	56,49
Unified monthly ticket of trains and 1 public transport mode in Riga (two ways)	Electric train	63,79
Unified monthly ticket of trains and 2 public transport modes in Riga (two ways)	Electric train	68,69
Unified monthly ticket of trains and 3 public transport modes in Riga (two ways)	Electric train	76,49
Unified working day ticket of trains and 1 public transport mode in Riga in 1 route (two ways)	Electric train	47,48
Unified working day ticket of trains and 1 public transport mode in Riga (two ways)	Electric train	53,28
Unified working day ticket of trains and 2 public transport modes in Riga (two ways)	Electric train	56,68
Unified working day ticket of trains and 3 public transport modes in Riga (two ways)	Electric train	62,78

Fig. 4.12. Two time slices on 15.08.2008 and 10.06.2009:
Fare price between the stations Jelgava and Riga-Pasazieru

4.2. Implementation of the Research Results in the Data Warehouses Systems on the Latvian Railway

In the process of investigation there has been developed the conceptual approach towards the corporate data storage and analyzing in the Latvian Railway information systems. This approach is based on the data warehouses technology, implemented in practice and in the author's works [1, 2, 14]. These works demonstrate the solutions for some important practical tasks, among them there are such issues as the bulk data processing system designing, the development of the designing fundamentals of the data warehouses in the systems of the decision making support on the railway, providing the stored data integrity and security, etc. Furthermore the special attention has been paid to the problem of the temporal data employment in the warehouses. This can be explained by the fact that the time is the principal dimension in the data warehouses. The essential processes of the data warehouses functioning is connected the time: unloading, transformation, loading (Extract, Transform, Load, or ETL), further data processing and the data marts production. The arrangement of the efficient interaction with the data time measuring in the warehouses determines the data integrity and security, the quality of their processing (the number of errors), the simplicity of the implementation of the warehouses functioning processes the immediacy of the changes introduction to them.

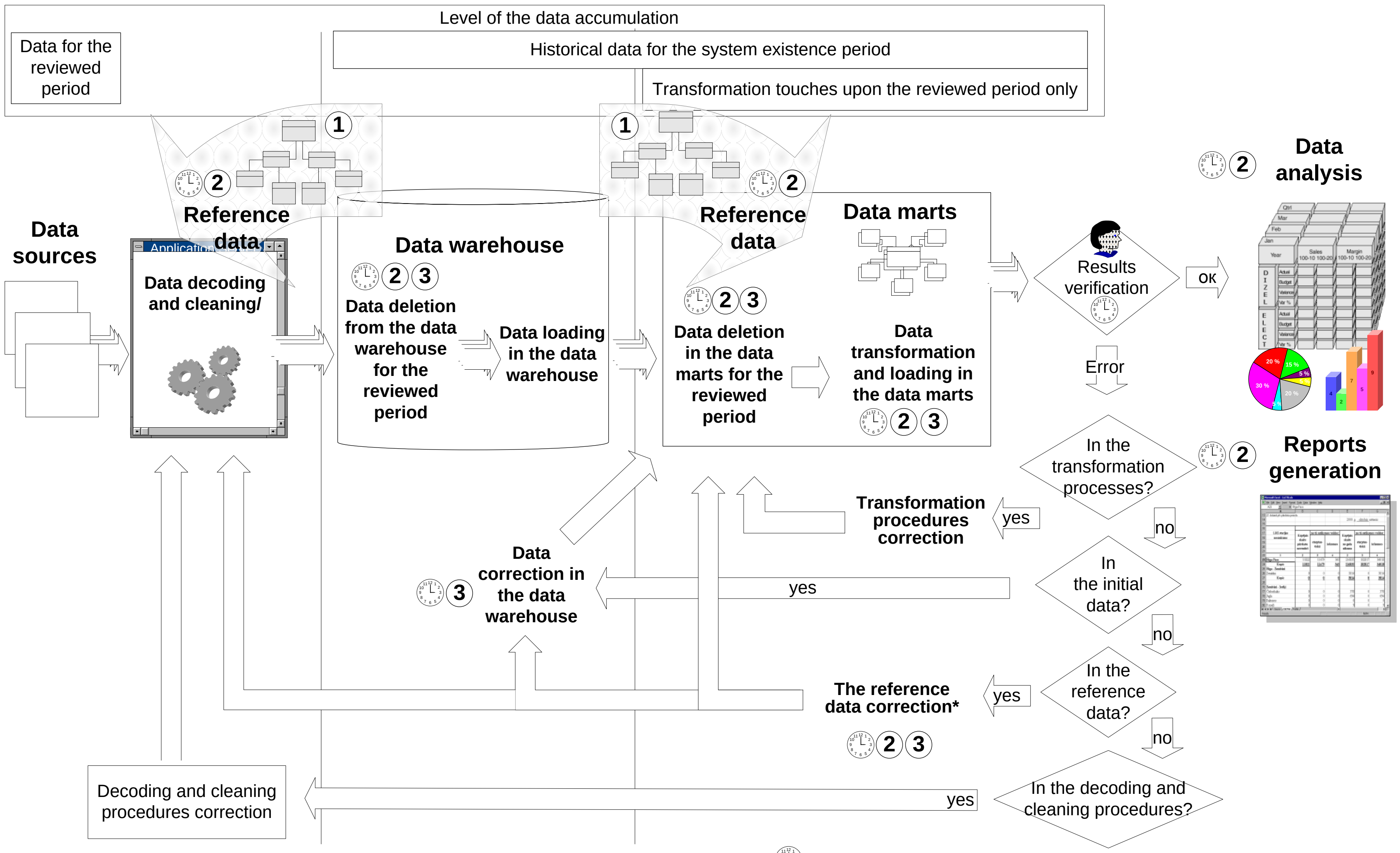
Chapter 1 states that the basic problem of the interaction with the temporal data lies in the fact that either the operator or the programming facilities applied might interact with the data from another time zone, not relating to the time range under processing. It is explained by the imperfection of the mechanisms of the interaction with the temporal dimension of the data, and sometimes by the complete absence of this dimension in the employed databases. This mostly refers to the reference data. This problem occurs on the different stages of the data processing in the warehouses. The technological process of the preparation the data for the analytical system of the passengers transportation on the Latvian railway is performed by Fig. 4.13 in a schematic form. The data from the operative *data sources* undergo the processes of decoding and cleaning up, then they are placed in the *data warehouse* and distributed among the *data marts*. Distribution is the complex of transformation with the participation of the reference data database, containing the data transformation rules dependant on time. After the data loading in the data marts the analyst estimates the results and makes the decision about the data readiness for the further analysis and the reports generation. In the process of estimation the analyst follows the aggregated data from the operational systems and other sources. If the result is unsatisfactory, the part of the technological process is repeated and completed with the data marts and data warehouses cleaning up procedures. Depending on the source of the error occurrence, the technological process is activated from the different points. The error source might be in the procedures of decoding and cleaning up, in the initial date, in the reference data database, in the processes of the data transformation.

The above mentioned procedures criticality increases under the condition of the malfunction of the specified order of the data loading in the analytical system, when the currently processed period is deeper in the past than the earlier processed period. In this situation the author has exposed the following problems:

- data transformation for the data marts is based on the data of the previous processed period, and consequently the part of the technological process is to be executed cyclically, starting with the currently processed period and to the extent of the last one in the system;
- procedures of the marts cleaning up and the data transformation are not intended for the data processing for any time period and they operate only in the specified time range; this time range is determined by the sequence of their usage;
- in the non-temporal database or in case of the erroneously derived time slice the reference data do not correspond to the processed period, therefore the data processing results might be false.

In the technological process of the information-analytical system functioning there have been exposed at least four critical procedures, in the process of executing which the probability of the malfunction of the data warehouse information resources integrity is rather high. They are the following:

- partial data deletion in the data warehouse;
- data correction in the data warehouse;
- reference data database correction;
- partial data deletion in the data marts.



