FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO



Spatio-Temporal Information Management and Visualization

Alexandre Miguel Barbosa Valle de Carvalho

PRODEI 2009

43)
SPA

U.PORTO PRODEI Programa Doutoral em Engenharia Informática

Spatio-Temporal Information Management and Visualization

A dissertation submitted to the

FACULTY OF ENGINEERING OF THE UNIVERSITY OF PORTO

for the degree of Doctor in Informatics Engineering

Presented by

ALEXANDRE MIGUEL BARBOSA VALLE DE CARVALHO

on the 7th de July, 14h30, Great Hall of DEC / FEUP

Supervised by

A. AUGUSTO DE SOUSA

Associate Professor at the Faculty of Engineering of the University of Porto

and CRISTINA RIBEIRO

Auxiliar Professor at the Faculty of Engineering of the University of Porto

President of the júri

FERNANDO NUNES FERREIRA
Faculty of Engineering of the University of Porto

Arguentes

MARCO OCTÁVIO TRINDADE PAINHO
Instituto Superior de Estatística e Gestão de Informação - Universidade Nova de Lisboa

MARIA BEATRIZ ALVES DE SOUSA SANTOS
University of Aveiro

JOSÉ MANUEL MATOS MOREIRA
University of Aveiro

ANTÓNIO FERNANDO VASCONCELOS CUNHA CASTRO COELHO
Faculty of Engineering of the University of Porto

Verificado em 21/7/09

Universidade do Porto
Faculdade de Engenharia
Biblioteca
Nº 101289
CDU 004(043) / CARa / SPA
Data 17 / 08 / 2009

Dedicated to my son and to my wife

Resumo

Desde a década de 1950 verificou-se um aumento progressivo da necessidade de utilizar ou de gerir grandes volumes de informação espacial, por um número crescente de domínios de aplicação. Ao longo deste período, a necessidade de existirem métodos eficientes para representar, armazenar, gerir e aceder a dados espaciais, permitiu identificar e constituir várias linhas de investigação que contribuíram para o desenvolvimento dos Sistemas de Informação Geográfica (SIG). A utilização de gerações de SIGs ajudou a provar que os Sistemas Gestores de Bases de Dados (SGBD) – utilizados previamente no âmbito da gestão de informação alfanumérica – podem ser considerados uma mais-valia para a gestão de informação espacial se a capacidade de representar e de gerir dados espaciais estiver contemplada no conjunto de funcionalidades disponibilizadas pelo SGBD. De facto, na actualidade, uma percentagem muito significativa de SIGs delegam a gestão de dados espaciais em SGBDs que proporcionam esta capacidade através de extensões ao núcleo. Esta aproximação à gestão de dados espaciais tem provado ser uma resposta adequada para um conjunto de domínios de aplicação, tais como a gestão urbana, a logística, a gestão de infra-estruturas, a gestão ambiental, a gestão do uso do solo, a gestão de tráfego, entre outras.

No entanto, também é possível constatar que as soluções actuais de gestão de dados espaciais, baseadas em SGBDs, proporcionam um suporte limitado relativamente à representação e gestão de dados temporais associados. Verifica-se que muitas das aproximações à representação e gestão de dados espaciais se limitam à caracterização do momento mais recente, sendo que os dados espaciais relativos à caracterização de outros períodos do passado são vulgarmente apagados, sobrepostos por dados mais actuais ou armazenados em sistemas de *backup* de mais difícil acesso. É ainda vulgar encontrar outras aproximações à gestão de dados temporais e espaço-temporais onde a gestão da complexidade inerente à natureza destes dados se encontra localizada ao nível da camada aplicacional.

Ambas as aproximações referidas resultam numa gestão deficiente de dados espaço-temporais: a primeira devido ao facto de ignorar e evolução temporal dos factos e a sua representação correspondente, e a segunda por manter a complexidade da gestão de dados espaço-temporais ao nível da cama de aplicação, em vez de explorar as potencialidades optimizadas de um SGBD.

Por outro lado, a investigação científica em representação e gestão de dados temporais, realizada nos últimos 20 anos, proporcionou um vasto conjunto de resultados, designadamente vários modelos conceptuais e de representação de dados temporais em bases de dados, várias propostas para linguagens de *query* temporais e um conjunto significativo de protótipos de SGBD temporais. Na intersecção desta área de investigação com a área dos SIG encontram-se várias possibilidades de investigação, relativamente à prossecução de métodos eficientes de gestão de dados espaço-temporais, bem como relativamente à visualização de informação espaço-temporal, uma vez que através de SGBD espaço-temporais se pode gerir, interrogar e obter mais facilmente maiores volumes de dados espaço-temporais.

Neste contexto, um dos problemas associados à visualização de conjuntos de dados espaço-temporais, reside no facto da aplicação de técnicas mais tradicionais de visualização (por exemplo, a visualização de instantes ou a animação sequencial) poder proporcionar uma representação limitada ou mesmo inadequada dos dados. Através destas técnicas pode ser difícil obter resultados capazes de proporcionar uma interpretação rápida e rigorosa dos dados relativos a instantes ou a vários períodos temporais. Por conseguinte, no contexto da visualização de informação espaço-temporal obtida a partir de SGBDs, considera-se ser necessário a investigação de métodos mais apropriados.

Dado o contexto anterior, o presente trabalho contribui para a gestão eficiente de dados espaço-temporais, propondo novas aproximações às actividades de representação, armazenamento e interrogação, e novos métodos para a visualização de dados espaço-temporais, relativos a instantes e a vários períodos de tempo.

Relativamente à gestão eficiente de dados espaço-temporais, a aproximação proposta introduz suporte para a gestão de dados temporais num SGBD espacial. A linguagem de *query* que resulta desta aproximação consiste na integração da extensão espacial Oracle Spatial na linguagem ATSQL2. Esta linguagem proporciona os meios adequados para a gestão de dados espaço-temporais, sem

introduzir limitações ao suporte proporcionado pelo Oracle Spatial e pelo ATSQL2 quando utilizados isoladamente. Para poder testar e avaliar esta aproximação foi desenvolvido um protótipo de SGBD que implementa a nova linguagem. O conjunto de testes realizado permitiu validar que a nova linguagem proporciona o suporte para as dimensões temporais *valid time* e *transaction time*, semântica sequencial e semântica não sequencial em *queries*, bem como *upward compatibility* e *temporal upward compatibility*. Os resultados obtidos permitem concluir que esta aproximação proporciona o suporte para a gestão de dados espaço-temporais ao nível do SGBD, mantendo os sistemas legados funcionais. Desta forma, a aproximação proposta pode ainda ser considerada uma alternativa viável para o desenvolvimento de novas aplicações com requisitos de gestão espaço-temporal e, simultaneamente, ser uma opção viável para a gestão de informação espacial, necessária em aplicações mais antigas.

No que respeita à visualização de informação espaço-temporal recolhida de um SGBD espaço-temporal, o presente trabalho propõe um novo modelo designado por *Temporal Focus+Context Visualization Model*. No núcleo deste modelo encontra-se a adaptação de uma técnica de visualização de informação, da classe *distorted view*, designada por *Focus+Context*. No entanto, no modelo proposto, a técnica *Focus+Context* explora a informação temporal, enquanto a versão original explora a dimensão espacial ou alternativamente, um espaço de atributos. Desta forma, é possível comprimir grandes volumes de dados, de períodos temporais distintos e obtidos a partir de vários conjuntos de dados, numa mesma visualização.

Dentro deste modelo de visualização encontra-se um conceito designado por *Temporal Degree of Interest* (TDOI) que exprime, de forma quantitativa, o interesse do utilizador por um item de informação temporal. O TDOI é calculado para cada um dos registos de um conjunto de registos, baseado no atributo *valid time* desse registo, bem como no conjunto de funções TDOI que o utilizador cria, de modo a caracterizar um particular grau de interesse temporal.

Na etapa de mapeamento da *pipeline* de visualização, o valor do TDOI correspondente a um registo é utilizado para controlar a representação das variáveis gráficas associadas aos dados desse registo, tais como o grau de transparência, a cor, ou o estilo de desenho. Deste modo, de acordo com o grau de interesse temporal de um registo, a representação visual dos dados pode ser enfatizada ou esbatida, proporcionando assim diferenças na representação que permitem aumentar a quantidade de informação que é apresentada simultaneamente, correspondente a diferentes períodos de tempo, bem como manter a compreensão dos resultados visuais.

Este modelo foi testado com vários cenários de visualização e os resultados demonstram que, por análise visual, é possível identificar as características temporais dos dados pertencentes a um conjunto de registos que representa uma evolução temporal, e de acordo com as funções definidas pelo utilizador para especificar o seu grau de interesse temporal. Deste modo, considera-se que a utilização do modelo proposto é adequada para vários domínios de aplicação onde se requer a análise visual de dados espaço-temporais, tais como, no planeamento urbano e na gestão de infra-estruturas, desde que sejam desenvolvidas funções de mapeamento do TDOI adequadas ao controlo das variáveis gráficas, considerando as especificidades de visualização nesses domínios.

Este modelo pode ser facilmente ampliado, através da incorporação de técnicas de *rendering* mais complexas, tais como *sketch drawing*, texturas, *blur* ou outras técnicas de *rendering* não fotorrealístico. Considera-se que através da incorporação destas técnicas é possível identificar mais facilmente outros aspectos temporais dos dados ou relações temporais entre os dados. A adequabilidade da utilização destas técnicas pode ser desenvolvida a título de trabalho futuro, mas desde já se identifica a necessidade de contributos de áreas tais como a semiologia, o design de informação e do estudo da percepção humana. Adicionalmente, há ainda lugar para a investigação de possibilidades de utilização deste modelo para a visualização de representações de dados que evoluem continuamente ao longo do tempo.

Abstract

Since the 1950s there have been an increasing number of application domains requiring the use of large volumes of spatial data. Such use proved to require efficient methods to represent, store, manage and retrieve spatial data, which led to the research that characterized the evolution of Geographic Information Systems (GIS), starting with monolithic applications using proprietary file formats. Several generations of GIS have helped to prove that database management systems (DBMS) used for managing alphanumeric data could be a valuable asset in GIS if spatial data management support would also be provided as an extension. In fact, most of today's spatial data management in GIS relies on conventional object-relational DBMS with a spatial extension in order to effectively satisfy the requirements of application domains such as urban management, logistics, critical infrastructure management, environmental management, land use management, traffic, among many others.

However, current spatial DBMS provide limited support in what concerns efficient temporal and spatio-temporal data management. Temporal snapshot approaches, in which only the most recent state is kept, are a common practice in GIS; in other approaches the temporal aspects of data are typically managed at the application level. Both these approaches lead to inefficient temporal data management: the first by hiding temporal evolution and the second by keeping the complexity of temporal data management at the application level.

Research in Temporal Database Management over the last 20 years has provided several conceptual data models of time in databases, several Temporal Structured Query Languages, and a set of temporal database management system prototypes. Moreover, the use of spatio-temporal DBMS in GIS triggers new research possibilities concerning information visualization, as larger spatio-temporal data sets are managed.

The visualization of spatio-temporal record sets with traditional visualization techniques, such as snapshots of time or using animation can be hard to interpret. These approaches lack the ability to meaningfully display several instants or periods of time onto the same visualization. Hence, in the context of spatio-temporal information visualization retrieved from STDBMS new methods are required.

This work is a contribution to the efficient spatio-temporal data management, proposing new approaches to the tasks of spatio-temporal data representation, storage and retrieval, and new methods to the visualization of spatio-temporal information, covering several instants or periods.

Concerning efficient spatio-temporal data management, the proposed approach extends a spatial DBMS with temporal support. The resulting query language seamlessly integrates the spatial extension Oracle Spatial within the ATSQL2 language and provides spatio-temporal support without harming the support provided by Oracle Spatial and by ATSQL2. To test and evaluate this approach, a fully-fledged prototype implementing the language system has been developed; tests have been performed considering support for *valid time*, transaction time, sequenced and non-sequenced semantic queries, as well as upward and temporal upward compatibility. The results show that spatio-temporal data management is feasible while keeping legacy applications fully functional.

Concerning the visualization of spatio-temporal information directly gathered from a spatio-temporal DBMS, this work proposes the new Temporal Focus+Context Visualization Model. In this model, Focus+Context, a well-known distorted view technique from the field of Information Visualization, is adapted to be used with time rather than with space or other attributes. Temporal Focus+Context allows large amounts of data from distinct periods of time and from several record sets to be compressed into a single visualization result.

Underlying the visualization model is the proposed concept of Temporal Degree of Interest (TDOI) that quantitatively expresses how much the user is temporally interested on an information item. The TDOI is calculated for each record and is based on the record *valid time* attribute as well as on user-defined visual analysis goals, described by one or more proposed TDOI functions. In the mapping stage of the visualization pipeline, the TDOI is used to control the representation of graphical variables, such as transparency, color, drawing style. Hence, according to the corresponding temporal degree of interest the representation of data can be enhanced or dimmed, allowing the compression of data from several distinct periods of time onto the same visualization result while keeping their meaningfulness.

The model has been applied to several test scenarios and the results demonstrate that visual analysis can depict the temporal features of data from a spatio-temporal record set containing information that discretely evolves over time and according to a user-specified temporal degree of interest. This model is therefore suited to be used in several domains where visual analysis of spatio-temporal data is required, such as urban planning, infrastructure and facility management, provided that appropriate mapping functions are developed.

The developed method can be easily extended with more complex rendering techniques, such as sketch drawing, textures, blur or other non-photorealistic enhancement techniques, to convey the temporal aspects of data by replacing the original graphical features of the record data. The appropriateness of these and other rendering techniques require further research, where contributions from the fields of semiology, information design and human perception are expected. The possibility of using this model for data evolving continuous time also requires further evaluation.

Résumé

Depuis les années 1950, il y a eu un accroissement progressif du besoin d'utiliser ou de gérer de grandes quantités d'information spatiale, par un nombre croissant de domaines d'application. Le long de cette période, le besoin de l' existence des méthodes efficaces pour représenter, stocker, gérer et accéder à des données spatiales a permis d' identifier et de constituer plusieurs lignes de recherche qui ont contribué pour le développement des systèmes d' Information Géographique (SIG). L'utilisation des générations de SIG a aidé à prouver que les Systèmes de Gestion de Bases de Données (SGBD) - utilisés, préalablement, au domaine de la gestion de l'information alphanumérique - peuvent être considérés un atout pour la gestion de l'information spatiale dès que la capacité de représenter et de gérer des données spatiales soit inclue à l'assemblage des fonctionnalités disponibilisés par le SGBD.

En effet, aujourd'hui, un pourcentage très expressif de SIGs délègue la gestion des données spatiales au SGBDs, qui permet cette capacité au moyen des extensions au noyau. Ce rapprochement à la gestion des données spatiales a été une réponse adéquate à un assemblage de domaines d'application, tels que la gestion urbaine, la logistique, la gestion des infrastructures, la gestion de l'ambiant, la gestion de l'usage du sol, la gestion du trafic routier, etc.

Cependant, il est aussi possible constater que les solutions actuelles de gestion des données spatiales, appuyées aux SGBDs, proportionnent un appui limité, en relation à la représentation et à la gestion de données temporelles associées. On vérifie que plusieurs approches à la représentation et à la gestion des données relatives aux caractéristiques d'autres périodes du passé sont ordinairement effacées, superposés par des données plus actuelles ou stockées dans des systèmes de backup d'accès plus difficile.

Il est encore courant de trouver d'autres rapprochements à la gestion de données temporelles et spatio-temporelles, où la gestion de la complexité inhérente à la nature de ces données, où se place au niveau de la couche de l'application.

Les deux rapprochements sous – dits provoquent une gestion déficiente des données spatio-temporelles: la première due au fait d'ignorer l'évolution temporelle des faits et sa correspondante représentation et la seconde pour maintenir la complexité de la gestion des données spatio-temporelles au niveau de la couche de l'application, au lieu d'exploiter les potentialités optimisées d'un SGBD.

D'un autre côté, la recherche en représentation et gestion de données temporelles, au cours des derniers vingt ans, a fourni un ample assemblage de résultats, nommément plusieurs modèles conceptuels et de représentation de bases de données temporelles, de plusieurs propositions de langages de *query* temporels et un assemblage significatif de prototypes de SGBD temporels.

Dans l'intersection de ce domaine de recherche avec celui des SIG, se trouvent plusieurs possibilités de recherche concernant l'obtention des méthodes efficaces de gestion des données spatio-temporelles, aussi bien que la visualisation de l'information spatio-temporelle, une fois qu' à travers le SGBD spatio-temporels on peut gérer, interroger et obtenir plus aisément des volumes de données spatio-temporelles plus grands.

Dans ce contexte, un des problèmes associés à la visualisation des assemblages de données spatio-temporelles se trouve dans le fait de l'application des techniques plus traditionnelles de visualisation (par exemple la visualisation d'instants ou l'animation séquentielle) pouvoir proportionner une représentation limitée ou même inadéquate des données. À travers ces techniques, il peut être facile d'obtenir des résultats capables de proportionner une interprétation rapide et rigoureuse des données relatives à des instants ou à de plusieurs périodes temporelles.

Par conséquent, dans le contexte de la visualisation de l'information spatio-temporelle obtenue du SGBD, on considère être nécessaire la recherche des méthodes plus adéquates.

Dans ce contexte, cet ouvrage contribue à la gestion efficace des données spatio-temporelles, en proposant de nouvelles approches aux activités de représentation, de stockage et d'interrogation, et de nouvelles méthodes pour la visualisation des données spatio-temporelles, relatives à des instants et à des plusieurs périodes de temps.

En ce qui concerne la gestion des données spatio-temporelles, l'approche proposée introduit l'appui pour la gestion des données spatio-temporelles dans un SGBD spatial. Le langage de *query* qui résulte de cette approche consiste à l'intégration de l'extension spatiale Oracle Spatial dans le langage ATSQL 2. Ce langage proportionne les moyens convenables à la gestion des données spatio-temporelles, sans introduire des limitations au support facilité par l'Oracle Spatial et par l'ATSQL2, quand sont utilisés isolément.

L'ensemble d'essais réalisés a prouvé que le nouveau langage proportionne le support pour les dimensions temporelles *valid time* et *transation time*, sémantique séquentielle et sémantique non séquentielle en queries, ainsi que *upward compatibility* et *temporal upward compatibility*.

Les résultats obtenus permettent conclure que cette approche proportionne le support pour la gestion des données spatio-temporelles au niveau du SGBD, en soutenant ces systèmes fonctionnels.

De cette façon, l'approche proposée peut encore être considérée une alternative viable pour le développement de nouvelles applications avec des conditions de gestion spatio-temporelle, et, simultanément, être une option viable pour la gestion de l'information spatiale, nécessaire en des applications plus anciennes.

En ce qui concerne la visualisation de l'information spatio-temporelle recueillie d'un SGBD spatio-temporel, ce travail propose un nouveau modèle désigné par *Temporal Focus + Context Vizualization Model*. Au noyau de ce modèle se trouve l'adaptation d'une technique de visualisation d'information de la classe distorted view, désignée par *Focus + Context*.

Cependant, dans le modèle proposé, la technique *Focus + Context* explore l'information temporelle, tandis que la version originale développe la dimension spatiale, ou alternativement, un espace d'attributs. De cette façon, il est possible comprimer de grands quantités de données, de périodes temporels distincts et obtenus de plusieurs assemblages de données, dans une même visualisation.

Dans ce modèle de visualisation, on trouve un concept désigné de *Temporal Degree of Interest* (TDOI) qu'exprime, de façon quantitative, l'intérêt de l'utilisateur par un item d'information temporelle.

Dans la phase de la configuration du mappe du pipeline de la visualisation, la valeur du TDOI correspondant à un enregistrement est utilisée pour contrôler la représentation des variables graphiques associées aux données de cet enregistrement, tels que le degré de transparence, la couleur, ou le style du dessin.

De cette façon, d'accord avec le degré de l'intérêt temporel d'un enregistrement, la représentation visuelle des données peut être mise en relief ou dégradé, proportionnant, ainsi, des différences dans la représentation, qui permettent augmenter la quantité d'information qui est présentée simultanément, correspondante à de différentes périodes de temps, tel que maintenir la compréhension des résultats visuels.

Ce modèle a été testé avec plusieurs situations de visualisation et les résultats en démontrent qu'à travers l'analyse visuelle c'est possible identifier les caractéristiques temporelles des données appartenant à en ensemble d'enregistrement qui représente une évolution temporelle, et, d'accord avec les fonctions définies par l'utilisateur, pour spécifier le degré de son intérêt temporel.

De cette façon, on considère que l'utilisation du modèle proposé est adéquat à plusieurs domaines d'application où s'exige l'analyse visuelle de données spatio-temporelles, tels que la planification urbaine et la gestion des infrastructures dès que soient développées des fonctions d'établissement du TDOI adaptées au contrôle des variables graphiques, considérant les spécificités de la visualisation dans ces domaines. Ce modèle peut être facilement amplifié, à travers l'incorporation de techniques de *rendering* plus complexes, tells que *sketch drawing*, des textures, *blur* ou d'autres techniques de *rendering* non-photoréaliste.

On considère qu'à travers l'incorporation de ces techniques c'est possible d'identifier plus facilement d'autres aspects temporels des données ou de relations temporelles entre les données.

L'adéquation de l'utilisation de ces techniques peut être développée à titre de travail futur, mais on identifie d'avance le besoin de la contribution des domaines tels que la sémiologie, le design d'information et l'étude de la perception humaine. Il y a encore lieu pour la recherche de la possibilité de l'utilisation de ce modèle pour la visualisation de la représentation des données qui évoluent continuellement tout le long du temps.

Keywords

Geographic information systems

Database management systems

Spatio-temporal data management

Spatio-temporal query-languages

Information visualization

Geovisualization

Spatio-temporal information visualization

Categories and subject descriptors (ACM CCS)

Primary classification

H. Information Systems

H.2 DATABASE MANAGEMENT

H.2.3 LANGUAGES

Subjects: *Data description languages*

Data manipulation languages, query languages

H.2.4 SYSTEMS

***Subjects**: Query Processing*

H.5 INFORMATION INTERFACES AND PRESENTATION (I.7)

H.5.2 User Interfaces (D.2.2, H.1.2, I.3.6)

***Subjects**: Graphical user interfaces (GUI), Theory and methods*

Additional classification

I. Computing Methodologies

I.3 COMPUTER GRAPHICS

I.3.6 Methodology and Techniques

Subjects: Interaction techniques

Acknowledgments

I am deeply indebted to Professor Doutor António Augusto Sousa and Professor Doutor Cristina Ribeiro, both my supervisors, for their constant guidance and continuous support at all the times.

I thank FEUP and INESC Porto for providing the logistics and means that have enabled the research undertaken in this dissertation.

I thank all my colleagues at USIC-INESC Porto for all those intellectually stimulating discussions that have helped me improving this dissertation.

Last but not least, I thank my family for their unconditional support throughout this work.

Table of contents

1 INTRODUCTION 1

1.1 PROBLEM DEFINITION 3
1.2 RESEARCH OBJECTIVES 4
1.3 INNOVATIVE CONTRIBUTIONS 5
1.4 OUTLINE 6

2 SPATIAL AND TEMPORAL DATABASE MANAGEMENT SYSTEMS 9

2.1 GEOGRAPHIC INFORMATION SYSTEMS 10
2.2 SPATIAL DATABASE MANAGEMENT SYSTEMS 12
2.2.1 Spatial Query Languages 14
2.2.2 Spatial DBMS 15
2.3 TEMPORAL DATABASE MANAGEMENT SYSTEMS 17
2.4 REPRESENTATION OF TIME IN DATABASES 19
2.4.1 Data Models for Representing Time-varying Data 21
2.5 TEMPORAL EXTENSIONS TO SQL 23
2.6 ARCHITECTURAL APPROACHES TO THE IMPLEMENTATION OF TEMPORAL DBMS 25
2.7 TEMPORAL AND SPATIO-TEMPORAL DBMS IMPLEMENTATIONS 26
2.8 CONCLUSIONS 31

3 THE ACTION! SPATIO-TEMPORAL DATABASE MANAGEMENT SYSTEM 33

3.1 FEATURED SPATIO-TEMPORAL DATA MANAGEMENT 34
3.2 REPRESENTATION SCHEME 37
3.3 SPATIO-TEMPORAL QUERY LANGUAGE 38
3.3.1 SQL Data Definition Statements 40
3.3.2 SQL Data Modification Statements 41
3.3.3 Temporal Coalescing of Spatially-enabled data 49
3.4 THE ARCHITECTURE OF ACTION! 50
3.4.1 Support for Spatial Data Types, Operators and Functions 53
3.4.2 Support for Table and Column aliases 54
3.4.3 Support for Temporal Operators Performing over Spatial Data 54
3.5 TEST AND RESULTS 58
3.5.1 Spatio-temporal Support for Snapshot Relations 58
3.5.2 Spatio-temporal Support for Valid Time Relations 62
3.6 CONCLUSIONS 72

4 INFORMATION VISUALIZATION 75

4.1 THE VISUALIZATION PIPELINE 78
4.2 FOCUS + CONTEXT TECHNIQUES 80
4.3 MULTIPLE REPRESENTATIONS 84
4.4 INFORMATION VISUALIZATION TAXONOMIES 85
4.5 VISUALIZATION OF LINEAR STRUCTURES 87
4.6 VISUALIZATION OF HIERARCHICAL STRUCTURES 93
4.7 VISUALIZATION OF NETWORKS 98
4.8 VISUALIZATION OF VECTOR SPACES 99
4.9 VISUALIZATION OF MULTIDIMENSIONAL DATA 103
4.10 TIME-DEPENDENT MULTIVARIATE INFORMATION VISUALIZATION 105
4.10.1 Static Visual Representations 109
4.10.2 Dynamic Visual Representations 114
4.11 GEOVISUALIZATION 115
4.12 CONCLUSIONS 119

5 INTERACTIVE VISUALIZATION OF SPATIO-TEMPORAL INFORMATION 125

5.1 PROBLEM DEFINITION 127
5.2 THE TEMPORAL F+C VISUALIZATION MODEL 129
5.3 TEMPORAL DEGREE OF INTEREST 130
5.4 MULTIPLE TDOI FUNCTIONS 131

5.5 MAPPING OF THE TEMPORAL DEGREE OF INTEREST 133
5.6 APPLICATION OF THE TEMPORAL DEGREE OF INTEREST 134
5.7 TEST SCENARIOS 135
5.8 CONCLUSIONS 142

6 CONCLUSIONS AND FUTURE WORK 145

6.1 GIS AND SPATIAL DATABASE MANAGEMENT SYSTEMS 146
6.2 SPATIO-TEMPORAL REQUIREMENTS FOR GIS 147
6.3 ACTION! - A STDBMS 149
6.4 VISUALIZATION OF INFORMATION 151
6.5 INTERACTIVE VISUALIZATION OF SPATIO-TEMPORAL INFORMATION 153
6.6 FUTURE WORK 155

List of figures

Figure 1: *Action!* modules and dataflow overview. 53
Figure 2: The *valid time* of tupple t_1 from *R1* and t_2 from *R2*. 56
Figure 3: The temporal set difference between tuple *t1* and *t2*. 56
Figure 4: A graphical representation of the contents of Table 10. 67
Figure 5: The database facts regarding Poland, Portugal, and the Czech Kingdom. 70
Figure 6: The spatio-temporal results from the sequenced *valid time* spatial union query. 71
Figure 7: Spatio-temporal results from the non-sequenced *valid time* spatial union query. 72
Figure 8: Chi's Information Visualization Data State Reference Model. 78
Figure 9: Santos and Brodlie Information Visualization Pipeline. 79
Figure 10: Steinberg's Parochial New Yorker's view of the World. 82
Figure 11: Map view showing rivers in focus and the context as blurred objects. 83
Figure 12: Leung's taxonomy of presentation techniques for large volumes of data. 85
Figure 13: An illustration of the Seesoft visualization technique. 88
Figure 14: An illustration of the TileBar visualization technique. 89
Figure 15: An illustration of the Table Lens visualization technique. 90
Figure 16: An illustration of the Lifestreams visualization technique. 90
Figure 17: An illustration of the 3D perspective wall visualization technique. 91
Figure 18: An illustration of the Document Lens visualization technique. 92
Figure 19: An example of the WordsEye visualization technique. 92
Figure 20: An example of the WebTOC visualization technique. 93
Figure 21: An illustration of thr 3D cone tree visualization technique. 94
Figure 22: An illustration of the Tree Map visualization technique depicting a computer directory structure. 95
Figure 23: An illustration of the Cushion Treemap visualization technique. 95
Figure 24: An illustration of the Information Cube. Overview (left) and an inner cube (right). 96
Figure 25: The use of Hyperbolic Browser technique for displaying a hierarchical structure.. 97
Figure 26: An illustration of the 3D Hyperbolic Browser visualization. 97
Figure 27: An example of a SemNet 3D visualization for a semantic network. 98
Figure 28: An illustration of the SHriMP prototype main window. 99
Figure 29: a Venn diagram transformation into an iconic display, according to the InfoCrystal visualization technique. 100
Figure 30: A SPIRE ThemeView showing topic distribution in a large document space. 100
Figure 31: A VIBE display. 101
Figure 32: An example of the VisIslands visualization technique (visual clusters of search results). 102
Figure 33: Two perspectives of an InfoSky Browser 2002 visualization. 103
Figure 34: The *Carte Chronologique*. 106
Figure 35: Priestley's Chart of Biography. 106

Figure 36: Lambert graphical analysis of periodic variation regarding soil temperature. 107
Figure 37: Minard's visual history of Napoleon's Russian Campaign. 107
Figure 38: Ibry's diagram of train schedules from Paris to Lyon 108
Figure 39: Categorization schema for visual methods for analyzing time-oriented data. 108
Figure 40: Graphs for viewing time: sector, stacked bar, circle (left to right). 109
Figure 41: An illustration of thr ThemeRiver visualization technique. 111
Figure 42: An illustration of a Spiral Display visualization. 111
Figure 43: An illustration of the Calendar View visualization technique. 112
Figure 44: An illustration of the Spiraclock visualization technique. 112
Figure 45: an example of a *TimeWheel* visualization. 113
Figure 46: An illustration of the MultiComb visualization. 114
Figure 47: Instant focus and interval focus with no context. 128
Figure *48*: Illustration of the TDOI calculation 131
Figure 49: Two constant functions used in the TDOI calculation for three records. 131
Figure 50: Multiple TDOI functions. 132
Figure 51: The Temporal F+C visualization pipeline. 133
Figure 52: TDOI calculation for a record where *I* is sliding from future to past. 135
Figure 53: The temporal F+C configuration window. 136
Figure 54: View of the political history of Europe dataset. 137
Figure 55: States of a building evolution across time. 139
Figure 56: Bounding box view of 3D geometry in blocks 139
Figure 57: Spatio-temporal building construction evolution. 140
Figure 58: Illustration of function FI where I=[535-538]. 141
Figure 59: Evolution of the construction. 141

List of tables

Table 1: The temporal compatibility between SQL statements and target tables. 39
Table 2: The query results according to the temporal context in Table 1. 43
Table 3: Execution of an *insert* statement (depends of temporal context defined in Table 1). 44
Table 4: Contents of table *countries2*, prior to the *delete* statement from SQL code 9. 45
Table 5: The new rows introduced by the *delete* statement. 45
Table 6: The table *countries2* after *transaction time* support has been added. 46
Table 7: The table *countries2* after the delete command is executed. 47
Table 8: The contents of *countries2* after and *update* statement 48
Table 9: The result of bi-temporal coalescing of *countries2* after state depicted in Table 8. .. 50
Table 10: The contents of *valid time* table *countries4*. 66

List of SQL statements

SQL Code 1: The creation of table *countries2* with spatial and *valid time* support. 37

SQL Code 2: A spatial metadata declaration for column *boundary* of table *countries2*. 38

SQL Code 3: The creation of table *countries2* with support for spatial entities as well as *valid time* support. 40

SQL Code 4: The temporal upgrade of table *countries2* for *transaction time* support. 41

SQL Code 5: A non-sequenced semantics query statement. 42

SQL Code 6: A sequenced semantics query statement. 42

SQL Code 7: A sequenced spatial bi-temporal statement performing over a coalesced bi-temporal table *countries2*. 43

SQL Code 8: A spatio-unitemporal *insert* statement on bitemporal table *countries2*. 44

SQL Code 9: A sequenced *delete* statement. 45

SQL Code 10: a spatio-temporal update statement performing over bitemporal table *countries2*. 47

SQL Code 11: The bi-temporal coalescing of table *countries2*. 49

SQL Code 12: The *valid time* creation of a table with an attribute of spatial type mdsys.sdo_geometry. 54

SQL Code 13: The creation of a spatial index for column boundary of table *countries2*. 54

SQL Code 14: A sequenced semantics query that retrieves a shape consisting on the union of all geometries for each time period in [1100-1350). 54

SQL Code 15: The use of mdsys.sdo_geometry with nested spatial data types. 54

SQL Code 16: A sequenced semantics query that retrieves rows where Portuguese and Polish borders have met. 54

SQL Code 17: translation of the *valid time* set difference operator into conventional statements. 56

SQL Code 18: The translation of the *valid time* set difference operator into conventional statements and providing support for spatial data. 57

SQL Code 19: The mapping of *valid time* coalescing operation on relation *R1* with a spatial attribute, *C1*, into conventional statements. 58

SQL Code 20: A spatio-temporal sequenced semantics query. 70

SQL Code 21: spatio-temporal non-sequenced semantics query. 71

List of SQL tests

Test 1: The snapshot creation of table *countries2* with support for spatial data. 59

Test 2: The insertion of spatial metadata for column boundary of table *countries2*. 59

Test 3: The index creation for column *boundary*. 59

Test 4: The insertion of a tuple in table *countries2* with facts for country Portugal. 60

Test 5: The retrieval of all the rows of table *countries2*. 60

Test 6: The temporal upgrade of snapshot table *countries2* using non-sequenced semantics. 61

Test 7: Table *k* is the temporal upgrade of a select performed over table *countries2*. After that, unitemporal coalescing is performed for table *k*. 62

Test 8: The contents of *valid time* table *countries2*. 63

Test 9: The snapshot select statement performing over coalesced *valid time* table *countries2*. 64

Test 10: The results for the unitemporal coalescing of Test 9. 64

Test 11: The temporal demotion of *valid time* table *countries2*. 64

Test 12: The snapshot select performed over temporally demoted *countries2*. 65

Test 13: The snapshot select performed over *countries4*. Question is: "what is the current distance between Portugal and Poland?" 68

Test 14: A sequenced semantics select statement asking: "across the time, what has been the current distance between Portugal and Poland?" 69

Test 15: The non-sequenced select statement performed over *countries4* for the largest area for polish borders across time 69

Test 16: A snapshot select statement performed over *valid time* table *countries4*. Query deals with only the current state of data. 70

1 INTRODUCTION

Over the last decades we have seen considerable developments in enabling technologies concerning the tasks associated with geographic information: from acquisition to communication and storage, from processing to presentation and analysis. Accomplishments in areas such as remote sensing and automated data acquisition, networks and hardware, alphanumeric and spatial Database Management Systems (DBMS), information visualization and scientific visualization have been pushing forward the requirements of today's information systems, supporting critical domains of modern societies, such as urban management, traffic, environmental management, among others.

In a wide set of application domains regarding information, problems have been increasing in size and complexity and are also becoming more demanding in what concerns the appropriate representation and management of the temporal aspects of data, such as the rate at which data is captured or processed, or the amount of distinct temporal states that are used in domain-specific analysis. The increasing demand for temporal data management is the immediate consequence of the intrinsic temporal nature of events, whether they result from natural causes or from human activity and of the desire to use all the information available, from distinct temporal states, in domain-specific queries and analysis.

From the perspective of data management systems, both alphanumeric and spatial database technology have provided solid and robust means of keeping track of the most recent state of data for any particular application domain. However, these systems have proved to be inefficient regarding the management of the temporal features of the data.

Hence, most of the times, two approaches have been followed: the first is to manage the most recent state of facts, along with their alphanumeric and spatial features and the second is to time stamp facts with the help of simple data types such as date and time. Both of these approaches proved to have severe limitations concerning the management of temporal aspects of data, since the first approach disregards information from the past which means that, for example, no historic data can be used in queries supporting decisions where temporal evolutions could make a difference.

The second approach (Date 2000) provides a limited management of historical data since problems emerge when handling some constrains and queries, due the lack of proper temporal support from the DBMS. This limitation is mainly due to the use of instants for time stamping where periods would do a much better job (Date 2000) and the lack of proper temporal data management by the database server, such as the ability to issue statements benefiting for upward compatibility (UC) and temporal upward compatibility (TUC) (Snodgrass, Böhlen et al. 1996; Blair, Jensen et al. 1997), involving sequenced semantics (SEQ) and non-sequenced semantics (NSEQ), and using temporal functions (Böhlen, Snodgrass et al. 1996) and operators (F.Allen 1983). Hence, most of the times, the support

for handling temporal data has to be provided at the application level, having drawbacks such as the increase of program complexity, the inefficiency of the algorithms used, sometimes not optimized, and the burden of traffic between the DBMS layer and the application layer.

The limitations resulting from inappropriate temporal data management are more severe in application domains where time is considered an intrinsic dimension of the information being managed, for instance, in application domains that deal with geographic and geospatial information.

The last decade has witnessed the intense use of Geographic Information Systems (GIS) for managing spatial information but, frequently, only the most recent state of facts is addressed while previous states are disregarded or deleted. The underlying reason concerns the database management system supporting the spatial information and its inability to address the information from the past as promptly and easily as the information from the present.

Among other application domains, urban data management, from the central government to the local authorities, has suffered from the lack of adequate temporal support by the database server. One immediate consequence is that the development of applications concerning urban data management and dealing with time-dependent data need to include a set of functions and operators which supports the handling of temporal data. The development of these functions and operators increases program complexity and, consequently, development time. Furthermore there is no guarantee that the implementation of the temporal functions and operators is optimized. Finally, since these functions and operators are available at the application level, extra time is spent in data transfers between the application and the database server, regarding the retrieval of data, to be the input of the functions and operators, and the storage of results. This is more evident in application domains involving geographic information, such as in urban data management, due to the large size of the data sets.

Spatio-temporal DBMS are a more appropriate solution for tasks involving spatio-temporal information representation, management, query, analysis and presentation. The spatial and temporal data management provided directly by a database overcomes the problems just identified since spatial and temporal functions and operators are provided directly by the DBMS. Hence, applications do not need to provide this kind of support, reducing in both program complexity and in the development time and costs. Application performance also increases since intermediate results no longer need to be transferred between the database server and the application level. Finally, it is also expected that the implementation of the set of spatial and temporal operators provided by the DBMS is optimized.

1.1 Problem Definition

The use of spatio-temporal database technology in Geographic Information Systems, through Spatio-Temporal Database Management Systems (STDBMS), can provide the appropriate mean to easily represent, manage, and query information from distinct temporal states (present or past). Moreover STDBMS facilitate certain tasks that are considered time consuming when they are performed in a database management systems without proper temporal support, for instance to query information that involves several distinct temporal states with multiple spatial and temporal selection criteria.

However, until recently fully-fledged STDBMS have been lacking, since in the last few years the research on temporal data management and reasoning has been performed separately from the research in GIS. Moreover, the community researching on temporal data management has provided several models for representing, querying and indexing time, without the existence of a consensus on one unified model widely accepted. Consequently, there are several temporal SQL languages proposed (most of which resulted in a DBMS prototype) and the lack of a standard temporal structured query language (although there is a proposal).