The reference data correction* – depending on the fact what data have been corrected or replenished, the technological process recommences from the different stages. It is sometimes convenient to correct the data in the warehouse, but not to decode and to reset them again

- Time dimension support in the queries to the data
1 - Model AOID
2 - Method of providing the user's temporal environment
3 - Method of the access control to the temporal data on the relation tuples level

Fig. 4.13. Diagram of the data warehouse augmenting and the data preparation for the analysis

Accordingly, the quality of this system functioning directly depends on the availability of the:

- time dimension in the reference data database;
- unsophisticated and convenient facilities for the temporal slices obtaining (the operation which will complicate the data processing code not in a great extent);
- adjustment facilities for the processing period for the data processing procedures simplification; as well as the prevention of the unpremeditated changes of other periods data.

Implementation of the methods and models developed in the thesis permits to solve the mentioned issues: to decrease the risk of the data warehouse integrity breaking, to simplify the data transformation and preparation for the analysis for any time period and to increase the immediacy and safety of these operations. The marks 🕒 and ①, ②, ③ in Fig. 4.13 perform the implementation areas of the developed models and methods in the process of the temporal data processing, and among them there are: the queries to the temporal reference data, the temporal sampling and aggregation, the data deletion of the specified time period, the access delimitation to the data referring to the unprocessed time range. There is the detailed consideration of the certain methods implementation in this scheme.

For the solution of the task of entering the definite time slice, corresponding to the processed data, it is proposed to employ *the method of providing the user's temporal environment*. This task is in demand at many stages of the technological process, starting up with the decoding stage and ending with the stage of data transformation in the marts and the results verification.

For the access delimitation to the data warehouse, referring to the non-processed time ranges, it is offered to apply the *method of the access delimitation to the relation systems data on the relation tuples level* [6, 13, 15] on the purpose of the prevention of the data unpremeditated changes. The same method is suggested for employing for the reference data database changes as well.

Whereas the reference data database is a temporal one, the processes of the data changing by the user, occurring in the definite time slice, are not trivial. If the operator does not determine the values of the time attributes for the changed data, the new object version, appearing under the condition of the data changes, has the operation time equal to the user's time range, described in his *time environment*. Accordingly, the reference data changes do not touch upon the other periods.

Conclusions of the Fourth Chapter

1. The employment of the developed models and methods of the information system SAR in the arrangement of scheduling the passenger trains has permitted:

   - to simplify significantly the information systems development;
   - to organize the data of the schedule and tariffs in the way which is the most space saving and safe from the point of view of the integrity damaging, and taking into consideration all the revealed temporal peculiarities of the data (the model with the object abstract identifier, the form Historical-Classic, the form Historical-Overlapped, the model of periodicity and the specific calendar);
   - to transform the segments of the centralized database of the classifiers into the temporal ones, providing at the same time the temporal up-ward compatibility with other applications developed before (the Rollback form, the Historical-Classic form, the method of automatic saving of the previous states of the objects, the method of providing the temporal environment for the user);
   - to provide the correct access to the active data versions (the logical rules method);
   - to provide the data temporal integrity (the method of providing the temporal integrity for the Historical-Classic form, the method of automatic saving of the previous states of the objects);
   - to provide the interference immunity for the history of the schedule data, tariffs and classifiers (the method of restricting the access to the relation systems data on the level of the relation tuples);
   - to achieve the demanded speed of performance by the way of employing the different indexation methods, including the clusterization procedure.

2. The employment of the developed models and methods in the information systems *APFIS* has permitted:

   - to decrease the risk of damaging the warehouse data integrity;
   - to simplify the data transformation and the data preparation for the analysis for any period of time;
   - to increase the efficiency and safety of these operations;
   - to modernize the centralized databases of the railway classifiers and codifiers.

3. The results of implementing the developed models and methods in the information systems of the passenger trains scheduling *SAR* and the Financial-Statistical information system of the passenger transportation *APFIS* on the Latvian State Joint Stock Company “Latvijas dzelzceļš” prove their practical application. At the same time the results of this investigation can be sought for by other information systems of the transport enterprises.

4. The complex implementation of the models and methods, developed in the process of this thesis research, in the data warehouses systems represent the unique method of designing the system of bulk data volumes processing, taking into consideration the temporal component of the data and the system processes.

CONCLUSIONS

The accomplished research principal results can be presented in the terms of the conclusion:

1. There have been determined the fundamental groups of the temporal data in the information systems of the Latvian Railway; the interconnection of these data and the issues under consideration have been examined. There have been defined the problems of employment of the temporal data in the operative information systems, including the following: the temporal data models imperfection; insufficient support of the data integrity; hardships of the data privacy provision; the absence of the efficient temporal data processing procedure; incompleteness of the processes of the data preparation to analysis.

2. There have been developed the series of the theoretical fundamentals in the field of the temporal databases, including: the new form of the temporal relation *Historical-Overlapped* introduction (relation with the crossed time segments); designing of the unique classification of the methods and models, subjected to the employment in the temporal systems under consideration; the formal characterisation of the temporal integrity restrictions for the relation temporal forms (the condition of the avoiding crossing of the time segments).

3. There have been suggested the fundamentals of the temporal databases designing in the relation environment, employing the temporal data model with the object abstract identifier and the temporal logic.

4. There have been designed the unique models and methods, oriented on the operation with the temporal objects with account for the railway transport tasks peculiarities:

- temporal relation model with the crossed lifespans;
- temporal object model with the periodicity and the specific calendar;
- method of providing the user’s temporal environment;
- methods of determining the temporal object active version, including: the logical rules method and the temporal elements method;
- method of the interval data transformation;
- method of the temporal data protection with consideration for the enhanced railway information systems requirements for the data integrity, privacy and accessibility.

5. The correctness of the proposed models and methods has been tested in various tasks of the information-analytical systems of the Latvian Railway, primarily in the train scheduling system. The obtained results confirm the validity of the models and the applicability of the methods.

6. The investigation results have allowed the development of the interactive information system for the passenger trains scheduling for the Latvian Railway, as well as achievement of the growing security of the temporal data of the passenger statements

processing system and upgrading the centralized database of the railway classifiers and codifiers. The proposed system of the scheduling support can serve as a basis of the railway transportation information system, but not only; it can be also used as the part of the management system, allowing the trains schedule simulation (by the way of automatic generation of recommendations for the analyst about the trains transfer on another day of the week after assigning the new day-off); it can be a part of the analytical system, supposed for the passenger traffic analysis and forecasting.

7. The research results have been approved in the Latvian Railway information systems. However, they can be implemented in the information systems of other railway companies. Moreover in many cases the obtained results have the universal nature and can be applied in temporal information systems regardless their destination.

8. The investigation results are employed in the process of lecturing in the courses "Databases and Databases Systems" (Study programme "Bachelor of Natural Sciences in Computer Science") and "Databases Modern Technologies" (Study programme "Master of Natural Sciences in Computer Science") in Transport and Telecommunication Institute.