From the described scenario, it is of no surprise that the increasing demand for proper temporal support in GIS has not yet been properly satisfied by appropriate temporal DBMS. The few STDBMS implementations available are prototypes, most of them resulting from scientific research initiatives.

Until the present, enterprises have made important investments in worldwide used GIS applications. These previous investments should be protected as much as possible in a scenario of technological upgrade to a STDBMS. Consequently, the STDBMS should provide support for legacy applications and provide spatio-temporal support for new applications. However, how to provide fully-fledged temporal support in GIS while keeping previous investments done in legacy application may still be considered a challenge, since only in the last few years synergies became to be explored between temporal data management and spatial data management in what concerns storage and retrieval from a single DBMS.

Considering the problem just described about, regarding temporal data management in GIS while keeping previous investments in legacy applications, it is one of the goals of the current work to prove the following statement:

> *Statement 1: the storage and retrieval of spatio-temporal information in a GIS can be provided through a conventional spatial DBMS, by extending it with temporal support for both valid time and transaction time and by preserving important features in temporal DBMS such as upward compatibility, temporal upward compatibility, sequenced semantics and non sequenced-semantics.*

Proving this statement may have a great impact in the layer responsible for data storage and retrieval in current GIS, where support for spatial data is already provided and spatio-temporal data management is increasingly demanded. Moreover, an approach to a DBMS that provides full spatio-temporal data management while keeping legacy applications working, without having to make changes, is highly desirable.

While the use of STDBMS empowers GIS with more appropriate tools for efficient spatio-temporal data management, it also opens new research possibilities in spatio-temporal information visualization, as exploratory data analysis can be performed over larger spatio-temporal datasets since they become more easily manageable and retrievable.

In the visualization of spatio-temporal datasets directly gathered from a temporal DBMS through a query statement, each object may be composed of a set of distinct temporally non-overlapping states that represent its temporal evolution. How to meaningfully display the set of states for each object constitutes a research challenge. In this work it is believed that the visualization of objects represented in a spatio-temporal data set may welcome user-defined parameters in order to display information according to user visualization goals. In other words, the user mental map regarding both the information and the visualization goals should somehow be reflected on the visualization process, in order to emphasize the part of information the user is interested, while also displaying the remaining information with less emphasis, but which constitute an information context. Such context can provide the additional clues to identify hidden patterns and correlations that are usually goals of exploratory data analysis.

In the current work, it is believed that information visualization techniques used to display abstract data, more specifically distorted view Focus+Context techniques, may provide appropriate answers for the previous problem. Hence, based on the previous considerations, it is one of the goals of the current work to prove the following statement:

> *Statement 2: Focus+Context visualization techniques provide appropriate solutions for the visualization of spatio-temporal information directly gathered from STDBMS in the context of exploratory data analysis.*

1.2 Research Objectives

Generally, problem-based applied research has been undertaken to prove *statement 1* and only after, with the help of the prototype system developed as a proof of concept, research was undertaken to prove *statement 2*.

More specifically, in order to prove *statement 1*, the following objectives have been defined:

1. To identify what is expected in a database management system supporting a GIS, regarding spatial and temporal support for data management;

2. To review the state of the art on the distinct approaches for providing spatial and temporal support in current DBMS technology;
3. To propose an approach for providing support for temporal data management based on current spatial DBMS used in GIS.
4. To develop a prototype system of a spatio-temporal database management system that is to be used as a proof of concept;
5. To test, validate and conclude about the strengths and the weaknesses of the proposed approach.

Furthermore, to prove *statement 2* the following objectives have been defined:

6. To review general-purpose information visualization possibilities;
7. To identify distinct approaches for the visualization of spatio-temporal information;
8. To propose an approach for the visualization of spatio-temporal information retrieved from a STDBMS;
9. To develop a prototype system of the researched visualization model to serve as a proof of concept;
10. To test, validate and conclude about the strengths and the weaknesses of the proposed model.

1.3 Innovative Contributions

In this dissertation an innovative approach is proposed to provide temporal support (*valid time* and *transaction time*) in GIS, based on settled concepts from the fields of temporal data management, spatial data management and Geographic Information Systems (Carvalho, Ribeiro et al. 2006).

A spatio-temporal query language has resulted from the researched approach, starting with ATSQL2 and extending it with spatial constructs such as spatial data types, spatial functions and operators, and spatial indexing. The effectiveness of the proposed query language has been demonstrated through the development and test of a database management system prototype named *Action!*. The correctness of the spatio-temporal support provided by the prototype has been tested on a small data set, built to illustrate the spatial-temporal features that become available.

It has been demonstrated that a GIS can rely on a STDBMS to manage spatial and temporal information while maintaining legacy application unharmed. In order to do so, the spatio-temporal database management must support important features such as the ability to provide upward compatibility regarding time and space. In other words to be able to deal with

applications that do not require spatio-temporal support at all or only require support for time or for space, independently. On the other hand new applications can be developed that take full benefit from the spatio-temporal data management provided.

As a side contribution for the research on spatio-temporal data management a Temporal Web Feature Service specification (Rocha and Carvalho 2006) has been proposed in the XML Application and Associated Technologies Conference, 2006. The aim of this work was to propose temporal extensions to the Filter Encoding Specification (OGC04-095 2005) by providing temporal support in Web Feature Service (WFS) requests, such as the ability to use sequenced semantics, non-sequenced semantics and temporal functions and operators such as temporal coalescing. Through this proposal, WFS servers could retrieve data from STDBMS.

Concerning the visualization of spatio-temporal information collected from a STDBMS, through queries, this work proposes an innovative approach named Temporal Focus+Context Visualization Model (Carvalho, Ribeiro et al. 2008) (Carvalho, Ribeiro et al. 2008). It is based on a Temporal Degree of Interest (TDOI) calculated for each record in the query result set. The TDOI input parameters are the record *valid time* attribute, as well as some user-defined requirements driven by application-specific or domain-specific goals concerning data presentation, exploration and analysis. In the mapping stage of the visualization pipeline, the degree of interest is used to control graphical properties, such as transparency and color, or to control more complex rendering attributes, such as sketch drawing for edges or other non-photorealistic enhancement techniques (Pomaska 2004).

By carefully defining user scenarios, large amounts of data from distinct periods of time and from several record sets can be compressed onto the same dynamic presentation as they are depicted with distinct graphical configurations. This visualization model has been tested with three historic-temporal data sets. The evaluation of results shows that the proposed approach is able to capture the relevant aspects of their domains.

1.4 OUTLINE

This dissertation is organized in six chapters besides the bibliography. After this introduction, Chapter 2 reviews the state of the art in spatial and temporal data management while Chapter 3 concerns the innovative contributions on spatio-temporal data management that have been researched in the current work. Next, Chapter 4 reviews the state of the art in information visualization, while Chapter 5 presents the innovative contributions on spatio-temporal information visualization directly retrieved from a STDBMS. Finally, Chapter 6 presents the most important conclusions from this work and uncovers perspectives of future work. By default, each chapter starts with an introduction and ends with the most important conclusions that can be retained from its contents.

After chapter 6, the document contains the bibliographic references and ends with the three annexes that are provided in an appending compact disk containing:

- Annex A contains the syntax of the proposed spatio-temporal SQL, resulting in the integration of ATSQL2 constructs and Oracle Spatial. It is expressed in the syntax formalism of ANother Tool for Language Recognition (ANTLR) and it has been used in the development of the prototype system;
- Annex B lists the sequence of tests that have been performed over the prototype system in order to validate the capability for spatio-temporal data management. Each test contains the following structure: the spatio-temporal SQL statement, the corresponding snapshot-equivalent set of SQL statements and the corresponding results retrieved DBMS core;
- Annex C contains a set of movies containing results on the application of the proposed Temporal Focus+Context Visualization model to two tests scenarios.

2 SPATIAL AND TEMPORAL DATABASE MANAGEMENT SYSTEMS

Until recently Spatial Database Management Systems have been providing appropriate means for the representation, management and retrieval of alphanumeric and spatial information in Geographic Information Systems (GIS) (Günter 1998). These systems are considered adequate for dealing with the most recent state of the facts managed by a GIS. However, important application domains where GIS are considered a key factor, such as urban data management, are increasing the demands for temporal support in GIS, in order to easily manage information about distinct temporal states.

Considering the relevance of the data management level in a GIS and the importance of GIS for a wide range of domain areas in modern society, new Database Management Systems have been developed aimed at providing appropriate means to represent, to store and to query new types of increasingly complex information.

This chapter addresses work done in the last three decades on spatial database management systems and in temporal database management systems. This is performed from the perspective of using these systems to support the tasks related to managing time-varying geographic information in a database management system.

The chapter is organized as follows: first several definitions of a Geographic Information System are provided. In order to do so, definitions and requirements of spatial DBMS are also provided. Next a definition and classification for spatial query languages is presented, along with query language proposals and implementations from the research and development communities. Furthermore, an overview of the main spatial DMBS is presented and conclusions are drawn regarding the support provided by spatial DBMS to GIS and their limitations in proper temporal data management. The second part of this chapter addresses temporal data management, starting with an overview of the results obtained by the research community on temporal databases. Before addressing temporal data models and query languages some key definitions regarding time are presented. Finally the most important implementations of temporal DBMS are surveyed. Where available, references are done to the support of spatial data by these DBMS. This chapter ends with the conclusions on temporal data management and spatio-temporal support for GIS.

2.1 GEOGRAPHIC INFORMATION SYSTEMS

Before moving on to the database management systems responsible for the representation, storage and retrieval of spatial data in a Geographic Information System a brief definition of a GIS and the role database management systems play in it is required.

GIS have been developed since the 1950's. Maguire *et al.* (Maguire, Goodchild et al. 1991) identify four distinct phases in the development of GIS until the 1990's, namely the pioneers phase, the scientific experimentation phase, the software developer phase and the user-centered phase.

Matos (Matos 2001) summarizes the evolution of GIS in five stages, starting in the 1960's and ending in 2000, namely:

- The pioneering stage is considered to be started with the Canadian Geographic Information System. By that time, the technology underlying GIS did not provide appropriate support for managing large data sets and, consequently, although potentially promising, the advantages of using GIS were not perceptible to the point of large-scale usage of these systems. However problems associated to the representation of geographic information started to emerge;
- The consolidation stage, from 1970 to 1980 is characterized by the awareness that geographic information modeling has a set of properties and characteristics that are common to a wide range of application domains establishing contact points in the solutions provided for these domains.
- The development stage, from 1980 to 1990, is characterized by the effort made by application vendors to promote the commercial use of GIS. According to Matos, actions performed in this context were frequently adequate and strategic but sometimes opportunistic and leading to some false expectations from the organizations who bought GIS technology, especially due to technological limitations and the lack of geographic information data sets by that time.
- The reconversion and data acquisition stage, from 1990 to 1995, is characterized by a predominant effort to overcome the limitations identified in the previous stage. The main efforts were made in enabling both the collection of large data sets and the overcome of technological limitations in dealing with these data sets. The users of GIS at this stage were predominantly public institutions and universities.
- The last stage, from 1995 to 2000, is a transition of GIS from the public institutions and universities to small companies, due to the lower cost of solutions and finally the usage of GIS by the common citizen, through the internet, taking advantage of location-based services. This last stage draws an important milestone in GIS as the revenue of using geographic information largely justifies the cost of data acquisition. Moreover this stage has given birth to the upcoming of Geomatics – the discipline of gathering, storing,

processing, and delivery of geographic information, or spatially referenced information - based on the construction of a coherent theoretical formalism.

In the last stage mentioned above GIS moved on to a distributed environment configuration, based on multi-layer service-oriented architectures, working over the internet and characterized by interoperable systems that are able to manage spatial data from heterogeneous data sources.

Distributed GIS is mainly the result of the work undertaken by the Open Geospatial Consortium (OGC) in defining new standards for the interconnection and interoperability of distinct GIS, aimed at providing location-based services, operating over the internet, through a common set of interfaces accepted by most GIS researchers, developers and vendors. In other words, the aim of OGC has been to specify ways for applications to store and access geographic information from multiple heterogeneous data sources, so that the data can be used to support diverse applications through a common feature model, data store and information access interface.

Among other contributions, the OpenGIS and later the OGC have been making critical contributions by defining specifications such as the OpenGIS Simple Features Specifications For SQL (OpenGIS 1999), version 1.1 in 1999, Geographic Markup Language, version 1.0 in 2000 (OGC 2000), the Web Map Service, version 1.0 also in 2000 (OGC 2000) and the Web Feature Service version 1.0 in 2002 (OGC 2002), among other standards and recommendations. Over the years this consortium has developed more standards, based on the increasing demands of the GIS industry, user and research community, such as the CityGML (OGC 2008), aimed at encoding the representation, storage and exchange of virtual 3D city and landscape and implemented as an application schema of the Geography Markup Language version 3.1.1 (GML3).

One of the consequences of the heterogeneous stages through which GIS has passed over the last decades is that its definition has also suffered considerable changes over this time. Moreover, the definition of GIS depends on the perspective through which GIS is being defined. For instance, Cowen (Cowen 1988) states that there are four distinct perspectives to the definition of GIS, namely a process-oriented and function-oriented perspective, an application- or domain-oriented perspective, a toolset-oriented perspective and a database-oriented perspective.

Earlier definitions, such as the one suggested by Dueker (Dueker 1979) regard GIS as a special case of information systems where the database consists of observations on spatially distributed features. This author also states that the purpose of GIS is to manipulate data in order to retrieve other data for *ad hoc* queries and analysis. Later Ozemoy *et al.* (V.M.Ozemoy, Smith et al. 1981), defines GIS as an automated set of functions that provides professionals with advanced capabilities for the storage, retrieval, manipulation and display of geographically located data. More recently, GIS has been regarded as a database system in

which most of the data is spatially indexed and upon which a set of procedures are operated in order to answer queries about spatial entities stored in the database. Carter (J.R.Carter 1989) focuses on the institutional perspective by defining a GIS as an institutional entity reflecting an organizational structure that integrates technology with a database, expertise and continuing financial support over time. The Environmental Systems Research Institute (ESRI) defines GIS as a system that integrates hardware, software, and data for capturing, managing, analyzing, and displaying all forms of geographically referenced information. For ESRI, a GIS can be viewed in three ways: the database view, the map view and the model view. Oliveira (Oliveira 2007) defines a GIS as an active object or set of interrelated active objects that processes representations of geo-referenced entities, activities and phenomena through the use of a spatial coordinate system.

In an attempt to gather the up-to-date ideas embedded on the previous definitions, it is possible to state that a GIS encompasses expertise encoded in a set of computational tools aimed at providing appropriate support for the management of spatially dependent information, including capture, synthesis and acquisition, validation, storage and retrieval, presentation and exploratory data analysis. Across the decades, this set of computational tools has taken distinct forms, from monolithic applications working with proprietary file formats to most current distributed service-oriented architectures and interoperable data formats. The aim of a GIS is the overall management of spatial information and the appropriate support for exploratory data analysis and decision making. These are the primary reasons for the success of using GIS in a wide range of application domains, such as government, transport management, facility and infrastructures management, critical structure assessment, urban management, environmental management, insurance industry, telecommunications/location-based services.

Considering the technological aspects of data storage and management in a GIS, the database management system plays a critical role, enabling the storage and handling of spatial data. However, GIS have not always relied on database management systems for their data repository.

2.2 Spatial Database Management Systems

Until the middle 1980s GIS used to be monolithic systems, taking care of storage, query and visualization of geographic information in a tightly integrated manner (Günter 1998). By that time GIS technology suffered from severe problems at the conceptual level, namely:

- The lack of proper data management and data retrieval support, only available through application programming interfaces external to the GIS core and provided by the GIS vendors;

- Persistency problems regarding the representation of the structure of information: much of the information regarding the data structure was built-in at the application level and not represented at the database level;
- Management of concurrent access to GIS resources: due to the complexity of implementation of access management to resources, security schemas, transaction support, most of the GIS technology available in the 1980s was confined to single user access;
- Data retrieval and fault tolerance: GIS technology was not prepared to overcome system failures, such as power loss. Consequently information could become corrupted, and therefore unusable due to data inconsistencies;
- Most of the GIS products did not offer support for the preservation of semantic integrity, since the underlying available database management systems hadn't support for triggers or similar techniques.

In order to efficiently overcome the above limitations, GIS technology started to adopt database management systems, previously used in application domains considered "more traditional" which only needed the management of alphanumeric data. Therefore, by the end of the 1980s, new GIS technology emerged which totally relied on spatial data management systems through commercially available database technology. However, as stated by Günter (Günter 1998), in order for a standard DBMS to be able to efficiently manage spatial data, some limitations need to be overcome regarding the spatial nature of data and the increasing requirements of having to deal with more relations, such as larger records and longer transactions. The task of integrating spatial support in alphanumeric DBMS was considered a complex task due to the nature of spatial information, namely:

- Complex structure, especially regarding the geometrical representation;
- Expected dynamic behaviour of spatial objects;
- Increase in computational requirements to account for spatial operators.

In short, when used in the context of GIS, traditional DMBS lacked for spatial data types, a spatial query language, spatial operators and functions and multi-dimensional access methods for the efficient construction of spatial indexes. Only after these limitations were overcome it became possible to designate a DBMS as being a spatial DBMS (Günter 1998).

Hence, for Gütting, a spatial DBMS requires the following extensions (Gütting 1994):

- Representations for the spatial data types of a spatial algebra;
- Spatial data types and operations supporting a data definition and query language;
- A set of efficient algorithms for spatial operators, such as the spatial join;
- Spatial index structures and access operations for spatial indexes.

Gütting considers that the only clean way to accommodate these extensions is through an integrated architecture based on the use of an extensible DBMS.

2.2.1 Spatial Query Languages

Until 1994 there was not a unified widespread adopted spatial algebra and query language (Gütting 1994). At that time a unified, consensual, canonical set of spatial operators had not yet been achieved, although the sets provided by distinct technological solutions were very similar. Currently, the scenario on available spatial query languages is similar, but the set of spatial data types and operators has been established. However, the implementations of spatial data definition and spatial data manipulation are distinct among the available spatial DBMS.

Aufaure-Portier and Trépied (Aufaure-Portier and Trépied 1996) present a survey on spatial query languages for GIS that identifies five categories of query languages, namely:

- Natural languages, such as KID (J.Chae and S.Lee 1995);
- Extensions of SQL, such as PostGIS (PostGIS 2008), GéoSabrina (Larue, Pastre et al. 1993), Spatial SQL (Egenhofer 1994), GISQL (Costagliola, Tortora et al. 1995);
- Tabular languages, such as QBE (Zloof 1977), OOQBE (F.Staes, L.Tarantino et al. 1991), PICQUERY+ (F.A.Cardenas, I.T.Ieong et al. 1995);
- Graphical languages, such as OHQL (E.Andonoff, C.Mendibourne et al. 1993) and SNAP (Bryce and Hull 1986);
- Visual languages such as CIGALES (Aufaure-Portier 1992),

Aufaure-Portier and Trépied also point out advantages and disadvantages of each of these categories and conclude that the ideal approach of a spatial query language is a hybrid interface harmoniously combining the visual language approach, he natural language approach and the hypermedia approach, since:

- Natural languages appear as an attractive approach for end-user but also presents some important drawbacks (e.g., verbosity and ambiguities) and consequently, can only be used as a complement to another approach;
- Extensions of SQL clearly can only be used by people having some knowledge in database theory. The authors also state that the extended spatial SQL approach seems to be the best formal support to retrieve spatial data but that they may be hidden into a graphical or visual language for the end-user;
- Hypermedia approaches to SQL allow a user-friendly browsing of the information base but cannot be considered as a query language and must also be combined with another approach;

- Visual languages seem to be the more promising approach but still have limitations such as the absence of formalisation, semantic ambiguities and limited expressive power.

In fact, the evolution of query languages demonstrated that the most widely used approach has been developed by extending SQL, such as Oracle Spatial (Oracle Corporation 2001) and PostGIS (PostGIS 2008), where both extend SQL3 (ANSI.X3H2-91-183 1991), through the use of abstract data types (Gütting and Schneider 1995).

On the other hand, the upcoming of distributed GIS has also provided for the working grounds of research towards a spatial query languages aimed at distributed environments and dealing with heterogeneous data sources, such as extending Xquery to GML documents (Jihong 2005).

2.2.2 Spatial DBMS

Spatial database management systems supporting GIS followed two approaches, from an architectural point of view, for extending database systems with support for high-level spatial data definition and spatial querying: the layered architecture and the dual architecture (Gütting 1994). In the first approach, the spatial support is developed on top of an extensible DBMS. Early systems following this approach represent spatial data types through a structure where complex geometric objects are broken into more elementary shapes until lines and points are reached. Other systems use a layered architecture which represents spatial data through the use of long/blob fields in conventional DBMS.

On the other hand, dual architecture approaches integrate two rather independent subsystems: the DBMS, which handles non-spatial data, and a spatial subsystem, which stores and manipulates the representation of spatial objects. This approach has been followed in the 1990's by most commercial GIS (e.g., ARC/INFO, SICAD) as well as by some research prototypes. The main advantage is that dual architectures promise higher degrees of freedom in defining spatial data types, structures and algorithms.

However, as datasets became larger and more complex, the main disadvantage of dual architecture spatial DBMS became clear: the inefficiency of query processing, due to the lack of optimization that results from having to query two systems independently.

More recently and due to the disadvantages of the dual architecture approach, spatial DBMS have returned to the layered architecture approach, taking advantage of SQL3 (ANSI.X3H2-91-183 1991) features, such as the use of abstract data types. Examples of these implementations are:

- Oracle Spatial (Oracle Corporation 2001), a commercial spatial extension to Oracle DBMS, consisting of an SQL schema with operators and functions aimed at storage, retrieval, update and query of spatial features in an Oracle database. Oracle spatial supports a spatial indexing system, procedures for area-of-interest,

spatial join queries and spatial analysis operations, a topological data model for working with nodes, edges and faces in a topology, a network data model and support for the management of georaster data. The free version of Oracle (Express edition), targeting academic groups and small companies, also supports Oracle Spatial;

- PostGIS (PostGIS 2008), the open source FLOSS licensed spatial extension for PostgreSQL (Rowe and Stonebraker 1987) with spatial support. This extension provides OGC compliance, over 300 spatial functions and operators, no geodetic support, support for 2D data and limited support for 3D data and GIST (r-tree variant) spatial indexing. It is considered one of the best spatial DBMS available;

- MySQL spatial extensions (Karlsson 2004), currently in version 2.0, an open source COSS licensed spatial extension for the MySQL 4.1 DBMS which is OGC compliant and support a subset of OGC Simple Features Specifications for SQL. MySQL implements the data type geometry plus some spatial functions that haven't been implemented according to the OGC specifications. Functions that test spatial relationships, mostly related to spatial analysis, are limited to work with minimal bounding rectangles rather than the actual geometries. MySQL versions earlier than 5.0.16 only supported spatial data in MyISAM tables. As of MySQL 5.0.16, InnoDB, NDB, BDB, and ARCHIVE also support spatial features;

- Microsoft SQL Server 2008 DBMS (Microsoft 2008). This DBMS provides OGC compliance and over 70 spatial functions that are a subset of OGC Simple Features Specifications for SQL, geodetic support, support for 2D and a spatial indexing mechanism based on B-trees;

- IBM DB2 Spatial extender (IBM 2008), implements the OGC Simple Features for SQL with Types and Functions Specification and the ISO SQL/MM Part 3 Spatial standard. DB2 provides data types so that spatial data can be represented as geometric shapes (such as points, lines and polygons), SQL access to spatial data for loading, retrieving and indexing.

Most of these DBMS implementations provide the means for application development and spatial data visualization through open source projects such as GeoTools (GEOTOOLS 2008), GeoServer (GeoServer 2001), GRASS (GRASS 1982) and uDIG (Refractions Research Inc. 2001) or through commercial packages such as ESRI ArcGIS (ESRI 2004).

However, other spatial DBMS exist that still follow the extensible approach architecture, such as DEDALE (Grumbach, Rigaux et al. 1998; Grumbach, Rigaux et al. 2000) which is one of the first implementations of a database system based on the linear constraint model (a paradigm for the representation of all sorts of data in a unified framework) (Revesz 1998; Grumbach, Koubarakis et al. 2003). Key features of this prototype system are:

- The ability to represent multi-dimensional point sets as linear constraint relations;
- An extension to handle interpolated spatial data (S. Grumbach 2000);
- The support of geometric objects embedded in a *d*-dimensional space where one of the attributes can be defined as a function or a subset of the other existing attributes. Moreover this function can result from a linear interpolation based upon a finite set of sample values (Grumbach, Rigaux et al. 2000);
- The support for moving objects, based on the previous feature, where a trajectory can be represented by a sample of points with time and location.

Moreover, in the past few years, there was an increase in demand for temporal data management for most of the application domains already managing spatial data. The lack of proper temporal support provided by spatial database management has been motivating approaches between the geographic information research community and the temporal data management research community.

2.3 Temporal Database Management Systems

Databases provide the means for the storage and the retrieval of data about properties of objects and relations among objects, concerning relevant statements or facts from the real world (or a modelled reality, also designated by a mini-world (Dyreson, Grandi et al. 1994)) and from the perspective of a particular domain. Intrinsic to the occurrence of facts, natural or human made phenomena, is the element of time.

According to Jensen (Jensen 2000), most of the applications domains where database technology plays an important role require proper temporal data management, as the information being managed in these domains encompasses a strong temporal dependency. Examples of these applications are financial applications (portfolio management, budgeting, billing, accounting, and banking); record-keeping applications (personnel, medical-records, insurance policies, and inventory); travel applications (airline, train, and hotel reservations); project management applications (scheduling); and scientific applications (trend analysis). These areas manage substantial quantities of time-varying data (Böhlen, Jensen et al. 2000).

For these applications, until recently, and according Böhlen *et al.*, the management of time-varying data has mostly relied on the application level because conventional (snapshot) DBMS did not provide the proper support for managing time-varying data. The lack of this support in DBMS can be justified by several reasons, namely:

- For most of the application domains semi-temporal and non-temporal approaches were considered satisfactory. Concerning the first approach, the temporal data management relies on non-temporal DBMS, where time is represented through primitive data types, such as date or time, but queries involving temporal constraints are complex to formulate (Zaniolo, Ceri et al. 1997). The second approach, also relying on non-temporal DBMS,

considers only the most recent state of facts, thus allowing the management of data by providing a snapshot (Dyreson, Grandi et al. 1994) of the present time. In this approach and regarding an object, information about previous facts is replaced by more recent information when available. If the object ceases to exist, its corresponding information is simply deleted;

- The initial lack of consensus on a reference data model and query language, in the research community, and the disagreement inside the ISO committee about the temporal extensions to SQL3.

Current applications, dealing with time-varying data, demand that appropriate storage and retrieval of temporal data should be provided by the database layer. Despite the fact that there is no standard regarding a temporal extension to SQL, the fact is that, according to Jensen and Snodgrass (Jensen and Snodgrass 1999) three decades of active research on temporal databases have provided solid, consistent research and some implementations that are able to provide the desired temporal support. For these authors, work performed since the 1980s resulted in approximately 2000 papers listed in a series of seven cumulative bibliographies (Wu, Jajodia et al. 1998), a glossary of terminology (Dyreson, Grandi et al. 1994), a book surveying the most relevant issues in temporal databases (Tansel, Clifford et al. 1994) and workshop proceedings. The SQL/temporal extension to SQL3 was also proposed (Jensen and Snodgrass 1999). This new part was accepted at the Ottawa meeting in January, 1995 as Part 7 of the SQL3 specification (Snodgrass 2006).

Snodgrass *et al.* (Snodgrass 2006) provide a brief description regarding the development of SQL/Temporal starting with the *Road Map of Additions to SQL/Temporal*, presented to the international committee (ISO/IEC JTC 1/SC 21/WG 3 DBL) where the authors provide answers to important questions, namely:

- Why temporal?
- Is SQL (SQL-92, SQL3) adequate?
- Why TSQL2?
- Why not explicit time stamps?
- Wouldn't a different language design also work?
- What about the new features introduced in SQL3?
- Doesn't this break the relational model?

By that time, the authors concluded that temporal databases required a new way of thinking but a solution was already designed by a team of temporal database experts. The comprehensive and consistent solution proposed already had a formal semantics, a detailed discussion of the design decisions and a prototype implementation: TSQL2. Afterwards, discussions started on adding valid time and transaction time support to SQL/Temporal,

resulting in 1996 in two change proposals, ANSI-96-501, on Adding Valid Time to SQL/Temporal, and ANSI-96-502, on Adding Transaction Time to SQL/Temporal, which were unanimously accepted by ANSI and forwarded to ISO committee. However, due to disagreements within the ISO committee as to where temporal support in SQL should go, the project responsible for temporal support was cancelled near the end of 2001, halting the efforts related to the working draft, "Part 7, SQL/Temporal" until the present time.

Regarding interactions with the research community on spatial database systems, it can be said that since 1990 the two communities started to work more integrated. According to Koubarakis et al. (Koubarakis, Sellis et al. 2003), researchers from both areas had felt that there were important connections in the problems addressed by each area, as well as techniques and tools used for their solution. However, according to these authors, with the exception of pioneering papers, little systematic interaction and synergy between these areas had occurred. Two of the most important interactions happened in the 1990's:

- Starting in 1994, the American initiative designated "Time in Geographic Space", undertaken by the National Center for Geographic Information;
- Starting in 1996, the CHOROCHRONOS project, funded by the European Commission DG XII Science, Research and Development contract FMRX-CT96-0056.

For both projects, research goals concentrated on:

- Ontology;
- Structure and representation of time and space;
- Models and languages for STDBMS;
- Graphical user interfaces for spatio-temporal information;
- Storage structures;
- Indexing techniques and query processing algorithms for spatio-temporal databases;
- Architectures for spatio-temporal databases.

The final goal concerned the application of STDBMS in realistic problems, targeting traditional applications such as GIS, but also more advanced ones where the connection to a standard database approach was not obvious.

2.4 Representation of Time in Databases

The nature of time has been researched from multiple points of view, for instance, philosophical, mathematical, physical, psychological, where interesting questions have been formulated (Date 2000), such as:

- The philosophical study of the dichotomy and duality about events and states;

- The scientific study of the representations of time;
- The psychological study that clarifies the moment of time designated by *now*.

Regarding the first question, an event is considered an instantaneous fact that occurs at a particular instant (time point on an underlying time axis) which instantiates an object or a set of properties of an object. Furthermore, a state is considered a set of instants for which a set of properties from an object are kept unaltered. Hence, it is not possible to represent states without the occurrence of events and the occurrence of an event implies the existence of states.

The research on the nature of time has also provided distinct understandings of how to model and, consequently, of how to represent it in a database (Zaniolo, Ceri et al. 1997; Date 2000; Jensen 2000). Zaniolo *et al.* (Zaniolo, Ceri et al. 1997) provide an overview of distinct time models for characterizing the time domain. For these authors, the models of time in a temporal logic represent time as an arbitrary set of instants with an imposed pair order, to which axioms can be introduced in order to provide more refined models. Examples are the linear model, branching model and cyclic model (Zaniolo, Ceri et al. 1997). Other axioms can also be placed about the density of the time line (continuous time model, discrete time model and dense time model), or about the granularity of time by defining the smallest non-decomposable unit of time (designated by *chronon* or *time quanta*[1]).

Although required, the characterization of the domain of time values is not enough to represent the temporal information of facts. Appropriate data types are needed for representing:

- An *instant*, or in other words, a time point on an underlying time axis;
- A *time period* or *time interval*: the time between two instants. In a system that models a time domain using *chronons* a time period may be represented by a set of contiguous *chronons*;
- A *time element*: a finite union of periods (Dyreson, Grandi et al. 1994; Zaniolo, Ceri et al. 1997).

Furthermore, the instantiation of *intervals*, *periods* and *instants* often requires the use of special time values retrieved from current-time variables, such as *now*. This variable, also designated as *until changed*, is a database time variable that has the meaning of "at the present time" (Clifford, Dyreson et al. 1997). Clifford *et al.* provide a clear semantics for the variable *now* as well as other current-time variables such as *now-relative*, and *now-relative indeterminate*. Their overall conclusion is the recommendation that time stamps involving

[1] Date points out that there is no consensus in the nomenclature for defining the unit of time Date, C. (2000). Temporal Databases (Chaper 22). An Introduction to Database Systems, Addison-Wesley.

current-time variables should be allowed to be stored as values of columns, for conventional and temporal databases, as well as implicit *valid* and *transaction timestamps*, for temporal databases. Moreover, other special time values should be supported such as *forever*. This value represents a special *valid time* instant that is the largest valid *chronon* supported in the *valid time* line. According to Dyreson *et al.* (Dyreson, Grandi et al. 1994) *forever* has no *transaction time* semantics and it is used as the upper boundary of time periods that represent a state which will cease to exist in an indeterminate instant in the future.

According to Tryfona *et al.* (Tryfona, Price et al. 2003) the temporal representation of facts of objects can be performed from three distinct temporal aspects: the *valid time*, the *transaction time* and *the existence time*. The *valid time* of a fact is the time when the fact is considered to be true in the modelled reality, which can be in the past, present or the future (Dyreson, Grandi et al. 1994; Date 2000; Tryfona, Price et al. 2003). The *transaction time* of a fact is the time when the fact is part of the current state in the database or, in other words the time when a fact has been represented as being true in the database (Dyreson, Grandi et al. 1994; Date 2000; Jensen 2000). It is important to state that the *transaction time* aspect of a fact may not be required on a particular application domain, since this aspect is mostly used for traceability and accountability purposes. According to Tryfona *et al.*, the third aspect - the *existence time* - does not directly apply to facts but to objects and can be formulated in terms of the *valid time*. It refers to the time when an object exists and in this way be a union of the *valid time* of facts regarding an object.

The management of each of these aspects in databases has been proposed by Snodgrass et al. on the report "*support of valid time in SQL/Temporal*" (Snodgrass, Böhlen et al. 1996) and on the report "*support of transaction time in SQL/Temporal*" (Snodgrass, Böhlen et al. 1996). *Valid time* and *transaction time* spaces may be used independently, alternatively or altogether in the temporal characterization of the facts for an object since they are orthogonal.

2.4.1 Data Models for Representing Time-varying Data

The concepts such as the domain of time, time data types and the *valid time* and *transaction time* aspects are required to temporally characterize facts. However, a temporal data model and language is required at the database level in order to add appropriate time support to time-varying structured data. According to Jensen et al. (Jensen, Soo et al. 1994) adding time to the relational model has been a daunting task. To these authors the ideal temporal data model encompasses too many requirements, some of which are mutually incompatible, and furthermore, some of the desired temporal characterization lacks efficient implementation. Such difficulties can be seen in the number of distinct language designs and underlying data models, over 40, published in the literature (Snodgrass 1996). In an attempt to unify temporal data models, Snodgrass *et al.* proposed the Bitemporal Conceptual Data Model (BCDM) to capture the essential semantics of time-varying relations. However, this

model has no particular concern with efficiency regarding presentation, storage and retrieval. For instance, in the BCDM, a time stamp associated with a tuple may encompass an arbitrary set of bitemporal chronons. As such, a relation represented in this conceptual model is non-First Normal Form (1NF) which, according to the authors, represents difficulties if directly implemented.

Jensen *et al.* (Jensen, Soo et al. 1994) rely on existing representational data models for presentation, storage and retrieval of temporal data. The representational data model concerns the appropriate representation of temporal data on the relational model by following an approach that either time stamps tuples or time stamps attributes or uses a combination of both.

Jensen *et al.* also suggest exploiting the equivalence mappings between the BCDM and the representational models, based on Gadia's concept of *informational content* (Gadia 1986). Here, if two temporal databases implementing distinct temporal data models capture the same reality, then they should contain the same *informational content*. It can be said that two temporal relation instances have the same *informational content* if they have *snapshot equivalence*. According to Gadia, *weakly equal* relations are every two distinct temporal relation instances, with identical set of non-temporal attributes, where for each distinct used time value (in *valid time* and in *transaction time*) their *snapshots* are identical.

Based on *informational content* and *snapshot equivalence*, it is possible to establish mappings between instances of distinct representational temporal data models. Hence, mappings between five representational bitemporal data models have already demonstrated (Jensen, Soo et al. 1994; Zaniolo, Ceri et al. 1997), namely:

- Ben-Zvi's Tuple Timestamped Representation Scheme (Ben-Zvi 1982) where five time stamps temporally characterize each tuple;
- Jensen's Back-log Based Representation Scheme (Jensen, Mark et al. 1999) where each tuple is time stamped with a single transaction *chronon*);
- McKenzie's Attribute Value Timestamped Representation Scheme (Leslie Edwin Mckenzie 1988) where states are time stamped with *transaction time* and attributes are time stamped with *valid time*;
- Snodgrass's Tuple Timestamped representation scheme (Snodgrass 1987; Snodgrass 1993) where each tuple is time stamped with *valid time* and *transaction time*;
- Bhargava's Attribute Value Timestamped Representation Scheme (Bhargava and Gadia 1989) where each attribute in a tuple is time stamped with *valid time* and with *transaction time*;

However, in order to take full benefit from temporal data models, the query language provided by a DBMS should support temporal definition and temporal modification statements

and should also comply with other temporal data management requirements. In the following the most relevant temporal query languages extending SQL will be presented. Before that, it is important to refer that an important part of the research on temporal data models has moved on into time varying semi-structured data, mostly addressing issues about time-varying XML documents and schemas (Amagasa, Yoshikawa et al. 2000; Vimercati 2002; Snodgrass, Dyreson et al. 2008).

2.5 TEMPORAL EXTENSIONS TO SQL

According to Böhlen *et al.* (Böhlen, Jensen et al. 2000) few precise query language requirements have been proposed by other authors, although widely found in the research literature. The most relevant of these requirements are to provide temporal functions and operators (such as *temporal join*), sequenced semantics and non-sequenced semantics, and also to provide:

- *Temporal upward compatibility* (TUC), which allows the use of temporal relations in a database that serves new applications without having harmful impact on existing legacy code. Here, legacy applications should be able to deal with temporal relations and the snapshot SQL statements should work exclusively with the current state of facts. According to Böhlen et al. the objective is to make it possible to gradually develop temporally-enabled applications without invalidating existing applications;
- *S-reducibility*, which implies that for all non-temporal queries *q*, a temporal query *q'* exists such that at each snapshot, the result of the temporal query reduces to the result of the original query. In other words, a relation with a temporal dimension can be perceived as a sequence of non-temporal relations to which regular SQL statements apply. Hence, a regular SQL statement can be applied to each snapshot state of a temporal relation and the results can be combined into a single temporal relation;
- *Interval preservation* and *non-restrictiveness* which aim at the preservation of intervals as much as possible, and permit the manipulation of interval-valued time stamps using built-in predicates and functions;

For these authors, most of the temporal query languages proposed in the literature consist of extensions to SQL.

TQuel is a temporal query language proposed by Snodgrass (Snodgrass 1987; Snodgrass 1993) which supersets the Quel Query Language (by that time counterpart to SQL) for the relational database system Ingress, and which was designed to be a minimal extension, both syntactically and semantically of Quel by incorporating three additional temporal constructs. TQuel is upward compatible to Quel and the three temporal constructs provided in Quel have direct semantic analogues in Quel.

TOSQL (Gad 1986) extends SQL with the specification of a query's time aspects. The extensions include the clauses AT, WHILE, DURING, BEFORE, AFTER, ALONG, and AS_OF. The default options are defined syntactically such that a query that omits the temporal portion retains the standard meaning of the corresponding SQL select operation. TUC, while not defined explicitly, was clearly a concern when designing TOSQL. A large part of TOSQL respects TUC, but statements of the form *select p from r* violate TUC because they also return the relation's time stamp(s). S-reducibility is not supported. The built-in time-related processing is restricted to the above clauses and can be overwritten by stating the clauses explicitly, which makes TOSQL non-restrictive and interval-preserving.

TSQL (Navathe and Ahmed 1989) is a superset of SQL, extending the latter with, for example, WHEN, TIME-SLICE, and MOVING WINDOW clauses. TSQL satisfies neither TUC (no adequate defaults for the above-mentioned clauses) nor S-reducibility. The restriction to coalesced (Böhlen, Snodgrass et al. 1996) relation instances also breaks interval preservation. Apart from enforced coalescing, TSQL is non-restrictive.

HSQL (Sarda 1990) is again a superset of SQL. The retrieval facilities are enhanced with support for coalescing (COALESCES, COALESCE ON), time-slicing (FROMTIME t1 TOTIME t2), and unfolding (EXPAND). The concurrent product provides built-in snapshot-reducible semantics for joins (and products), but not for sub queries, aggregates, set difference, and integrity constraints, for instance. Like TOSQL, the design of HSQL takes TUC into consideration. HSQL is interval-preserving and non-restrictive and it does not support S-reducibility.

SQL-92 (Melton and Simon 1993) provides only quite limited support for temporal data. SQL-92 is not temporal upward-compatible with itself (legacy statements are not restricted to the current time). The S-reducibility property is not satisfied (no built-in processing of temporal joins, for example). Because SQL-92 does not treat time with special semantics, it is trivially interval-preserving and non-restrictive.

TempSQL (Bhargava and Gadia 1993) is an extension of SQL defined over relations where attribute values are temporal assignments, that is, partial functions from time into some value domain. Temporal expressions can be used in (nested) expressions. The SELECT-FROM-WHERE statement is extended with a WHILE clause that may be used for specifying the time domain of a tuple. TempSQL is restrictive in the sense that set operations have an enforced, built-in temporal semantics. TempSQL provides automatic coalescing, violating interval preservation.

IXSQL (Lorentzos and Mitsopoulos 1997) provides support for generic interval data in SQL. It extends SQL-92 with a REFORMAT AS and a NORMALIZE ON clause. The reformat clause is used to specify a sequence of UNFOLD and FOLD operations, which convert a set of intervals into a set of constituent points, and vice versa. The NORMALIZE operation is a syntactic abbreviation of the reformat clause, and it coalesces a relation. For the same reason

as for SQL-92, it does not satisfy TUC. No support for S-reducibility is provided. IXSQL is non-restrictive and, to some degree, it is also interval-preserving (intervals are not preserved by UNFOLD because interval boundaries are lost when unfolding a relation).

TSQL2 (Snodgrass, Ahn et al. 1994) was designed by a commission of eighteen researchers, that already took part in the previous proposal of one or more temporal data models. Their goal was to consolidate the data models and the query languages previously proposed and try to bring consensus to the temporal extension of SQL-92 by determining which data model would be best suited for future research. Özsoyoğlu and Snodgrass (Özsoyoğlu and Snodgrass 1995) consider TSQL2 to be a second generation temporal query language designed by first generation query language researchers. Regarding the domain of time values, TSQL2 provides a finite linear timeline that can be assumed discrete, continuous or dense behaviours for the representation of time values (Zaniolo, Ceri et al. 1997). The chronon granularity is variable with the possibility to group consecutive chronons into granules and providing the proper support for operations on temporal data with distinct granules. Conceptually, the subjacent data model used in TSQL2 is the BCDM, but at the implementation level other models are used. TSQL2 employs syntactic defaults. It adds a VALID clause to SQL-92 for specifying the time stamp of the result. If the VALID clause is omitted from a query, intersection semantics is assumed. By default, TSQL2 returns *valid time* relations. To retrieve a non-temporal relation, SELECT SNAPSHOT has to be specified. TSQL2 neither satisfies TUC (*valid time* relations are returned by default) nor S-reducibility (e.g., sub queries violate this (M. Böhlen 1995)) an also is not interval preserving because coalesced instances are enforced. While some operations come with built-in implicit temporal semantics (e.g., set operations applied to temporal relations) TSQL2 is non-restrictive because implicit time stamps can be rendered explicit.

The proposals of SQL/Temporal to SQL3 (Snodgrass, Böhlen et al. 1996; Snodgrass, Böhlen et al. 1996; Snodgrass 1998), submitted to the SQL standardization committee describe early work on temporal statement modifiers which were designed to fulfil TUC and S-reducibility. Interval preservation is not guaranteed because time stamps of snapshot-reducible queries are left unspecified in the nondeterministic definition of the language. Non-restrictiveness is achieved via non-sequenced statements. Extended S-reducibility is not satisfied because sequenced modifiers may only be applied to statements without explicit references to time dimensions.

2.6 Architectural Approaches to the Implementation of Temporal DBMS

Architectural approaches for implementing temporal database management systems fall into three categories (Breunig, Türker et al. 2003; Koubarakis, Sellis et al. 2003): the *layered* architecture approach, the *monolithic* architecture approach and the *extensible* architecture

approach. Each of these architectures contains advantages and disadvantages comparing with the others which are, in short:

- *Layered* architectures implement advanced features and services with specific requirements in a layer that resides over an underlying off-the-shelf DBMS. Applications which require these advanced features and services communicate with the DBMS through the layer interface. Authors find that the main advantage of this architecture is to provide very fast development of new features. On the other hand, they consider that, due to a clear separation of the layer from the DBMS core, transaction management and query optimization are only guaranteed for the DBMS level (K. Torp 1997).
- *Monolithic* architectures overcome these problems, since every specific feature and service is provided directly by the DBMS kernel. Yet, this architecture carries a daunting task which is the implementation process (Koubarakis, Sellis et al. 2003).
- *Extensible* architectures try to combine the best from the two previous architectures: the support for application-specific or domain-specific features is placed within the DBMS but outside of its core. Generally, this is only possible if the DBMS supports the development of extensions. Extensible architectures can benefit from the support for transaction management, indexing, and query optimization. Moreover, the development of an extension is less complex than the development of new functions inside the DBMS kernel.

For each of these architectures there are several implementations. However, one might suggest that general-purpose DBMS will soon start providing temporal data management by following the extensible approach. This statement can be fundamented in the evolution of how general-purpose DBMS have been providing support for spatial data. Despite previous architectural approaches, current DBMS vendors are following the extensible approach, such as Microsoft SQL Server (Microsoft 2008) or Oracle (Oracle Corporation 2008), among others.

2.7 Temporal and Spatio-temporal DBMS Implementations

Vassilakis *et al.* (Vassilakis, Georgiadis et al. 1996) present several design criteria in adding temporal functionality to a snapshot database (Snodgrass 1987), such as complete temporal functionality, full snapshot DBMS compatibility, level of implementability, data integrity, degree of performance, resource consumption and portability across different platforms. These authors also present, evaluate and point out several DBMS that have been implemented according to the three architectural approaches presented on the previous section. The survey in (Böhlen 1995), on temporal database systems, classifies several existing systems – ARCADIA, Calanda, ChronoLog, HDBMS, TDBMS, TempCASE, TempIS, TimeDB, TimeIT, TimeMultiCal, and others - grouping them in families according to carefully thought criteria. The temporal support described includes the temporal operations available,

the temporal statements allowed, the database technology, the time stamp format, the temporal dimensions supported, among others. This author points out important conclusions on the tested implementations, like the dominance of the relational model with time stamped tuples and the fact that *valid time* dimension has been the focus of attention, leaving *transaction time* to a second place. In the remaining of this section, and complementary to the list addressed by Böhlen, other temporal database management systems are presented and characterized, such as, CONCERT, DEDALE, SECONDO, TIGER, TEMPOS and TRIPOD[2].

CONCERT (Blott, Relly et al. 1996; Relly and Röhm 1999) follows the extensible system architecture and consists of an object manager placed on the top of a database system, exporting advanced query functionality outside the database. According to Breunig et al. (Breunig, Türker et al. 2003) CONCERT propagates a new paradigm called "exporting database functionality" whereas opposed to the data being loaded and owned by the database. Here, data may reside outside the database in external repositories or archives. From the database system perspective, data is viewed as abstract objects with only a few central object methods known. This paradigm is considered extremely useful for spatio-temporal environments as there is no single most useful format to represent spatio-temporal data. Data indexing in CONCERT is based on the conceptual behaviour of data. For this purpose, four behaviour classes are provided: SCALAR, RECORD, LIST and SPATIAL. To index data, CONCERT provides B-Trees for ordered data and R-Trees for spatial data. In this way the provided classes cover all relevant concepts that are the basis of current indexing techniques. A special aspect of the CONCERT approach is the fact that indexing is performed for abstract data types provided its conceptual behaviour class. Further, each specific data type can then be mapped to one of the four concepts provided by the CONCERT framework. Spatio-temporal indexing structures can be implemented and plugged into CONCERT independently of the data type indexed by just using the SPATIAL concept typical operations.

SECONDO (Dieker and Güting 2000) is a data model independent environment for the implementation of non-standard database systems. According to Breunig et al. (Breunig, Türker et al. 2003) it offers the advantage of both the extensible system and the toolkits approach while avoiding their disadvantages. The core component of SECONDO is a generic "database system frame" that can be filled with implementations of a wide range of data models, such as the relational, object-oriented, graph-oriented, sequence-oriented or spatio-temporal database models. In order to accomplish this in SECONDO data model, independent components and mechanisms are separated from data-model dependent parts. For that, the system frame has the descriptive algebra (for data model and query language

[2] An exception is made for TimeDB as the concepts from this DBMS have been the working grounds for the spatio-temporal data management proposed in chapter 3.

definition), the execute algebra (for data structure and operations definition) and the rules (to enable the formulation of query plans which map descriptive algebra terms into corresponding execute algebra terms). A general formalism proposed by Gütting (Güting 1993), designated Second Order Signature serves the purpose of enabling the development of the previous referred components. According to Gütting, the key idea of Second-Order Signature (and algebra) is a system of two coupled many-sorted signatures, where the top-level signature offers type constructors and the bottom-level signature provides polymorphic operations over the types defined as terms in the top level. Hence the top level can be used to define a data or representation model, while the bottom level can be used to describe a query processing algebra. Breunig *et al.* regard the SECONDO core component as being capable of providing a formal basis to describe generic query plan algebra giving a clear algorithm for translating a query plan into an operator tree, functional abstraction, and including complete type checking, type mapping and resolution of operator overloading.

TIGER (M.H. Böhlen 1998) is a temporal database system which focuses on temporal database extension but which, according to Böhlen *et al.*, can be extended for spatio-temporal data access. According to Breunig *et al.* (Breunig, Türker et al. 2003) this system is implemented as a layer to the Oracle DBMS. Key features of this system are:

- enhanced temporal support through statement modifiers: upward compatibility, temporal upward compatibility, sequenciality, non-sequenciality;
- intelligent reuse of existing database technology as it is built over Oracle DBMS;
- seamless extension of database systems with external modules into the layer;
- support for temporal aggregation, temporal coalescing and temporal difference through external modules.

The implementation of TIGER employs different programming paradigms such as C++, C, Java and Prolog. The implementation is based on the architecture of Chronolog (Böhlen 1994) and, according to Breunig *et al.* (Breunig, Türker et al. 2003) it is also related to TimeDB. TIGER is the first system that employs external modules, is accessible online and can be used for distance learning.

TEMPOS (M. Dumas 2000) is a temporal prototype system which extends the ODMG object model (Cattell, Barry et al. 2000) to add temporal support. This system tries to overcome important limitations of previous extensions such as TAU, TOOBIS and T_ODMG. According to Dumas, these extensions where mainly limited in what concerns (a) the support for legacy applications, (b) the independence between the abstract and representational views of each new concept, during the modelling and language designs and (c) the lack of formal semantics in the data model and the query language.

TRIPOD (Tony Griffiths 2001) addresses the object data model in temporal databases, in an attempt to overcome the fact that much of the research work is aimed at the relational data model, with object data models receiving far less consideration. TRIPOD is a spatio-historical object model that uses a specialized mechanism, called a *history*, to maintain knowledge about entities that change over time. Key features of its proposal include:

- consistent representations of primitive spatial and temporal data types;
- a component-based design in which spatial, temporal and historical extensions are formalized incrementally, for subsequent use together or separately;
- a formally specified data model; The model can be used directly during the design of spatio-historical applications, but also forms the basis of an implementation activity developing a spatio-historical object database management system.

Griffiths states that the key principles underpinning the TRIPOD project are *orthogonallity* and *synergy*. The former means that the different extensions to the ODMG standard (Cattell, Barry et al. 2000) should be coherent in isolation: the TRIPOD system should be effective when the information represented concerns only the spatial dimensions or only temporal dimensions. The *synergy* principle concerns the fact that the system should allow the use of spatial and temporal functionality in a seamless integrated manner.

Oracle DBMS is a commercial database management system which has been used to develop temporal extensions (Böhlen, C.S.Jensen et al. 1998; Steiner 1998). Since its version *9i* it has been providing features for temporal support. According to Snodgrass (Snodgrass 2006):

- Oracle 9i included support for *transaction time* as it allowed applications to access prior *transaction time* states of the database; In Oracle 9i, modifications and conventional queries are temporally upward compatible;
- Oracle 10g extended flashback queries to retrieve all the versions of a row between two *transaction times* (a key-*transaction time*-range query) and allowed tables and databases to be rolled back to a previous *transaction time*, discarding all changes after that time;
- The Oracle 11g Workspace Manager includes the *period* data type, *valid time* support, *transaction time* support, support for bitemporal tables, and support for sequenced primary keys, sequenced uniqueness, sequenced referential integrity, and sequenced selection and projection, in a manner that Snodgrass considered quite similar to that proposed in SQL/Temporal. However, as seen in *Oracle white paper* (Oracle Corporation 2007):
    - If *valid time* is enabled for a table, each row contains an added column to hold the *valid from* and *until* period associated with the row, as an object type.

- The user activates a session variable with the *SetValidtime* procedure that acts as a filter on subsequent queries. This support for *valid time* is clearly distinct from the support provided in other query languages that are in the base of the SQL/temporal proposal for SQL3.

TimeDB (Steiner 1998) is a temporal prototype system that follows a layered architecture, uses ATSQL2, provides bitemporal statements, supports upward compatibility and temporal upward compatibility, allowing legacy data and code to maintain the usability (Böhlen 1995). These concepts and constructs from SQL/Temporal have been implemented in the TimeDB version that uses Oracle. These ideas have also made their way into design patterns for things that change with time. Vassilakis (Vassilakis, Georgiadis et al. 1996) regards TimeDB as begin part of the family of implementations that provide temporal functionality at the client side. To this author, such family has high implementability but abandons full data integrity and consumes a considerable amount of resources (such as memory allocation). Vassilakis also suggests (and presents an example of this situation) that implementation considerations may lead designers to limit the amount of temporal functionality offered by the extended DBMS, if they are not building a full temporal DBMS. On the other hand, according to Torp (K. Torp 1997), developing a full-fledged DBMS that supports a temporal superset of SQL is a complex task that only the major vendors can expect to accomplish. This author states – regarding TimeDB - that layered technology promises much faster development. Lastly, although a client-side temporal implementation, TimeDB is not penalized by limited temporal functionality or the requirement to develop snapshot operations inside the layer, since all operations (and temporary results) are totally dealt by the underlying DBMS. Steiner considers the translation algorithm of TimeDB a general one, in the sense that it can be used to translate different temporal query and modification languages into standard SQL statements (Steiner 1998). This algorithm creates and uses tables to store temporary results that are produced by a statement and used by the next one. Hence, TimeDB does not retrieve intermediate results, avoiding performance problems due to exchange of temporary data transfers with the underlying DBMS.

Other spatio-temporal extensions exist, such as GeoToolkit (Balovnev, Breunig et al. 1997), a layered system architecture to provide a spatio-temporal extension of ObjectStore that has been design to support geological database applications, especially those which are intrinsically space and time dependent (Breunig, Türker et al. 2003). For instance, GeoDeform works with GeoToolkit and it is an application based on the concept of geological time-space modelling.

Finally, application tools that deal with *transaction time* have also been developed, most of which allow the compilation and view of how rows from one database change over time. This compilation serves as the starting point to selectively back out or replay changes or to query the changes over the data. Examples of these applications (Snodgrass 2006) are LogExplorer

(for Microsoft SQL Server), Tempos Time Navigator for IBM DB2, Oracle, MSSQL Server, Sybase, IBM data propagator and MarkLogic Server.

2.8 CONCLUSIONS

Definitions of a GIS have been evolving over the decades, depending on scientific, technological, organizational and social aspects, as well as on the perspective from which GIS has been addressed: as a collection of data; as a repository of maps or a set of tools able to generate maps; and finally, as a mean to model the reality properly dealing with the spatial characterization of objects and events.

In the past, the role of storage and retrieval of spatial data in GIS was performed by systems that adopted either a monolithic architecture or a layered architecture. Both had disadvantages since in the first approach the application development was confined to the GIS vendor. Furthermore, data was recorded in file formats which were only known by the GIS vendor and, consequently, data interchange with other GIS was not easy. With the layered architectural approach a system had to handle two subsystems carrying optimization problems in tasks where a seamless integration would be desirable, for instance, in the evaluation of a query that had alphanumeric and spatial restrictions.

Currently in GIS, DBMS perform a critical role in the storage, management and retrieval of spatially dependent information. This is true since the early stages related to capture, synthesis and acquisition of information, going through validation, storage and retrieval, and until visualization, presentation and exploratory data analysis. Generally, current DBMS supporting GIS are extensions of a relational or object-relational database management system, which provides spatial data types and operators, spatial functions and spatial index mechanisms for multi-dimensional access optimization. One important aspect in what seems to be this consensual approach of extending a DMBS is that DBMS developers have been making an effort to increasingly incorporate ISO and OGC standards in their solutions, thus contributing to higher degrees of interoperability and interchange of spatial data between these systems.

Furthermore, in the last years there has been an increase in the demand of efficient temporal data management for most of the application domains managing spatial data. The fact is that time stamping spatial data with dates is starting to become insufficient for solving problems and providing answers that involve spatial and temporal attributes seamless integrated. Moreover, without proper support for temporal data management, such as the support for *valid time*, *transaction time*, temporal functions and operators, spatial SQL queries become too complex to formulate if certain temporal requirements are involved. Consequently, the research community on spatial databases started to work more closely with the research community addressing temporal data management.

In fact, built-in temporal support on query languages, provided by temporal database management systems seem to be one of the most appropriate approaches to support time-varying data in application domains for which time is critical to manage. However, the development of fully-fledge temporal databases management systems is considered a very complex task. Nonetheless, this goal has been continuously pursued over the last three decades. In order to accomplish it, the research community has provided several distinct points of view for reasoning about the nature of time, a wide range of temporal representational data models, and attempts to unify these data models. The research community also provided several query language designs, which try to fulfil the largest number of requirements such as the support for *valid time* and *transaction time*, upward compatibility, temporal upward compatibility, snapshot reducibility, interval preserving, sequenced and non-sequenced semantics, temporal operators and functions such as temporal join or coalesce, respectively. Their implementations use several architectural approaches, although providing temporal support as an extension to SQL seems to be the common and generalized approach. ATSQL and its implementation over the relational model in TimeDB seem to be one of the best suited solutions for temporal data management. Moreover, other approaches have been done to temporal data management, such as frameworks that have been proposed for developing temporal (and spatial) DBMS, such as SECONDO or DEDALE.

In spite of all the developments, commercial DBMS still lack proper temporal support. An exception can be presented which regards Oracle: recently, this DBMS started to provide *valid time* support, however in ways that seem not to totally comply with the recommendation of SQL/Temporal to SQL3.

One of the possible reasons for this generalized lack of temporal support can be associated with the fact the SQL/Temporal has not yet been approved as a standard by ISO. From the perspective of GIS, the current demand for spatio-temporal DBMSs has no answers from the market.

There is still no DBMS that is able to provide the required spatial support (section 2.7) seamless integrated with the temporal support provided by languages such as ATSQL2 or defined in SQL/Temporal.

3 THE ACTION! SPATIO-TEMPORAL DATABASE MANAGEMENT SYSTEM

Geographic Information Systems play an important role in a wide range of application domains, both for the management of spatial data and as a decision support tool. Until recently, priority has been given to the spatial issues, leaving the temporal aspects of data much outside GIS. To manage spatio-temporal data in a convenient and efficient manner within these systems, common approaches have been to ignore time and dealing with the most recent state of facts, or to manage the temporal aspect of data at the application level. Such approaches compromise the efficient management of spatio-temporal data information overweighting the application layer instead of the DBMS.

To our knowledge no current DBMS for GIS provides full temporal data management over the *valid time* and *transaction time* domain. In order to overcome this limitation this chapter reports a contribution to the introduction of temporal support in GIS by proposing a solution and the corresponding DBMS proof of concept - *Action!* - whose query language is based on ATSQL2 (Steiner 1998) and Oracle Spatial (Oracle Corporation 2001). The resulting query language provides the support to store, manage and query spatial data over both the *valid time* and *transaction time* dimensions. It also provides support for legacy applications and, in this way, allows the protection of previous investments in GIS by still keeping operational legacy geographic information applications disregarding the temporal features.

This chapter starts by describing the spatio-temporal data management features that become available from the integration of ATSQL and Oracle Spatial query languages into a single spatio-temporal query language. Next, the data representation scheme is described, regarding the temporal and spatial dimensions. Furthermore, the proposed query language is presented in terms of the spatial and temporal features that are provided. Details are given for statements such as *CREATE*, *SELECT*, *INSERT*, *UPDATE* and *DELETE*. Next, the document presents the key ideas on how to provide the featured data management in a DBMS as well as the adopted architectural approach. Furthermore, the translation of spatio-temporal SQL statements into snapshot equivalent statements is presented and discussed. This chapter also addresses the key issues and choices done regarding the prototype system which implements the adopted data model and proposed query language. Moreover, this chapter presents a set of tests on the spatio-temporal data management features that become available in *Action!*. Each test describes the spatio-temporal feature that is being evaluated and also consists of one or more spatio-temporal statements, their corresponding snapshot-equivalents and the results that are retrieved. The current chapter ends with the main conclusions that can be drawn from the proposed spatio-temporal data management.

3.1 FEATURED SPATIO-TEMPORAL DATA MANAGEMENT

Efficient temporal data management is still a problem in certain application domains involving GIS. Typically GIS follow two approaches regarding the management of the temporal aspects of data. The first is to manage only the most recent state of facts and their spatial constraints and the second is to time stamp facts with the help of simple data types like date/time. Both these approaches entail severe limitations concerning the management of temporal aspects of spatial data (Date 2000). The snapshot approach disregards information from the past which means that, for example, no historic data can be used in queries supporting decisions where temporal evolutions could make a difference.

The semi-temporal approach provides a simple management of historical data, but problems emerge when handling some constraints and queries due the lack of proper temporal support, mainly because of the use of instants for time stamping where time periods would do a much better job (Date 2000). Furthermore, in this context, the lack of temporal support by the database server is considered the strongest limitation. For instance, the ability to issue statements benefiting for upward compatibility and temporal upward compatibility (Snodgrass, Böhlen et al. 1996), involving sequenced semantics and non-sequenced semantics, and using temporal functions (Böhlen, Snodgrass et al. 1996) and operators over instants and intervals (F.Allen 1983). Hence, due to the lack of proper temporal support by the DBMS, applications requiring temporal data management features have these features provided at the application level and not at the DBMS level. This approach has drawbacks such as the increase of program complexity, the inefficiency of the algorithms used, sometimes not optimized, and the burden of traffic between the snapshot DBMS layer and the application layer.

There is an increasing demand for *valid time* support in a considerable number of application domains, especially for those that deal with geographic information. These application domains would benefit from proper temporal data management in order to more easily support technical analysis based on temporal evolutions instead of temporal snapshots.

There are also situations where it is required to know when the facts have been recorded in the database. Such situations are related to auditing and involve the management of another dimension, the *transaction time* (Snodgrass, Böhlen et al. 1996).

Moreover, despite the willingness to adopt *valid time* and *transaction time* support, the technological upgrade to a full STDBMS may also require the protection of previous investments. Due to this fact, the STDBMS should provide upward compatibility regarding the support for spatial and temporal data.

In order to overcome the lack of support for *valid time* and *transaction time* in a geographic information environment, while keeping support for legacy applications, the current work has studied previous researched methods and techniques for spatial data management, which

ultimately resulted in the proposal of the spatio-temporal database management system designated by *Action!*. It is important to state that *Action!* has been thought for supporting applications where the representation of discrete changes of spatial objects, over time, is considered an adequate representation of spatio-temporal data. There is a wide range of geographic information domains that have such requirements, for example, the management of urban data, land ownership and facility management where buildings, roads, pavements, public gardens, urban equipment, traffic signs, land blocks and infra-structures evolve over time. The facts about these objects require the representation of alphanumeric, spatial and temporal data, suited to be used by analysis and decision support applications which address all the temporal states of data instead of only the most recent state.

The spatio-bitemporal data management contribution of this work has its core in a new query language that supersets ATSQL2 and Oracle Spatial. The underlying reason that fundaments this new query language is that, among several temporal query languages available, the temporal features provided by ATSQL2 have been considered the most acceptable, according to the requirements in section 2.7.

The choice for Oracle Spatial has been based on the spatial support provided, but also on the syntax similarity and compatibility with ATSQL2. Among the possible candidate spatial query languages that have been studied, taking into account the requirements just described, Oracle Spatial, developed on top of Oracle Cartridges, was identified as being one of the most well suited SQL languages. Oracle Spatial and ATSQL2 are both supersets of SQL, and a systematic analysis showed that there are no statement modifiers with conflicting semantic between these languages. Another reason for choosing Oracle spatial was the kind of spatial support provided which is considered efficient and robust. The major inconvenient of Oracle Spatial is the fact that it is not open source and, until recently was not freely available. The second inconvenience has been removed since, as a consequence of the high competition in the market regarding off-the-shelf database solutions, Oracle currently provides Oracle Spatial freely available and distributable in Oracle Express (targeted for small companies), with limitations regarding the number of processors used or the number of simultaneous connections.

The proposed query language provides data definition and data modification statements that, taking advantage of the temporal support provided by ATSQL2, allow the integration in these statements of the spatial support provided by Oracle Spatial. In short, the new query language encompasses the spatial data types, the spatial schema, the spatial functions and spatial operators from Oracle Spatial into ATSQL2.

Moreover, the use of spatial features integrated into ATSQL2 required an evaluation of the possibility to include these features into Steiner translation algorithm (Steiner 1998). The most important underlying reason is the fact that, in the case of sequenced semantics queries, the relational algebra operators used are the temporal counterparts (temporal select, temporal

project, temporal cross-product, temporal set operators). From the evaluation resulted that the modifications performed to the translation algorithm are related to the inclusion of the support for spatial attributes and functions and operators within the snapshot-equivalent mappings that convert temporal algebra operators into conventional algebra operators. By addressing these modifications, the spatio-temporal query language provides the following features:

- legacy SQL data definition and data modification statements;
- support for spatial data by means of spatial data types and variables, spatial operations and functions and spatial indexing mechanisms;
- support for temporal data types, operations and functions from ATSQL2;
- Sequenced *valid time* and *transaction time* statements involving spatial attributes, spatial data types, spatial operators and spatial functions;
- Non-sequenced *valid time* and *transaction time* statements involving spatial attributes, spatial data types, spatial operators and spatial functions;
- *Valid time* and *transaction time* coalescing of spatio-temporal relations which include spatial attributes.

As it can be regarded from the previous features, more than providing complementary features for space and for time, the underlying languages that contribute to the proposed spatio-temporal query language work seamlessly integrated.

Regarding the integration, the snapshot-equivalent mappings for the temporal relational algebra operators, such as *temporal join*, *temporal select* and *temporal project*, have been studied and analysed in order to determine the impact of using spatial attributes, spatial functions and spatial operators within these mappings. The temporal relational algebra operators replace the conventional operators when a sequenced semantics modifier is used in a temporal SQL statement, such as the word *validtime* starting a statement. In this situation, the statement should be evaluated over all the states of data, instead of just the current state.

Based on the snapshot-equivalent mappings defined by Snodgrass and Böhlen (Snodgrass, Böhlen et al. 1996), it can be said that it is possible to translate the temporal operators to snapshot-equivalent statements (that work with conventional relational algebra operators) regardless of the use of spatial attributes and spatial functions. For such purpose, in the translation, spatial attributes should be regarded as any other non-temporal attribute and the same goes for operators and functions.

Finally it is expected that the proposed spatio-temporal SQL language should inherit the features already provided by ATSQL2 and orthogonally to these features should provide representation of geometry, the use of spatial coordinate systems and the manipulation and retrieval of 2D/3D data through a set of spatial functions and spatial operators.

An STDBMS implementing the proposed spatio-temporal query language should enable organizations to start managing spatio-temporal data while also keeping appropriate data management for legacy applications. This approach can be regarded as a solution for systems where new applications can explore the spatio-temporal support provided while old applications can benefit from the support given by *standard* SQL, *spatial* SQL and ATSQL2.

3.2 REPRESENTATION SCHEME

The *Action!* prototype STDBMS implementing the spatio-temporal data management described on the previous section adopts Snodgrass Tuple Timestamped Representation Scheme. As previously presented in section 2.4.1, in Snodgrass Tuple Timestamped Representation scheme, a relation providing *valid time* support requires an additional attribute of data type *time period*, in the relation schema, to represent a *valid time* instant or interval, per tuple. The same goes for *transaction time*. Alternatively, the attribute of type time period can be replaced by two attributes of type time instant, thus allowing an equivalent representation.

In *Action!*, as in ATSQL2, *valid time* support and *transaction time* support can be added to tables through the use of *AS VALIDTIME* and *AS TRANSACTIONTIME* modifiers, placed at the end of a *CREATE TABLE* statement. For each time dimension *Action!* creates two new attributes of type integer, which are designated *VTS_TIMEDB*, *VTE_TIMEDB* for *valid time* and *TTS_TIMEDB*, *TTE_TIMEDB* for *transaction time*.

The reason for keeping the attribute designations used in TimeDB has to do with compatibility of data representation and storage between the two systems. This choice has proved very useful for *valid time* support testing and validation purposes, as it allowed running tests for TimeDB in *Action!*. For the same reason *Action!* also provides the use of the TimeDB calendar implementation which is distinct of the Gregorian calendar. Alternatively, *Action!* also supports the use of a Gregorian calendar implementation. However, the selection of the calendar used in a database instance should be performed before any insertion, updating or retrieval of data.

Action! relies on the spatial data model defined in Oracle Spatial for the representation of 1D, 2D and 3D geometric objects. For this purpose, a relation can store spatial data through the definition of spatial attributes in its schema that use the *MDSYS.SDO_GEOMETRY* data type. As an illustration, consider the following statement which creates a table, *countries2*, with support for *valid time* and spatial data:

```
CREATE TABLE countries2 (name char(20), capital char(20), boundary
mdsys.sdo_geometry) AS VALIDTIME
```

SQL Code 1: The creation of table *countries2* with spatial and *valid time* support.

This statement creates a new table *countries2* with columns: VTS_TIMEDB, VTE_TIMEDB, NAME, CAPITAL, and BOUNDARY. The first two columns (VTS_TIMEDB,

VTE_TIMEDB) are used to provide per-tuple *valid time* support while the last column (BOUNDARY) provides support for spatial data.

Furthermore, for every spatial attribute defined in a new table there is the need to insert geometry metadata in table *user_sdo_geom_metadata*. Satisfying this requirement enables the DBMS to deal with the spatial attribute properly, when referred in spatial functions or operators. For instance, the following statement, created by the user, inserts metadata in table *user_sdo_geom_metadata* about the spatial attribute *boundary*, comprising information such as the spatial coordinate system, the spatial boundaries and the spatial precision.

```
INSERT INTO user_sdo_geom_metadata VALUES('countries2','boundary',
MDSYS.SDO_DIM_ARRAY( MDSYS.SDO_DIM_ELEMENT('X', 0, 1000, 0.005),
MDSYS.SDO_DIM_ELEMENT('Y',  0,  1000, 0.005)), NULL);
```

SQL Code 2: A spatial metadata declaration for column *boundary* of table *countries2*.

Moreover, other spatial data is generated, regarding each spatial attribute for which an index is created. Indexes are mandatory for the use of spatial functions. However their data is managed by Oracle Spatial and it is not directly available to the user.

This spatio-temporal representation scheme stores and manages alphanumeric, temporal and spatial data. In order to do so, spatio-temporal SQL definition and modification statements are required.

3.3 SPATIO-TEMPORAL QUERY LANGUAGE

Regarding the possible spatial and temporal data management that an SQL statement can perform, *Action!* supports four groups of statements:

1. Standard SQL statements, which do not require spatial or temporal support;
2. Spatial SQL statements, which do not require temporal support;
3. Temporal SQL statements, which are translated into snapshot-equivalent SQL statements using the Steiner's translation algorithm. This means that each of these statements can be translated into sets of statements that match the group of statements presented in 1;
4. Spatio-temporal SQL statements which are translated into snapshot-equivalent spatial SQL statements thanks to the modifications that have been performed to Steiner's translation algorithm. The management of this group of statements can be considered the major contribution of *Action!*. For the other groups, Oracle Spatial and ATSQL2 already provide support. Here, a modified version of Steiner's translation has been used, which accommodates the necessary changes to integrate spatial support within temporal support. However, the principle remains the same: spatio-temporal statements are translated into their snapshot-equivalents, which fall into group of statements 2.

Hence, a STDBMS implementing this language has to manage four groups of SQL statements but, due to the translation that takes place within the STDBMS, only the support for standard SQL and spatial SQL is required in the STDBMS core.

Action! also supports the creation of several database instances; each can contain relations with distinct support: snapshot, spatial support, temporal support or spatio-temporal support. By combining the previous groups of SQL statements with the possible temporal nature of relations the following cases have been identified, summarizing the compatibility between an SQL statement being issued and the tables involved in it.

		Table temporal support			
		snapshot	*valid time*	*transaction time*	*valid and transaction time*
Statement temporal nature	*snapshot*	Case 1	Case 2	Case 3	Case 4
	valid time		Case 5		Case 6
	transaction time			Case 7	Case 8
	valid and transaction time				Case 9

Table 1: The temporal compatibility between SQL statements and the target tables.

In the Table 1, empty cells denote temporal incompatibility between SQL statement being issued and the tables involved in it. For instance a *valid time* statement cannot be performed over a snapshot table. The cells with cases identified by numbers correspond to distinct compatible execution contexts.

The spatio-temporal features of *Action!* can be grouped into SQL definition and SQL modification statements. For each statement the cases of temporal compatibility summarized in the table above are addressed with distinct possible results.

As defined in section 1.1, *UC* is used to denote upward compatibility, *TUC* is used to denote temporal upward compatibility, *SEQ* is used to denote the use of a sequenced semantics modifier and *NSEQ* is used to denote the use of a non-sequenced semantics modifier.

3.3.1 SQL Data Definition Statements

The modification statements provided by the STDBMS are *CREATE TABLE and ALTER TABLE* commands which act as their snapshot counterparts except for features that can be supported by ATSQL2 and Oracle Spatial:

- The *CREATE TABLE* gets from ATSQL2 the possibility to provide temporal support for *valid time* and *transaction time*. The *ALTER TABLE* command also inherits from ATSQL2 the possibility to temporally promote a snapshot table or a table with support for one temporal dimension;
- Using Oracle Spatial, the *CREATE TABLE* and *ALTER TABLE* allow columns with support for spatial data.

Hence, the creation of a table with *valid time* support or with *transaction time* support depends only on the use of the *AS VALIDTIME* or the *AS TRANSACTIONTIME* modifier at the end of the statement, respectively. If any of these statement modifiers are omitted the table is created without temporal support. The following SQL Code 3 creates a table with *valid time* support which also provides support for the representation of spatial entities in column *boundary*:

```
create table countries2 (name char(20), capital char(20), boundary
mdsys.sdo_geometry) as validtime;
```

SQL Code 3: The creation of table *countries2* with support for spatial entities as well as *valid time* support.

Moreover, the temporal upgrade of a snapshot table can be performed through the *ALTER* statement by appending the modifier *ADD VALIDTIME*, for *valid time*, or *ADD TRANSACTIONTIME*, *for transaction time*. If a table is to be temporally upgraded on both dimensions two *ALTER TABLE* commands are required, one for each dimension. Regarding the data representation scheme, the temporal upgrade on each temporal dimension inserts two new attributes in the table schema: *VTS_BEGIN*[3] and *VTS_END* for *valid time* and *TTS_BEGIN* and *TTS_END* for *transaction time*. If the table does not already provide support for that dimension an error is presented. For each row in a table, the pair of attributes allows the representation of a time period, defined by its end points. As in other models it is assumed that the end time instant is considered excluded from the time period. The temporal upgrade of a table inserts values in the time period of each row, but the values depend on the temporal dimension for which the table was temporally upgraded. The time period of each row, regarding *valid time*, is defined as starting with the value of the database variable *now*, at execution time, and ending with the value of the database constant *forever*. For the

[3] For the purpose of compatibility with the TimeDB data model, the same names where used in the attribute identification regarding the *valid time* and transaction time.

transaction time, the time period of each row is defined as starting with the instance of database variable *now* at execution time and ending with an abstract reference to *now*.

For instance, after executing the *CREATE STATEMENT in* SQL Code 4, the following *ALTER* statement adds *transaction time* support to table *countries2*, temporally promoting it to spatio-bitemporal support.

```
alter table countries2 add transactiontime;
```

SQL Code 4: The temporal upgrade of table *countries2* for *transaction time* support.

Regarding the CREATE TABLE and the *ALTER TABLE* statements, the following operations are executed:

- For *CREATE TABLE* statements, check for table existence. If it does not exist, create the table with the schema defined in the statement. At this moment *AS VALIDTIME* and AS *TRANSACTIONTIME* are ignored;
- For *ALTER TABLE* statements, check for table existence and check the temporal compatibility between the table and the statement;
- If *valid time* support is being added to a table, by modifiers *AS VALIDTIME* or *ADD VALIDTIME*, two new columns, *VTS_TIMEDB* and *VTE_TIMEDB*, are inserted in the table schema in order to allow the representation of a *valid time* period for each row. The VTS_TIMEDB and *VTE_TIMEDB* are filled with the value of variable *now* and the constant *forever*, respectively, for already existing tuples. This means the information is considered to hold from the instant the command was issued to an uncertain instant in the future;
- If *transaction time* support is being added to a table, through *AS TRANSACTIONTIME* or *ADD TRANSACTIONTIME*, two columns, *TTS_TIMEDB* and *TTS_TIMEDB*, are inserted in the table schema to allow the representation of a *transaction time* period for each row. New *TTS_TIMEDB* and *TTE_TIMEDB* are filled with the current value of variable *now* and the reference to variable *now*, respectively, for already existing tuples. This means the information is considered to be true in the database from the instant the command was issued to the present instant of time.