References

List of the Author's Published Works

1. Kopytov E., Petoukhova N., Demidov V. Methodology of Huge Data Volume Processing System Development for Analysis of Latvian Railway Passengers Transportation. In: *Proceedings of VI International Conference "TransBaltica 2001"*, June 7-8, 2001, Riga, Latvia. pp. 201-208.
2. Petukhova N. Methodology of the System Development for Huge Data Volume Processing. In: *Informācijas tehnoloģija: zinības un prakse. VI konference. Konferences materiāli. 2001. gada 28. novembrī. Rīga, Latvija.* Riga: DATI, 2001, pp.65-70. (in Russian)
3. Petoukhova N. Increase of Safety of Data in Relational Database for Records Level. In: *Science and Technology – the Step into the Future: Programme and Abstracts. Research-Practice and Educational-Methodological Conference, Riga, Latvia, May 2-3, 2002.* Riga: TTI, 2002, p. 26.
4. Kopytov J., Demidovs V., Petukhova N. Use of Temporal Databases Principles in Information Systems of the Latvian Railway. In: *Proceedings of VII International Conference "TransBaltica 2002"*, June 12-14, 2002, Riga, Latvia, pp. 183-190. (in Russian)
5. Petukhova N. Detailed Access to Data as a Tool of Increase of Information System Safety. In: *Programme and Abstracts of the International Conference "Reliability and Statistics in Transportation and Communication (RelStat'02)", October 17-18, 2002*. Riga, Latvia, 2002, pp. 73-74. (in Russian)
6. Petuhova N. Method of Access Provision to Relational Systems Data on Relation Row Level. *Transport and Telecommunication*. Volume 4, No 1, 2003, pp. 45-52. (in Russian)
7. Kopitovs J., Demidovs V., Petukhova N. Method of Temporal Databases design Using Relational Environment. In: *Proceedings of the 43rd International Scientific Conference of Riga Technical University*, Riga Technical University, Riga, Latvia. 2002, p. 19.
8. Kopitovs J., Demidovs V., Petukhova N. Method of Temporal Databases design Using Relational Environment. In: *Scientific proceedings of Riga Technical University – Computer Science: Applied Computer Systems*, Series #5, Issue #13. Riga: RTU, 2002, pp. 236-246.
9. Kopitovs J., Demidovs V., Petukhova N. Application of Virtual Data Models in Decision Support Systems. In: *Abstracts of the International Scientific and Technical Conference "Civil Aviation on the Modern Stage of Evolution of Science, Technics and Society"*. April 17-18, 2003, Moscow: MSTUCA, pp. 181-182. (in Russian)
10. J. Kopitovs, V. Demidovs, N. Petukhova. Virtual Models in Forecasting Systems of Railway Transportation. *The international conference: Modelling and Simulation of Business Systems (MOSIBUS 2003).* Vilnius, Lithuania, May 13-14, 2003, pp. 265-268.
11. Kopitovs J., Demidovs V., Petukhova N. Principles of Creating Data Warehouses in Decision Support Systems of Railway Transport. *Sixth International Conference on Computing Anticipatory Systems*: *CASYS '03*. Liège, Belgium, August 11-16, 2003, p.8 of Symposium 8.

12. Petukhova N. Principles of Development of Complex Security Systems in Relational Databases on the Example of Latvian Railway. In: *Proceedings of the International Conference* “*Reliability and Statistics in Transportation and Communication (RealStat’03)”,* October 16-17. 2003, Riga, Latvia. pp. 28-29.
13. Petukhova N. Principles of Development of Complex Security Systems in Relational Databases on the Example of Latvian Railway. In: *Transport and Telecommunication*. Latvia, Riga: TTI, 2004, Volume 5, Nr. 2. pp. 36-44.
14. Kopytov E., Demidovs V., Petoukhova N. Principles of Creating Data Warehouses in Decision Support Systems of Railway Transport. In: *Computing Anticipatory Systems: CASYS 2003 - Sixth International Conference*. Published by The American Institute of Physics, Conference Proceedings 718, 2003, pp. 497-507.
15. Petukhova N. Development of a Complex Security System in Relational Databases for Railway Transport. In: *Scientific proceedings of Sixth International Baltic Conference on Computer Science and Information Technologies - Databases and Information Systems: DB&IS 2004.* Scientific Papers University of Latvia. June 6-9, 2004, Volume 673. pp. 139-150.
16. Petukhova N. Relational Approach to Information Security in Temporal databases. In: *Programme and Abstracts of the 5th International Conference* “*Reliability and Statistics in Transportation and Communication (RealStat’05)”. 13-14 October 2005.* Latvia, Riga: TTI, 2005, pp. 62-63.
17. Petukhova N. The Problems of Information Security Management in Temporal Databases, *RESEARCH and TECHNOLOGY - STEP into the FUTURE*. Riga, Volume 1, No 4, 2006, Latvia. pp. 115-116.
18. Petukhova N. N. Problems of the Information Security Providing in the Temporal Databases. *Transport and Telecommunication*, Volume 7, No 3, 2006, pp. 504-513. (in Russian)
19. Petukhova N. Investigation of Temporal Data Models in Information Systems on the Railway. In: *Proceedings of the 7th International Conference* “*Reliability and Statistics in Transportation and Communication (RelStat’07).* October 24-27, 2007, Riga, Latvia, pp. 168-177.
20. Kopytov E., Demidovs V., Petukhova N. Modelling of Railway Schedule in Temporal Databases. In: *Proceedings of the International Conference ”Modelling of Business, Industrial and Transport System”*. May 7-10, 2008, Riga, Latvia, Transport and Telecommunication Institute, pp. 107-116.
21. Kopytov E., Demidovs V., Petukhova N. Method of Railway Schedule Supporting in Temporal Databases. In: *Proceedings of the 5-th International Conference ”Research, Development and Application of High Technologies in Industry”.* April 28-30, 2008, St. Petersburg, Russia, Polytechnic University, pp. 78-79 (in Russian).
22. Kopytov E., Demidovs V., Petukhova N. Method of Multiversion Railway Schedule Creation in Information Systems. In: *Proceedings of the 6-th International Scientific-Practical Conference ”Research, Development and Application of High Technologies in Industry”.* 16-18 October 2008. St. Petersburg, Russia, Polytechnic University, pp. 17-24 (in Russian).

23. Petukhova N. Temporal Models and Methods of Transport Information Systems. In: *Proceedings of the International Scientific Conference of Junior Researchers "Science – Lithuania's Future. TRANSPORT*". TECHNIKA, Vilnius Gediminas Technical University. 2008. pp. 319-325. (in Russian)
24. Kopytov E., Demidovs V., Petukhova N. Application of Temporal Elements in the Railway Schedule Systems. *Transport and Telecommunication*, 2008, Volume 9 No 2, pp. 14-23.
25. Kopytov E., Petukhova N. Application of Temporal Databases in Modeling of Railway Processes. In: *Proceedings of the International Symposium on STOHASTIC MODELS in RELIABILITY ENGENEERING, LIFE SCIENCE and OPERATIONS MANAGEMENT (SMRLO'10)*, Beer Sheva, Israel, SCE – Shamoon College of Engeneering, pp. 586-594
26. Kopytov E., Petukhova N. Application of Temporal Databases in Modeling of Railway Processes. In: *Book of Abstracts of the International Symposium on STOHASTIC MODELS in RELIABILITY ENGENEERING, LIFE SCIENCE and OPERATIONS MANAGEMENT (SMRLO'10).* February 8-11, 2010, Beer Sheva, Israel, SCE – Shamoon College of Engeneering, p. 135.
27. Kopytov E., Petukhova N., Demidovs V. Methods for Railway Schedule Periodicity Support in Temporal Databases. In: *Proceeding of 9th JOINT CONFERENCE ON KNOWLEDGE-BASED SOFTWARE ENGINEERING (JCKBSE'10). Kaunas, Lithuania, August 25-27, 2010*. Kaunas: Tehnologija, 2010, pp. 178-191.

Regulatory Documents and Concepts, Worked out with the Author's Participation

28. *Concept of Developing the Information System "Electronic Booking in the International Trains" (SAR-R)*. Riga: VAS LDz ISC, 2008, 18 p. (in Latvian)
29. *Requirements Specification of the Information System "Electronic Booking in the International Trains" (SAR-R)*. Riga: VAS LDz ISC, 2009, 43 p. (in Latvian)
30. *Technology of the Information System "Electronic Booking in the International Trains" (SAR-R)*. Riga: VAS LDz ISC, 2010, 50 p. (in Latvian)
31. *Requirements Specification of the Information System Modernisation "Registration of the Hardware and Software" (APNIS)*. Riga: VAS LDz ISC, 2008, 150 p. (in Latvian)
32. *Concept of Development of the Information System „Interactive Schedule of the Trains Traffic"*. Riga: VAS LDz ISC, 2007, 54 p. (in Latvian)

Other Sources

33. *European transport policy for 2010: time to decide. White paper.* Brussels, European Communities. 2001, 119 p., Available at: http://ec.europa.eu/transport/white_paper/index_en.htm
34. Latvijas Republikas satiksmes ministrijas oficiāla mājas lapa. Available at: http://www.sam.gov.lv/
35. *Transporta attīstības pamatnostādnes 2007.-2013.gadam (informatīvā daļa).* Latvijas republikas Ministru kabineta 2006.gada 12.jūlija rīkojums Nr.518.
36. *State Joint Stock Company "Latvijas Dzelzceļš" basic performance indicators in 2004-2009*. Available at: http://www.ldz.lv/?object_id=1928

37. Simon A. *Strategic Database Technology: Management for the Year 2000*. San Francisco: Morgan Kaufmann Publishers, 1995, 446 p.