3.3.2 SQL Data Modification Statements

The spatio-temporal modification statements provided by the STDBMS are *INSERT, UPDATE and DELETE*. Besides temporal compatibility issues between the statement and the tables used, each of the supported statements provides distinct behaviours and results depending on the temporal nature of the statement and the temporal support of the tables being used, as also illustrated in Table 1. Orthogonal to these behaviours is the support for spatial data modification, either as spatial selection, in the WHERE clause, as the projection of spatial attributes.

A *SELECT* statement is transformed into an equivalent algebra expression where the algebraic operators used depend on the temporal semantics of the query statement and of the temporal support of the referenced tables. Conventional algebraic operators are used except with sequenced semantics queries for one or more temporal dimensions. In this situation temporal relational algebraic operators are used, such as temporal projection, temporal selection or temporal cross-product. As defined in ATSQL2, non-sequenced semantics is activated through the use of the *NONSEQUENCED* modifier before the *SELECT* command. Moreover, for the use of sequenced semantics the *VALIDTIME* or *TRANSACTIONTIME* modifier should be placed before the *SELECT* command.

As an illustration the following *select* statements use non-sequenced and sequenced semantics, respectively, to select all the rows of table *countries2* (it is assumed the table provides support for *valid time*):

```
nonsequenced validtime select * from countries2;
```

SQL Code 5: A non-sequenced semantics query statement.

```
validtime select * from countries2;
```

SQL Code 6: A sequenced semantics query statement.

While the execution of SQL Code 5 is performed through conventional relational operators, the execution of SQL Code 6 is performed through temporal relational operators, in the example temporal projection. The lack of support for temporal relational operators, from the underlying DBMS, requires that each temporal relational operator is translated into an equivalent expression, defined through standard relational operators, whose results are snapshot-equivalent to the results produced by the temporal relational operator.

For a *SELECT* command, the resulting rows of a record set are described in the following Table 2, having in mind the distinct compatibility scenarios from Table 1 .

Results	Description
Case 1	Selects rows (UC).
Case 2	Selects rows that are currently *valid time* (*valid time* TUC).
Case 3	Selects rows that are current in the database (*transaction time* TUC).
Case 4	The intersection between Case 2 and Case3 (bi-temporal TUC).
Case 5	SEQ: If no scope is provided in the statement header, the execution retrieves all rows from the table. Otherwise retrieves the rows that correspond to the defined scope. In both cases temporal algebra operators are used which require proper translation in order to be executed with standard relational algebra operators. NSEQ: If no scope is provided in the statement header, the execution retrieves all rows from the table, otherwise returns all rows and time stamps each row with the defined scope. In both cases standard relational algebra operators are used.
Case 6	The intersection between cases 5 and 3.
Case 7	Similar to case 5, but for *transaction time*.
Case 8	The intersection between cases 2 and 7.
Case 9	The intersection between cases 5 and 7.

Table 2: The query results according to the temporal context in Table 1.

For results 5, 6, 7, 8, 9 temporal algebraic operators are used for *valid time* and for *transaction time* when sequenced semantics is specified in the query statement.

```
validtime and transactiontime select sdo_geom.sdo_area(a.boundary,
0.005) from countries2 a(period);
```

SQL Code 7: A sequenced spatial bi-temporal statement performing over a coalesced bi-temporal table *countries2*.

The execution SQL Code 7 first performs bitemporal coalescing of table *countries2*, aliased to *a*. This operation is triggered by the *(period)*, as defined in ATSQL2. The temporary table resulting from the coalescing operation is then used to project the area of the spatial attribute *a.boundary*. In the result, for each tuple, along with the numeric values, two time periods are also displayed regarding the *valid time* and the *transaction time*. All the states of facts recorded in the temporary table are accounted in this operation.

For INSERT statements the behaviour is described in Table 3:

Results	Description
Case 1	Inserts row (UC).
Case 2	Inserts row setting the *valid time* period to an interval starting at the current instant and ending in *forever* (*valid time* TUC).
Case 3	Inserts row setting *transaction time* period to an interval starting at the current instant and ending in *now* (*transaction time* TUC);
Case 4	Inserts row setting the *valid time* according to Case 2 and the *transaction time* according to Case 3 (bi-temporal TUC)
Case 5	If no scope is provided in the statement header, the execution inserts a row with *valid time* according to Case 2. Otherwise the *valid time* of the inserted row corresponds to the scope provided in the statement.
Case 6	Inserts a row setting *valid time* according to Case 5 and *transaction time* according to Case 3.
Case 7	Similar to Case 3. Scope definition in *transaction time* is forbidden.
Case 8	Inserts a row setting *valid time* according to Case 2 and the *transaction time* according to Case 7.
Case 9	Inserts a row setting *valid time* according to Case 5 and the *transaction time* according to Case 7.

Table 3: Execution of an *insert* statement (depends of temporal context defined in Table 1).

The SQL code 8 illustrates an insert statement that uses non-sequenced semantics to insert a tuple containing alphanumeric and spatial data, defining the *valid time* period to be [1250-1590[.

```
nonsequenced validtime period [1250-1590) insert into countries2
values ('portugal', 'lisboa', mdsys.sdo_geometry (2003, null, null,
mdsys.sdo_elem_info_array (1,1003,1),…,80.0,607.0)));
```

SQL Code 8: A spatio-unitemporal *insert* statement on bitemporal table *countries2*.

In the *delete* statement, for each row targeted for removal, the result can be the removal of the row, the update of the row or even the insertion of new rows. To illustrate this behaviour, regarding *valid time*, consider the following table instance *countries2*, which has *valid time* support:

valid time	NAME	CAPITAL	BOUNDARY
[1143-01-01 - 1150-01-01)	Portugal	Guimaraes	(a polygon)
[1150-01-01 - 1183-01-01)	Portugal	Porto	(a polygon)
[1183-01-01 - 1213-01-01)	Portugal	Coimbra	(a polygon)
[1213-01-01 - 1250-01-01)	Portugal	Coimbra	(a polygon)
[1250-01-01 - 1315-01-01)	Portugal	Lisboa	(a polygon)
[1315-01-01 - 1590-01-01)	Portugal	Lisboa	(a polygon)
[1640-01-01 - 1900-01-01)	Portugal	Funchal	(a polygon)
[1900-01-01 - forever)	Portugal	Lisboa	(a polygon)

Table 4: Contents of table *countries2*, prior to the *delete* statement from SQL code 9.

In this table a short history of Portugal is presented, from 1143 to 1590 and from 1640 until the present time. The column *boundary* contains the political boundary of Portugal, defined as a polygon, for each *valid time* period. Assume the polygon for column *boundary* and first row of *countries2* has an area smaller than 4500. Now consider the following *delete* statement is issued over *countries2*:

```
validtime period [1145-1149) delete from countries2
where sdo_geom.area(boundary, 0.05) > 4500;
```

SQL Code 9: A sequenced *delete* statement.

The *delete* statement affects the row with *valid time* [1143-1150), but only in what concerns [1145-1149). The result is that the *delete* statement removes the row but, in turn, inserts two new rows corresponding to the *valid time* intervals not included in the *delete* scope. Each non-temporal attribute in the new rows is filled with the corresponding value from the deleted row. Hence, the first row of table *countries2* would be deleted and in turn the rows presented in Table 5 would be inserted.

valid time	NAME	CAPITAL	BOUNDARY
[1143-01-01 – 1145-01-01)	Portugal	Guimaraes	(a polygon)
[1149-01-01 – 1150-01-01)	Portugal	Guimaraes	(a polygon)

Table 5: The new rows introduced by the *delete* statement.

The example *valid time delete* statement in SQL code 9 resulted in several conventional *insert*, *delete* and *update* statements that are issued to the server-side of the STDBMS. These statements contemplate all situations where, for each targeted row, the scope defined in the *delete* statement intersects, contains or is contained in the time period of the row.

The use of a *delete* statement in a table where *transaction time* is supported does not remove any row. Instead, for each row targeted for deletion the upper limit of the *transaction time* period is set to the instant the *delete* statement is executed. Semantically this means that the row is no longer current in the database.

Assume that table *countries2* depicted in Table 4 is temporally upgraded to *transaction time* on the 12th July 2006, as illustrated in Table 6. Hence, each row is time stamped with a *transaction time* period [2006-07-12 - *now(tt)*) where *now(tt)* represents the database variable *now*, meaning that the row is considered to be current in the database.

transaction time	valid time	NAME	CAPITAL	BOUND
[2006-07-12 - now(tt))	[1143-01-01 - 1150-01-01)	Portugal	Guimaraes	(a polygon)
[2006-07-12 - now(tt))	[1150-01-01 - 1183-01-01)	Portugal	Porto	(a polygon)
[2006-07-12 - now(tt))	[1183-01-01 - 1213-01-01)	Portugal	Coimbra	(a polygon)
[2006-07-12 - now(tt))	[1213-01-01 - 1250-01-01)	Portugal	Coimbra	(a polygon)
[2006-07-12 - now(tt))	[1250-01-01 - 1315-01-01)	Portugal	Lisboa	(a polygon)
[2006-07-12 - now(tt))	[1315-01-01 - 1590-01-01)	Portugal	Lisboa	(a polygon)
[2006-07-12 - now(tt))	[1640-01-01 - 1900-01-01)	Portugal	Funchal	(a polygon)
[2006-07-12 - now(tt))	[1900-01-01 - forever)	Portugal	Lisboa	(a polygon)

Table 6: The table *countries2* after *transaction time* support has been added.

The *valid time delete* statement presented in SQL Code 9 and executed on the 30th July 2006 would produce the following results:

- The *transaction time* of row with *valid time* [1143-1150) would be updated by setting the end limit of the *transaction time* to the instant the statement is executed. Semantically this row would be considered no longer current in the database;
- The rows displayed in Table 5 would be inserted. The *transaction time* of each inserted rows would start at the instant the statement is executed and would end at database variable *now*. Semantically these two new rows would be considered current in the database.

Table 7 displays the table with spatio-bitemporal support after the *valid time delete* command has been executed.

transaction time	*valid time*	NAME	CAPITAL	BOUND
[2006-07-12 – 2006-07-30)	[1143-01-01 - 1150-01-01)	Portugal	Guimaraes	Polygon
[2006-07-12 - now(tt))	[1150-01-01 - 1183-01-01)	Portugal	Porto	Polygon
[2006-07-12 - now(tt))	[1183-01-01 - 1213-01-01)	Portugal	Coimbra	Polygon
[2006-07-12 - now(tt))	[1213-01-01 - 1250-01-01)	Portugal	Coimbra	Polygon
[2006-07-12 - now(tt))	[1250-01-01 - 1315-01-01)	Portugal	Lisboa	Polygon
[2006-07-12 - now(tt))	[1315-01-01 - 1590-01-01)	Portugal	Lisboa	Polygon
[2006-07-12 - now(tt))	[1640-01-01 - 1900-01-01)	Portugal	Funchal	Polygon
[2006-07-12 - now(tt))	[1900-01-01 - forever)	Portugal	Lisboa	Polygon
[2006-07-30 - now(tt))	[1143-01-01 - 1145-01-01)	Portugal	Guimaraes	Polygon
[2006-07-30 - now(tt))	[1149-01-01 - 1150-01-01)	Portugal	Guimaraes	Polygon

Table 7: The table *countries2* after the delete command is executed.

As in the delete statement, the *update* statement is also translated into a sequence of conventional *insert*, *delete* and *update* statements that are issued to the server-side of the STDBMS. For instance, with the current contents of spatio-bitemporal table *countries2*, an *update* statement is issued on the 1st August 2006, for *valid time* period [1200-1590) where the boundary length is greater than 100 units, setting the capital to *Porto* (SQL Code 10).

```
validtime period [1200-1590) update countries2 set capital =
'Porto' where sdo_geom.sdo_length(boundary, 0.05) > 100;
```

SQL Code 10: a spatio-temporal update statement performing over bitemporal table *countries2*.

Assuming that in Table 7 the Portuguese boundary length is greater than 100 units during period [1183-1590), 4 rows are affected by the *update* command. However, the row with *valid time* [1183-1213) should only be affected in what concerns the time period [1200–1213). Hence, regarding this row, the following operations are executed:

- The affected row is set to no longer being current in the database: *transaction time* [2006-07-12 – 2006-08-01);

- A new row is inserted representing the same fact as the affected row, *valid time* [1183-1200) and *transaction time* [2006-01-08 – now(tt));
- Another new row with *valid time* [1200-1213) is inserted, reflecting the fact that the capital is Porto. Transaction time of this row is also [2006-01-08 – now(tt));

Table 8 illustrates the contents of *countries2* after the *update* statement has been executed.

transaction time	***valid time***	**NAME**	**CAPITAL**	**BOUND**
[2006-07-12 – 2006-07-30)	[1143-01-01 - 1150-01-01)	Portugal	Guimaraes	Polygon
[2006-07-12 - now(tt))	[1150-01-01 - 1183-01-01)	Portugal	Porto	Polygon
[2006-07-12 – 2006-08-01)	[1183-01-01 - 1213-01-01)	Portugal	Coimbra	Polygon
[2006-07-12 - 2006-08-01)	[1213-01-01 - 1250-01-01)	Portugal	Coimbra	Polygon
[2006-07-12 - 2006-08-01)	[1250-01-01 - 1315-01-01)	Portugal	Lisboa	Polygon
[2006-07-12 - 2006-08-01)	[1315-01-01 - 1590-01-01)	Portugal	Lisboa	Polygon
[2006-07-12 - now(tt))	[1640-01-01 - 1900-01-01)	Portugal	Funchal	Polygon
[2006-07-12 - now(tt))	[1900-01-01 - forever)	Portugal	Lisboa	Polygon
[2006-07-30 - now(tt))	[1143-01-01 - 1145-01-01)	Portugal	Guimaraes	Polygon
[2006-07-30 - now(tt))	[1149-01-01 - 1150-01-01)	Portugal	Guimaraes	Polygon
[2006-08-01 – now(tt))	[1183-01-01 - 1200-01-01)	Portugal	Coimbra	Polygon
[2006-08-01 – now(tt))	[1200-01-01 - 1213-01-01)	Portugal	Porto	Polygon
[2006-08-01 – now(tt))	[1213-01-01 - 1250-01-01)	Portugal	Porto	Polygon
[2006-08-01 – now(tt))	[1250-01-01 - 1315-01-01)	Portugal	Porto	Polygon
[2006-08-01 – now(tt))	[1315-01-01 - 1590-01-01)	Portugal	Porto	Polygon

Table 8: The contents of *countries2* after and *update* statement

Rows targeted for update that are fully contained in the *valid time* period of the statement are logically deleted and for each of these rows a new row reflecting the capital Porto is inserted.

3.3.3 Temporal Coalescing of Spatially-enabled data

For unitemporal coalescing *Action!* uses the nested-loop coalescing operator proposed by Böhlen *et al.* (Böhlen, Snodgrass et al. 1996). This operator requires to repeatedly update rows in a table (until no changes occur) setting the upper bound of the time interval of tuples for which value-equivalent tuples exist with an overlapping time period having a higher upper bound (Steiner 1998).

As previously said, *Action!* adopts the translation algorithm used in TimeDB, exploring the possibility to keep the temporal support already provided while adding support for spatial data management. At the conceptual level, performing unitemporal coalescing over a relation with spatial attributes should hold with the translation algorithm used in TimeDB since spatial attributes are regarded as any other non-temporal attributes, for which value-equivalent comparison is valid. However, regarding implementation issues in the *Action!* prototype, such comparison between two spatial values requires a specific function provided by Oracle Spatial: *sdo_geom.sdo_relate*.

The decision to perform unitemporal or bitemporal coalescing depends on the temporal support provided by the relation being coalesced and the temporal modifiers used in the *select* statement. For instance, the following statement:

```
validtime and transactiontime
(select * from countries2)(period);
```

SQL Code 11: The bi-temporal coalescing of table *countries2*.

requires *countries2* to be a table with bitemporal support. In this situation bi-temporal coalescing takes place meaning that first unitemporal coalescing is performed regarding *valid time* and secondly unitemporal coalescing is performed regarding *transaction time*. To illustrate the coalescing of a relation with bi-temporal support suppose that the statement presented in SQL Code 11 is executed when table *countries2* presents the contents in Table 8. It is assumed that the polygon describing the boundary, besides having a length greater than 100 as previously said, was kept unchanged for the period [1183-1590).

The result of bitemporal coalescing is depicted in Table 9 where, due to the fact that these rows are value-equivalent, current rows with *valid time* [1200-1213), [1213-1250), [1250-1315) and [1315-1590) are coalesced in one row with *valid time* [1200-1590). Furthermore, logically deleted rows with *valid time* rows [1183-1213), [1213-1250) are coalesced into [1183-1250) and [1250-1315), [1315-1590) are coalesced into [1250-1590).

transaction time	*valid time*	NAME	CAPITAL	BOUND
[2006-07-12 – 2006-07-30)	[1143-01-01 - 1150-01-01)	Portugal	Guimaraes	Polygon
[2006-07-12 - now(tt))	[1150-01-01 - 1183-01-01)	Portugal	Porto	Polygon
[2006-07-12 – 2006-08-01)	[1183-01-01 - 1250-01-01)	Portugal	Coimbra	Polygon
[2006-07-12 - 2006-08-01)	[1250-01-01 - 1590-01-01)	Portugal	Lisboa	Polygon
[2006-07-12 - now(tt))	[1640-01-01 - 1900-01-01)	Portugal	Funchal	Polygon
[2006-07-12 - now(tt))	[1900-01-01 - forever)	Portugal	Lisboa	Polygon
[2006-07-30 - now(tt))	[1143-01-01 - 1145-01-01)	Portugal	Guimaraes	Polygon
[2006-07-30 - now(tt))	[1149-01-01 - 1150-01-01)	Portugal	Guimaraes	Polygon
[2006-08-01 – now(tt))	[1183-01-01 - 1200-01-01)	Portugal	Coimbra	Polygon
[2006-08-01 – now(tt))	[1200-01-01 – 1590-01-01)	Portugal	Porto	Polygon

Table 9: The result of bi-temporal coalescing of *countries2* after state depicted in Table 8.

3.4 THE ARCHITECTURE OF ACTION!

Action! adopts a combination of the layered architectural approach and the extensible approach. Previously, both approaches have been presented as alternatives. However, the developed prototype system adopts both approaches due to implementation issues. Hence, in *Action!*, the layered architectural approach is used for the spatio-temporal support on the client side, through an application programming interface that connects directly to the DBMS. In *Action!*, the extensible approach corresponds to the spatial support, based on Oracle Spatial, inside the DBMS.

Spatio-temporal support should be provided following the extensible approach in order to benefit from features already provided by the DBMS such as transactions and concurrent access to resources. The selection of the layered architecture instead of developing a spatio-temporal extension over the spatial extension in Oracle has to do with the higher implementation difficulties regarding the second choice, already identified by several authors. This means that the implementation of *Action!* has the benefits of the layered architecture but also suffers some of the disadvantages of this approach (section 2.6). TimeDB, one of most well known implementation of ATSQL2, followed a similar architectural approach, supporting several DBMS among which Oracle (without spatial data management).

Nonetheless, the principles previously described about the proposed spatio-temporal query language remain valid with the chosen architecture where a spatio-temporal statement is issued by an application to the client-side part of the STDBMS, through the API. The client-side part is responsible for translating the statement into a set of snapshot-equivalent spatial SQL statements which are then issued to Oracle Spatial. During the execution of these statements temporary tables are created to store intermediate results. For spatio-temporal SQL statements a record set containing the results is kept on the server-side of the STDBMS, as another temporary table. This record set can be accessed by the application, through the API, based on commands such as *open(), next(), getValue() and close()*. This last command deletes the temporary table that kept the resulting record set.

Considering this workflow, notice that *Action!* heavily relies on the client-side part, to which an application can connect to issue SQL statements and retrieve results from it, as seen in section 3.3. Although four types of SQL statements can be issued by applications, the client-side only issues standard SQL or Spatial SQL to Oracle. It might seem that, regarding standard SQL and spatial SQL statements issued by applications, the client-side part of the STDBMS acts as a simple proxy. However, the identification of the temporal and the spatial features and requirements demands that every statement issued from applications to the API is scanned, parsed and an internal tree-like representation is constructed, corresponding to the statement execution plan.

For queries, the top node of this execution plan can be one of the following nodes:

- A relational algebra set operator: *union*, *difference*, *intersect*, each possibly having as argument another temporal set operator or an algebra expression (*select*, *project*, *cross product*);
- A relational algebra expression: *select*, *project*, *cross product*;
- A temporal relational algebra set operator (*valid time* and *transaction time domains*): *temporal union*, *temporal difference*, *temporal intersect*, each possibly having as argument another temporal set operator or a temporal algebra expression (*temporal select*, *temporal project*, *temporal cross product*);
- A temporal relational algebra expression *valid time* and *transaction time*: (*temporal select*, *temporal project*, *temporal cross product*);

Moreover, each of these nodes in the representation tree can refer to spatial attributes, spatial operators and spatial functions.

For other SQL data manipulation statements the top node assumes the expression containing the statement, such as the conventional *insert*, *update*, *delete* or their temporal counterparts for *valid time* and for *transaction time*.

For each statement issued by an application through the API and after completing its representation, the client-side of the STDBMS generates the set of standard SQL or Spatial SQL statements that are to be issued to the server-side part of the STDBMS. This task can be a straightforward construction when the statement issued by an application is a standard SQL or spatial SQL and is performed over tables with no temporal support. Otherwise, this task can involve:

- the translation of temporal operators when the original statement is temporal SQL or spatio-temporal SQL;
- the translation of standard SQL or spatial SQL statements that are performed over tables with temporal support;

In order to generate the set of statements that are issued to the server-side part of the STDBMS the tree corresponding to the execution plan is traversed from the leaves to the top root node and for each temporal relational algebra operator a translation takes place that generates snapshot-equivalent statements using conventional relational algebra operators.

The purpose of this translation is to enable sequenced semantics queries, non-sequenced semantics queries, derived tables, operations over intervals and instants, unitemporal and bitemporal coalescing. Steiner translation algorithm implementation in TimeDB 2.0b (the prototype version evaluated in this work) has demonstrated the support for these temporal features for *valid time*.

Steiner's original translation algorithm has been extended in *Action!* to encompass *transaction time* support and support for spatial data. Hence, transversally to the temporal features of Steiner translation algorithm, management for spatial data types, the spatial schema, the spatial indexing mechanism and the set of spatial functions and spatial operators, is required.

Figure 1 illustrates the dataflow through top-level modules, from the time a spatio-temporal SQL statement is issued to the STDBMS and until the results are retrieved (literal values or spatio-temporal record sets).

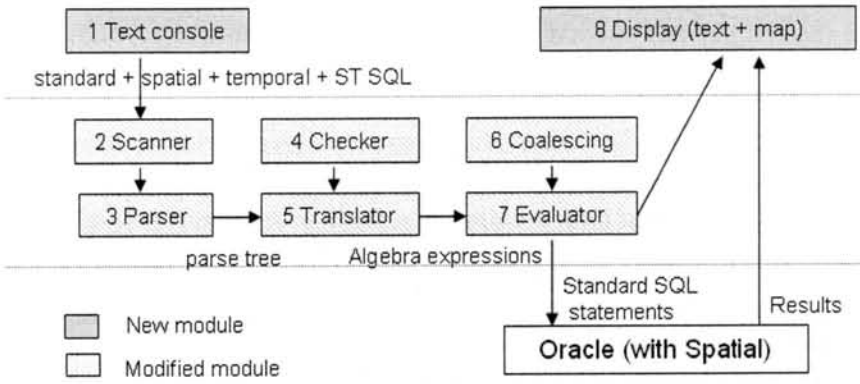


Figure 1: *Action!* modules and dataflow overview.

The following subsections address the modifications that are required in Steiner's translation algorithm, in order to provide spatio-temporal support.

3.4.1 Support for Spatial Data Types, Operators and Functions

In order to provide spatial support, the modified version of the translation algorithm encompasses spatial related goals that, in short, are the capacity to scan, parse and translate statements that refer or use the underlying spatial data model, the geometry data types, the spatial (and aggregate) operators and functions, and the use of spatial index mechanism.

This allows that spatial data types, operators and functions can be used within the spatio-temporal query language presented in section 3.3, in order to provide spatial projections and selections. To this purpose, new features have been added to the translation algorithm, which now provides:

- address the *mdsys* schema, in order to access and manipulate data and structures like geometry metadata tables (*user_sdo_geom_metadata*) and index metadata (*mdsys.user_sdo_index_metadata*);
- management of spatial data types, such as *mdsys.sdo_geometry* or *mdsys.spatial_index* and aggregate spatial data types such as *mdsys.sdoaggrtype* (this data type is required to provide support for spatial aggregate functions such as *sdo_aggr_union*);
- management of nested spatial data types and nested spatial functions where it is common for a spatial data type, function or operator to have several spatial arguments, each one being another data type or spatial function with its own arguments;
- management of spatial data type object methods, such as *get_gtype*, from *mdsys.sdo_geometry* data type;
- management of spatial indexes, as the indexation of spatial columns is required for the use of a considerable number of spatial functions, such as *sdo_filter* or *sdo_within_distance*.

Incorporating these features in the translation algorithm provides the ability for dealing with statements requiring spatial support (often integrated with temporal support), as illustrated in SQL Code 12 to SQL Code 15.

```
create table countries (name char(20), capital char(20), boundary
mdsys.sdo_geometry) as validtime;
```

SQL Code 12: The *valid time* creation of a table with an attribute of spatial type mdsys.sdo_geometry.

```
CREATE INDEX countries_idx ON countries (boundary) INDEXTYPE IS
mdsys.spatial_index;
```

SQL Code 13: The creation of a spatial index for column boundary of table *countries2*.

```
validtime period [1100-1350)
select sdo_aggr_union (mdsys.sdoaggrtype(c.boundary, 0.005)) from
countries c;
```

SQL Code 14: A sequenced semantics query that retrieves a shape consisting on the union of all geometries for each time period in [1100-1350).

```
nonsequenced validtime period [1250-1590) insert into countries
values ('portugal', 'lisboa', mdsys.sdo_geometry (2003, null, null,
mdsys.sdo_elem_info_array (1,1003,1),…,80.0,607.0));
```

SQL Code 15: The use of mdsys.sdo_geometry with nested spatial data types.

3.4.2 Support for Table and Column aliases

The management of spatial data types and functions can be considered a complex task since the translation algorithm has to be aware of the nature of each argument used in a spatial construct. For instance, consider the use of a sequenced semantics query such as presented in SQL Code 14 where argument *c.boundary* is used in the spatial union function. For this statement, sequenced semantics is used due to the *valid time* modifier at the head of the statement. Hence, this statement needs to be translated into a *snapshot-equivalent* set of statements for which attribute *boundary* of table alias *c* needs to be properly used. Another example of this requirement is presented in SQL Code 16:

```
validtime select a.name from countries a, countries b
where a.name = 'portugal' and b.name = 'poland' and
sdo_geom.sdo_distance(a.boundary, b.boundary, 0.005) <=0;
```

SQL Code 16: A sequenced semantics query that retrieves rows where Portuguese and Polish borders have met.

In the previous statement, two alias are defined, *a* and *b*, referring to the same table *countries*, in order to be used as arguments in the spatial function *sdo_geom.sdo_distance*.

3.4.3 Support for Temporal Operators Performing over Spatial Data

Steiner's translation algorithm translates temporal statements into the equivalent temporal algebra expressions using temporal set operators (temporal union, temporal intersection,

temporal difference), where each argument to one of these algebra expressions can be a simple temporal algebra expression or the result of another temporal set operator.

The integration of spatial functions and spatial operations into the translation algorithm is not a major problem: spatial projection and spatial selection operations are, from the perspective of temporal algebra, regarded as simple non-temporal projection and selection operations. Moreover, spatial data types are managed by the translation algorithm, in what concerns temporal algebra, as other non-temporal data types such as *number* or *character strings*. However, column references used as arguments in spatial data types or in spatial functions and operators require translation, as seen in section 3.3.

Furthermore, changes are required in order to properly translate the temporal set operators and the temporal *coalescing*. From the perspective of temporal algebra, each spatial attribute is dealt as any other non-temporal attribute. In all of these operators a comparison between tuples takes place: for each non-temporal attribute, the values of tuples are compared through the use of the equal operator, such as, in the following expression:

$table_1.column_a = table_2.column_a$

However, regarding spatial attributes the equal operator should be replaced by a spatial attribute *sdo_geom.relate* such as in the following expression:

$sdo_geom.relate(table_1.column_a$, *'equal'*, $table_2.column_a$, *<a spatial tolerance>) = 'equal'*

Through *sdo_geom.relate*, two spatial geometries can be compared and determined if they are equal, provided a given spatial tolerance. Hence, the snapshot-equivalent translation for temporal set operators and temporal coalescing requires that the snapshot equivalent mappings can accommodate the use of spatial operator *sdo_geom.relate* every time a spatial comparison is required. For instance, consider SQL Code 17 that is Steiner's translation of the *valid time* set difference operation into plain SQL.

```
INSERT INTO R1
SELECT a0.vts_#$, a1.vts_#$, a0.A1, a0.A2, ..., a0.An
FROM R1 a0, R2 a1
WHERE a1.vts_#$ > a0.vts_#$ AND
      a1.vts_#$ < a0.vte_#$ AND
      a0.A1 = a1.B1 AND
      a0.A2 = a1.B2 AND
      ... AND
      a0.An = a1.Bn;

INSERT INTO R1
SELECT a1.vte_#$, a0.vte_#$, a0.A1, a0.A2, ..., a0.An
FROM R1 a0, R2 a1
WHERE a1.vte_#$ > a0.vts_#$ AND
      a1.vte_#$ < a0.vte_#$ AND
      a0.A1 = a1.B1 AND
      a0.A2 = a1.B2 AND
      ... AND
      a0.An = a1.Bn;

DELETE FROM R1 a0 WHERE EXISTS (
```

```
SELECT a1.* FROM R2 a1
WHERE ((a0.vts_#$ >= a1.vts_#$ AND a0.vts_#$ < a1.vte_#$) OR
       (a1.vts_#$ >= a0.vts_#$ AND a1.vts_#$ < a0.vte_#$)) AND
        a0.A1 = a1.B1 AND
        a0.A2 = a1.B2 AND
        ... AND
        a0.An = a1.Bn);
```

SQL Code 17: translation of the *valid time* set difference operator into conventional statements.

Here the arguments for this set operator are relations *R1* and *R2* (which must be union-compatible). According to Steiner (Steiner 1998) the unitemporal set difference returns the tuples in *R1* with time intervals during which no value-equivalent tuple in *R2* can be found. At the end of execution, the results are kept in *R1*.

In order to better illustrate the previous SQL Code 17, consider the following Figure 2.

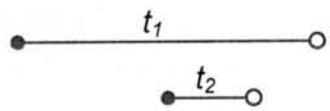


Figure 2: The *valid time* of tupple t_1 from *R1* and t_2 from *R2*.

Figure 2 illustrates the temporal existence of two tuples: *t1* from table *R1* and *t2* from table *R2*. The temporal set difference between *t1* and *t2* is illustrated in the following Figure 3:

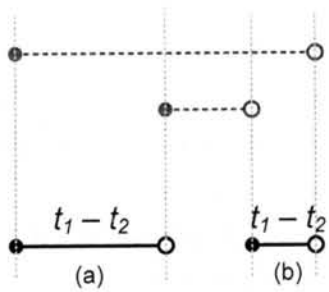


Figure 3: The temporal set difference between tuple *t1* and *t2*.

where:

- (a) is a tuple considered in the first *insert* statement in SQL Code 17 which addresses possible non-overlapping part of tuple *t1* at the beginning of the *valid time* interval of tuple *t2*. This tuple is inserted into table *R1*.
- (b) is a tuple resulting from the second statement in SQL Code 17 which addresses possible non-overlapping part of tuple *t1* at the end of the *valid time* interval of tuple *t2*. This tuple is inserted in table *R1*.

The *delete* statement removes all the tuples in table *R1* for which value-equivalent tuples in *R2* exist having overlapping time intervals, leaving in *R1* the result of the temporal set difference.

Regarding the modifications of this algorithm in order to accommodate spatial attributes, and assuming that attributes *A2* from *R1* and *B2* from *R2* are spatial attributes the previous translation becomes:

```
INSERT INTO R1
SELECT a0.vts_#$, a1.vts_#$, a0.A1, a0.A2, ..., a0.An
FROM R1 a0, R2 a1
WHERE a1.vts_#$ > a0.vts_#$ AND
      a1.vts_#$ < a0.vte_#$ AND
      a0.A1 = a1.B1 AND
     (sdo_geom.relate(a0.a2, 'equal', a1.b2,
                      <a spatial tolerance>) = 'equal') and
      ... AND
      a0.An = a1.Bn;

INSERT INTO R1
SELECT a1.vte_#$, a0.vte_#$, a0.A1, a0.A2, ..., a0.An
FROM R1 a0, R2 a1
WHERE a1.vte_#$ > a0.vts_#$ AND
      a1.vte_#$ < a0.vte_#$ AND
      a0.A1 = a1.B1 AND
     (sdo_geom.relate(a0.a2, 'equal', a1.b2,
                      <a spatial tolerance>) = 'equal') AND
      ... AND
      a0.An = a1.Bn;

DELETE FROM R1 a0 WHERE EXISTS (
SELECT a1.* FROM R2 a1
WHERE ((a0.vts_#$ >= a1.vts_#$ AND a0.vts_#$ < a1.vte_#$) OR
       (a1.vts_#$ >= a0.vts_#$ AND a1.vts_#$ < a0.vte_#$)) AND
        a0.A1 = a1.B1 AND
       (sdo_geom.relate(a0.a2, 'equal', a1.b2,
                        <a spatial tolerance>) = 'equal') and
        ... AND
        a0.An = a1.Bn);
```

SQL Code 18: The translation of the *valid time* set difference operator into conventional statements and providing support for spatial data.

Generalizing, the translation algorithm has to be aware of the nature of all attributes in order to provide appropriate comparison when the values being compared correspond to spatial attributes.

For instance, the following block of SQL code corresponds to the algorithm that maps the *valid time* coalescing operation into a set of conventional operators (Steiner 1998), modified to support spatial data for an attribute *C1* of table *R1*, which is being coalesced:

```
** REPEAT **
UPDATE R1 A0 SET (A0.vte_#$) = (
       SELECT MAX(A1.VTE_#$) FROM R1 A1
       WHERE A0.C0 = A1.C0 AND
             (sdo_geom.relate(A0.C1, 'equal', A1.C1,
                              <a spatial tolerance>) = 'equal') AND
             ... AND
             A0.CN = A1.CN AND
             A0.VTE_#$ >= A1.VTS_#$ AND A0.VTE_#$ < A1.VTE_#$

       )
WHERE EXISTS (SELECT * FROM R1 A1
              WHERE A0.C0 = A1.C0 AND
                    (sdo_geom.relate(A0.C1, 'equal', A1.C1,
                                     <a spatial tolerance>) = 'equal')
```

```
AND
                    ... AND
                    A0.CN = A1.CN AND
                    A0.VTE_#$ >= A1.VTS_#$ AND
                                 A0.VTE_#$ < A1.VTE_#$);

** UNTIL no updates **

DELETE FROM R1 A0 WHERE EXISTS (
       SELECT * FROM R1 A1
       WHERE A0.C0 = A1.C0 AND
             (sdo_geom.relate(A0.C1, 'equal', A1.C1,
                              <a spatial tolerance>) = 'equal') AND
             ... AND
             A0.CN = A1.CN AND
             A1.VTS_#$ < A0.VTS_#$ AND A1.VTE_#$ = A0.VTE_#$);
```

SQL Code 19: The mapping of *valid time* coalescing operation on relation *R1* with a spatial attribute, *C1*, into conventional statements.

In conclusion, it has been found that the changes required to the temporal algebra operators in order to accommodate the spatial support in temporal relations, during the mapping of these operators to snapshot-equivalent SQL, consist in supporting value comparison for spatial attributes. This operation is required to provide support for the comparison between the contents of two tuples, where a spatial attribute is present.

3.5 TEST AND RESULTS

Having settled on the spatio-temporal support provided by *Action!* and on the modifications that are required in the translation algorithm to accommodate spatial data management, tests have been performed. The extensive list of tests can be found in Annex B.

The tests are divided into groups, according to the temporal nature of the table that supports the data. In every group spatial data can be found in column *boundary*. The test scenario represents a modelled reality of the political evolution of four countries of Europe, namely Portugal, Poland, Czech Republic and Slovakia. The decision for a small set of data has only to do with a more easily controlled evaluation of results. The selection of these countries has to do with the fact that Portugal has kept almost the same border configuration across the centuries while the others countries have not. Moreover it is desirable to have countries that are physically separated in order to test spatial operators such as distance.

In the following sections several tests and their respective results are presented and discussed. For each of the tests we present the spatio-temporal SQL statement that is issued by an application, the spatio snapshot-equivalent SQL that results from the translation algorithm that is issued to Oracle and, where available, the results of the execution.

3.5.1 Spatio-temporal Support for Snapshot Relations

Test 1 creates a snapshot table, *countries2* with three columns: *name*, *capital* and *boundary*, where the boundary is of type geometry. This table provides the means to

represent data about political facts where the capital city of country *x* is *y* and the country has a border configuration *z*.

```
CREATE TABLE countries2 (name char(20), capital char(20), boundary
mdsys.sdo_geometry);
```

Snapshot equivalents

```
create table countries2 (name char(20), capital char(20), boundary
mdsys.sdo_geometry)
```

Test 1: The snapshot creation of table *countries2* with support for spatial data.

For Test 1, as no temporal support is required, the snapshot equivalent statement is the same original statement.

After the creation of the table, and since there is a spatial column, the corresponding metadata is inserted into the appropriate table *user_sdo_geom_metadata*. The statement in Test 2 specifies that the boundaries of all the geometry are within in a rectangle (0, 0, 1000, 1000) and that no coordinate system is being adopted for this column (NULL).

The difference to the statement being issued to the underlying STDBMS is that the statement contains the table schema for *user_sdo_geom_metadata* as *Action!* retrieves this metadata from the Oracle and uses it for realising the schema of a table in order to determine its temporal support or to resolve the projections used in *SELECT* *.

```
INSERT INTO user_sdo_geom_metadata VALUES('countries2','boundary',
MDSYS.SDO_DIM_ARRAY( MDSYS.SDO_DIM_ELEMENT('X', 0, 1000, 0.005),
MDSYS.SDO_DIM_ELEMENT('Y', 0, 1000, 0.005)), NULL);
```

Snapshot equivalents

```
insert into user_sdo_geom_metadata (TABLE_NAME, COLUMN_NAME,
DIMINFO, SRID) values ('countries2', 'boundary',
MDSYS.SDO_DIM_ARRAY(MDSYS.SDO_DIM_ELEMENT('X', 0, 1000, 0.005),
MDSYS.SDO_DIM_ELEMENT('Y', 0, 1000, 0.005)), NULL)
```

Test 2: The insertion of spatial metadata for column boundary of table *countries2*.

In order to benefit from full-fledged spatial support, it is required to create an index for the data in column *boundary* (Test 3). Indexes are required by several spatial operators in Oracle spatial.

```
CREATE INDEX countries_idx ON COUNTRIES2 (boundary) INDEXTYPE IS
mdsys.spatial_index;
```

Snapshot equivalents

```
create index countries_idx on COUNTRIES2(boundary) indextype is
mdsys.spatial_index
```

Test 3: The index creation for column *boundary*.

Table *countries2* can now be filled with data. Test 4 inserts some facts for country Portugal, stating that its capital city is Lisboa and that the border limits correspond to a

specific polygon (multi-polygons are also available). Since this is a standard spatial SQL performed over a snapshot relation the SQL being issued to Oracle is equal to the original SQL.

```
INSERT INTO countries2 VALUES ('Portugal', 'Lisboa',
MDSYS.SDO_GEOMETRY (2003, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY
(1,1003,1), MDSYS.SDO_ORDINATE_ARRAY(80.0,607.0, 92.0,601.0,
98.0,593.0, 89.0,592.0, 89.0,586.0, 76.0,583.0, 76.0,588.0,
58.0,587.0, 55.0,580.0, 42.0,588.0, 43.0,597.0, 46.0,613.0,
43.0,628.0, 38.0,644.0, 33.0,654.0, 30.0,662.0, 30.0,671.0,
37.0,677.0, 44.0,685.0, 43.0,697.0, 39.0,712.0, 50.0,710.0,
59.0,711.0, 71.0,706.0, 69.0,693.0, 77.0,682.0, 72.0,673.0,
78.0,663.0, 69.0,646.0, 81.0,645.0, 82.0,636.0, 77.0,630.0,
82.0,628.0, 83.0,616.0, 80.0,607.0)));
```

Snapshot equivalents

```
insert into countries2 (NAME, CAPITAL, BOUNDARY) values
('Portugal', 'Lisboa', MDSYS.SDO_GEOMETRY(2003, NULL, NULL,
MDSYS.SDO_ELEM_INFO_ARRAY(1, 1003, 1),
MDSYS.SDO_ORDINATE_ARRAY(80.0, 607.0, 92.0, 601.0, 98.0, 593.0,
89.0, 592.0, 89.0, 586.0, 76.0, 583.0, 76.0, 588.0, 58.0, 587.0,
55.0, 580.0, 42.0, 588.0, 43.0, 597.0, 46.0, 613.0, 43.0, 628.0,
38.0, 644.0, 33.0, 654.0, 30.0, 662.0, 30.0, 671.0, 37.0, 677.0,
44.0, 685.0, 43.0, 697.0, 39.0, 712.0, 50.0, 710.0, 59.0, 711.0,
71.0, 706.0, 69.0, 693.0, 77.0, 682.0, 72.0, 673.0, 78.0, 663.0,
69.0, 646.0, 81.0, 645.0, 82.0, 636.0, 77.0, 630.0, 82.0, 628.0,
83.0, 616.0, 80.0, 607.0))) --updates: 1
```

Test 4: The insertion of a tuple in table *countries2* with facts for country Portugal.

After the insertion of one tuple, a *select* statement can be issued which retrieves all the contents of *countries2* (Test 5). Being a snapshot table, conventional relational algebra operators are used.

```
SELECT * FROM countries2;
```

Snapshot equivalents

```
create table temp_relation_4 (NAME, CAPITAL, BOUNDARY) as select
relation_alias_1.NAME, relation_alias_1.CAPITAL,
relation_alias_1.BOUNDARY from countries2 relation_alias_1
```

NAMECAPITAL BOUNDARYROWID

Portugal Lisboa PolygonAAAG1nAABAAAKC6AAA

1 rows

Test 5: The retrieval of all the rows of table *countries2*.

As it can be seen in Test 5, the SQL issued to Oracle creates a temporary relation where the results of the query are stored. This relation remains on the system after the statement is issued and until an application closes the record set that retrieves information from it. Hence, in this statement and in the following statements the name *temp_relation_XX* is a table used to store intermediate results, during the evaluation of a query that, for instance, has inner queries.

In the sequel the term *Polygon* has been used in substitution of the term *oracle.sdoapi.geom.Polygon*.

The result of Test 5 consists of a single tuple where values are presented for column *name*, *capital* and *boundary* and *rowid*. Regarding the last column, it is automatically inserted in the temporary table that corresponds to the final results of a statement issued by an application to *Action!* The goal of this column is to provide the means to identify and distinguish every record in a record set (is used in the data retrieval from a record set, during the visualization of spatio-temporal data).

Action! also provides support for snapshot relations to be temporally promoted in *valid time* or in *transaction time*. The following Test 6 presents a temporal upgrade of table *countries2*, through non-sequenced *valid time* semantics. Here, all the records in *countries2* are temporally upgraded to *valid time* and for each row the *valid time* attribute is set to period [1999-2001).

```
nonsequenced validtime period [1999-2001) SELECT * from countries2;
Snapshot equivalents
create table temp_relation_6 (VTS_TIMEDB, VTE_TIMEDB, NAME,
CAPITAL, BOUNDARY) as select 915148800000 VTS_TIMEDB, 978307200000
VTE_TIMEDB, relation_alias_3.NAME, relation_alias_3.CAPITAL,
relation_alias_3.BOUNDARY from countries2 relation_alias_3
validtime NAME CAPITAL BOUNDARY ROWID
[1999-01-01 - 2001-01-01)Portugal Lisboa
oracle.sdoapi.geom.PolygonAAAG1xAABAAAKEKAAA
1 rows
```

Test 6: The temporal upgrade of snapshot table *countries2* using non-sequenced semantics.

By this mean, a snapshot table can be temporally upgraded and, for instance, participate as an inner query of a query that involves the temporal dimension for which the upgrade was performed. For instance, the statement in Test 7 starts by retrieving column *name* and *boundary* for snapshot table *countries2*. The result is aliased to *k* which is temporally upgraded in *valid time* for time period [1999-2001). After that, unitemporal coalescing of table *k* is performed as *k* provides *valid time* support.

```
validtime (nonsequenced validtime period [1999-2001) SELECT k.*
from (select name, boundary from countries2) k) (period);
```

Snapshot equivalents

```
create table temp_relation_22 (name, boundary) as
select name, boundary
from countries2 relation_alias_8

create table temp_relation_23 (VTS_TIMEDB, VTE_TIMEDB, name,
boundary) as
select 915148800000 VTS_TIMEDB, 978307200000 VTE_TIMEDB, k.name,
k.boundary
from temp_relation_22 k

create table temp_relation_24 (VTS_TIMEDB, VTE_TIMEDB, name,
boundary) as
select greatest(relation_alias_9.VTS_TIMEDB) as VTS_TIMEDB,
least(relation_alias_9.VTE_TIMEDB) as VTE_TIMEDB,
relation_alias_9.name, relation_alias_9.boundary
from temp_relation_23 relation_alias_9
where greatest(relation_alias_9.VTS_TIMEDB) <
      least(relation_alias_9.VTE_TIMEDB)

update temp_relation_24 a0 set a0.VTE_TIMEDB = (select
max(a1.VTE_TIMEDB) from temp_relation_24 a1 where a0.VTE_TIMEDB >=
a1.VTS_TIMEDB and a0.VTE_TIMEDB < a1.VTE_TIMEDB and a0.name =
a1.name and sdo_geom.relate(a0.boundary, 'equal', a1.boundary,
5.0E-4) = 'EQUAL') where exists (select * from temp_relation_24 a1
where a0.VTE_TIMEDB >= a1.VTS_TIMEDB and a0.VTE_TIMEDB <
a1.VTE_TIMEDB and a0.name = a1.name and
sdo_geom.relate(a0.boundary, 'equal', a1.boundary, 5.0E-4) =
'EQUAL')

delete from temp_relation_24 a0 where exists ( select * from
temp_relation_24 a1 where (a1.VTS_TIMEDB < a0.VTS_TIMEDB and
a1.VTE_TIMEDB = a0.VTE_TIMEDB) and a0.name = a1.name and
sdo_geom.relate(a0.boundary, 'equal', a1.boundary, 5.0E-4) =
'EQUAL')

drop table temp_relation_22
```

validtime name boundary ROWID

```
[1999-01-01 - 2001-01-01)Portugal
           oracle.sdoapi.geom.Polygon AAAG2UAABAAAKCyAAA

1 rows
```

Test 7: Table *k* is the temporal upgrade of a select performed over table *countries2*. After that, unitemporal coalescing is performed for table *k*.

3.5.2 Spatio-temporal Support for Valid Time Relations

Consider that all the rows in the previous snapshot table *countries2* have been deleted, the table has been permanently upgraded to *valid time* with statement *alter table countries2 add valid time* and that three rows have been inserted. The *valid time* statement presented in Test 8 retrieves all the rows, for all states, for table *countries2*. Since sequenced semantic is being used, due to the *valid time* modifier at the start of the statement, the algebra operators are the temporal select and temporal project. Consequently a translation of these operators takes place in *Action!* as seen in the SQL that is issued to Oracle. In this particular translation the spatial attribute *boundary* is being dealt as the other non-temporal attributes *name* and *capital*.

```
validtime select * from countries2;

Snapshot equivalents

create table temp_relation_45 (VTS_TIMEDB, VTE_TIMEDB, NAME,
CAPITAL, BOUNDARY) as select greatest(relation_alias_4.VTS_TIMEDB)
as VTS_TIMEDB, least(relation_alias_4.VTE_TIMEDB) as VTE_TIMEDB,
relation_alias_4.NAME, relation_alias_4.CAPITAL,
relation_alias_4.BOUNDARY
from countries2 relation_alias_4
where greatest(relation_alias_4.VTS_TIMEDB) <
      least(relation_alias_4.VTE_TIMEDB)

validtimeNAMECAPITALBOUNDARYROWID

[1900-01-01 - forever)      Portugal Funchal
oracle.sdoapi.geom.PolygonAAAG6nAABAAAKciAAA

[1500-01-01 - 1590-01-01)Portugal Lisboa
oracle.sdoapi.geom.PolygonAAAG6nAABAAAKciAAB

[1640-01-01 - 1900-01-01)Portugal Funchal
oracle.sdoapi.geom.PolygonAAAG6nAABAAAKciAAC


3 rows
```

Test 8: The contents of *valid time* table *countries2*.

Conventional and spatial SQL can also be issued for tables with *valid time* support. In this case the results of an SQL statement should be equivalent to the same statement issued over a snapshot table (temporal upward compatibility). The statement in Test 9 demonstrates this feature. Here unitemporal coalescing is being performed over *valid time* table *countries2* and after a snapshot select is being performed over the intermediate results.

```
Select * from countries2 (period);

Snapshot equivalents:

create table temp_relation_49 (VTS_TIMEDB, VTE_TIMEDB, NAME,
CAPITAL, BOUNDARY) as select VTS_TIMEDB, VTE_TIMEDB, NAME, CAPITAL,
BOUNDARY from countries2

update temp_relation_49 a0 set a0.VTE_TIMEDB = (select
max(a1.VTE_TIMEDB) from temp_relation_49 a1 where a0.VTE_TIMEDB >=
a1.VTS_TIMEDB and a0.VTE_TIMEDB < a1.VTE_TIMEDB and a0.NAME =
a1.NAME and a0.CAPITAL = a1.CAPITAL and
sdo_geom.relate(a0.BOUNDARY, 'equal', a1.BOUNDARY, 5.0E-4) =
'EQUAL') where exists (select * from temp_relation_49 a1 where
a0.VTE_TIMEDB >= a1.VTS_TIMEDB and a0.VTE_TIMEDB < a1.VTE_TIMEDB
and a0.NAME = a1.NAME and a0.CAPITAL = a1.CAPITAL and
sdo_geom.relate(a0.BOUNDARY, 'equal', a1.BOUNDARY, 5.0E-4) =
'EQUAL')

update temp_relation_49 a0 set a0.VTE_TIMEDB = (select
max(a1.VTE_TIMEDB) from temp_relation_49 a1 where a0.VTE_TIMEDB >=
a1.VTS_TIMEDB and a0.VTE_TIMEDB < a1.VTE_TIMEDB and a0.NAME =
a1.NAME and a0.CAPITAL = a1.CAPITAL and
sdo_geom.relate(a0.BOUNDARY, 'equal', a1.BOUNDARY, 5.0E-4) =
'EQUAL') where exists (select * from temp_relation_49 a1 where
a0.VTE_TIMEDB >= a1.VTS_TIMEDB and a0.VTE_TIMEDB < a1.VTE_TIMEDB
and a0.NAME = a1.NAME and a0.CAPITAL = a1.CAPITAL and
sdo_geom.relate(a0.BOUNDARY, 'equal', a1.BOUNDARY, 5.0E-4) =
'EQUAL')

delete from temp_relation_49 a0 where exists ( select * from
temp_relation_49 a1 where (a1.VTS_TIMEDB < a0.VTS_TIMEDB and
```

```
a1.VTE_TIMEDB = a0.VTE_TIMEDB) and a0.NAME = a1.NAME and a0.CAPITAL
= a1.CAPITAL and sdo_geom.relate(a0.BOUNDARY, 'equal', a1.BOUNDARY,
5.0E-4) = 'EQUAL')

create table temp_relation_50 (NAME, CAPITAL, BOUNDARY) as select
relation_alias_8.NAME, relation_alias_8.CAPITAL,
relation_alias_8.BOUNDARY from temp_relation_49 relation_alias_8
where relation_alias_8.VTS_TIMEDB <= 1152894352547 and
relation_alias_8.VTE_TIMEDB > 1152894352547 --updates: 0

drop table temp_relation_49

NAMECAPITAL  BOUNDARY                              ROWID

Portugal Funchal oracle.sdoapi.geom.PolygonAAAG7AAABAAAKfqAAA


1 rows
```

Test 9: The snapshot select statement performing over coalesced *valid time* table *countries2*.

The result of the unitemporal coalescing is a temporary table that contains the results presented in Test 10 where time periods have been maximized for non-temporal attributes having value equivalent data.

```
validtime                    NAME     CAPITALBOUNDARY
[1640-01-01 - forever)       Portugal Funchal  Polygon
[1500-01-01 - 1590-01-01)    Portugal Lisboa   Polygon
2 rows
```

Test 10: The results for the unitemporal coalescing of Test 9.

After the unitemporal coalescing has been performed the snapshot select is performed over the temporary table, meaning that only the rows that are considered to be current in the modelled reality are retrieved. Moreover, due to the non-temporal nature of the *select* statement, the *valid time* period for each of the resulting rows is omitted. Hence, the result is a single row which has been selected by the fact that its *valid time*, [1640-01-01 – forever) is considered current in the modelled reality. In Test 9 the evaluation of current rows has used the chronon *1152894352547* which represents the time instant the statement was evaluated.

A *valid time* table can also be temporally demoted with the use of the *alter table* statement, as in Test 11:

```
alter table countries2 drop validtime;
Snapshot equivalents
alter table countries2 drop (VTS_TIMEDB, VTE_TIMEDB)
```

Test 11: The temporal demotion of *valid time* table *countries2*.

Internally, the temporal demotion corresponds to the removal of the columns that support the corresponding temporal dimension which, in the previous test, are columns *VTS_TIMEDB* and *VTE_TIMEDB*. Consequently sequenced semantics is no longer available for this table

unless a temporal upgrade is performed as in Test 6. Considering the table contents in Test 8 the snapshot select statement performed over the now snapshot table *countries2* provides the results in Test 12.

```
select * from countries2;
Snapshot equivalents
create table temp_relation_54 (NAME, CAPITAL, BOUNDARY) as select
relation_alias_11.NAME, relation_alias_11.CAPITAL,
relation_alias_11.BOUNDARY from countries2 relation_alias_11 --
updates: 0
NAMECAPITAL BOUNDARYROWID
Portugal Funchal    oracle.sdoapi.geom.Polygon AAAG7UAABAAAKfCAAA
Portugal Lisboa     oracle.sdoapi.geom.Polygon AAAG7UAABAAAKfCAAB
Portugal Funchal    oracle.sdoapi.geom.Polygon AAAG7UAABAAAKfCAAC
3 rows
```

Test 12: The snapshot select performed over temporally demoted *countries2*.

In these results the non-temporal attributes are kept unaltered, but the *valid time* attributes have been deleted.

Despite the fact that spatial attributes have been managed until now, the previous tests have not taken considerable advantage of the spatial support provided by *Action!*, besides spatial data representation and storage. The following tests demonstrate how it is possible to take the full advantage of the spatial support provided by *Action!*. To such purpose, consider that a table, *countries4*, has been previously created with the same schema as *valid time* table *countries2* and that it contains the following rows:

VALIDTIME	COUNTRY	CAPITAL	BOUNDARY
[1025/01/01 - 1040/01/01)	Poland	Gniezno	Polygon
[1040/01/01 - 1150/01/01)	Poland	Cracow	Polygon
[1150/01/01 - 1250/01/01)	Poland	Cracow	Polygon
[1250/01/01 - 1350/01/01)	Poland	Cracow	Polygon
[1350/01/01 - 1450/01/01)	Poland	Cracow	Polygon
[1450/01/01 - 1596/01/01)	Poland	Cracow	Polygon
[1596/01/01 - 1650/01/01)	Poland	Warsaw	Polygon
[1650/01/01 - 1795/01/01)	Poland	Warsaw	Polygon
[1918/01/01 - 1939/01/01)	Poland	Warsaw	Polygon
[1945/01/01 - forever)	Poland	Warsaw	Polygon
[1198/01/01 - 1250/01/01)	Czech Kingdom	Prague	Polygon
[1250/01/01 - 1350/01/01)	Czech Kingdom	Prague	Polygon
[1350/01/01 - 1450/01/01)	Czech Kingdom	Prague	Polygon
[1450/01/01 - 1550/01/01)	Czech Kingdom	Prague	Polygon
[1550/01/01 - 1621/01/01)	Czech Kingdom	Prague	Polygon
[1918/01/01 - 1939/01/01)	Czechoslovakia	Prague	Polygon
[1945/01/01 - 1993/01/01)	Czechoslovakia	Prague	Polygon
[1993/01/01 - forever)	Czech Republic	Prague	Polygon
[1940/01/01 - 1945/01/01)	Slovakia	Bratislava	Polygon
[1993/01/01 - forever)	Slovakia	Bratislava	Polygon
[1143/01/01 - 1200/01/01)	Portugal	Guimaraes	Polygon
[1200/01/01 - 1250/01/01)	Portugal	Coimbra	Polygon
[1250/01/01 - 1590/01/01)	Portugal	Lisboa	Polygon
[1640/01/01 - 1900/01/01)	Portugal	Lisboa	Polygon
[1900/01/01 - forever)	Portugal	Lisboa	Polygon

Table 10: The contents of *valid time* table *countries4*.

Table 10 stores information about the spatio-temporal evolution of four countries, namely, Portugal, Poland, Czech Kingdom, Czechoslovakia, Czech Republic and Slovakia. In the represented reality, Poland started to exist in 1040 and still exists at present. However, during 1795 and until 1980 Poland did not exist. A similar situation occurred with Portugal, founded in 1143 and existing at present time, except for the period, 1590-1640. Czech Republic and Slovakia are two other countries existing at the present time, which resulted from the separation of Czechoslovakia in 1993. Czechoslovakia is a country emerging in the XX century, more specifically in 1918. However, centuries before, a Czech Kingdom had existed, from 1198 and until 1621.

The reality just described corresponds to a simplified approximation of the History of Europe for these countries. Spatio-temporal data results from the digitalisations of maps that, for a precise date, depicted the political borders of Europe. Alphanumerical information has been entered according to the map contents. The use of this domain for testing the spatial support provided by the STDBMS has to do with the familiarity that readers may have in interpreting political maps.

Figure 4 represents the geographic data corresponding to Table 10, where for every state and regarding each of the countries represented in *countries4*, the political borders have been drawn. In Figure 4 it is possible to see that Portugal had stable political borders. Other countries, Poland, Czech Kingdom, Czechoslovakia, Czech Republic and Slovakia, have developed quite differently. For instance, Poland's political borders have been greatly enlarged and have also been greatly reduced across de centuries.



Figure 4: A graphical representation of the contents of Table 10.

Using this data Test 13 inquires the current distance between Portugal and Poland. This snapshot statement performed over a table with *valid time* support is a demonstration of the upward compatibility capability of *Action!*.

```
SELECT a.name, b.name, SDO_GEOM.SDO_DISTANCE(a.boundary,
b.boundary, 0.005) FROM countries4 a, countries4 b WHERE a.name =
'Portugal' AND b.name = 'Poland';
```

Snapshot equivalents

```
create table temp_relation_4 (name, name1, attrib_alias_0) as
select a.name, b.name, SDO_GEOM.SDO_DISTANCE(a.boundary,
b.boundary, 0.005) from countries4 a, countries4 b where
a.VTS_TIMEDB <= 1152891813125 and a.VTE_TIMEDB > 1152891813125 and
b.VTS_TIMEDB <= 1152891813125 and b.VTE_TIMEDB > 1152891813125 and
a.name = 'Portugal' AND b.name = 'Poland'

namename attrib_alias_0ROWID

Portugal Poland 493.42577962648AAAG0VAABAAAKHKAAA


1 rows
```

Test 13: The snapshot select performed over *countries4*.
Question is: "what is the current distance between Portugal and Poland?"

Action! also provides the means to issue a similar question that takes into account, not only the present spatial configuration of the political borders of Portugal and Poland, but all the previous states, concerning these two countries. In order to do so, a sequenced semantic query is required and for this matter, the select statement requires that the *valid time* modifier is placed at the start of the statement, as shown in Test 14:

```
validtime SELECT a.name nome1, b.name nome2,
SDO_GEOM.SDO_DISTANCE(a.boundary, b.boundary, 0.005)
FROM countries4 a, countries4 b
WHERE a.name = 'Portugal' AND b.name = 'Poland';
```

Snapshot equivalents

```
create table temp_relation_5 (VTS_TIMEDB, VTE_TIMEDB, nome1, nome2,
attrib_alias_1) as select greatest(a.VTS_TIMEDB, b.VTS_TIMEDB) as
VTS_TIMEDB, least(a.VTE_TIMEDB, b.VTE_TIMEDB) as VTE_TIMEDB, a.name
nome1, b.name nome2, SDO_GEOM.SDO_DISTANCE(a.boundary, b.boundary,
0.005) from countries4 a, countries4 b where greatest(a.VTS_TIMEDB,
b.VTS_TIMEDB) < least(a.VTE_TIMEDB, b.VTE_TIMEDB) and a.name =
'Portugal' AND b.name = 'Poland'
```

validtimenome1nome2attrib_alias_1ROWID

```
[1143/01/01 - 1150/01/01) Portugal  Poland  505.466121515577
AAAWR4AAFAAAyGkAAA

[1150/01/01 - 1200/01/01) Portugal  Poland  509.417314193383
AAAWR4AAFAAAyGkAAB

[1200/01/01 - 1250/01/01) Portugal  Poland  508.205667028616
AAAWR4AAFAAAyGkAAC

[1250/01/01 - 1350/01/01) Portugal  Poland  526.459875014231
AAAWR4AAFAAAyGkAAD

[1350/01/01 - 1450/01/01) Portugal  Poland  525.380814267137
AAAWR4AAFAAAyGkAAE

[1450/01/01 - 1590/01/01) Portugal  Poland  524.604613018224
AAAWR4AAFAAAyGkAAF

[1640/01/01 - 1650/01/01) Portugal  Poland  525.457895553963
AAAWR4AAFAAAyGkAAG

[1650/01/01 - 1795/01/01) Portugal  Poland  522.820236792724
AAAWR4AAFAAAyGkAAH
```

```
[1918/01/01 - 1939/01/01) Portugal  Poland  485.362750939954
AAAWR4AAFAAAyGkAAI
[1945/01/01 - forever)    Portugal  Poland  493.42577962648
AAAWR4AAFAAAyGkAAJ
10 rows
```

Test 14: A sequenced semantics select statement asking: "across the time, what has been the current distance between Portugal and Poland?"

In this test it is possible to see that all the states of the two countries are intersected and when a common time period exists, the distance between the two countries for that time period is calculated. For instance, the first row provides the distance between these two countries for the time period [1143/01/01 - 1150/01/01), which is the intersection of time period [1040/01/01 - 1150/01/01) for Poland with time Period [1143/01/01 - 1200/01/01) for Portugal. The second row in the results corresponds to the use of the second state for Poland [1150/01/01 - 1250/01/01) with the first state for Portugal, resulting in a concrete distance value for time period [1150/01/01 - 1200/01/01).

Action! is also able to perform questions where all the temporal states are used to produce a result which is based on specific spatial criteria. Test 15 uses conventional relational operators (selection, projection) to go through all the states of polish history and retrieve the row where spatial function *sdo_geom.sdo_area* retrieves the largest area, concerning the political borders.

```
nonsequenced validtime
select max(sdo_geom.sdo_area(c.boundary, 0.005)) maiorFronteira
from countries4 c
where c.name = 'Poland';
Snapshot equivalents
create table temp_relation_8 (maiorFronteira) as select
max(sdo_geom.sdo_area(c.boundary, 0.005)) from countries4 c where
c.name = 'Poland'
maiorFronteira ROWID
68186 AAAG0YAABAAAKHqAAA
1 rows
```

Test 15: The non-sequenced select statement performed over *countries4* for the largest area for polish borders across time.

In the previous test, the snapshot-equivalent statements issued to the underlying database resemble conventional SQL. Consider that the snapshot-equivalent SQL in Test 15 is issued to *Action!* instead of Oracle. In this situation a translation takes place as this is a snapshot select statement that is being executed for a table with *valid time* support. Hence the translation ensures that only the current state should be accounted for, during the execution, as seen in Test 16.

```
select max(sdo_geom.sdo_area(c.boundary, 0.005)) from countries4 c
where c.name = 'Poland'
```

Snapshot equivalents

```
create table temp_relation_9 (attrib_alias_1) as select
max(sdo_geom.sdo_area(c.boundary, 0.005)) from countries4 c where
c.VTS_TIMEDB <= 1222342013708 and c.VTE_TIMEDB > 1222342013708 and
c.name = 'Poland'

attrib_alias_1 ROWID

26028.5 AAAWR7AAFAAAyGsAAA

1 rows
```

Test 16: A snapshot select statement performed over *valid time* table *countries4*. Query deals with only the current state of data.

The statement in Test 16 retrieves the largest area for Poland, but regarding only the current state of facts. Since this state contains only one row for the representation of Poland, its area is returned as the maximum area of polish borders.

Spatial aggregate types and functions are also provided by *Action!* in the context of spatio-temporal data management. If spatial aggregate functions are used in sequenced semantic queries, the spatial function is performed for each temporal state identified in the query execution. For example, the sequenced semantics query presented in SQL Code 20 retrieves the temporal evolution of the spatial union for the political borders of all the countries represented in *countries4*, during time period [1100-1350):

```
validtime period [1100-1350)
select sdo_aggr_union(mdsys.sdoaggrtype(c.boundary, 0.005))
from countries4 c;
```

SQL Code 20: A spatio-temporal sequenced semantics query.

Figure 5 displays the rows involved in the query execution as horizontal lines where dots correspond to the time instants that delimit each row time period. Vertical lines correspond to time instants that identify a state for which data concerning a country has changed.

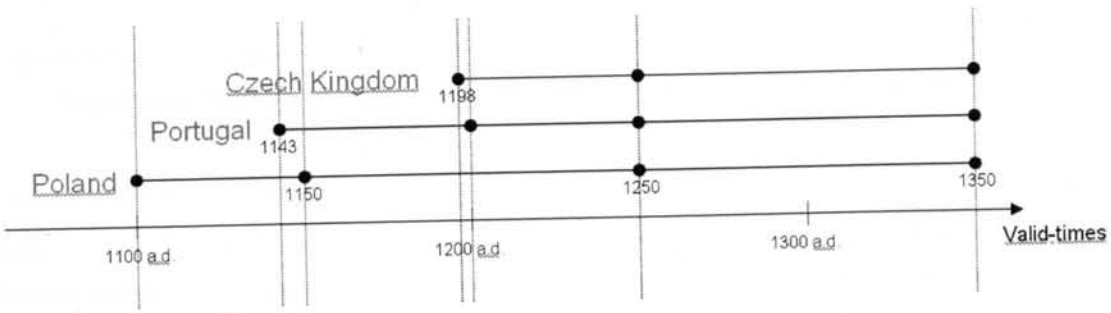


Figure 5: The database facts regarding Poland, Portugal, and the Czech Kingdom.

The following Figure 5 depicts the results of the statement in SQL Code 20, with the six states present in the statement result. Besides the first state, from 1100 to 1143, each of the

remaining states contains a multi-polygon geometry which is the spatial union of the countries present in that state.

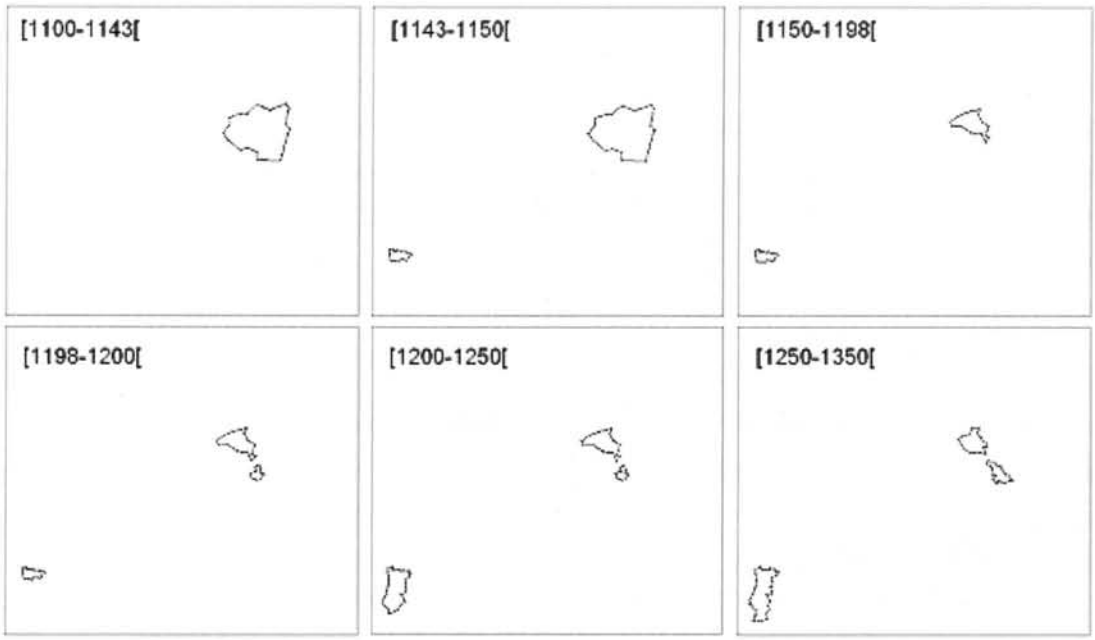


Figure 6: The spatio-temporal results from the sequenced *valid time* spatial union query.

A similar query can be performed to retrieve the spatial union off all the political boundaries for countries existing during [1100-1350). This query, presented in SQL Code 21, uses non-sequenced semantics instead of sequenced semantics:

```
nonsequenced validtime period [1100-1350)
select sdo_aggr_union(mdsys.sdoaggrtype(c.boundary, 0.005))
from countries c;
```

SQL Code 21: spatio-temporal non-sequenced semantics query.

The result from the previous non-sequenced query is a temporally time stamped row, with time period [1100-1350) containing a multi-polygon geometry that corresponds to spatial union of all the political boundaries represented in *countries4* for time period [1100-1350) (Figure 7).

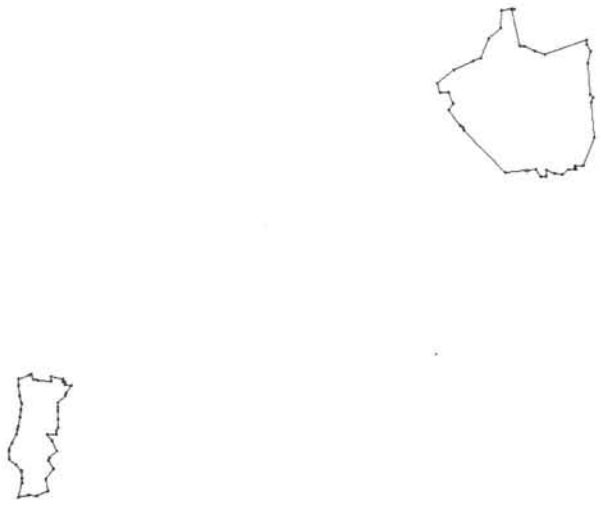


Figure 7: Spatio-temporal results from the non-sequenced *valid time* spatial union query.

3.6 CONCLUSIONS

This chapter has presented an approach for spatio-temporal data management that, from the perspective of the data representation scheme, is based on Snodgrass tuple time stamped representation scheme, and from the perspective of the query language corresponds to the integration of spatial support provided by Oracle Spatial within ATSQL2. Through the proposed SQL language the resulting spatio-temporal data management is more than the mere sum of spatial plus temporal support: it results in a seamless integration where spatial selections and projections can be integrated into temporal sequenced and temporal non-sequenced semantics defined in ATSQL2. Moreover the resulting query language maintains upward compatibility regarding Oracle Spatial and ATSQL2.

A prototype STDBMS, *Action!*, has been researched and developed based on an architectural approach that combines the layered approach and the extensible approach. Consequently, *Action!* is divided into two components. On the client-side part an application can connect through an API and issue spatio-temporal SQL statements. The client is responsible for the management and translation of each statement into a set of snapshot-equivalent statements, which use conventional relational operators, and which can be successfully processed and executed by Oracle and its spatial extension. The server-side part of *Action!* is supported on Oracle Spatial and it is responsible for executing the statements delivered by the client-side part and for returning the corresponding results.

At the level of translating spatio-temporal SQL statements nine distinct temporal scenarios have been identified, considering the temporal compatibility between the statements and the targeted tables. Each of these scenarios requires an appropriate processing of the SQL statement in order to map the temporal semantics of the statement into snapshot-equivalent statements that are issued to Oracle.

Regarding the spatial support provided in the proposed solution, it has been determined that it is managed orthogonally to the temporal support, during the translation process: the spatial support provided concerns the ability to use spatial attributes or results from spatial functions or operators and the ability to constrain selections through spatial functions and operators in the context of spatio-temporal queries or when dealing with spatio-temporal relations. Regarding snapshot-equivalent mappings, the spatial attributes, functions and operators are managed as any other non-temporal attribute, function and operator. Hence, each snapshot-equivalent SQL statement resulting from translation is embedded with references to spatial attributes, spatial functions and operators. These conventional-spatial SQL statements are then issued to the database server and are successfully processed and executed by the spatial extension of Oracle.

In the development of the *Action!* prototype the translation algorithm proposed by Steiner has been extended to provide management for spatial tables, spatial indexes, spatial data types and spatial functions and operators. Moreover, all the operators of the temporal relational algebra were modified in order to provide appropriate value-comparison between spatial attributes.

The correctness of the spatio-temporal features provided by *Action!* has been tested. The tests have been performed based on a small data set that was developed to illustrate the spatio-temporal features of *Action!*. Tests were separated into two groups:

- spatio-temporal support for snapshot relations, which intended to demonstrate upward compatibility of the system and the ability to temporally upgrade snapshot relations. In each of these tests spatial requirements are present in order to demonstrate the spatial support seamless integrated with temporal support.

- spatio-temporal support for *valid time* relations, which intends to demonstrate temporal upward compatibility of the system and the ability to use sequenced and non-sequenced semantics mingled with spatial requirements.

Based on the spatio-temporal data management features presented in this chapter it is possible to say that the integration of Oracle Spatial with ATSQL2 can be considered suitable for playing the role of storage and retrieval of data, evolving discretely over time and space, in a geographic information domain for which the need to manage the temporal aspects of data is starting to emerge. In fact, by enhancing a spatially-enabled conventional database management system with temporal support as it has been described over this chapter, a STDBMS can be achieved that provides spatial data management for valid time and for transaction time and preserves important features such as upward compatibility, temporal upward compatibility, sequenced semantics and non sequenced-semantics.

More spatio-temporal data management features could be available if additional features defined in SQL3 are included, concerning assertion definition, cursor expressions and fetch

statements. In order to do this further study is required on the use of these features along with the already provided spatio-temporal data management features. It is believed that the development of these and other spatio-temporal features may require that spatio-temporal support is totally developed on the server-side.

Considering the wider context that underlies this research for a STDBMS, it is possible to state that *Action!* also constitutes an appropriate tool to support the research on time-dependent information visualization. Through the proposed spatio-temporal query language the user is able to issue a wide variety of spatio-temporal queries that provide the working grounds for researching on spatio-temporal information visualization. This part of the work is addressed later, in Chapter 5.

However, before moving onto the visualization of time-dependent data, retrieved from a STDBMS, an introduction on the research in Information Visualization is required. Hence, the next chapter presents the most relevant work performed in Information Visualization, going in depth on the visualization of time-dependent data.

4 INFORMATION VISUALIZATION

The work presented on the previous chapter, concerned a contribution for a spatio-temporal database management system named *Action!* which allows the efficient representation, management and query of spatio-temporal data sets and the retrieval of record sets that comprise the temporal evolution regarding a set of entities. The availability of efficient spatio-temporal data management poses challenging problems in what concerns the visualization and exploratory data analysis of spatio-temporal record sets. For instance, ambiguities or difficulties of interpretation may arise in the visual representation onto the same display result of two distinct states about an entity that has occupied the same location in distinct intervals of time.

In order to provide solutions for the visualization of spatio-temporal data sets, the current chapter surveys the interdisciplinary research area of Information Visualization, where a considerable number of visual representation techniques have been proposed, to help people observe, understand, explore and analyse large-scale collections of abstract data.

Card *et al.* regard information visualization as "*...the use of interactive visual representations of data to amplify cognition. This means that the data is transformed into an image; it is mapped to screen space. The image can be changed by users as they proceed working with it. This interaction is important as it allows for constant redefinition of goals when new insight into the data has been gained.*" (Stuart K. Card 1999). For these authors, through the use of 2D and 3D color graphics and animation, visualization can be an efficient mean to display the structure of information, allowing users to navigate through the structure and to interact with the information (Stuart K. Card 1999).

In describing information visualization, Plaisant focuses on compactness, meaning the ability to graphically compact large amounts of information in such a manner that allows the observer to more effectively discover, make decisions or provide explanations about patterns, individual or groups of information items (Plaisant 2004).

In fact, information visualization is a very wide research field encompassing a heterogeneous set of domains from human activity. In the last decades, information visualization has been addressed by the computer science research community, but it had already been the working grounds for psychological, semiotic and social studies related with the understanding of the human visual system, the perception of signs and how cultural characteristics from distinct social environments change the way information is perceived. Therefore, it expected that authors outside of computer science and technological domains have performed wide and in-depth studies on information visualization (Bertin 1967; Tufte 2001), having several of these studies been the working grounds for subsequent computer science research.

Successful visualization techniques can reduce the time it takes to understand the underlying data, to find relationships, and to get the information searched. Information visualisation seeks to present information visually, in essence to offload cognitive work to the human visual perception system, taking advantage of the human eye's broad bandwidth pathway into the human brain.

According to Treisman, one special property of our visual system is the pre-attentive (that is, without the need for focused attention) processing of visual information, which is performed automatically on the entire visual field detecting basic features (patterns, colors, closure, line ends, contrast, tilt, curvature and size) of objects in the display. Treisman states that these simple features are extracted from the visual display in the pre-attentive system and later joined in the focused attention system into coherent objects (Treisman 1985).