38. Зиндер Е. З. Проектирование баз данных: новые требования, новые подходы. Материалы конференции *“Корпоративные базы данных ‘96”*. Available at: http://www.citforum.ru/database/kbd96/41.shtml

39. Кузнецов С. Тенденции в мире систем управления базами данных. Материалы конференции *“Корпоративные базы данных ‘96”*. 1996, Available at: http://www.citforum.ru/database/kbd96/48.shtml

40. Кузнецов С., Гринев М. *Управление данными: достижения и проблемы*. Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению "Информационно-телекоммуникационные системы", 2008, 48 с.

41. Navathe S. B. and Ahmed R. A Temporal Relational Model and a Query Language. *Information Sciences*, Volume 49, 1989, pp. 147-175.

42. Richard T. Snodgrass. *Developing Time-Oriented Database Applications in SQL*, Morgan Kaufmann Publishers, Inc., San Francisco, 1999, 504+xxiii pages.

43. Richard T. Snodgrass. Special Series on Temporal Database, Part 1: Of Duplicates and Septuplets. *Database Programming & Design*. June 1998, pp. 46-49.

44. S. B. Navathe and R. Ahmed. Temporal Extensions to the Relational Model and SQL. *Temporal Databases: Theory, Design, and Implementation*, eds. A. Tansel, et al., The Benjamin/Cummings Publishing Company, Inc., 1993, pp. 92-109.

45. The PostgreSQL Global Development Group. Available at: http://developer.postgresql.org

46. PostgreSQL Tutorial, Chapter 5. Advanced Postgres SQL Features. Available at: http://www.ninthwonder.com/info/postgres/tutorial/c0503.htm

47. Stonebraker M. and Rowe L. The design of POSTGRES. In: *Proceedings of the ACM SIGMOD Conference of Data*, Washington, DC, 1986, pp. 340-355.

48. Stonebraker M., Anton J., and Hirohama M. Extendability in POSTGRES. *IEEE Database Engineering*, 10 (2), June 1987, pp. 16-23.

49. Tansel A. U., Clifford J., Gadia S. *Temporal Databases: Theory, Design, and Implementation*. Redwood City, CA: The Benjamin/Cummings Publishing Company. 1993, 633 p.

50. Etzion O., Jajodia S., and Sripada S. (eds.). *Temporal Databases: Research and Practice. Lecture Notes in Computer Science*, Volume 1399, Springer-Verlag 1998, 429 p.

51. Date C. J., Darwen H., Lorentzos N. A. *Temporal Data and the Relational Model: A Detailed Investigation into the Application of Interval and, Relation Theory to the Problem of Temporal Database Management*. Morgan Kaufmann, 2002, 480 p.

52. Jensen C. *“Temporal Database Management”*. Dr. techn. thesis, defended on 14.04.2000, 1328 p., Available at: http://www.cs.auc.dk/~csj/Thesis/

53. Jensen C. “*TDB Glossary*”. http://www.cs.aau.dk/~csj/Glossary/

54. Klassen C. *What is a Temporal Database*?
Available at: http://citm.utdallas.edu/publications/whitepapers/wp_temporaldb.htm

55. Ben-Zvi J. *The Time Relational Model*. PhD thesis, Computer Science Dept., UCLA, 1982.

56. Steiner A. *A Generalisation Approach to Temporal Data Models and their Implementations*. Doctoral Thesis. ETH No.12434, Department of Computer Science, ETH Zurich, Switzerland, 1998, 163 p.
Available at: http://www.globis.ethz.ch/people/former/steiner/thesisSteiner

57. Böhlen M. H. Temporal Database System Implementations. *ACM SIGMOD.* Volume 24, Issue 4, December 1995, pp. 53-60.

58. Segev A., Jensen C. S., Snodgrass R. T. Report on The 1995 International Workshop on Temporal Databases. *ACM SIGMOD*. Volume 24, Issue 4, December 1995, pp. 46-52.

59. Soo M. D., Sripada S. M. *The TSQL2 Temporal Query Language*. Kluwer Academic Publishers, 1995. Available at ftp://ftp.cs.arizona.edu/tsql/tsql2/bookspec.pdf, A TSQL2 Commentary ftp://ftp.cs.arizona.edu/tsql/tsql2/eval.pdf

60. Bjørn Skjellaug. *Temporal Data: Time and Relational Databases*. Research Report 246. UNIVERSITY OF OSLO, Department of Informatics Oslo, Norway. April 1997.

61. Jensen C. S. and Soo M. D. and Snodgrass R. T. Unifying Temporal Data Models via a Conceptual Model, *Information Systems*, 1994, Volume 1, pp. 513-547.

62. Terenziani P. Symbolic User-Defined Periodicity in Temporal Relational Databases. *IEEE Transactions on Knowledge and Data Engineering*. February 2003, Volume 15, Issue 2, pp. 489-509.

63. Terenziani P., A. Luca. Temporal reasoning about composite and/or periodic events. *Journal of Experimental & Theoretical Artificial Intelligence (Taylor & Francis)*. March 2006, Volume 18, No 1, pp. 87-115.

64. Luca A. *Representing and reasoning with classes and instances of possibly periodic events. Theory, algorithms and applications.* Dr. techn. thesis, 2005, 118 p.

65. Terenziani P., Snodgrass R. Reconciling Point-based and Interval-based Semantics in Temporal Relational, *A TimeCenter TechnicalReport*. June 2001, 46 p., Available at: http://www.di.unito.it/~terenz/TECH-REP/TimeCenter-TR60.pdf

66. TimeCenter official site, 2000-2010. Available at: http://www.cs.aau.dk/TimeCenter

67. Костенко Б.Б., Кузнецов С.Д. История и актуальные проблемы темпоральных баз данных. *Труды Института системного программирования*, т. 13, часть 2, М., ИСП РАН, 2007, с. 77-114.
Available at: http://www.citforum.ru/database/articles/temporal/

68. David B. Lomet, Roger S. Barga, Mohamed F. Mokbel, German Shegalov, Rui Wang, Yunyue Zhu. Immortal DB: transaction time support for SQL server. *SIGMOD Conference 2005*. pp. 939-941.

69. Allen J. F., Maintaining knowledge about temporal intervals. *Communications of the ACM*, Volume 26 n.11, Nov. 1983, pp. 832-843.

70. Wiederhold, G., Fries J.F., Weyl S. Structured Organization of Clinical Data Bases. In: *Proceedings of the AFIPS National Computer Conference*. AFIPS. 1975, pp. 479-485.