Taking advantage of pre-attentive processing, information visualization can greatly improve intuitiveness of representations yielding in a faster and more natural way of acquiring information (Kosara, Miksch et al. 2002; Perera, Goodman et al. 2007). Healey suggests that, typically, tasks that can be performed on large multi-element displays in less than 200 to 250 milliseconds are considered pre-attentive (Healey, Booth et al. 1993). According to Healey this is because eye movements take at least 200 milliseconds to initiate, but it also means that the pre-attentive system can provide visual information results in a single glimpse.

Hence, while machines excel at simulations, data filtering, and data reduction, humans are experts at using their highly developed pattern-recognition skills to look through the results for regions of interest, features, and anomalies. Compared to computer programs, humans are especially good in seeing unexpected and unanticipated emergent properties (van Dam 2000).

Information visualization is also seen as the visualization of *abstract data* (structured and unstructured), this meaning data that has no inherent mapping to space (Stuart K. Card 1999; Voigt 2002). According to Card, the process is not straightforward, since one has to find a good way to map data values to screen space. According to Robertson, information access and management is difficult in large information spaces, because it is hard to visualize what is there and how the various parts are related (Robertson, Mackinlay et al. 1991).

Card *et al.* also state that information visualization should be seen in contrast to scientific visualization, since the second deals with physically-based data (Stuart K. Card 1999). According to these authors, in scientific visualization, data is defined in reference to space coordinates, which makes it relatively easy to visualize it in an intuitive way. Card *et al.* suggest geographic data and computer tomography data of a body as two examples of scientific data. Such data can result from real-world observations and from computer simulations. Tory and Möller share the same opinion about the nature of scientific data when they state that it involves an inherent physical component (Tory and Möller 2004).

Despite the fact that information can be of abstract or of physically inherent nature, the fact is that information visualization concerns the process of mapping data onto a visual metaphor. According to Tufte, this mapping process becomes more effective if it is able to communicate complex ideas in a clear, precise and efficient manner, by complying with the following requisites (Tufte 2001):

- to present the information undistorted regarding its original meaning;
- to induce the observer to focus on the presented contents, instead of on the methodology, the design or the technology used to visualize the information;
- to keep coherency when presenting great amounts of information;
- to encourage or make possible for the observer to compare distinct parts of the presented information;
- to present the information through mechanisms which facilitate the visualization at progressively distinct levels of detail;
- to be well integrated with other descriptions of the information set, such as verbal or statistical.

Still regarding the visualization mapping process, Maletic *et al.* (Maletic, Marcus et al. 2002) consider that five dimensions of interest should be addressed, namely:

- Tasks – why is the visualization needed?
- Audience – who will use the visualization?
- Target – what is the data source to represent?
- Representation – how to represent it?
- Medium – where to represent the visualization?

Mackinlay also addressed the effectiveness of mapping data to a visual metaphor. For such purpose, this author defined two criteria, namely, expressiveness and effectiveness (Mackinlay 1986). The first, expressiveness, refers to the capability of the metaphor to visually represent all the information it is desired to visualize. The second criteria relates to how good is the visual metaphor as a way of representing such information. These criteria were used in 2D mappings, but can also be applied for 3D mappings. Marcus *et al.* consider that a solid data characterization is required in order to satisfy the expressiveness and effectiveness criteria for mapping data to a visual metaphor, where the goal is to maximize the amount of data that can be represented with an accent on the user's information seeking goals (Marcus, Feng et al. 2003). This author also states that the power of a visualization representation is also derived from its semantic richness, simplicity, and level of abstraction.

4.1 THE VISUALIZATION PIPELINE

In information visualization, the visualization pipeline describes the step-wise process of creating visual representations of data. A traditional perspective of this pipeline is to regard it as having three stages: pre-processing, mapping and rendering (Tory, Campo et al. 1997). The first stage regards operations such as putting the raw data in a specific format, applying a smoothing filter, interpolating missing values, or correcting erroneous measurements. The mapping stage comprises the transformation of the prepared data from the previous stage into the visualization space. This transformation should not only allow the representation of spatial relationships between data items as intuitive as possible but also to aid the observer in understanding those relationships. According to Mackinlay this is the most critical step for achieving expressiveness and effectiveness (Mackinlay 1986), where:

- expressiveness determines whether a graphical language can express the desired information;
- effectiveness determines whether a graphical language exploits the capabilities of the output medium and the human visual system.

The third stage considers the rendering of the mapped information, resulting in a visual presentation (image data).

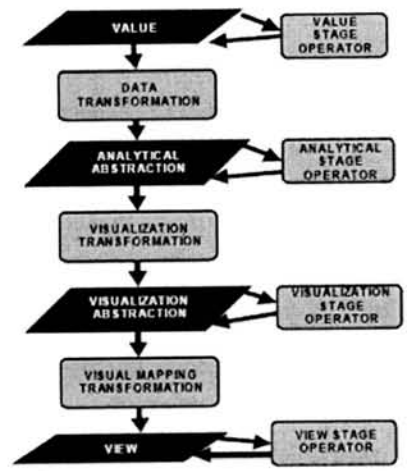


Figure 8: Chi's Information Visualization Data State Reference Model.

Other authors have distinct perspectives of the visualization pipeline. For instance, Chi (Chi 2000) considers 4 data stages (see Figure 8) - value, analytical abstraction, visualization abstraction, and view - and 3 process stages - data transformation, visualization transformation and visual mapping transformation - where:

- data transformation: generates some form of analytical abstraction from the value (usually by extraction);

- visualization transformation: takes an analytical abstraction and further reduces it into some form of visualization abstraction, which is a visualizable content;
- visual mapping transformation: takes information that is in a visualizable format and presents a graphical view.

Santos and Brodlie (Santos and Brodlie 2004) introduce an extended data flow model to accommodate the visualization process (see Figure 9). In contrast to the common visualization pipeline, these authors replace the pre-processing stage by two separate components distinguishing between data analysis and filtering.

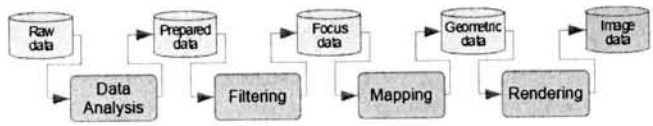


Figure 9: Santos and Brodlie Information Visualization Pipeline.

The data analysis stage is computer-centered and provides some computational methods to analyze or enrich the data. Typical functions for this purpose are interpolation functions, Principle component Analysis[4] or Multidimensional Scaling (Santos and Brodlie 2004). According to these authors, the filtering-component is user-centered, and provides different functionality for selecting data of interest.

According to Campo (Campo, Orosco et al. 1997), several authors have identified other stages in the information visualization process, such as the abstraction stage, prior to the mapping stage, where semantic filtering is performed; the interaction stage, mingled with the rendering stage, which addresses the user interaction processes and how the information visualization should maintain consistency, and finally the visualization state which records the user made operations and the current state of visualization.

Carmo and Cunha (Carmo and Cunha 1998) suggest that the filtering step can reduce the information amount, but also can generate new information by building new variables that result from functions applied to previously existing variables. These authors also state that filtering operations can happen in distinct steps of the visualization process. For instance, the most relevant filtering process will be performed at the pre-process stage, through user-described semantic criteria defined through functions, variable sets, domain restriction, etc. However, filtering may also occur in the mapping stage. For instance, depending on the chosen representation, some visual elements representing information can have an unperceived representation if they are too small. Finally, filtering may also occur at the rendering stage where the rendered results may be conditioned by the physical properties of

[4] The Principal Component Analysis (PCA) is a technique commonly applied to reduce the number of variables and to detect structures in the relationships between variables, in multi-dimensional data sets.

the output device, such as the display resolution, and the consequent limitation of the density of information. To this purpose, Frank and Timpf suggest that the control of information density can be performed through the "amount of ink", meaning the ratio between drawn pixels/total empty pixels (Frank and Timpf 1994). Through this solution, distinct display resolutions will provide distinct results.

Finally, filtering operations are also present at the interaction stage, and can have repercussions on all other previous stages of the visualization pipeline, namely:

- by controlling the amount of rendered information, through user observation of the resulting image, stopping the rendering process when the density of represented information starts to become unperceivable. This technique is more efficient if the rendering of information elements considers a decreasing sorting of information relevance;
- by switching the adopted representation model that is responsible for the mapping stage or influencing it through the modification in the data selection criteria that are at the base of the process;
- by loading a new dataset or making slight modifications to the dataset.

The overall aim of filtering operations over several stages of the visualization pipeline is to allow a more manageable representation of large information sets. Regarding the mapping stage, Leung rand Apperley realize that one of the problems associated with large computer-based information systems is the relatively small window through which an information space can be viewed. Common problems regard how to locate a given item of information, how to interpret the item and how to relate it with the other items, if the first cannot be seen in its full context (Leung and Apperley 1994).

The following section addresses the problem of how to provide manageable representations of large information sets and to allow the user to have detailed information of a particular item while keeping available a representation of the informational context.

4.2 Focus + Context Techniques

Distorted views rely on a principle of information visualization called *focus + context* (F+C, also called Detail-in-Context or Detail-and-Context). Here the most important data, at the focal point, is displayed at full size and detail, or through other visual characterization that expresses maximum interest from the user. The area around the focal point - the context - helps make sense of how the important information relates to the entire data structure. Yet, context regions far from the focal point may be displayed with smaller degree of interest or even selectively omitted.

According to Card (Card, Mackinlay et al. 1999), F+C start from three premises: first, the user needs both overview (context) and detailed information (focus) simultaneously. Second,

information needed in the overview may be different from the information needed in the detail. Third, these two kinds of information can be combined into a single (dynamic) display, much as in human vision. Therefore, F+C techniques provide the ability to have the information of interest in the foreground and all the remaining information in the background simultaneously visible or, as several authors metaphorically say: *"seeing a tree without missing the forest"*.

Keahey (Keahey 2003) suggests the basic idea about F+C techniques is to show selected regions in greater detail (focus), preserve global view at reduced detail (context) and avoid occlusion of information (all information is visible simultaneously). Kosara *et al.* share the same opinion, by stating that it is the goal of F+C techniques to point out relevant information items to the user, but also to provide a quick understanding of these items in the context of more information items (Kosara, Miksch et al. 2002).

Hence, F+C techniques provide an effective mean of doing micro/macro readings on information. According to Tufte, micro/macro designs for graphically presented information can be adapted to every process of information visualization, including topographical views and *landscape panoramas* (Tufte 2001). Such designs allow the presentation of great amounts of complex information, with high levels of detail and organization, through multiple and frequently hierarchical levels of contextual readings.

There are several ways of displaying information by taking advantage of F+C. For instance, the way humans observe the world is empirically a F+C technique, where objects in the foreground are naturally focused. The human visual system is also keen in focusing at a specific distance of observation (depth of field) while the surroundings, at different distances, are seen unfocused. Another way to achieve F+C is to display information by using an overview-plus-detail presentation, which rely on two windows, where the first shows a rough overview of the complete information space and neglects details, while the second window shows a small portion of the information space and visualizes details. The overview contains a rectangle that can be moved and sized and its contents are showed in the detail view. Both images can be shown either sequentially or in simultaneously.

Other F+C techniques rely on distorting the visualization of information by displaying information items distinctively, according to some criteria of interest to the observer. One of the first techniques applying this F+C principle and using digital technology, was suggested by *Furnas* and designated by *The Fisheye Concept* (Furnas 1986). This technique is motivated by the idea of a mental map that the user creates, according to subjective perception of the information being displayed.

For instance, the "Parochial New Yorker's view of the World", drawn by Steinberg in March 29, 1976, and presented in Figure 10, illustrates how new yorkers see the United States of America from their perspective where, according to their subjective perceptions, things get less interest the more far they are from New York. In Steinberg's drawing the degree of interest is visually mapped in terms of level of detail.



Figure 10: Steinberg's Parochial New Yorker's view of the World.

In the technique presented by Furnas, aimed at the display of large information structures, the basic strategy resides on a degree of interest (DOI) function that assigns a number to each point in the structure, expressing how much the user is interested in viewing that point. For each point in the structure, this function considers its *a priori* importance and the distance from the point to the current observation point. Filtering takes place when the user defines a numerical threshold (semantic user defined criteria) from which points will be rendered. No suggestion is made, concerning the DOI results influencing the representation model, besides showing/hiding information, at the mapping stage. Therefore this technique resides mainly at the pre-processing stage.

Sarkar and Brown (Sarkar and Brown 1992) have extended Furnas work by addressing the mapping stage in more detail. They provide distinct graphical interpretations to fisheye views by introducing layout considerations into the fisheye formalism. Here, the position, the size, and the level of detail of an object being displayed depend on user-specified functions that work with the object distance to the observation point. These authors have explored a fisheye lens approach for viewing and browsing graphs, showing a quite large area of interest with detailed information, while the remain of the graph is successively smaller and represented with less detail.

Bartram *et al.* (Bartram, Ho et al. 1995) considers distorted-view approaches, such as the Fisheye views, to have great potential to overcome the problem of viewing large information sets on small screens, but identifies several problems regarding focus, transition between views and user disorientation from excessive distortion. In order to overcome these and other limitations Bartram *et al.* propose a constrained *fisheye* technique for viewing and navigating in large information spaces called *continuous zoom.* According to the authors, it allows the display of detail and context in a flexible, timely and effective manner, where the main goals are:

- The use of multiple detailed views concurrently;
- Information, whether in a detailed view or summary view should never be occluded and therefore overlapping is not appropriate;
- Spatial distortion is minimized where possible by maintaining relative locations between information nodes to reduce user disorientation;
- Support efficient display reconfiguration with a minimum of required user interaction, but leave the user in full control at all times.
- Transition between views should be smooth;

To Bartram *et al.* F+C falls into three classes, namely:

- traditional pan/zoom techniques, which uniformly scale the entire space and allow scrolling but suffer from the keyhole problem (the inability of seeing all the word in high detail simultaneously or the limitation of looking at the world through a keyhole);
- multiple window techniques, which suffer from the extra space required to show the overview and from the user effort to mentally integrate detail and context;
- distorted view methods, which reconfigure the display *in situ* to emphasize certain details within a larger context, but can introduce issues of focus, transition between views and user disorientation from excessive distortion;

Kosara *et al.* (Kosara, Miksch et al. 2002) suggest another F+C technique based on the optical effect of *Depth of Field* (DOF). In photography, for instance, the DOF determines which depth range is used for focusing: objects at the DOF range from the camera are depicted sharply while objects progressively closer or farther from the DOF distance are increasingly blurred. In the context of information visualization, the authors define *semantic depth of field* as the process that blurs an object based on its current relevance to a particular visualization, relying on the principle that unblurred objects receive the viewer's attention.

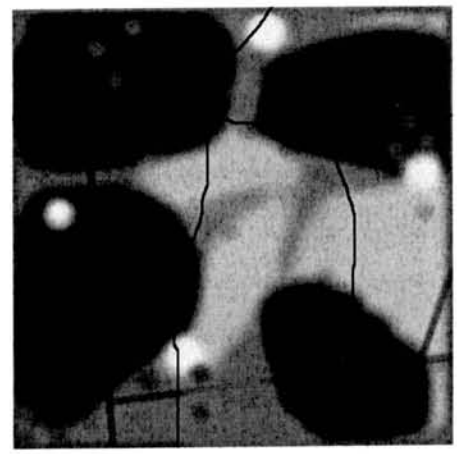


Figure 11: Map view showing rivers in focus and the context as blurred objects.

This principle is based on the fact that blurring an object removes the high spatial frequencies and reduces the contrast. Hence, the human pre-attentive processing system extracts less information from its representation and attention is directed to unblurred objects (see Figure 11).

4.3 MULTIPLE REPRESENTATIONS

The degree of interest calculated for an information item can be visually expressed with a wide set of visual clues, ranging from the simplest ones, based on scale or color, to more complex ones, based on level of detail, where higher degrees of interest should correspond to higher levels of detail.

In order to accommodate distinct levels of detail some F+C techniques have relied on multiple representations for each information item, where each distinct representation is targeted for a particular range of the user's degree of interest. Moreover, all the representation instances can be of a single nature, for example graphical, or of distinct natures. One possibility could be the use of a highly detailed vector data representation, while the corresponding intermediate detailed representation could be raster iconic and the low detailed representation could be textual, such as a linguistic tag.

The visualization technique for abstract information, presented by Fairchild (Fairchild 1993), corresponds to the use of multiple representations, of the same nature, for representing an information item through a set of icons. For each information item, the generation process goes through the identification of a set of graphical properties, based on its attributes. These attributes are used to produce a vector of icons that contains distinct item representations, considering increasing levels of interest. A DOI function is then applied considering the *a priori* item importance and the distance to the information item in focus (in a way similar to the *fisheye* views approach), being the result an index to the above vector.

This concept of automated representation for information visualization is also adopted by Frank (Frank and Timpf 1994) in the multi-scale tree where, for each cartographic object, multiple cartographic representations are automatically generated and used according to the scale of visualization.

On the other hand, Stonebraker (Stonebraker, Chen et al. 1993) relies on a set of multiple representations, with distinct natures, for an information item presented in the Tioga system.

Bartram's continuous zoom technique (Bartram, Ho et al. 1995), also considers multiple representations of distinct nature. In this approach, a cluster may show iconic representations or pictorial representations, according to the current degree of interest of the information item.

4.4 Information Visualization Taxonomies

In the 1990's, research on information visualization had reached the state where a number of successful designs and techniques had been proposed. In order to understand the differences among the designs, and to suggest new possibilities, several authors proposed taxonomies that enable the classification of information visualization techniques. The primary working grounds for most of the early taxonomies was based on Bertin's comprehensive work performed on the semiology of graphics (Bertin 1967).

Attempts to understand parts of the design space were made by Keller (Keller and Keller 1994) which listed techniques used in scientific visualization. Chuah and Roth developed a framework to enable the comparison between systems and to reuse previous design elements, as well as to guide their design of a toolkit for constructing visualization interfaces through the composition of interaction primitives (Chuah and Roth 1996). In this characterization, the authors have decomposed interactive systems into semantic primitives which were called Basic Visualization Interactions (BVIs). These authors provided a classification for the various primitives and described how they can be composed to form new systems.

Shneiderman suggests that a useful starting point for designing advanced graphical user interfaces is the Visual Information-Seeking Mantra: overview first, zoom and filter, then details-on-demand (Shneiderman 1996). This author proposed a task-by-data-type taxonomy considering 1D, 2D, 3D data, temporal and multi-dimensional data, tree and network data and seven tasks: overview, zoom, filter, details-on-demand, relate, history, and extract.

<table>
<tr><td colspan="4">Large volumes of data</td></tr>
<tr><td colspan="2">Inherently graphical data</td><td colspan="2">Non-graphical data</td></tr>
<tr><td colspan="2">direct ↓ ← Graphical abstraction</td><td colspan="2">↓ direct</td></tr>
<tr><td colspan="2">Large information space (graphical)</td><td colspan="2">Large information space (non-graphical)</td></tr>
<tr><td>Distorted view (detail in context)</td><td>Non-distorted view (detail while little or no context)</td><td>Distorted view (detail in context)</td><td>Non-distorted view (detail while little or no context)</td></tr>
<tr><td>Encoding spatial transformation (geometric)</td><td>Zooming windowing</td><td>Data suppression (abstraction and thresholding)</td><td>Paging / clipping</td></tr>
</table>

Figure 12: Leung's taxonomy of presentation techniques for large volumes of data.

Leung and Apperley (Leung and Apperley 1994) suggested a taxonomy of visual presentation techniques that, at a top level, distinguishes non-graphical data from inherently graphical data. For each class these authors present the transformations required (direct

transformation/graphical abstraction) and the visualization technique (distorted view/non-distorted view) that are more suitable for presentation (see Figure 12).

Card provided a data-oriented taxonomy (Card and Mackinlay 1997) which was subsequently expanded in (Card, Mackinlay et al. 1999). This taxonomy divides the field of visualization into several subcategories: scientific visualization, GIS, multi-dimensional plots, multidimensional tables, information landscapes and spaces, node and link, trees, and text-forms.

Other taxonomies exist centered on the dimensionality of information, such as the OLIVE (On-line Library of Information Visualization Environments) (OLIVE 1997), which classifies techniques using the Shneiderman taxonomy (Shneiderman 1996).

Chi also suggested another taxonomy that classifies information visualization techniques from the data type point of view. Yet, this author integrates it with the processing operators that are inherent in visualization techniques, and that could be described using the Data State Model (Chi 2000). In Chi's paper, 36 techniques are classified and organized according to the following groups in scientific visualization, GIS, multi-dimensional plots, multidimensional tables, information landscapes and spaces, node and link, trees, and text-forms.

On the presentation perspective, Tufte divides the graphical representation of quantitative information into four distinct classes, namely: data maps, time-series, space-time narratives and relation graphics (Tufte 2001).

In short, in the 1990's, a wide set of designs and techniques had been proposed, leading to the need for classification systems that enabled the characterization and comparison between the techniques. Since then several taxonomies have been presented with substantial differences among them, as they address information visualization techniques from distinct perspectives such as, functional, interaction-centered, data-type centered, domain-centered, etc.

For the purpose of presenting the relevant streams of research in the field of information visualization, in the following sections, a data structure classification approach will be used, which is adapted from Shneiderman taxonomy mentioned above (Shneiderman 1996). Hence, techniques will be presented according to the underlying structure of information they intend to display. For this matter, information is seen organized in the following structures:

- Linear structures, such as tables, program source code, alphabetical lists, chronologically ordered items;
- Hierarchical structures such as trees;
- Networked structures, such as general graph structures, hypermedia node-link graphs, semantic networks, webs.

- Vector spaced structures, mainly from the field of information retrieval, addressing text document corpus, word frequencies, similarity measures between documents, document clustering.
- Spatial structures: inherently 2D or 3D information such as floor plans, maps, CAD models, which have an inherent form of representation due to their physical characteristics. However, abstract-based information can be overloaded onto spatial structures.
- Multidimensional structures: integration metadata attributes such as type, size, author, modification date, among others. Items with *n* attributes become points in *n*-dimensional space.
- Time-dependent structures or time-dependent information, which has been detached from multidimensional structures since the term multidimensional refers to the dimensionality of the independent variables, while the term multivariate refers to the dimensionality of the dependent variables of a data set (Bergeron, Cody et al. 1993; Wong and Bergeron 1997). In time-dependent structures the parameter time is predominant and subdues the dimension for other associated variables, leading to the consideration that this class of structure corresponds to multivariate data over time.

The next sub-sections go into each of these categories describing information visualization techniques that have been proposed over the years. However, before doing so it is important to refer that several of the visualization techniques that are presented could be mapped to more than one of these categories. Just as an example, some vector spaced structures, from information retrieval, can also be regarded as multivariate structures. Time-dependent structures and multivariate structures have coincident domains. Information retrieval spatial-system based visualization techniques could also be mapped into the spatial structures categories.

4.5 VISUALIZATION OF LINEAR STRUCTURES

Linear structures regard one-dimensional data which occurs in the form of sequential lists, most often text based (OLIVE 1997). Typical examples include program listings, documents with many lines, and document search results. Users of systems that visualize such data will sometimes wish to:

- further search for more specific results;
- visualize global data about the character of the list they are viewing;
- visualize how a particular element in the list compares to others;
- easily access individual elements in long lists.

The common approach to visualizing linear structures is to provide methods for scrolling through the information until the desired information element is reached. Usually this is accompanied by some sort of ordering system to produce labels for the data, either by page or line, which facilitates the navigation. A typical WWW search result illustrates all these techniques (OLIVE 1997).

Common visual approaches compact the presentation of information and, by doing this, allow more data to fit on screen. Consequently, if more data is simultaneously displayed, the user can access that data much faster. Most often, the compression of information in a presentation is based on coding information with visual clues such as line length or color.

Seesoft (Eick, Steffen et al. 1992; Ball and Eick 1996) is a visualization technique for linear structures, more specifically, for large amounts of program source code (up to 50000 lines of code simultaneously, according to the authors). In this technique, lines of code are compressed down to rows of pixels, where the color of each row indicates a statistic of interest. However, the part of program code the user is interested in is displayed in a readable size. Figure 13 illustrates an example consisting of 38 files comprising 12037 lines of code. The age of source code lines is shown in gradient colors from red (newest lines) to blue (oldest lines). According to the authors, this technique allows the display of data derived from a variety of sources, such as version control systems that track the age, programmer, and purpose of the code, static and dynamic analyses. By means of direct manipulation and high interaction graphics, the user can manipulate this reduced representation of the code in order to find interesting patterns. Due to the capability of displaying the area of interest, while maintaining a context, this technique is considered a F+C technique.

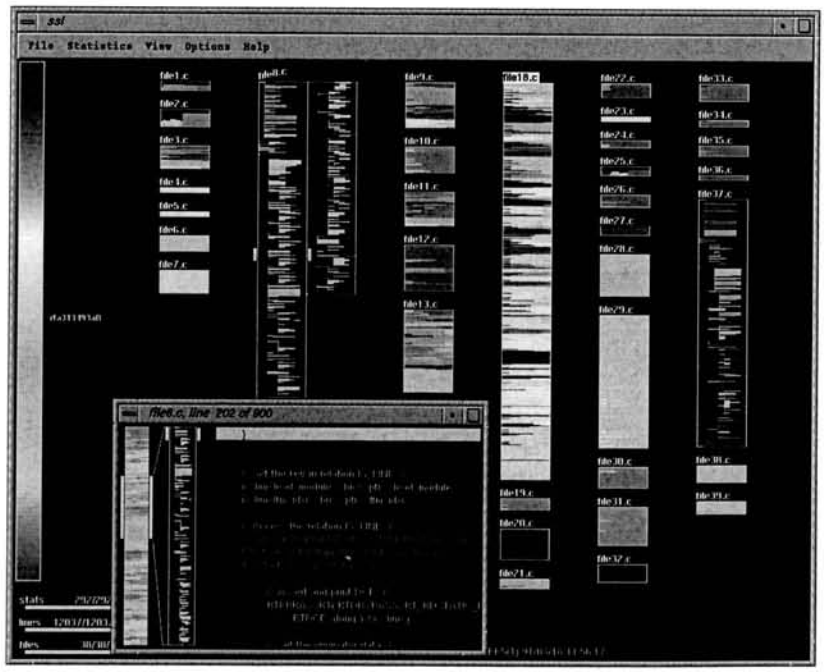


Figure 13: An illustration of the Seesoft visualization technique.

TileBars is an icon list based visual approach to information retrieval which allows users to simultaneously view the relative length of the retrieved text documents, the relative frequency of the query terms, and their distributional properties with respect to the document and to each other (Hearst 1995). For example, in Figure 14, each TileBar icon represents a single document and the length of the bar is proportional to the length of the document. Each greyscale block represents a segment of text as determined by the TextTiling method.

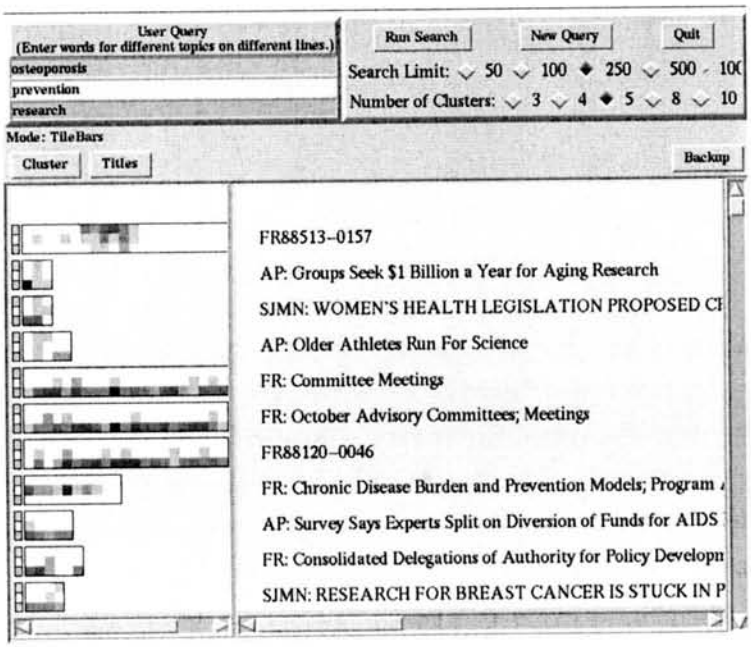


Figure 14: An illustration of the TileBar visualization technique.

Regarding linear structures such as regular matrixes, Rao and Card suggested the Table Lens technique for visualizing and making sense of large tables (Rao and Card 1995). According to Rao and Card, by fusing symbolic and graphical representations into a single manipulable F+C display and providing a small set of interactive operations (e.g. sorting), the Table Lens supports navigating around a large data space, easily isolating and investigating interesting features and patterns. For instance, in Figure 15, a graphical representation was selected for each of its columns, based on the type of variable in the column e.g. category, quantitative, or date. Graphical vocabulary including text, color, shading, length, and position to represent underlying cell values is utilized. It is based on a number of factors besides the variable type including, besides the data value itself, the kind of region and its specific geometry, as well as user options and results of user operations (e.g. search). According to the authors, due to data compression based on symbolic and graphical representation, the Table Lens technique supports much larger tables than conventional spreadsheets do (about 30 times as many cells).

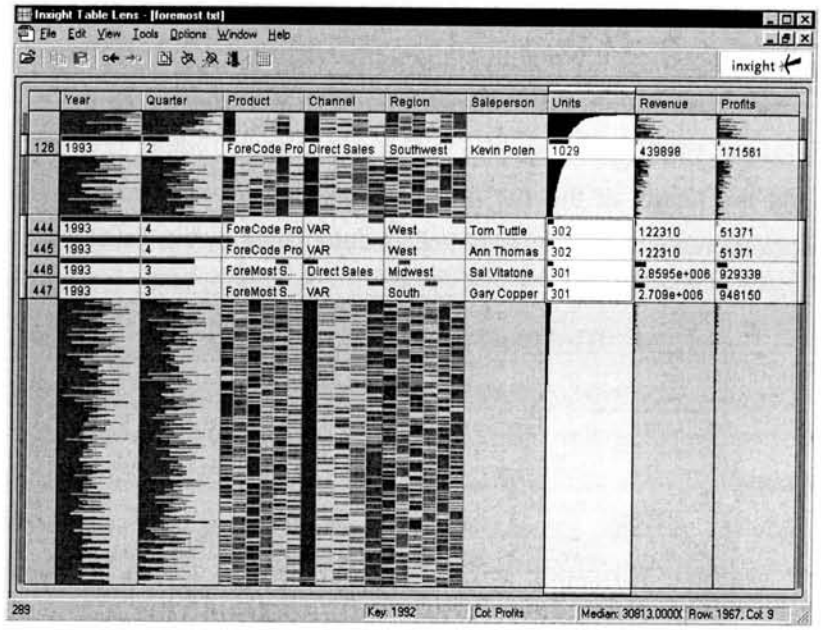


Figure 15: An illustration of the Table Lens visualization technique.

Freeman and Fertig presented a metaphor which aims at the task of dynamically organizing linear structured information, more specifically the user's personal electronic data, consisting in personal files, electronic mail, schedules, rolodex and financial data: the Lifestream (Freeman and Fertig 1995).

According to the authors, the Lifestream is built on a simple storage metaphor - a time-ordered stream of documents combined with several powerful operators - that replaces many conventional computer constructs (such as named files, directories, and explicit storage). In the process, the Lifestream provides a unified framework that subsumes many separate desktop applications to accomplish and handle personal communication, scheduling, and search and retrieval tasks (Figure 16).

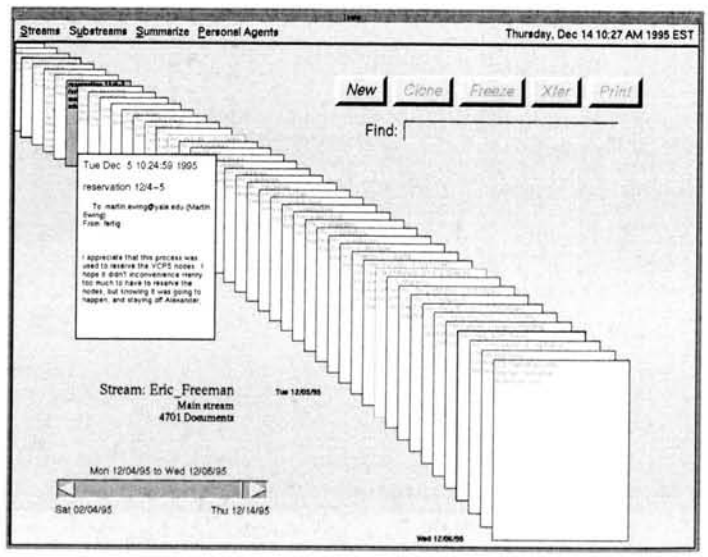


Figure 16: An illustration of the Lifestreams visualization technique.

The tail of the stream contains documents from the past and moving away from the tail and toward the present, the stream contains more recent documents (such as working documents or unread e-mail). Moving beyond the present and into the future, the stream contains documents that will require the user attention: reminders, calendar items, to-do lists.

Still about linear structures, Mackinlay, Robertson and Card proposed the perspective wall visualization technique - a 3D visualization metaphor (Mackinlay, Robertson et al. 1991). The Perspective Wall's physical metaphor of folding is used to distort an arbitrary 2D layout into a 3D visualization (the wall), while automatically retaining any 2D task-specific features. The center panel shows detail, while the two side panels, receding in the distance and shaded to enhance 3D perception, show the context information (see Figure 17).

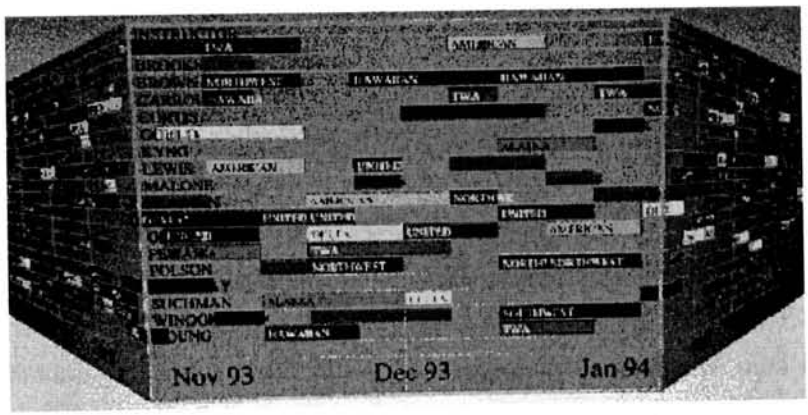


Figure 17: An illustration of the 3D perspective wall visualization technique.

Robertson and Mackinlay presented another F+C technique for the visualization of large rectangular presentations that allow the user to quickly focus on a part of a presentation while continuously remaining in context (Robertson and Mackinlay 1993): the Document Lens. The user grabs a rectangular lens and pulls it around to focus on the desired area at the desired magnification. The presentation outside the lens is stretched to provide a continuous display of the global context. According to the authors, this stretching is efficiently implemented with affine transformations allowing text documents to be viewed as a whole with an interactive visualization (see Figure 18).

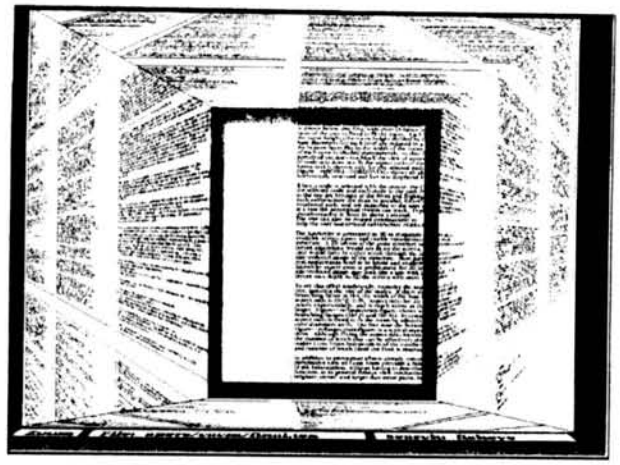


Figure 18: An illustration of the Document Lens visualization technique.

Other less conventional visualization techniques exist for linear structures, such as the WordsEye technique (Coyne and Sproat 2001). This technique works on written text, and automatically creates figurative 3D visualization of an English description of scene. For instance and according to the authors, Figure 19 corresponds to a visual representation of the sentence "*John uses the crossbow. He rides the horse by the store. The store is under the large willow. The small allosaurus is in front of the horse. The dinosaur faces John. A gigantic teacup is in front of the store. The dinosaur is in front of the horse. The gigantic mushroom is in the teacup. The castle is to the right of the store.*"

According to the authors, the text-to-scene information visualization technique WordsEye relies on a large database of 3D models and poses to depict entities and actions. Every 3D model can have associated shape displacements, spatial tags, and functional properties to be used in the depiction process. In this technique, significant work has been done in the linguistic analysis to evaluate the semantic representation of the input text, in order to depict consistent outputs.

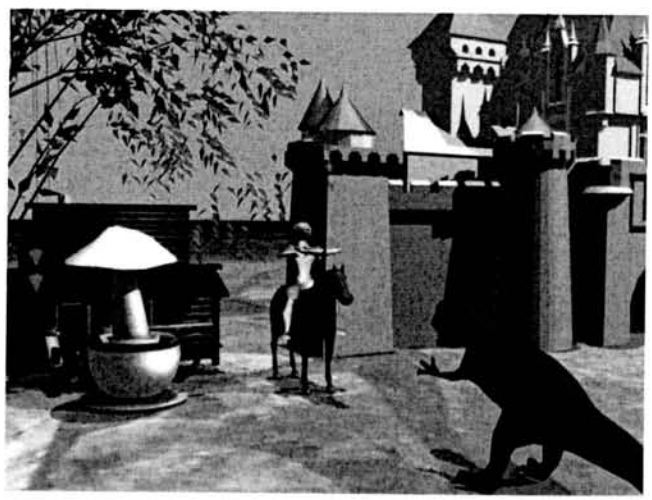


Figure 19: An example of the WordsEye visualization technique.

4.6 VISUALIZATION OF HIERARCHICAL STRUCTURES

In OLIVE (OLIVE 1997), hierarchical structures are defined as collections of data nodes where each node has a unique parent (node above it in the hierarchy), but may have many siblings (nodes below it in the hierarchy). In general, the nodes and the links between them can have multiple attributes. The entire structure of the hierarchy and its encompassing relations is also usually relevant. Operations on hierarchies are also vast, such as finding a particular node, viewing a node in the context of the entire hierarchy, examining the overall structure and relations of the tree, and even finding duplicates or anomalies within the tree structure. Hence, tasks can be applied to a single node, a link, a collection of nodes, or even to the entire structure.

Tree views are a common technique to display tree structures, based on two distinct display areas: the left area shows the structure while the right area shows the corresponding information items (files, documents) at a particular tree level. Most common examples are the tree controls used in computer applications to display the contents of a hard disk.

For website content visualization, the WebTOC is a visualization technique aimed at extending the traditional tree view with overlapped supplementary statistical information about the proportion of various media types present on each webpage (Nation, Plaisant et al. 1997). Here, colored bars are used to depict this proportion, while shadows on the bars indicate number of files (see Figure 20).

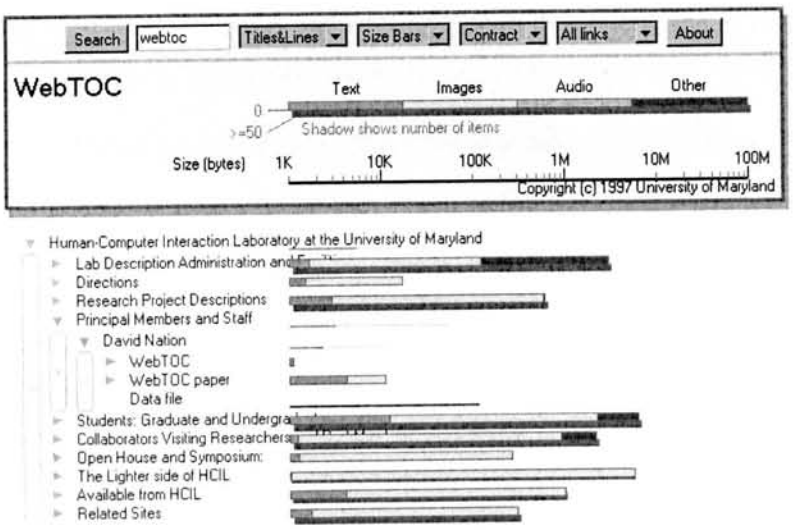


Figure 20: An example of the WebTOC visualization technique.

Robertson, Mackinlay and Card proposed a visualization method aimed at hierarchical structured information that contains some fisheye properties: the 3D Cone Tree (Robertson, Mackinlay et al. 1991). Figure 21 illustrates an example of an application of this technique

which, according to the authors, provides faster navigation and allows more information to be seen at one time: the Cone Tree shows the structure of an entire hierarchy at once, and the cones interactively spin around to retrieve occluded data.

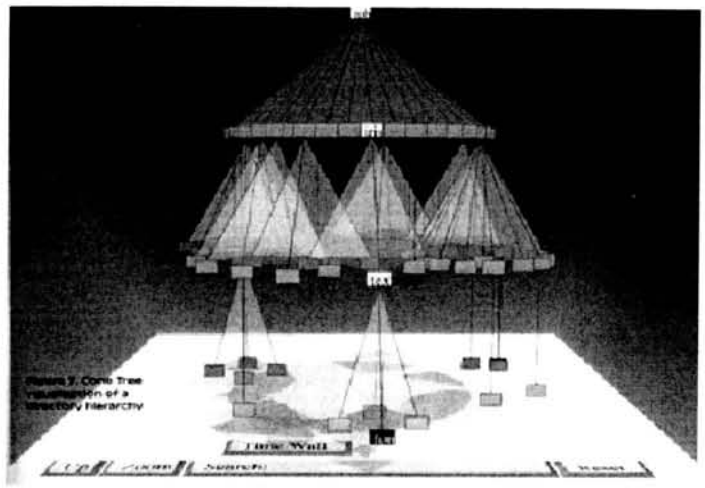


Figure 21: An illustration of the 3D cone tree visualization technique.

In order to display hierarchical information, the *cone* displays a tree where each node corresponds to the apex of a cone and the children of the node are positioned around the rim of the cone. The fact that the tree is detailed rendered in 3D, including shadows and transparency, provides the basic *fisheye* property of showing local information while also showing the entire context.

Aimed at the compact visualization of hierarchical structures, but having the space constrained to a specific limited area, Johnson and Shneiderman suggested the Tree-map (Johnson and Shneiderman 1991) which, according to the authors, is very effective in showing attributes of leaf nodes using size and color coding. The Tree-map technique enables users to compare nodes and sub-trees, even at varying depth in the tree, and help spot patterns and exceptions (see Figure 22).

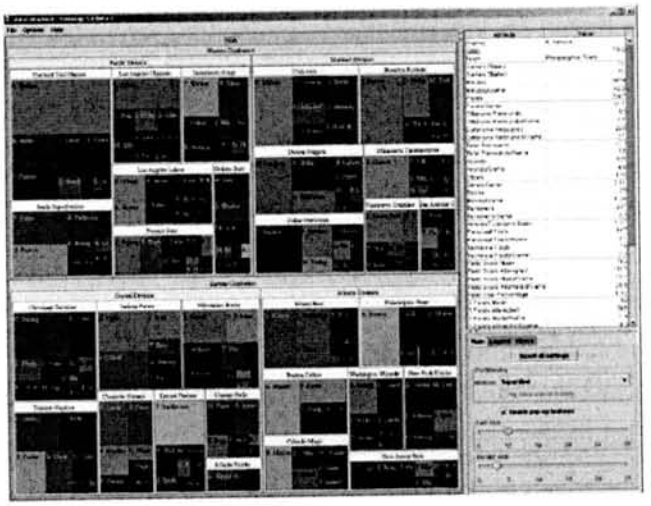


Figure 22: An illustration of the Tree Map visualization technique depicting a computer directory structure.

This technique was later extended by Wattenberg, in order to avoid excessively narrow strips, through the use of heuristics that slice up each rectangle into more evenly proportioned sub-rectangles (Wattenberg 1999).

van Wijk and Wetering also improved the base treemap visualization of hierarchical structures (Wijk and Wetering 1999). These authors regard treemaps as a compact, space-filling display of hierarchical information, based on recursive subdivision of a rectangular image space. van Wijk added intuitive shading to provide insight in the hierarchical structure. During the subdivision, ridges are added per rectangle, which are rendered with a simple shading model. The result is a surface that consists of recursive cushions. According to van Wijk and Wetering, Cushion Treemaps provide shading as a strong extra cue to emphasize the hierarchical structure (Figure 23).

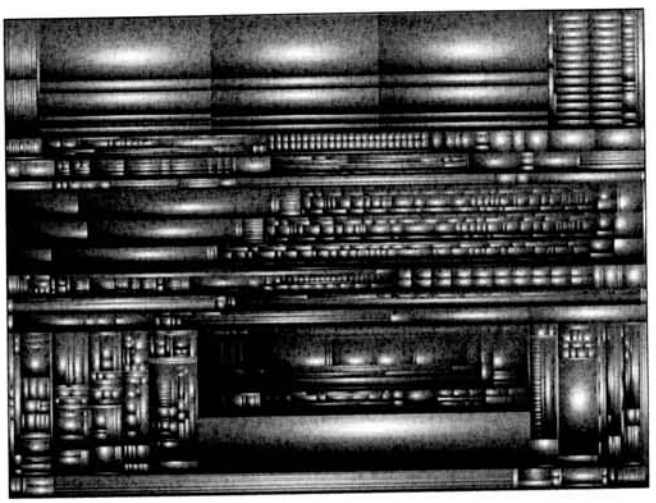


Figure 23: An illustration of the Cushion Treemap visualization technique.

Loosely based upon the 2D tree-map visualisations of Johnson and Shneiderman, Rekimoto presented the Information Cube (Rekimoto and Green 1993), a 3D visualization

technique which is based on the nested box metaphor that, according to the authors, is a familiar concept for users. According to these authors, we must face many kinds of information which are difficult to represent in 1D or 2D, because they are too complex and do not fit into the current displays size and resolution. Rekimoto presents large software structures, database schemes, and organization structures, such as huge project's PERT chart, traffic control information, and network management as domains where the shortage of display area occurs frequently. Hence by using semi-transparent rendering (Figure 24), the system controls the complexity of the information presented to the user. This transparency allows the user to see the inside, while hiding inner information gradually. Rekimoto states that, with several 3D interaction techniques provided by the system, the user can recognize and inspect the information structure intuitively, with the help of VR equipment or through a conventional display.

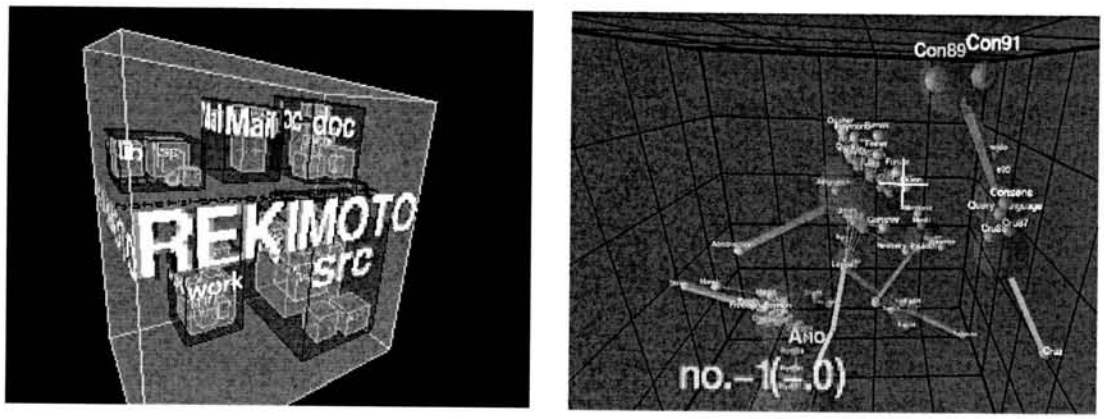


Figure 24: An illustration of the Information Cube. Overview (left) and an inner cube (right).

Also in the stream of space-constrained visualization for hierarchical structures, Lamping and Rao proposed a space-filling F+C technique. This allows the presentation of several levels of the hierarchy at once, by laying the structure on an hyperbolic plane, which is then mapped to the unit disc (Lamping and Rao 1994). As subtrees get far from the root, they become small until the point they are invisible in the display (Figure 25).

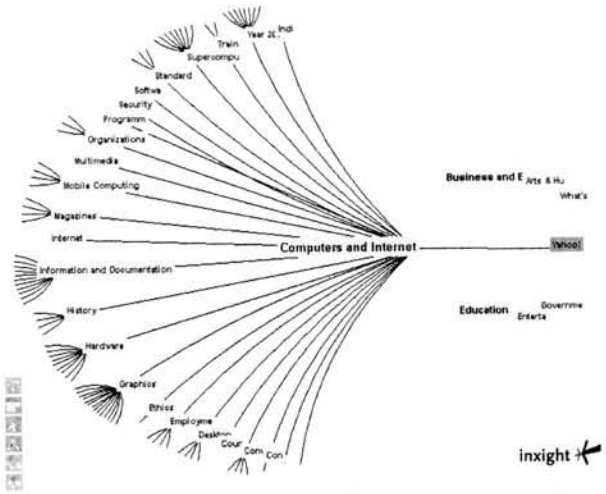


Figure 25: The use of Hyperbolic Browser technique for displaying a hierarchical structure.

Later this technique was improved by adopting a 3D visualization where the disk is replaced by a sphere (Munzner and Burchard 1995) (Figure 26).

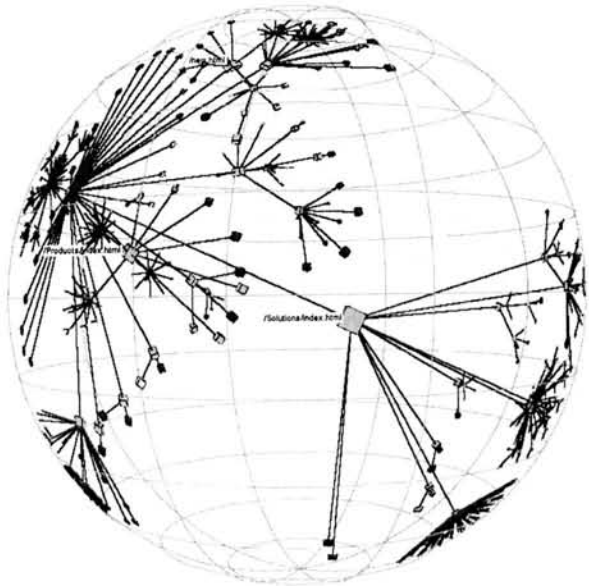


Figure 26: An illustration of the 3D Hyperbolic Browser visualization.

In short, information visualization techniques for displaying hierarchical structures try to provide, by different means, effective answers for two problems: to display large hierarchical structures over a constrained display-space and to display detailed data about the focused information while keeping available context information to let the user realise about its current location in the structure. Similar problems are dealt by information visualization techniques for network structures, presented in the next section.

4.7 VISUALIZATION OF NETWORKS

The SemNet (Fairchild, Poltrock et al. 1988) system is an early example of the exploitation of 3D visualization for network information structures, The structures visualized in SemNet were mostly large knowledge bases in the form of arbitrary graphs. However, depending on the network complexity, the results tended to be cluttered and the cognitive task of understanding the structure was quite difficult (see Figure 27).

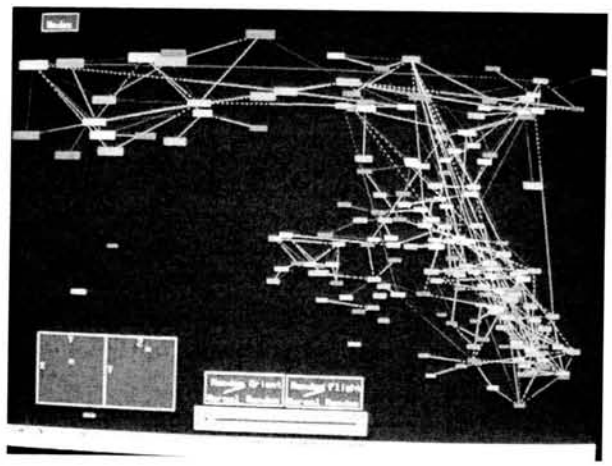


Figure 27: An example of a SemNet 3D visualization for a semantic network.

Narcissus (Hendley, Drew et al. 1995) is an information visualisation system for creating 3D visualisations of hyper structures. The main application of this technique is in the software visualisation area, helping to interpret the software structure by displaying the dependencies between the various components within a system. Later Hyperspace (Wood, Beale et al. 1995) extended the Narcissus information visualisation system by concentrating on the visualisation of hypertext document stores, based on web pages.

Simple Hierarchical Multi-Perspective (SHriMP) (Storey and Muller 1995) is a tool for visualising and exploring software architecture, by providing an overview of a software structure, allowing detailed views while maintaining a context within the graph structure. To such purpose, SHriMP employs both nested graph and fisheye techniques to provide both abstraction and focusing of detail as necessary (Figure 28).

GraphVisualizer3D (Franck and Ware 1994) uses 3D visualisation and virtual environment technology to provide a more readable and comprehensible graph visualisation, aimed at displaying predominantly directed graphs. The focus of this visualization technique has also been to visualise software structures (components and relationships within object oriented C++ programs).

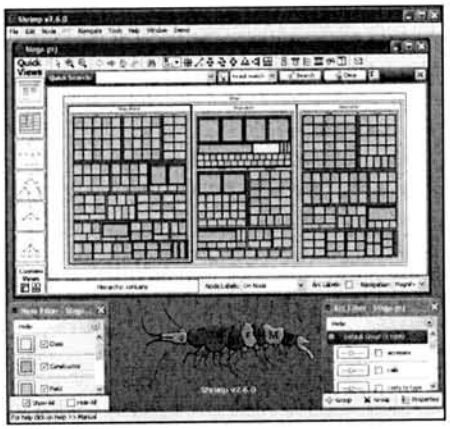


Figure 28: An illustration of the SHriMP prototype main window.

4.8 VISUALIZATION OF VECTOR SPACES

Several systems employ methods for mapping sets of information from a high-dimensional term space to a lower dimensional display space, preserving the high-dimensional distances as far as possible. Most of these systems fit within the field information retrieval, either for flat repositories or for hierarchically organised structures. In both these cases the purpose is to help users effectively access large collections of objects with the goal of satisfying the users' stated information needs (Croft and Turtle. 1992). The most common approaches to text retrieval are the Boolean Term Specification and the Similarity Search, being the last, according to Hearst, an umbrella term covering the vector space model, probabilistic models and any other approach which attempts to find the documents that are most prone to answer a query, based on the terms they contain (Hearst 1995).

According to Hearst, when a system retrieves a document in response to a query, it is important to indicate, not only how strong the match is (e.g., how many terms from the query are present in the document), but also how frequent each term is, how each term is distributed in the text and where the terms overlap within the document. This information is especially important in long texts since it is less clear how the terms in the query contribute to the ranking.

The need for this kind of distributional information has not been emphasized in the past, perhaps because, by then, researchers had not focused on long texts. In order to overcome these limitations, several techniques have been suggested which provide a visual approach that falls into one of three categories: Venn diagram, icon lists and mapping to a specific spatial system. All these approaches aim at providing the user with information that is more compact and faster to perceive than textual results.

InfoCrystal (Spoerri 1993) is an example of a reference point system that is based on the Venn diagram model. It is useful for determining the distribution of documents in a collection that satisfy each of the possible Boolean queries. A geometric shape is drawn, with the number of vertices corresponding to the number of query terms. The resultant subsets of the

search (combination of the query terms) are represented as the icons within the shape. Figure 29 illustrates the process of building an InfoCrystal display based on a 3-term query.

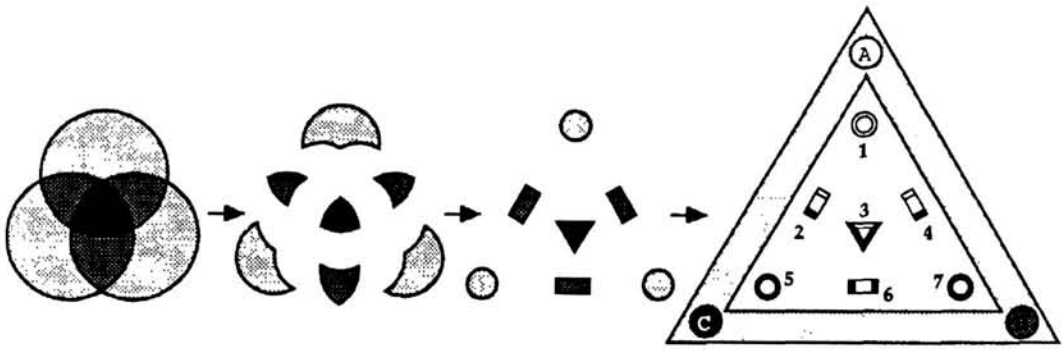


Figure 29: a Venn diagram transformation into an iconic display, according to the InfoCrystal visualization technique.

In SPIRE (Spatial Paradigm for Information Retrieval and Exploration) the goal of information retrieval stands, in contrast to the classical view of information retrieval, where techniques are usually applied to find text that the user expects to read (Wise, Thomas et al. 1995). Two visualisations are provided: SPIRE Galaxies display documents as stars in a galaxy, where documents which are close in high dimensional space are also close in the two-dimensional galaxy display. SPIRE ThemeView builds on the galaxy view by aggregating frequently occurring topical keywords from neighbouring documents and displaying the main themes in a thematic landscape. The layout of the concept space has been accomplished with a variety of clustering and dimensionality reducing algorithms (York, Bohn et al. 1995) and elevation of each feature in the map indicates the theme strength (Figure 30). SPIRE operates on flat, unstructured document collections.

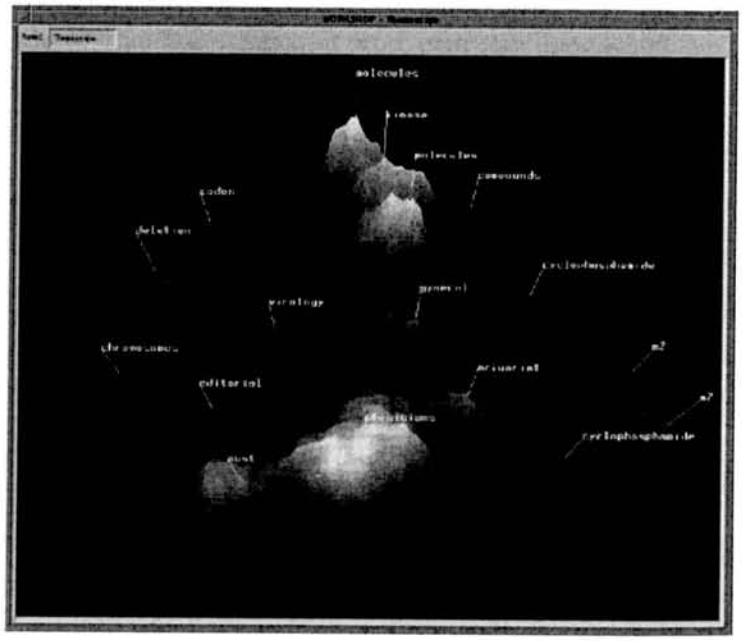


Figure 30: A SPIRE ThemeView showing topic distribution in a large document space.

According to Wise *et al.*, this visualization technique represented a change in paradigm for information retrieval since the original text document was transformed into a new visual representation that communicated by image instead of text.

In VIBE (Olsen, Korfhage et al. 1993) query terms, called Points of Interest (POI), are shown at the vertices of a figure. A POI may be as simple as a set of keywords, or may be defined by more complex functions. For every POI, each document is scored, for instance, according to the POI frequency. The POIs define an information space, and the document icons are mapped to this space according to the POIs score. Figure 31 illustrates a VIBE display of results where the POIs (query terms) form the vertices of the triangle and documents matching the query terms are displayed as rectangles. The documents located in the middle of the triangle have an even score on each POI. On the other hand, documents on top of a POI fully match the corresponding query term and have a zero score for the other POIs.

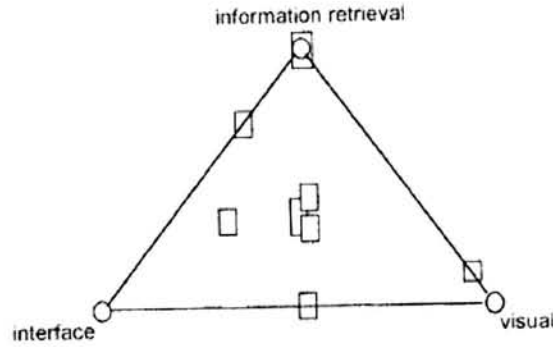


Figure 31: A VIBE display.

According to Morse (Morse, Lewis et al. 2002), VIBE can be used to represent the results of Boolean as well as vector queries. It allows users to interact with multidimensional displays (i.e., any number of POIs) and moderately large document collections.

VisIslands (Andrews, Sabol et al. 2001) is another spatial system-based information retrieval visualization technique. First, it builds a vector space model from the search result, and second, it maps the placement of the model to a 2D island scape (Figure 32). VisIslands use standard clustering techniques to group document sets returned in response to a search query on the fly. The clusters are used for more efficient similarity placement, by first placing cluster centroids, and then placing documents around them

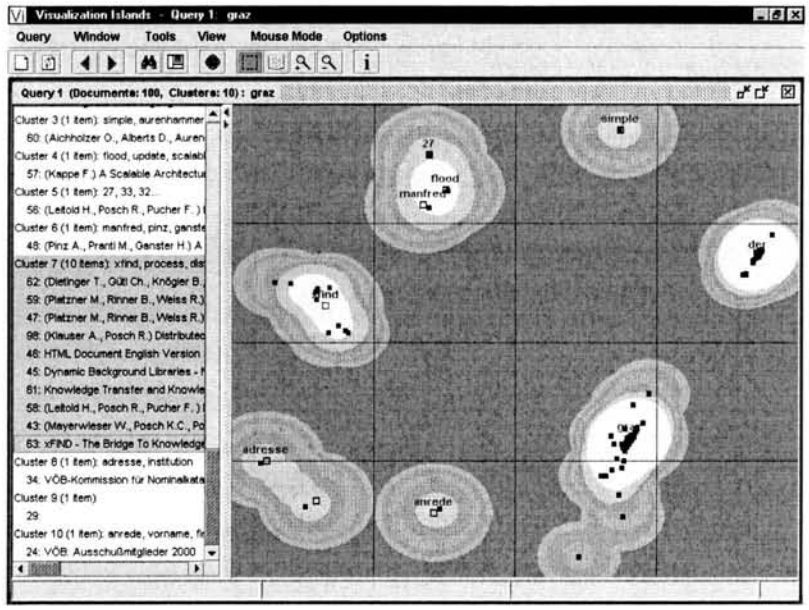


Figure 32: An example of the VisIslands visualization technique (visual clusters of search results).

The Bead system (Chalmers and Chitson 1992) employs a thematic landscape view. The information space is arranged based on inter-document similarity, forming a 2D landscape where users can navigate freely. Bead operates on flat document repositories and does not employ hierarchical structures.

InfoSky visual explorer is a system that enables users to interactively explore large, hierarchically structured document collections (Granitzer, Kienreich et al. 2004). Similar to a real-world telescope, InfoSky employs a planar graphical representation with variable magnification. Documents with similar contents are placed close to each other and displayed as stars, while collections of documents at a particular level in the hierarchy are visualised as bounding polygons. For instance, in Figure 33, two perspectives of the InfoSky Browser 2002 visualization are presented: on the left view, a context overview is displayed, and on the right view and through proper zoom, the red squared region on the left picture is presented in more detail.

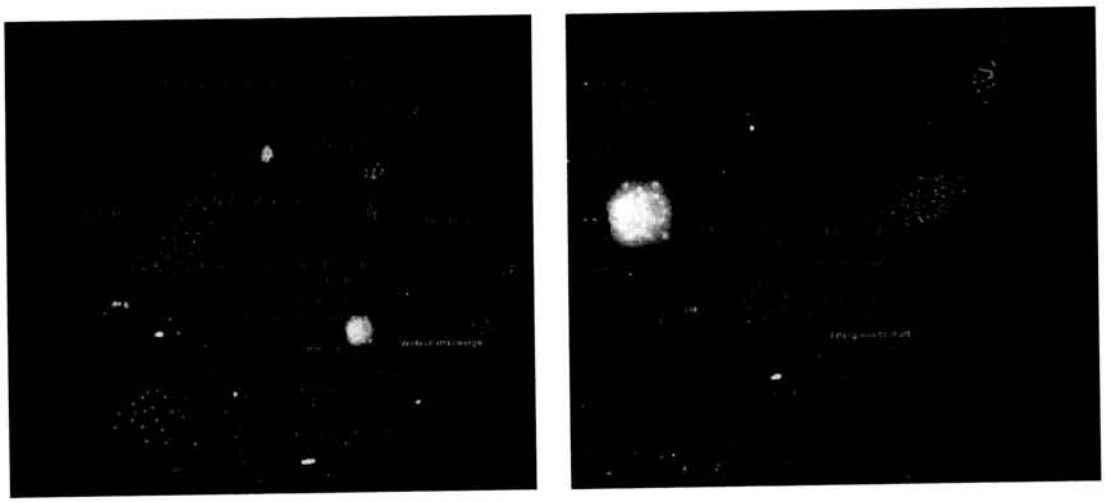


Figure 33: Two perspectives of an InfoSky Browser 2002 visualization.

According to the authors, InfoSky addresses challenges such as:

- to represent both the hierarchical organization of documents and inter-document similarity within a single, consistent visualization;
- to integrate both a global and a local view of the information space into one seamless visualization;
- to maintain an easy exploration and good scalability.

The authors also provide a comparative overview of several information retrieval visualization techniques, compared to InfoSky (Granitzer, Kienreich et al. 2004).

Still regarding visualization techniques for information retrieval, Morse *et al.* suggest that, although many techniques have been proposed, few have been tested and, furthermore, where testing has been performed, results were often inconclusive (Morse, Lewis et al. 2002). For such purpose, Morse *et al.* present a case study methodology (BASSTEP) for testing novel interfaces that uses bottom-up, stepwise testing to allow the evaluation of a visualization technique, itself, rather than restricting the evaluation to the system instantiating it. According to Morse *et al.*, BASSTEP provided clear results in a case study where five different display types used in existing information retrieval visualization systems were evaluated. From among text, tables, icons and graphs, the graphical methods (icon, graphs) were preferred when the complexity of the task is increased. The novel method proved to give the best performance in the most complex situation, a three-term task with vector data.

4.9 Visualization of Multidimensional Data

According to Wong, the field of multidimensional data addresses the study of relationships between multiple parameters (Wong and Bergeron 1997). Mathematically, these parameters

can be classified as dependent and independent (Keller and Keller 1994), where a variable is said to be dependent if it is a function of other variables, called the independent variables.

Bergeron clarifies the term multidimensional and distinguishes it from the term multivariate by stating that the first refers to the dimensionality of the independent variables, while the last refers to the dimensionality of the dependent variables (Bergeron, Cody et al. 1993).

Wong states that the emphasis in multidimensional multivariate data visualization (MDMV) shifts away from the strong mathematical definition of dependent and independent variables towards the broader definition of multiple variates, or factors, where the key goal is visually summarize a data set and find the hidden relationships between multiple dependent variables (Wong and Bergeron 1997) .

First stage in MDMV visualization started in 1782 when Crome used point symbols to show the geographical distribution in Europe of 56 commodities (Collins 1993). According to Wong's survey, this stage ended in 1976, before Tukey's work (Tukey 1977) which signified a new stage in scientific data visualization. This second stage is characterized by interactive visualizations of 2D and 3D data, although multivariate data started gaining more attention. Third stage is seen as the main expansion of this field, since most of the MDMV visualization concepts were developed during this stage, which include: grand tour methods (Buja and Asimov 1986), parallel coordinates (Inselberg, Reif et al. 1987), iconography (Pickett and Grinstein 1988), worlds within worlds (Feiner and Beshers 1990), dimension stacking (LeBlanc, O.Ward et al. 1990), hierarchical axis (Mihalisin, Gawlinski et al.), Hyperbox (Alpern and Carter 1991) and various ideas collected in (Cleveland 1993; Cleveland, McGill et al. 1993).

Some of these techniques attempt to show all dimensions and all variates visually as one display, whereas others aim at *direct manipulation graphics*, in which the user interactively selects subsets for display. One common approach for reducing the complexity of a multivariate function is to hold one or more of its independent variables constant. Each constant corresponds to take an infinitely thin slice of the world perpendicular to the constant variable's axis, reducing the world's dimension. In the third stage, virtual reality (Feiner and Beshers 1990) began to appear in the MDMV visualization literature.

According to Wong, some of the most recently developed tools are, each in a different way, elaborations of work done in previous stages. HyperSlice (van Wijk and van Liere 1993) is an attempt to combine the panel matrix of scatter plot matrix with direct manipulation of brushing (Becker and Cleveland 1987). AutoVisual (Beshers and Feiner 1993) is an extended version of worlds within worlds (Feiner and Beshers 1990) with a new rule-based interface. XmdvTool (Ward 1994) integrates four existing MDMV visualization tools: dimension stacking, scatter plot matrix, glyphs, and parallel coordinates into one system with enhanced n-dimensional brushing.

Despite this retrenchment stage, new techniques are being suggested, for instance starfield visualizations (Ahlberg and Wistrand 1995), relations among the variables in multivariate datasets as hyper-surfaces (Inselberg 1997), visualization with uniform treatment of all dimensions that provides an integrated overview of an entire multidimensional function space (Jayaraman and North 2002).

4.10 Time-dependent Multivariate Information Visualization

Temporal data can be analyzed using statistical tools. Many algorithms are developed to extract valuable information from the data (understand correlations, study trends, identify patterns, etc.). A complementary research avenue involves visual analysis of temporal data, which relies on interactive visualization of time-dependent information. The purpose of the visualization of time-dependent data is to present, detect and validate properties of time patterns and temporal behaviours of information items. According to MacEachren (Maceachren 2004), a well designed visualization can help in the answering of questions for unknown temporal data, such as:

- Does a data element exist in a specific time; (temporal existence)
- When does a data element exists on time; (temporal location)
- How long is the time span from the beginning to the end on the data element? (temporal interval)
- How often does a particular data element occur? (temporal texture)
- How fast is a data element changing or how much difference is there? (rate of change)
- In what order do data elements appear? (sequence)
- Do data elements exist together? (synchronization)

There are distinct views and definitions of what time is, which justifies the existence of distinct mathematical, psychological and social time models, among others. The perception of time has very specific properties, such as being continuous and being impossible to move in a backwards direction. The way humans perceive time is critical in the visual representation of this dimension. Mainly, such task has been accomplished through proper visual metaphors, each one addressing one or more properties of the characterization of time.

The visualization of time-dependent data has been one focus of interest over the centuries. For instance, first known elements of visualization of time-dependent data dated from the 11th century and are present in a text book from a monastery school. This book contains an illustration of planet orbits described over the years (Tufte 2001).

The earliest modern timeline was created by Jacques Barbeu-Dubourg in 1753. The *Carte Chronologique* (see Figure 34) was an annotated timeline of history on a 54-foot scroll and included the names and the descriptive events, grouped thematically, with symbols denoting

character (martyr, tyrant, heretic, noble, upright, etc.) and profession (painter, theologian, musician, monk, etc.) (Ferguson 1991).

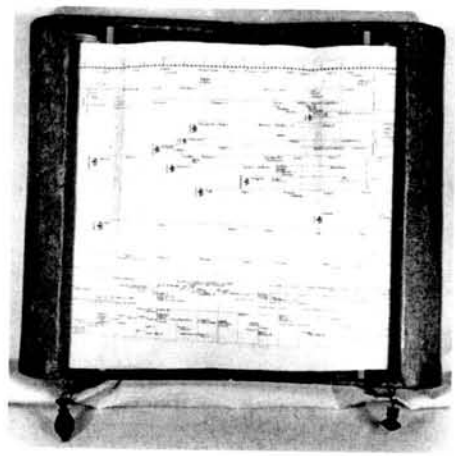


Figure 34: The *Carte Chronologique.*

Joseph Priestley was one of the first to attempt to graphically depict historical events (Priestley 1765). The *A Specimens of a Chart of Biography* was designed as a proof of concept for a much larger, more detailed version, showing the lifespan of 2000 famous people, from 1200 b.C. to 1750 a.C. (see Figure 35). Priestley is credited with inventing the idea of showing the lifespan of an entity (famous people in this context) by the length of a horizontal line. He also invented the convention of using dots (...) to indicate uncertain birth and death dates.

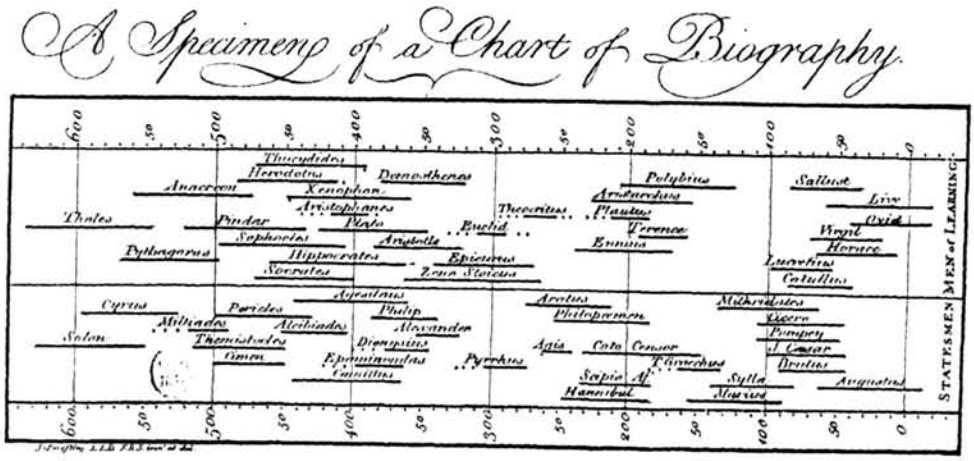


Figure 35: Priestley's Chart of Biography.

In 1779, Johann Heinrich Lambert developed a graphical analysis of periodic variation (in soil temperature), having the first semi-graphic display combining tabular and graphical formats.

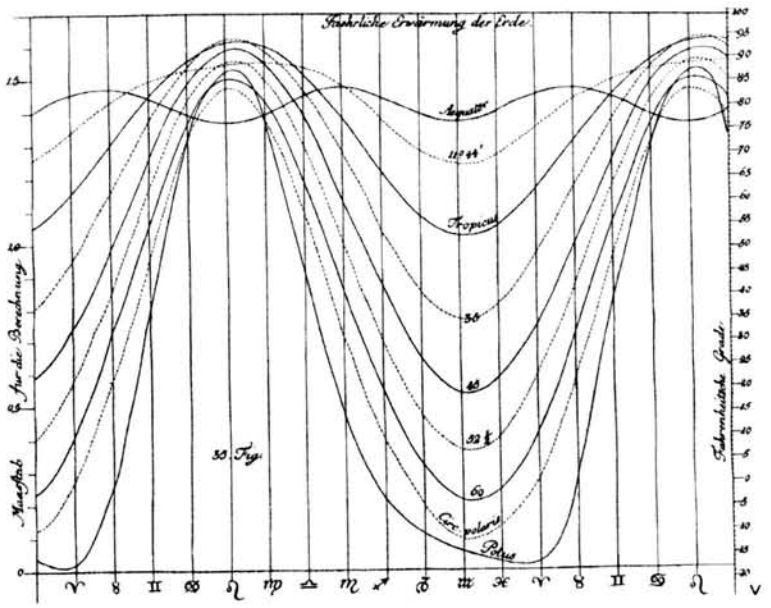


Figure 36: Lambert graphical analysis of periodic variation regarding soil temperature.

Charles Minard (1781-1870) illustrated the disastrous result of Napoleon's failed Russian campaign of 1812 (see Figure 37). The graph shows the size of the army by the width of the band across the map on its advance and retreat directions, with temperature on the retreat stage shown by the red line graph at the bottom. This chart provides the army location and the trajectory, its size and the weather temperature changing over the time span of the campaign. According to Tufte, this chart is still considered one of the best statistical charts ever drawn (Tufte 2001).

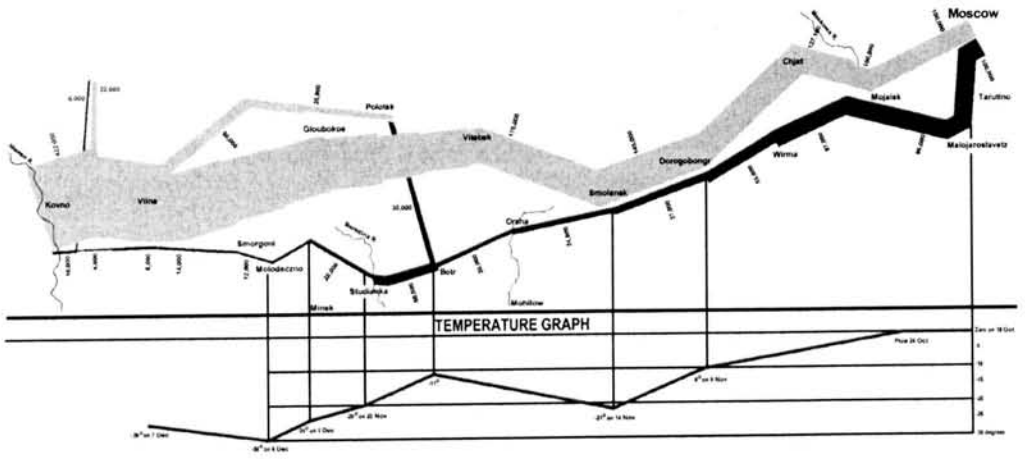


Figure 37: Minard's visual history of Napoleon's Russian Campaign.

Ibry created a diagram, showing the time schedules of trains from Paris to Lyon (Marey 1885). This design makes it easier to determine the fastest trains (steeper slopes), layovers (line breaks), and the intersection points of south-bound and north-bound trains (see Figure

38). The design works as a graphic calculator because the stations are spaced on the vertical axis in proportion to distance along the route.