71. Richard T. Snodgrass, Temporal Databases: Status and Research Directions. *ACM SIGMOD Record*, special issue on Directions for Future Database Research, W. Kim, ed., 19(4). December 1990, pp. 83-89.
72. Elmasri R., Lee J. Y.: Implementation Options for Time-Series Data. *Temporal Databases: Research and Practice. Lecture Notes in Computer Science*, Volume 1399, Springer-Verlag. 1998, pp. 115-128.
73. David Toman: Point-Based Temporal Extensions of SQL and Their Efficient Implementation. *Temporal Databases: Research and Practice. Lecture Notes in Computer Science*. Volume 1399, Springer-Verlag, 1998, pp. 211-237.
74. Ling Liu, Özsu M. T. (eds.): *Encyclopedia of Database Systems*. US: Springer, 2009, 3752 p.
75. Jensen C. S., Snodgrass R. T. (eds), *Temporal Database Entries for the Springer Encyclopedia of Database Systems*, TimeCenter TR-90, May 2008, 337 p.
76. Bair J., M. Böhlen H., Jensen C. S., and Snodgrass R. T. Notions of Upward Compatibility of Temporal Query Languages. *Wirtschaftsinformatik*, 39(1), February 1997, pp. 25-34.
77. Clifford J., Dyreson C., Isakowitz T., Jensen C. S., Snodgrass R. T. On the Semantics of “NOW” in Databases. *ACM Transactions on Database Systems*, 22, June 1997, pp. 171-214.
78. Elisa Bertino, Claudio Bettini, Elena Ferrari, Pierangela Samarati, A Temporal Access Control Mechanism for Database Systems, *IEEE Transactions on Knowledge and Data Engineering*, Volume 8 n.1, February 1996, p. 67-80.
79. Gadia S.K. A Homogeneous Relational Model and Query Languages for Temporal Databases. *ACM Transactions on Database Systems*, Volume 13, Issue 4, December 1988, pp. 418-448.
80. Gadia S. K. Applicability of Temporal Data Models to Query Multilevel Security Databases: A Case Study. *Temporal Databases: Research and Practice. Lecture Notes in Computer Science*, Volume 1399, Springer-Verlag, 1998, pp. 238-256.
81. Bhargava G., Gadia S. K. Relational Database Systems with Zero Information Loss. *IEEE Transactions on Knowledge and Data Engineering*, Volume 5 no.1, February 1993, pp. 76-87.
82. Gregersen H., Christian S. Jensen. Temporal Entity-Relationship Models-A Survey, *IEEE Transactions on Knowledge and Data Engineering*. Volume 11 no.3, May 1999, pp. 464-497.
83. Gendrano J., Huang B., Rodrigue J., Moon B., Snodgrass R., Parallel Algorithms for Computing Temporal Aggregates. In: *Proceeding of ICDE*, 1999.
84. Kline N., Snodgrass R. Computing Temporal Aggregates. In: *Proceeding of ICDE*, 1995, pp. 222-231.
85. Moon B., Lopez I., Immanuel V. Scalable Algorithms for Large Temporal Aggregation, In: *Proceeding of ICDE*, 2000.
86. Tuma P. *Implementing Historical Aggregates in TempIS*. Master's thesis, Wayne State University, Michigan, 1992.

87. Ye X., Keane J. Processing temporal aggregates in parallel. In: *Proceeding of Int. Conf. on Systems, Man, and Cybernetics*, 1997, pp. 1373-1378.

88. Yang J., Widom J. Incremental Computation and Maintenance of Temporal Aggregates. In: *Proceedings 17th International Conference on Data Engineering*, 2-6 April 2001, pp. 51-60.

89. Zhang D., Markowetz A., Tsotras V. J., Gunopulos D., Seeger B. Efficient Computation of Temporal Aggregates with Range Predicates. In: *Proceeding of PODS*, 2001, 9 p.

90. Guoqin Sun. PRaCoSy: *Data Structures for Time-table Projections*. Technical Report sgq/4/1, UNU/IIST, UNU/IIST, P.O.Box 3058, Macau, July 1994.

91. Guoqin Sun. *Formal Models of Time-table Input Tool*. Technical Report sgq/11/3, UNU/IIST, P.O.Box 3058, Macau, October 1994.

92. Kumar A., Tsotras, V.J., Faloutsos, C. Designing access methods for bitemporal databases, *IEEE Transactions on Knowledge and Data Engineering*, Jan/Feb 1998, Volume 10, Issue 1, pp. 1-20.

93. Snodgrass R. T. The Temporal Query Language TQUEL. *ACM Transactions on Database Systems*, 12(2), June 1987, pp. 247-298.

94. Jensen C. S., Mark L., Roussopoulos N., Sellis T. Using Caching, Cache Indexing, and Differential Techniques to Efficiently Support Transaction Time. *VLDB Journal*, 2(1), 1992, pp. 75-111.

95. Gadia S. K. *A Seamless Generic Extension of SQL for Querying Temporal Data*. Technical Report TR-92-02, Computer Science Department, Iowa State University, March 1992.

96. McKenzie E., Snodgrass R. T. *Supporting Valid Time in an Historical Relational Algebra: Proofs and Extensions*. Technical Report TR–91–15, Department of Computer Science, University of Arizona, Tucson, AZ, August 1991.

97. Torp K., Jensen C. S., Snodgrass R. T. Stratum Approaches to Temporal DBMS Implementation. In: *Proceedings of the International Database Engineering and Applications Symposium*. Cardiff, Wales, U.K., July 8–10, 1998, pp. 4-13.

98. *Temporal Table Support. Release 13.10, B035-1182-109A.* Teradata Corporation. September 2010, 238 p.

99. Latest Teradata Database Release Supports Big Data and the Convergence of Advanced Analytics. *HPCwire*. October 25, 2010.
Available at: http://www.hpcwire.com/offthewire/Latest-Teradata-Database-Release-Supports-Big-Data-and-the-Convergence-of-Advanced-Analytics-105685053.html

100. Skyt J., Jensen C.S., and Mark L., A Foundation for Vacuuming Temporal Databases. *Data and Knowledge Eng.*, Volume 44, No 1, January 2003, pp. 1-29.

101. *Workshop Proceedings "Temporal Data Mining: Algorithms, Theory and Applications" The Fifth IEEE International Conference on Data Mining (ICDM'05)*. November 27-30, 2005, Houston, Texas, USA.

102. DoD 5200.28-STD, *Trusted Computer System Evaluation Criteria*, TCSEC, 1985.

103. *Information Technology Security Evaluation Criteria*, ITSEC, Luxembourg: Office for Official Publications of the European Communities, 1991, 171 pp.

104. *Common Criteria for Information Technology Security Evaluation*, ISO/IEC 15408, 2009.

105. Lindblad S. Security Without Multiple Databases: Oracle's Virtual Private Database. *Oracle Open World 2001*. Berlin, 2001.

106. Козленко Л. Информационная безопасность в современных системах управления базами данных, *КомпьютерПресс* 3, 2002, p. 104-108.

107. Галатенко В.А., *Стандарты информационной безопасности: Курс лекций.* Москва: ИНТУИТ.ру (Интернет-Университет Информационных Технологий), 2004, 328 с. Available at: http://publish.abitu.ru/courseslibrary/ims/c_yfksu/secbasics/lectures/lecture+5.esp

108. Bell E., LaPadula L. J. *Secure computer systems: Unified exposition and multics interpretation*. Technical Report MTR-2997, The Mitre Corporation, Burlington Road, Bedford, MA 01730, USA.

109. *Preliminary information for International Symposium on Secure Software Engineering ISSSE 06 – IEEE*. Available at: http://www.ieee-security.org/Calendar/cfps/cfp-ISSSE06.html

110. Microsoft Corporation with Andy Ruth and Kurt Hudson. *Security+ Certification Training Kit*, Microsoft Corporation. 2003, 512 p.

111. Osborn S. Database Security Integration using Role-Based Access Control, *IFIP WG11.3 Working Conference on Database Security*. 2000, pp. 245 - 258.

112. Вьюкова Н., Галатенко В. Информационная безопасность систем управления базами данных. *Системы управления базами данных*. 1996, no.1. Open Systems Publishing House, Москва. с. 29-54. Available at: http://www.citforum.ru/database/kbd96/49.shtml

113. Demidovs V. *Development of Models of the Dataware of Decision Support Systems at the Railway*. Ph.D. thesis. Riga: Transport and Telecommunication Institute. 2006, 147 p.

114. Demidovs V. Use of Methods for the Increase of Data Reliability in Decision Support Systems of the Latvian Railway. In: *Transport and Telecommunication*, 2003, Volume 4, Nr.1. pp. 16-22. (In Russian).

115. Мухопад Ю.Ф. *Микроэлектронные информационно-управляющие системы*. Учебное пособие. Иркутск: ИрГУПС, 2004, 404 с.

116. *Системы автоматизации и информационные технологии управления перевозками на железных дорогах:* Учебник для вузов ж.-д. транспорта. Москва: Маршрут, 2006, 544 с.

117. *Проектирование информационных систем на железнодорожном транспорте*: Учебник для вузов ж.-д. транспорта. Лецкий Э.К., Крепкая З.А., Маркова И.В. и др. Москва: Маршрут, 2003, 408 с.

118. Gregory I. A Place in History: A Guide to Using GIS in Historical Research. 2002, Available at: http://hds.essex.ac.uk/g2gp/gis/index.asp

119. Wei Xu, Hou-Kuan Huang, Yong Qin. A spatio-temporal forecasting method of railway passenger flow. In: *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, 26-29 August. 2004, Volume 3, pp. 1550-1554. Available at: http://ieeexplore.ieee.org/servlet/opac?punumber=9459

120. Reggiani A., Schintler L. A. (eds.) *Methods and Models in Transport and Telecommunications: Cross Atlantic Perspectives*. Springer-Verlag Berlin Heidelberg, 2005, 364 p.

121. Lamma E., Mello P., Milano M. A distributed constraint-based scheduler. *Artificial Intelligence in Engineering*, Elsevier Science, Great Britain, April 1997, Volume 11, No 2, pp. 91-105.

122. Donggen Wang, Tao Cheng. A spatio-temporal data model for activity-based transport demand modelling. *International Journal of Geographical Information Science*. Volume 15, No 6, September 2001, pp. 561-585.

123. Wei Xu, Yong Qin, Hou-Kuan Huang. A new method of railway passenger flow forecasting based on spatio-temporal data mining. In: *Proceedings of the 7th international IEEE conference on intelligent transportation systems (ITSC 2004, Washington DC)*. 2004, pp. 402-405.

124. Barber F., Tormos P., Lova A., Ingolotti L., Salido M. A., Abril M. A Decision Support System for Railway Timetabling (MOM): the Spanish Case. *Comprail'2006*. WIT Trans. in The Built Environment: Computer in Railways, Volume 10, 2006, pp. 235-244.

125. Barber F., Salido M.A. Ingolotti L., Abril M., Lova A., Tormos P. An Interactive Train Scheduling Tool for Solving and Plotting Running Maps, *Current Topics in Artificial Intelligence LNAI 3040*. 2004, pp. 646-655.

126. Bussiecky M.R., Winter T. Zimmermann U.T. Discrete optimization in public rail transport. *Mathematical Programming* 79(3), 1997, pp. 415-444.

127. Chiu C.K., Chou C.M., Lee J.H.M., Leung H.F., Leung Y.W., A Constraint-Based Interactive Train Rescheduling Tool. *Constraints*. Volume 7, No 2. 2002, pp. 167–198.

128. Krueger H. Parametric Modelling in Rail Capacity Planning. *Canadian National Railway*. Montreal, Canada, 1998.

129. Lindner T. *Train schedule optimization in public rail transport*, Ph. Dissertation, Technische Universitat Braunschweig, Germany, 2000.

130. Oliveira E; Smith B. M. A job-shop scheduling model for the single-track railway scheduling problem. In: *Proceedings of the 19th Workshop of the UK Planning and Scheduling Special Interest Group - PlanSIG*, 2000, pp.145-160.

131. Salido M.A., Barber F. A Constraint Ordering Heuristic for Scheduling Problem, *1st Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2003)*, (2). 2003, pp. 476-491.

132. Weits, E. Railway Capacity and Timetable Complexity. In: *Proceedings of the 7th International Workshop on Project Management and Scheduling* (PMS '00), Osnabrück, Germany, 2000, 4 p.

133. Zuidwkij R.A., Kroon L.G. Integer Constraints for Train Series Connections, Erasmus Research Institute of Management (ERIM). Discussion Paper. 2002, 16 p.

134. *Proceeding of the International conference Algorithmic MeThods and Models for Optimization of RailwayS (ATMOS 2002). Electronic Notes in Theoretical Computer Science*. Elsevier, 2002, 67 p.

Appendix 1. Indicators of the Railways of the World States

Prepared using statistical data from *Online The World Factbook (2005)*
(see https://www.cia.gov/cia/publications/factbook/index.html)

Rank	Country	Railway network (km)	Country area (km^2)	Coverage density	Territory coverage		Population*
					km^2	%	
1	2	3	4	5	6	7	8
1	Germany	47,201	357,021	132.21	283,206	79.32%	82,422,299
2	Czech Republic	9,572	78,866	121.37	57,432	72.82%	10,235,455
3	Belgium	3,521	30,528	115.34	21,126	69.20%	10,379,067
4	Switzerland	4,583	41,290	111.00	27,498	66.60%	7,523,934
5	Hungary	7,937	93,030	85.32	47,622	51.19%	9,981,334
6	Slovakia	3,662	48,845	74.97	21,972	44.98%	5,439,448
7	Poland	23,072	312,685	73.79	138,432	44.27%	38,536,869
8	Austria	6,011	83,870	71.67	36,066	43.00%	8,192,880
9	United Kingdom	17,156	244,820	70.08	102,936	42.05%	60,609,153
10	Taiwan	2,497	35,980	69.40	14,982	41.64%	23,036,087
11	Netherlands	2,808	41,526	67.62	16,848	40.57%	16,491,461
12	Italy	19,459	301,230	64.60	116,754	38.76%	58,133,509
13	Japan	23,556	377,835	62.34	141,336	37.41%	127,463,611
14	Denmark	2,673	43,094	62.03	16,038	37.22%	5,450,661
15	Slovenia	1,229	20,273	60.62	7,374	36.37%	2,010,347
17	France	29,085	547,030	53.17	174,510	31.90%	60,876,136
18	Uganda	1,244	23,604	52.70	7,464	31.62%	28,195,754
19	Croatia	2,726	56,542	48.21	16,356	28.93%	4,494,749
20	Romania	11,385	237,500	47.94	68,310	28.76%	22,303,552
21	Kenya	2,778	58,265	47.68	16,668	28.61%	34,707,817
22	Ireland	3,312	70,280	47.13	19,872	28.28%	4,062,235
23	Serbia	4,135	88,361	46.80	24,810	28.08%	9,396,411
24	Korea North	5,214	120,540	43.26	31,284	25.95%	23,113,019
25	Israel	853	20,770	41.07	5,118	24.64%	6,352,117
26	Mozambique	3,123	80,159	38.96	18,738	23.38%	19,686,505
27	Bulgaria	4,294	110,910	38.72	25,764	23.23%	7,385,367
28	Cuba	4,226	110,860	38.12	25,356	22.87%	11,382,820
29	Ukraine	22,473	603,700	37.23	134,838	22.34%	46,710,816
30	Latvia**	2,303	64,589	35.66	13,818	21.39%	2,274,735
31	Korea South	3,472	98,480	35.26	20,832	21.15%	48,846,823
32	Azerbaijan	2,957	86,600	34.15	17,742	20.49%	7,961,619
33	Ecuador	966	28,356	34.07	5,796	20.44%	13,547,510
34	Moldova	1,138	33,843	33.63	6,828	20.18%	4,466,706
35	Portugal	2,850	92,391	30.85	17,100	18.51%	10,605,870
36	Spain	14,873	504,782	29.46	89,238	17.68%	40,397,842
37	Armenia	845	29,800	28.36	5,070	17.01%	2,976,372
38	Macedonia	699	25,333	27.59	4,194	16.56%	2,050,554
39	Lithuania	1,771	65,200	27.16	10,626	16.30%	3,585,906
40	Belarus	5,512	207,600	26.55	33,072	15.93%	10,293,011
41	Sweden	11,481	449,964	25.52	68,886	15.31%	9,016,596
42	United States	226,605	9,631,420	23.53	1,359,630	14.12%	298,444,215
43	Georgia	1,612	69,700	23.13	9,672	13.88%	4,661,473

1	2	3	4	5	6	7	8
44	Sri Lanka	1,449	65,610	22.09	8,694	13.25%	20,222,240
45	Estonia	958	45,226	21.18	5,748	12.71%	1,324,333
46	Cameroon	987	47,544	20.76	5,922	12.46%	17,340,702
47	Cote d'Ivoire	660	32,246	20.47	3,960	12.28%	17,654,843
48	Greece	2,571	131,940	19.49	15,426	11.69%	10,688,058
49	India	63,230	3,287,590	19.23	379,380	11.54%	1,095,351,995
50	Bangladesh	2,768	144,000	19.22	16,608	11.53%	147,365,352
51	South Africa	20,872	1,219,912	17.11	125,232	10.27%	44,187,637
52	Finland	5,741	338,145	16.98	34,446	10.19%	5,231,372
53	Albania	447	28,748	15.55	2,682	9.33%	3,581,655
54	New Zealand	4,128	268,680	15.36	24,768	9.22%	4,076,140
55	Botswana	888	60,037	14.79	5,328	8.87%	1,639,833
56	Syria	2,711	185,180	14.64	16,266	8.78%	18,881,361
57	Madagascar	854	58,704	14.55	5,124	8.73%	18,595,469
58	El Salvador	283	21,040	13.45	1,698	8.07%	6,822,378
59	Tunisia	2,153	163,610	13.16	12,918	7.90%	10,175,014
60	Norway	4,077	323,802	12.59	24,462	7.55%	4,610,820
61	Bosnia and Herzegovina	608	51,129	11.89	3,648	7.13%	4,498,976
62	Uruguay	2,073	176,220	11.76	12,438	7.06%	3,431,932
63	Argentina	31,902	2,766,890	11.53	191,412	6.92%	39,921,833
64	Turkey	8,697	780,580	11.14	52,182	6.69%	70,413,958
65	Dominican Republic	517	48,730	10.61	3,102	6.37%	9,183,984
66	Pakistan	8,163	803,940	10.15	48,978	6.09%	165,803,560
67	Togo	568	56,785	10.00	3,408	6.00%	5,548,702
68	Mexico	17,562	1,972,550	8.90	105,372	5.34%	107,449,525
69	Uzbekistan	3,950	447,400	8.83	23,700	5.30%	27,307,134
70	Chile	6,585	756,950	8.70	39,510	5.22%	16,134,219
71	Guyana	187	21,497	8.70	1,122	5.22%	767,245
72	Guatemala	886	108,890	8.14	5,316	4.88%	12,293,545
73	Thailand	4,071	514,000	7.92	24,426	4.75%	64,631,595
74	Vietnam	2,600	329,560	7.89	15,600	4.73%	84,402,966
75	Zimbabwe	3,077	390,580	7.88	18,462	4.73%	12,236,805
76	China	74,408	9,596,960	7.75	446,448	4.65%	1,313,973,713
77	Venezuela	682	91,205	7.48	4,092	4.49%	25,730,435
78	Malawi	797	118,480	6.73	4,782	4.04%	13,013,926
79	Honduras	699	112,090	6.24	4,194	3.74%	7,326,496
80	Australia	47,738	7,686,850	6.21	286,428	3.73%	20,264,082
81	Burma	3,955	678,500	5.83	23,730	3.50%	47,382,633
82	Malaysia	1,890	329,750	5.73	11,340	3.44%	24,385,858
83	Jordan	505	92,300	5.47	3,030	3.28%	5,906,760
84	Costa Rica	278	51,100	5.44	1,668	3.26%	4,075,261
85	Benin	578	112,620	5.13	3,468	3.08%	7,862,944
86	Russia	87,157	17,075,200	5.10	522,942	3.06%	142,893,540
87	Egypt	5,063	1,001,450	5.06	30,378	3.03%	78,887,007
88	Kazakhstan	13,700	2,717,300	5.04	82,200	3.03%	15,233,244
89	Iraq	2,200	437,072	5.03	13,200	3.02%	26,783,383
90	Turkmenistan	2,440	488,100	5.00	14,640	3.00%	5,042,920
91	Canada	48,467	9,984,670	4.85	290,802	2.91%	33,098,932
92	Senegal	906	196,190	4.62	5,436	2.77%	11,987,121
93	Panama	355	78,200	4.54	2,130	2.72%	3,191,319
94	Iran	7,256	1,648,000	4.40	43,536	2.64%	68,688,433

1	2	3	4	5	6	7	8
95	Morocco	1,907	446,550	4.27	11,442	2.56%	33,241,259
96	Ghana	953	239,460	3.98	5,718	2.39%	22,409,572
97	Tanzania	3,690	945,087	3.90	22,140	2.34%	37,445,392
98	Nigeria	3,505	923,768	3.79	21,030	2.28%	131,859,731
99	Brazil	29,252	8,511,965	3.44	175,512	2.06%	188,078,227
100	Guinea	837	245,857	3.40	5,022	2.04%	9,690,222
101	Tajikistan	482	143,100	3.37	2,892	2.02%	7,320,815
102	Indonesia	6,458	1,919,440	3.36	38,748	2.02%	245,452,739
103	Cambodia	602	181,040	3.33	3,612	2.00%	13,881,427
104	Bolivia	3,519	1,098,580	3.20	21,114	1.92%	8,989,046
105	Gabon	814	267,667	3.04	4,884	1.82%	1,424,906
106	Philippines	897	300,000	2.99	5,382	1.79%	89,468,677
107	Colombia	3,304	1,138,910	2.90	19,824	1.74%	43,593,035
108	Zambia	2,173	752,614	2.89	13,038	1.73%	11,502,010
109	Namibia	2,382	825,418	2.89	14,292	1.73%	2,044,147
110	Peru	3,462	1,285,220	2.69	20,772	1.62%	28,302,603
111	Congo Republic of the	894	342,000	2.61	5,364	1.57%	3,702,314
115	Sudan	5,978	2,505,810	2.39	35,868	1.43%	41,236,378
116	Kyrgyzstan	470	198,500	2.37	2,820	1.42%	5,213,898
117	Burkina Faso	622	274,200	2.27	3,732	1.36%	13,902,972
118	Angola	2,761	1,246,700	2.21	16,566	1.33%	12,127,071
119	Congo Democratic Republic of the	5,138	2,345,410	2.19	30,828	1.31%	62,660,551
120	Algeria	3,973	2,381,740	1.67	23,838	1.00%	32,930,091
121	Mongolia	1,810	1,564,116	1.16	10,860	0.69%	2,832,224
122	Paraguay	36	40,675	0.89	216	0.53%	6,506,464
123	Saudi Arabia	1,392	1,960,582	0.71	8,352	0.43%	27,019,731
124	Ethiopia	681	1,127,127	0.60	4,086	0.36%	74,777,981
125	Mali	729	1,240,000	0.59	4,374	0.35%	11,716,829
126	Nepal	59	147,181	0.40	354	0.24%	28,287,147
127	Nicaragua	6	129,494	0.05	36	0.03%	5,570,129

* 2006 year

** The Latvian Railway takes the 30th position in the world according to the railway network coverage density indicator among the countries with the area of more than twenty thousand square kilometers

Appendix 2. Reference on Natalia Petukhova Participation in the Development of Information Systems on the Latvian Railway

VALSTS AKCIJU SABIEDRĪBA
LDz LATVIJAS DZELZCEĻŠ
Reģ. Nr. 40003032065

Informatīvās skaitļošanas centrs

Turgeņeva ielā 21, Rīgā, LV-1547. Tālruņi: 67233902, 67233900. Fakss: 67234366. E-pasts: isc@ldz.lv
Norēķinu konts LV60RIKO0002013100262 A/S "DnB NORD Banka" kods RIKOLV2X

Rīgā

24 . 08 . 2010 Nr. ISC-23/176

Transporta un sakaru institūta
Promocijas padomei

Izziņa par piedalīšanos informācijas sistēmu izstrādē

Ar doto dokumentu apliecinām, ka Nataļja Petuhova (p.k. 200677-14453) strādā VAS „Latvijas Dzelzceļš" Informatīvās skaitļošanas centrā no 1999. gada 01. februāra un piedalījās šādu "Latvijas dzelzceļa" projektos:

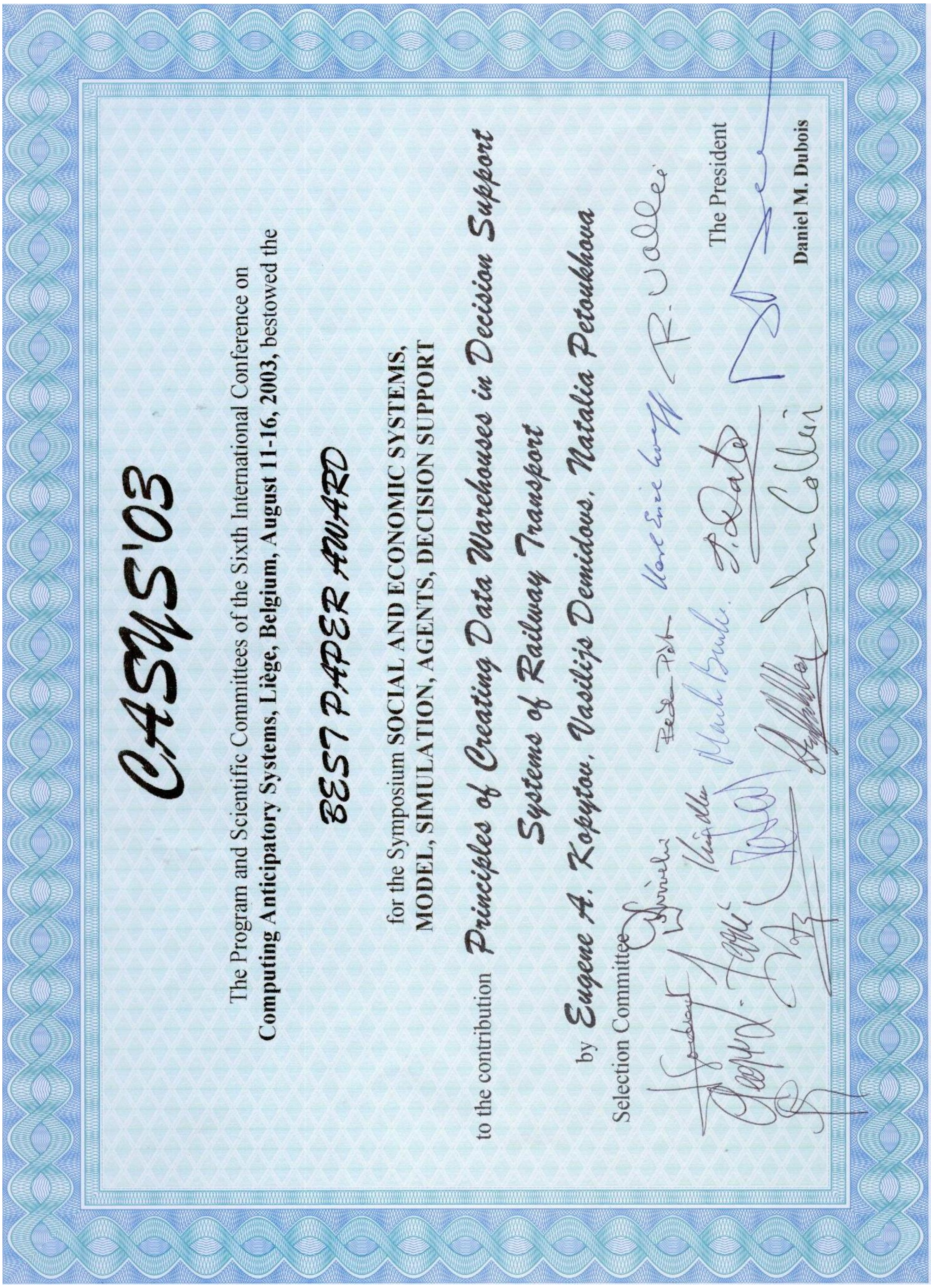
- Informācijas sistēmas „Iekšzemes satiksmes vilcienu kustības saraksts" izstrādē (datubāzes projektētāja, implementētāja);
- Informācijas sistēmas „Starptautiskās satiksmes vilcienu kustības saraksts" izstrādē (datubāzes projektētāja, implementētāja)
- Informācijas sistēmas „E-rezervēšana starptautiskajos vilcienos" izstrādē (sistēmanalītiķe, datubāzes eksperte);
- Informācijas sistēmas „Pārkāpumu analīze" izstrādē (sistēmas arhitekte, datubāzes un programmatūras izstrādātāja)
- „Aparatūras un programmatūras nomenklatūra" informācijas sistēmas izstrādē (projektu vadītāja, sistēmas arhitekts, datubāzes izstrādātāja)
- Kravu pārvadājumu KRAIS-2 informācijas sistēmas izstrādē (datubāzes eksperte, sistēmanalītiķe, implementētāja);
- Informācijas sistēmas „Normatīvi tehnisko dokumentu datu bāzes" izstrādē (sistēmanalītiķe, implementētāja);
- Informācijas sistēmas „LDz normatīvie akti" izstrādē (sistēmanalītiķe, implementētāja);
- Informācijas sistēmas „Apvienotā pasažieru pārvadājumu finanšu statistiskā sistēma" izstrādē (izstrādātāja);
- Augstas pieejamības sistēmas (HADR) tehnoloģijas ieviešana (sistēmanalītiķe, implementētāja);

ISC vadītājs R.Kolnejs

LATVIJAS REPUBLIKA SATIKSMES MINISTRIJA
Informatīvās skaitļošanas centrs
LDz
VALSTS AKCIJU SABIEDRĪBA LATVIJAS DZELZCEĻŠ

Demidovs 67234370

Appendix 3. The Best Paper Award of the sixth international conference on Computing Anticipatory Systems CASYS'03



CASYS'03

The Program and Scientific Committees of the Sixth International Conference on **Computing Anticipatory Systems, Liège, Belgium, August 11-16, 2003**, bestowed the

BEST PAPER AWARD

for the Symposium SOCIAL AND ECONOMIC SYSTEMS, MODEL, SIMULATION, AGENTS, DECISION SUPPORT

to the contribution *Principles of Creating Data Warehouses in Decision Support Systems of Railway Transport*

by *Eugene A. Kopytou, Vasilijs Demidous, Natalia Petoukhova*

Selection Committee

The President

Daniel M. Dubois

Appendix 4. The prize after Karlis Irbitis (doctoral scholarship) from Latvian Academy of Science, "Latvijas Gaisa Satiksme" corporation and Latvian educational fund "Education, science and culture"

Diploms

Latvijas Zinātņu akadēmija, VAS "Latvijas Gaisa Satiksme"
un Latvijas Izglītības fonda mērķprogramma
"Izglītībai, zinātnei un kultūrai"

apbalvo

Nataļju Petuhovu

ar diplomu un K. Irbīša stipendiju par darbu
Transporta informatīvo sistēmu drošības paaugstināšana.

LZA prezidents	VAS "Latvijas Gaisa Satiksme" prezidents	Latvijas Izglītības fonda mērķprogrammas IZK padomes priekšsēdētājs
Juris Ekmanis	Edijs Kizenbahs	Imants Meirovics

Rīgā 200_5_. gada _28.jūlijā_

Nr. 2748 /807

LGS
LATVIJAS GAISA SATIKSME