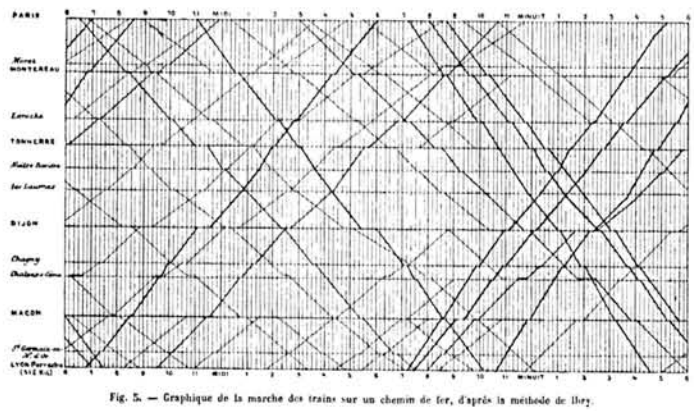


Fig. 5. — Graphique de la marche des trains sur un chemin de fer, d'après la méthode de Ibry.

Figure 38: Ibry's diagram of train schedules from Paris to Lyon.

The examples presented until now are considered effective solutions to the problem of displaying time-dependent information for a particular context.

Currently there is a wide range of techniques available to display time-dependent data. According to Müller and Schumann (Müller and Schumann 2003), in order to effectively provide a visual representation that can properly address time-dependent data, it is necessary to address the following questions:

- What kind of representation is best suited: *static* or *dynamic* representation?
- What is the underlying temporal model: foreseen (linear or non-linear, discrete or continuous) or unforeseen (event driven)?
- What kind of temporal aspects of data are for display: isolated variables or set of variables for multivariate correlations?

Aigner *et al.* (Aigner, Miksch et al. 2007) provide a systematic categorization for the graphical representation of time-oriented data analysis (see Figure 39).

Time	**Temporal primitives**	time points	time intervals	
	Structure of time	linear	cyclic	branching
Data	**Frame of reference**	abstract	spatial	
	Number of variables	univariate	multivariate	
	Level of abstraction	data	data abstractions	
Representation	**Time dependency**	static	dynamic	
	Dimensionality	2D	3D	

Figure 39: Categorization schema for visual methods for analyzing time-oriented data.

According to these authors, the visual methods for analyzing time-oriented data depend on a categorization that, at a top level, depends on three criteria: nature of time, source data, and representation, namely:

- nature of time: is time represented as points or as intervals? What time model is used: linear, cyclic, or branching?
- what is the nature of the source data: is it abstract information or has a spatial reference? Is it univariate or multivariate? How many levels of abstraction does it have?
- what is the adopted representation? Is it static or dynamic? Is it a 2D or a 3D representation?

The following sections address relevant work in graphically representing time, grouped according to the time dependency in the representation of time: *static* and *dynamic* representations. Temporal nature and underlying data categorizations are provided as needed.

4.10.1 Static Visual Representations

Static visual representations benefit from having the ability to allow the conclusion of quantitative facts as well as explore special data features, data values, time steps and positions according to the underlying time scales, without limitations. However, most often, the application of *static* visual representation techniques has been restricted to depict one variable or a limited number of variables, whereas the analysis of multivariate correlations is reduced. In order to compress more information onto a single chart, Bertin distinguished 8 visual variables (Bertin 1981; Bertin 1999). These variables could aid in the visual representation to encode nominal, ordinal or quantitative data visually. More recently, Mackinlay enhanced the ranking of these visual variables (Mackinlay 1986).

There is a wide range of examples regarding *static* visual representations. Harris stated that in a case of a 1D scale, the visual representation leads to a *Sequence Graph* while 2D and 3D scales lead to a *Time Series Graph* (Harris 1999). This author also states that the appropriateness of a technique also depends on the nature of available data, for instance, *Point Graphs* for point data, *Line Graphs* for continuous data and *Circle Graphs* for cyclic data (see Figure 40).

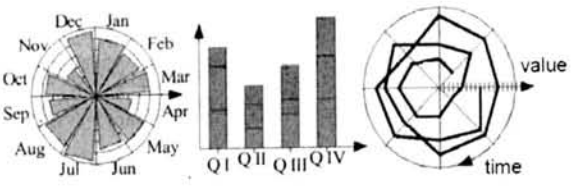


Figure 40: Graphs for viewing time: sector, stacked bar, circle (left to right).

The *Stacked Bar Chart* represents the amount of data for one variable as variable height bar slices that are present at each bar, depicting an individual time-step. This method also uses color or texture to code the values of each time-step. The *Stacked Bar Chart* belongs to a class of visualization techniques that map independent representation of data, for each time-step, into a single visual representation. Another example of this class is the *Parallel Coordinates* technique, which consists in multiple parallel 2D coordinate systems, one for each time-step, where values are plotted as bars. Each bar displayed on coordinate system is connected through lines to the previous and next bars in the corresponding coordinate systems. Color (Brewer 1994) and texture also help the user to better identify distinct entities displayed on the chart (Inselberg 1997).

In an effort to develop a systematic approach to the representation of change on maps, Monmonier categorizes methods for depicting time on static maps as falling in three types (Monmonier 1990): dance maps (single maps that depict change in an object's location over a time interval), change maps (single maps that depict change in a location's attributes during a time interval), and chess maps (small multiples that depict change in the state of things from one time to the next). Kraak extends this categorization to include dynamic representations of space-time phenomena (Kraak 1994) and Edsall *et al.* propose a categorization of changes in space-time phenomena along a continuum, ranging from change in location (but not attributes) to change in attributes (but not location) (Edsall and Peuquet 1997).

Given that time can be conceptualized as linear and as cyclic, cartographic representations of time are problematic (Maceachren 2004). Using the static "small multiple" technique (Bertin 1981; Tufte 2001), both conceptualizations can be incorporated in the same visual display, for example, by using an organization in which column positions of the small multiple matrix depict identical points in the cycle and the sequence across the entire time can be seen by scanning across each row in turn.

As opposed to the visualization of a single variable, the visualization of multivariate data over time allows the analysis of multivariate correlations, which can be of special interest. However, according to Müller, visualization of multivariate data over long periods of time requires a special effort (Müller and Schumann 2003).

The *ThemeRiver* is a visualization technique for analysis of time-related patterns, trends, and relationships across a large collection of documents (Havre, Hetzler et al. 2000). The themes are represented by a "river" that flows left to right through time. The river widens or narrows to depict changes in the collective strength of selected themes in the underlying documents. Individual themes are represented as colored streams flowing within the river. The theme currents narrow or widen to indicate changes in individual theme strength at any point in time. For instance, Figure 41 illustrates themes in Fidel Castro's speeches from 1959-mid 1961:

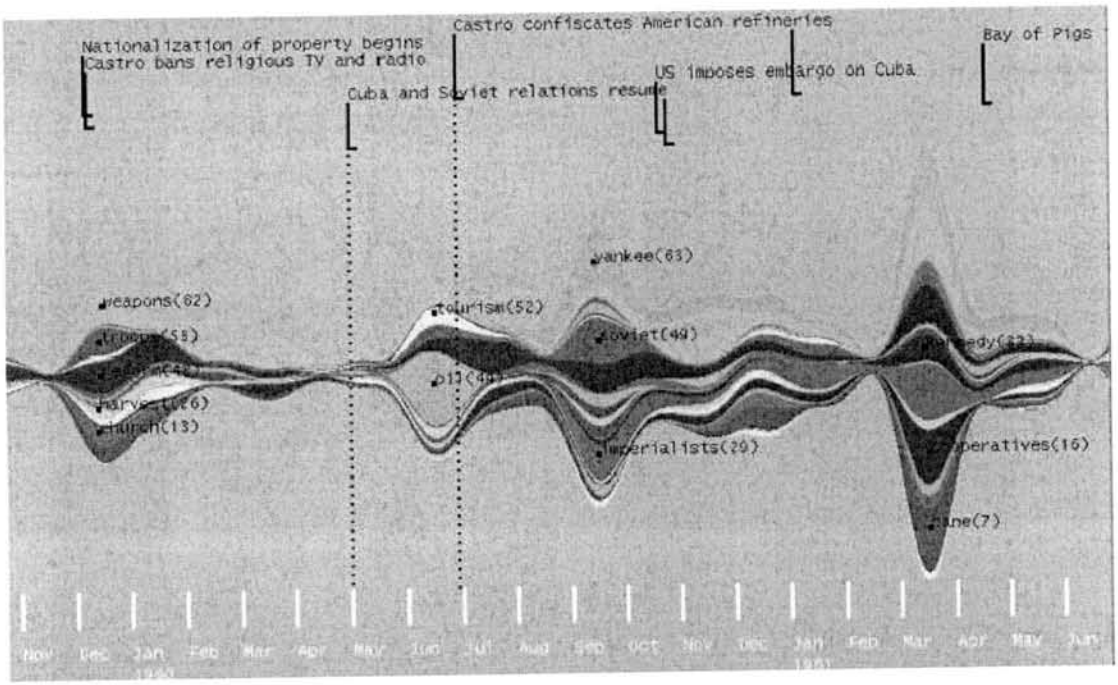


Figure 41: An illustration of the ThemeRiver visualization technique.

The *Spiral Graph* is a visualization technique which displays data along a spiral to highlight serial attributes along the spiral axis and periodic attributes along the radius (Carlis and Konstan 1998). This technique provides the means to compare and analyse temporal serial and cyclic data. The data is rendered over each radius as points, lines or bars, depending on the characteristics of data. For instance, Figure 42 presents a spiral display of year-month consumption percentage of food during a period of 1980-1988, rotated and zoomed to show one season of boundary lines.

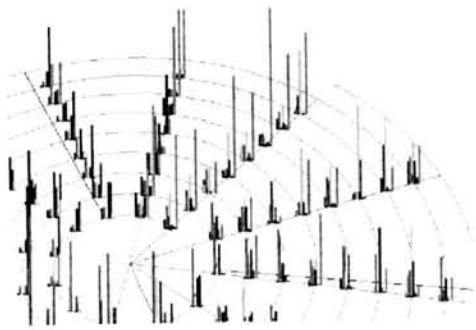


Figure 42: An illustration of a Spiral Display visualization.

The *Calendar View* is a visualization technique that uses an intuitive metaphor designed to present clusters of daily data values (van Wijk and van Selow 1999). Each selected day on the calendar is painted with a color on the button face that corresponds to the color being used to plot the data cluster on the graph (see Figure 43).

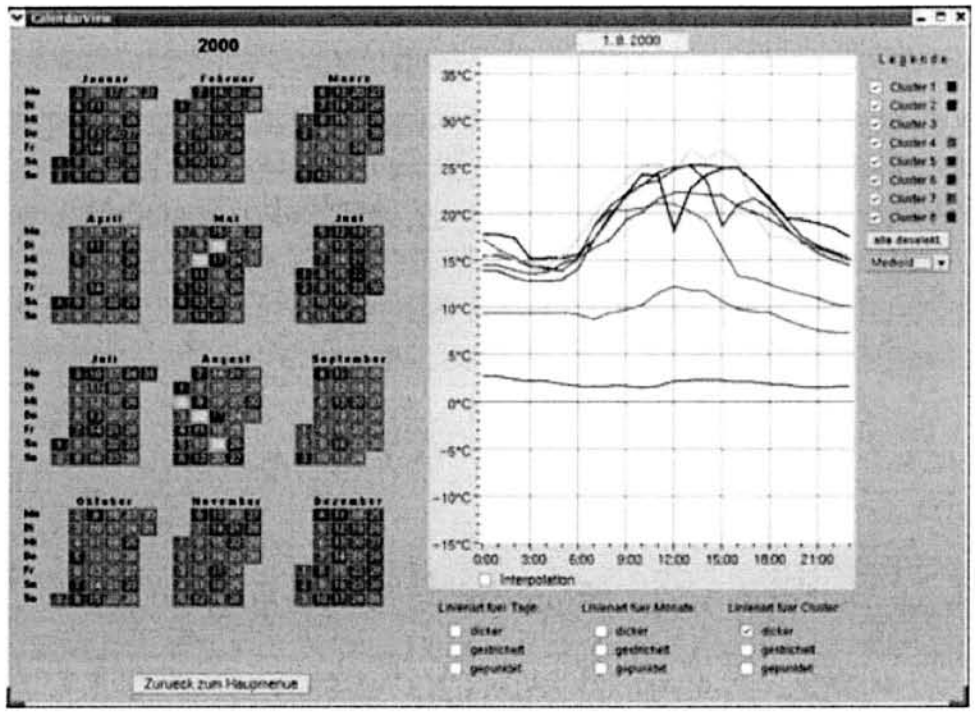


Figure 43: An illustration of the Calendar View visualization technique.

Other visualization techniques use well known objects as metaphors. Examples of these techniques are the *Lexis Pencils* (Keiding 1990; Brian and Pritchard 1997), and the *SpiraClock* (Dragicevic and Huot 2002) to display a continuous outlook of nearby events. Regarding the last one, Figure 44 presents a Spiraclock shown at different times. The middle picture shows that the short event will occur in 5 minutes (9:40) and the other one in 35 minutes (10:10).

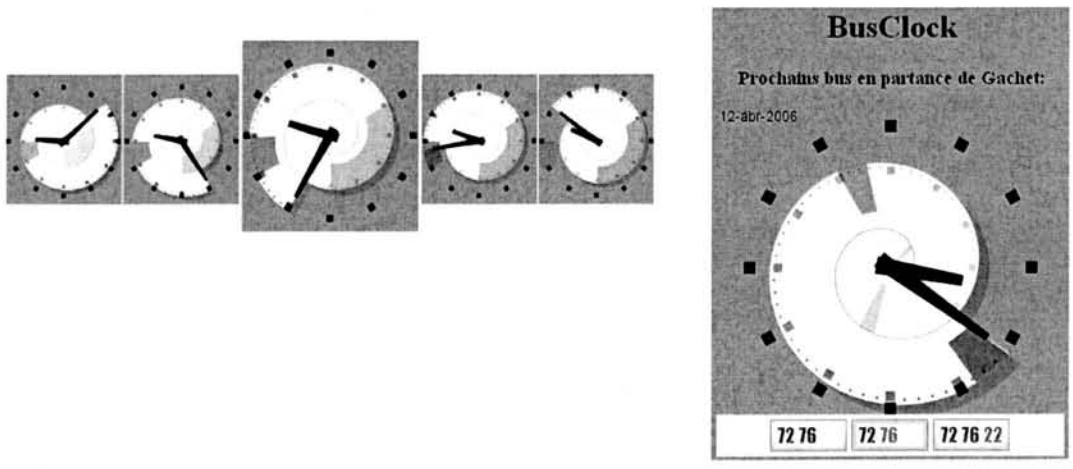


Figure 44: An illustration of the Spiraclock visualization technique.

Other approaches for displaying time-dependent data rely on linear axis-based visualization. More conservative approaches use only one time axis per variable, for instance Parallel Coordinates technique, making it possible to show multivariate data by using several

time axis. Other techniques rely on only one time axis for representing a multivariate dataset. Examples are the *Time-Wheel* technique where the time axis is presented on the center of the display and other axis (one per variable) are arranged circularly around it (Tominski, Abello et al. 2004). Each variable is addressed a distinct color. For the representation of pairs <time, corresponding value> line segments are drawn between the time axis and the location at variable's axis corresponding to the value. For instance, Figure 45 presents a TimeWheel representation showing the quantitative evolution of 6 variables over the timeline (center axis).

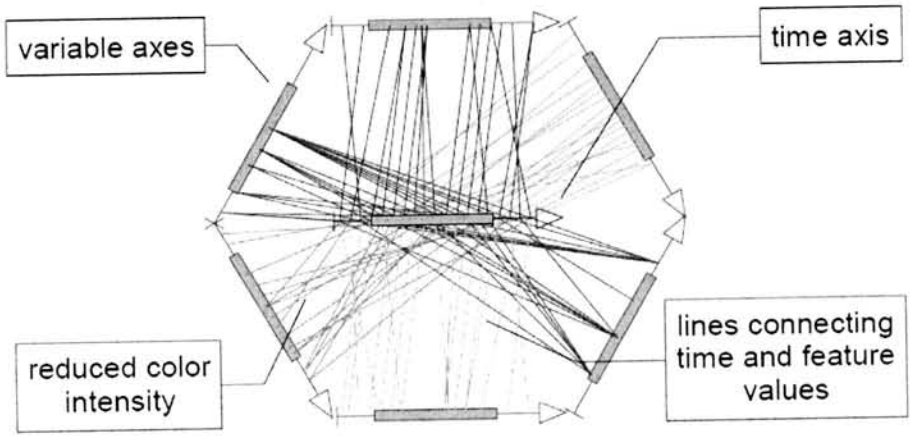


Figure 45: an example of a *TimeWheel* visualization.

These authors also propose another technique using multiple time axes, called *MultiComb*, where the main motivation is to make use of the expressiveness of line charts. In order to do so, this technique arranges plots radially, where each plot represents a depending variable and the dimension of reference directly. There are two variants of this technique: either the plots extend outwards from the graph central point, or the plots extend around the graph central point. For instance, in Figure 46, two MultiComb graphs are presented, where in the top graph the plot extends outwards from the central point and, in the bottom graph, the plots extends around the central point.

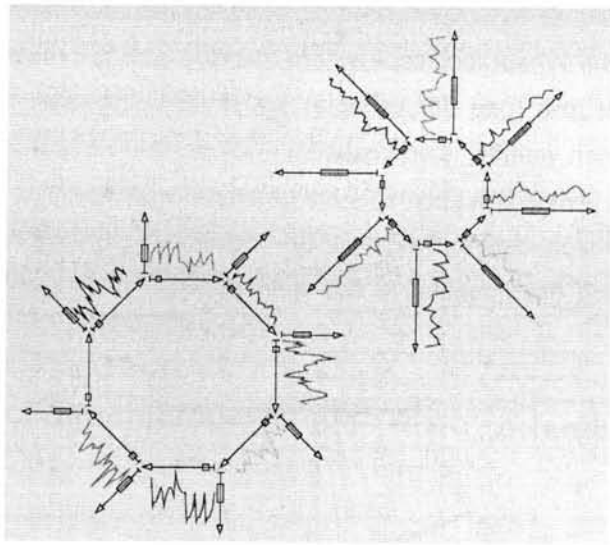


Figure 46: An illustration of the MultiComb visualization.

Interaction techniques are also present in *static* visual representations. Here the purpose of interaction is for the user to provide some change in the criteria that control the temporal aspects that underlie the visual representation of time-dependent data. For instance, two of the most well known interaction techniques are *temporal brushing* and *temporal focusing* (Monmonier 1990). *Temporal brushing* is based on the process of interactively selecting data items in order to render the appropriate graphical representation (Buja, McDonald et al. 1991; Tominski, Schulze-Wollgast et al. 2005). The first provides the means to brush a timeline in order to render the appropriate graphical representation, while the last restricts the dataset to be presented to a specific time frame.

4.10.2 Dynamic Visual Representations

Müller states that *dynamic* visual representations make direct use of time to depict some data. Here, the most natural approach is to map the temporal aspects of the data directly onto the time control of a dynamic representation, where the representation of data elements representation changes over the time (size, shape, color, texture, transformation of elements) (Müller and Schumann 2003). Possibly, a temporal scale factor can be regarded as another element of the representation, when only the relevant events of a narrative are to be shown; for instance, an animation of a few minutes showing a multi-million year collision between two galaxies.

For *dynamic* visual representations of time-dependent data it is important to know about the underlying data temporal model. Although there are several models of time available, as previously stated, most often the continuous linear time model or the discrete linear time model are adopted. Furthermore, in most cases, all the time-dependent data underlying the visualization is known before the dynamic representation takes place.

In fact, in much of the early work about map animation, time was regarded as linear and focused on techniques for generating map movies to represent change over time (Thrower 1961; Tobler 1970; Gould, DiBiase et al. 1990; Dorling and Openshaw 1992; Johnson 2004; Johnson 2005).

Considering *dynamic* visual representations of *continuous* linear time, most often data results from a time-dependent function *f(t)*, which is sampled in order to get data to be visualized. The degree of sampling depends on the desired rate of visualization. For instance, it may be suitable to sample with time-lapses, in real-time or in slow motion, depending on the overall temporal scope and the intuitive manageable rate of change.

On the other hand, for the *dynamic* visual representations of data available at discrete linear time steps, two approaches can be followed: to generate flashlight visualizations (snapshots) or to assume that the data available corresponds to a dataset sampled from a continuous linear temporal model, allowing the means to generate animations, and thus provide the perception of continuous movement. In order to do so, the time gaps between the sampled data must be appropriate for the animation purposes, regarding the overall data temporal scope of observation. To some extent, when the sampling rate is not enough, interpolation techniques can be applied, taking into consideration local temporal neighbourhood, needed to generate smooth animations from temporally sparse data (Acevedo and Masuoka 1997). Nevertheless, interpolation can have side effects such as aliasing regarding inadequate/insufficient underlying data.

Within the research stream regarding animation generation about changes in time, attention has also been directed to appropriate metaphors for representing change (Gersmehl 1990) and choice of symbolization methods for depicting changing phenomena with different spatial or attribute characteristics (DiBiase and MacEachren 1992).

There are, however, situations where changes in the data cannot be foreseen, since the temporal model is not linear and the events for which data is being captured have not occurred before the visualization starts. For instance, the visualization of time-dependant data from real-time simulation can carry this problem. According to Müller, the scenario of unpredictability about when events are going to happen and corresponding data will be created, raise problems in applying several techniques for *dynamic* visual representation, such as interpolation (Müller and Schumann 2003). However, general approaches from the field of information visualization can be applied in event-based visualization. For instance, Matkovic successfully applied level of detail and interaction techniques such as Focus+Context in real-time monitoring of processes (Matkovic, Hauser et al. 2002).

4.11 GEOVISUALIZATION

Geovisualization does not fit any of the taxonomies presented in section 4.4. Due to the physically-based nature of the underlying data, it is considered part of the scientific

visualization research domain. However, one of the goals of this work is to propose a new technique for the visualization of spatio-temporal information and, in order to do so, concepts from information visualization have been brought to the field of geovisualization. Moreover, many information visualization techniques and concepts are welcomed to support the visualization of spatial data overloaded with other multivariate data or the visualization of time-dependent spatial data. Hence, it is from this perspective that the remaining section addresses the field of geovisualization.

Spatial information regards any information about the shape and the relationships among entities that are associated with a location (known or able to be calculated), defined by coordinates addressed through a spatial reference system. Among the possible entities able to be spatially described are physical objects or phenomena (natural or artificial), administrative units or derived information (for instance, spatially referenced statistical information).

First spatial information visualization was supported on maps which are graphical representations for communicating geographically related information. The most ancient remaining of a map dates from 3800 b.C. and consists of a clay stone depicting water streams and other cartographic objects (Matos 2001). Through centuries, spatial information was used essentially for mapmaking, in cartography, being the last the science and art that broadly comprises all the steps needed to produce a map.

It is important to refer that in the past, maps were designed to be both the database and the presentation media. The advent of digital cartography and GIS, in the 1960s, split these tasks, giving birth to a new class of spatial information, geospatial and geographic information, which does not necessarily directly involve a graphical representation or map (OGC 2007). More generally, Geospatial and Geographic Information are mathematical representations (usually 2D or 3D data) whose geometry and object topology is described through the means of a reference coordinate system (which does not necessarily restricts to Earth surface). Toppe adds a temporal feature to the definition of geographic information which is the ability to suffer changes over time. According to this author, such changes can take so much time that they can be ignored or, in the other extreme, they can happen so fast that it is required to keep the rate of change.

Over the years, spatial information visualization has been considered a research stream within Scientific Visualization. The ground for this statement is the physically-based nature of spatial information. However, the visualization of geospatial and geographic information often requires integrated approaches from visualization in scientific computing (ViSC), cartography, image analysis, information visualization, exploratory data analysis (EDA), and GIS (MacEachren and Kraak 2000). Hence, for its unique requirements, the visualization of geospatial and geographic information is considered a research field by itself.

MacEachren first defined geovisualization as the use of concrete visual representations, whether on paper or through computer displays or other media to visualize spatial contexts and problems, so as to engage the most powerful of human information-processing abilities, relying on vision (DiBiase and MacEachren 1992). According to these authors, geovisualization takes advantage of interactivity and dynamism, inherent to it, to facilitate exploration, in a more effective and dynamic manner, than prior cartographic techniques (MacEachren, Boscoe et al. 1998). Later, MacEachren added that geovisualization emphasizes development and assessment of visual methods designed to facilitate exploration, analysis, synthesis, and presentation of geo-referenced information.

As previously said, geovisualization evolved from convergences in several disciplines, being the most relevant the Cartography, the Geographic Information Systems and the Information Visualization. However, before the establishments of such synergies, in the late 1980's and early 1990's, these disciplines evolved in parallel, meaning, closely related yet arguably independent to each other. For instance, efforts to develop geospatial data standards rarely included how to visualize the data, while computer graphics rendering libraries and standards evolved independently of geospatial data models. The lack of collaboration between both fields resulted in inefficiencies associated to geo-visualization, such as the registration of geospatial data within visualization software, difficulties to easily produce animation sequences within GIS environments and, perhaps more important, the lack of connections between the database and the visualization environment that supports display of geospatial data reliably (Rhyne, Ivey et al. 1994). According to these authors, as Information Visualization evolved and matured in the mid to late 1990's, geographers and cartographers began to actively participate in this field, since cartographic methods often depict the graphical representation of abstract and non-inherently spatial data.

Besides individual research contributions, the major contributes to the geovisualization field where mainly provided by three research groups (Rhyne, MacEachren et al. 2006), namely:

- Carto Project (from ACM Special Interest Group on Graphics), since 1996, aimed at examining how viewpoints and techniques from the computer graphics community could be effectively applied to cartographic and spatial data sets;
- Commission on Visualization and Virtual Environments (from International Cartographic Association), since 1999, aimed at addressing core research issues associated with extending cartographic methods into the visualization arenas;
- GeoVRML Working Group, since 1998, aimed at developing tools and recommending practices for the representation of geographic data using the Virtual Reality Modelling Language;
- X3D Geospatial Working Group (from Web3D Consortium's): focused on developing tools and recommended practice for the representation of geographic data using X3D. The

goal of this group is to enable geo-referenced data, such as maps and 3D terrain models, to be viewed in non-spatial applications, for instance, web browsers, as well as to integrate spatial and non-spatial data through web services.

Until 2006, these groups focused their research on four major aspects of geovisualization, namely, the representation of geospatial information, the integration of computational and visual geographic and cartographic methods, the creation of effective interface designs for geovisualization tools, and the study of the usability of geovisualization environments (MacEachren and Kraak 2000), (Rhyne, MacEachren et al. 2006).

Regarding time-dependent data that has an inherent physical component, such as in geospatial datasets, the geovisualization community has been developing new visualization techniques, based on concepts that often come from other research areas such as Information Visualization, Scientific Visualization and Cartography (Rhyne, MacEachren et al. 2006).

Among these techniques there are new possibilities to visually represent information through the use of animation. About these possibilities, MacEachren *et al.* (MacEachren, Boscoe et al. 1998) consider animation a well-suited visual tool for qualitative analysis and Harrower (Harrower 2002) states that the potential power of animated maps lies in their ability to represent change over time and thus facilitate an understanding of process rather than state. Preliminary research of the use of animation in geovisualization showed that animation can reveal subtle space-time patterns that are not evident in static representations, even to expert users who are highly familiar with the data (Patton and Cammack 1996). On the other hand, as the benefits of using dynamic map displays is not consensual in the geovisualization community, work is being performed to assess the effectiveness of these representations for learning, knowledge discovery and knowledge construction (Fabrikant and Goldsberry 2005).

Exploring animation, several techniques have been proposed in the context of geovisualization, such as observations, simulated walkthroughs and fly through over 2D/3D landscapes or cityscapes, performed in real-time or in differed time (Fabrikant and Josselin 2003; Tominski, Schulze-Wollgast et al. 2005; Harrower 2007; Harrower and Sheesley 2007; Harrower and Fabrikant 2008). Also, in order to improve the visual stimulation in animation, several authors have addressed the use of dynamic variables such as those proposed by DiBiase *et al.* (DiBiase, MacEachren et al. 1992), namely duration, order and rate of change, or such as those variables proposed by MacEachren *et al.* (MacEachren 1994), namely frequency, display time and synchronization.

However, one key issue for the success of the use of animation in geovisualization is the empowerment of the user with control over the animation and its flow (Koussoulakou and Kraak 1992). In fact, the increase of the user participation in the process of exploring the time-dependent geospatial data has resulted, in the last decade, in a paradigm change. Representation became part of a user-centered process of data exploration and analysis,

where highly dynamic and interactive representations of complex datasets and their temporal features can be rendered (Döllner 2000). In this process, according to Edsall *et al.* (Edsall, Andrienko et al. 2008), the user is able to incorporate his own abstractions and to refine mental models that lead to better exploration of data and the detection of previously hidden patterns and features.

4.12 Conclusions

This chapter has presented important contributions on information visualization - an interdisciplinary active field of research, addressing the study of the visual representation of large-scale collections of data and the use of graphical techniques to help the effective observation, understanding, exploration and analysis of data. The aim of information visualization is to offload the cognitive process to visual perceptive system by visually representing large amounts of data in such a way that the observation of data can be an adequate mean to effectively discover and provide explanations about patterns, individual or groups of information items and to effectively support the decision making process.

In order to accomplish this goal a sequence of stages must be performed, designated by the visualization pipeline, in a process that goes from the raw large dataset to the final visual representation. Each of the stages must be thought in the context of how to communicate complex ideas in a clear, precise and efficient manner while complying with usability and goal-driven requisites such as those presented by Tufte (Tufte 2001) and by Maletic (Maletic, Marcus et al. 2002).

There are distinct perspectives on the structure of the visualization pipeline since authors consider distinct stages. For instance, (Tory, Campo et al. 1997) divided the pipeline into three stages, Chi (Chi 2000) considers 4 data stages and 3 process stages. Santos and Brodlie replace the pre-processing of data by two separate components, distinguishing between data analysis and filtering; Carmo and Cunha (Carmo and Cunha 1998) go further by suggesting that filtering operations can happen in distinct steps of the visualization process. These authors argue that the filtering step can reduce the information amount, but can also generate new information by building new variables that result from functions applied to previously existing variables.

In short, the distinct perspectives on the stages that compose the visualization pipeline aim at transforming raw data into manageable, meaningful and more easily to interpret visual representations that make no distortions of the represented values in what concerns the original data. However, the accomplishment of this task is complex, especially in the presence of large datasets and in scenarios where the user is allowed to have detailed information of a particular item while keeping available a representation of the informational context.

For this purpose techniques have been researched that provide micro/macro visualizations, designated by Focus+Context, which allow the presentation of great amounts

of complex information, with high levels of detail and organization, through multiple and frequently hierarchical levels of contextual visualization. Generally Focus+Context fall into three approaches, namely traditional pan/zoom, multiple windows and distorted views. In the context of distorted views, the *Fisheye Concept* presented by Furnas (Furnas 1986) is a technique motivated by the idea of a mental map that the user creates, according to subjective perception of the information being displayed. Hence, each information item being presented is assigned a number resulting from a degree of interest function chosen by the user (user defined criteria). This function provides the expression of how much the user is interested in, for each information item. Filtering takes place when the user defines a numerical threshold from which items will be rendered. Calculations take place at the pre-processing stage and no assumptions are made regarding the mapping stage, meaning that the mapping strategy is independent of the presented model in what concerns how to map an information item based on its calculated degree of interest.

However, each of the Focus+Context approaches can present their own drawbacks. For instance, traditional pan/zoom techniques suffer from the keyhole problem; multiple window techniques suffer from the extra space required to show the overview and the user effort to mentally integrate detail and context; distorted view methods can introduce issues of focus, an issues of transition between views and the potential user disorientation in case of excessive distortion. Another common approach in the presentation of multiple levels of detail is the use of multiple representations for single or groups of information items.

One or more of these approaches are used in a considerable number of information visualization techniques, each of these aimed at providing an effective solution for a particular visualization problem. In the 1990's, as the number of proposed information visualization techniques was becoming considerably large, researchers felt the need for taxonomies that enabled the classification of techniques, and that provided an understanding of the differences between designs, in order to suggest new possibilities. However the wide range and variety of techniques available at that time did not allow the proposal of a single classification system, but of several systems (Santos and Brodlie 2004). These systems address the problem of classification of information visualization techniques according to distinct perspectives. Hence, there are taxonomies that classify techniques depending on (a) whether or not deal with inherently graphical data; (b) a set of subcategories divided by data-oriented goals: scientific visualization, GIS, multi-dimensional plots, multidimensional tables, information landscapes and spaces, node and link, trees, and text-forms; (c) the dimensionality of information; (d) the underlying primary data type, among others.

In order to present a wide set of information techniques in this chapter, a data structure approach has been adopted, which is adapted from Shneiderman taxonomy. Hence, techniques have been presented according to the following classification groups: linear structures, hierarchical structures, networked structures, vector spaced structures, multidimensional multivariate structures and finally time-dependent structures. The reason

why time-dependent information been detached from multidimensional structures is that the term multidimensional refers to the dimensionality of the independent variables, while the term multivariate refers to the dimensionality of the dependent variables of a data set (Bergeron, Cody et al. 1993; Wong and Bergeron 1997).

In time-dependent structures, the parameter time is predominant and subdues the dimension for other associated variables, leading to the consideration that this class of structure corresponds to multivariate data over time. The visualization of multivariate data over time allows the analysis of multivariate correlations, which can be of special interest. However, according to Müller, visualization of multivariate data over long periods of time requires a special effort (Müller and Schumann 2003).

For time-dependent information visualization, it is possible to state that there are a considerable number of successful examples of its practice since the 18th century: *La Carte Chronologique, A Specimens of a Chart of Biography,* Lambert graphical analysis of periodic variation, Napoleon's Russian campaign of 1812, Ibry's diagram of train schedules are examples of it. The success of these examples relies on the fact they use adequate methods for mapping and representing the time-dependent data at stake.

Currently the number of time-dependent information visualization techniques is vast and classification systems, such as the taxonomy proposed by Aigner *et al.* (Aigner, Miksch et al. 2007) are required, in order to provide a systematic categorization for the graphical representation of time-oriented data analysis. These authors developed a classification that, at a top level, depends on the nature of time (the underlying time model), the source data (abstract or physically-based), and the representation (static or dynamic).

In this chapter, the presented techniques for time-dependent information have been grouped according to the representation criteria, more specifically in static and in dynamic visual representations.

Hence, regarding static visual representations, it can be said that they benefit from having the ability to allow the conclusion of quantitative facts as well as explore special data features, data values, time steps and positions according to the underlying time scales, without limitations. However the application of these techniques has most often restricted to depict one variable or a limited number of variables, whereas the analysis of multivariate correlations is reduced. Such limitation can be reduced through the use of more visual variables which allow the compression of more information onto a single chart. Examples of static representations are the *Time Series Graph, Stacked Bar Chart, Parallel Coordinates*, or in the case of spatial data, *dance maps*, *change maps* and *chess maps*. Examples of multivariate static representations are the *ThemeRiver*, *Spiral Graph*, *Calendar View*, *Lexis Pencils*, *Time-Wheel*, and *MultiComb*, among others.

Regarding dynamic representations, they make direct use of time to depict data. The most natural approach is to map the temporal aspects of the data directly onto the time control of a dynamic representation, where the representation of data elements changes over the time (size, shape, color, texture, transformation of elements). For this representation, an important property regarding the underlying data is critical to be known, namely, the time model (such as, for instance, if it is a continuous linear time model or a discrete linear time), among others. Depending on the model, distinct approaches can be followed in the visualization of temporal data. A common case is the dynamic visual representations of data available at discrete linear time steps, for which two approaches can be followed: to generate flashlight visualizations (snapshots) or to assume that the data available corresponds to a dataset sampled from a continuous linear temporal model. The second approach allows the generation of animations providing the perception of continuous movement. Here, the time gaps between the sampled data must be appropriate for the animation purposes, regarding the overall data temporal scope of observation. Sometimes interpolation techniques can be used to generate smooth animations from temporally sparse data. However, the use of interpolation can have side effects such as aliasing regarding inadequate/insufficient underlying data.

Finally, this chapter addresses the interdisciplinary field of Geovisualization which is aimed at the visual presentation of spatial structures having an inherent form of representation due to their physical characteristics, such as it happens with spatio-temporal data. If the definition was limited to the previous statement, geovisualization would fall into the field of scientific visualization. However, in geovisualization, abstract-based information can be overloaded onto spatial structures and information visualization techniques can be successfully brought into this research area.

Among the several possibilities to visually represent spatio-temporal information, the use of animation techniques has been studied. On one hand, it is considered a well-suited visual tool for qualitative analysis and provides the ability to represent change over time and thus facilitate an understanding of process, rather than of state. Preliminary research of the use of animation in geovisualization showed that animation can reveal subtle space-time patterns that are not evident in static representations, even to expert users who are highly familiar with the data. On the other hand, other authors may disagree with the total effectiveness of dynamic map displays, making the issue depend on the interaction and data-exploration tools that are available to the users to explore data. These authors argue that further studies are required on the effectiveness of animation techniques for learning, knowledge discovery and knowledge construction.

A wide range of animation techniques have been proposed in the context of geovisualization, such as observations, simulated walkthroughs and fly throughs over 2D/3D landscapes or cityscapes, performed in real-time or in differed time. Some of these techniques also focus on the improvement of the visual stimulation in animation through the

use of dynamic variables such as those proposed by DiBiase *et al.* (DiBiase, MacEachren et al. 1992), or those variables proposed by MacEachren *et al.* (MacEachren 1994).

The contents presented in this chapter intend to provide the background for chapter 5 which addresses the second important contribution of the current work, namely the interactive visualization of spatio-temporal data that is gathered from a STDBMS. To achieve such visualization, a model is defined that uses several of presented concepts, emphasizing, Focus+Context distorted view techniques, the *Fisheye Concept*, and the dynamic representation of time-dependent information considering a linear discrete time model.

5 INTERACTIVE VISUALIZATION OF SPATIO-TEMPORAL INFORMATION

The last decades have witnessed significant developments in the technological support for information management, from data gathering to information visualization and analysis. Information systems supporting a wide variety of application domains have been encompassing contributes of many scientific areas, as a response to the increase of requirements from these systems.

One of these requirements is the efficient management of information over time. This is a consequence of the intrinsic temporal nature of events, whether they result from natural causes or from human activity. From the perspective of data management, alphanumeric and spatial databases have provided solid and robust means of keeping track of the most recent state of data for any particular application domain.

However, many applications require that, besides the most recent state, previous temporal states are also kept. Alphanumeric and spatial databases have pushed the burden of such temporal data management to the application layer, due to the lack of proper temporal data types, operators and functions, indexes and query language extensions. This has lead to undesired complexity and performance problems.

The last decade has witnessed the fast development on temporal data management, resulting in approaches to reduce most of the complexity and performance problems referred above. Several temporal database management system prototypes now offer support for *valid time* (Snodgrass, Böhlen et al. 1996), *transaction time* (Snodgrass, Böhlen et al. 1996), continuous or discrete change, as well as historic-temporal data or predictive data. These accomplishments have a direct impact on how information is represented, kept and queried (Jensen and Snodgrass 1999). Among other application domains, those that handle geographic and geospatial information can benefit from temporal database support, to allow efficient representation, management and access of information on the evolution of spatial objects.

To this purpose the current work has surveyed spatial and temporal database management systems in Chapter 2 and a STDBMS – *Action!* - has been proposed, in chapter 3. Through the spatio-temporal SQL language that is provided by the *Action!* the user is able to issue statements concerning the representation, storage and management of spatio-temporal data. But perhaps the most important fact is that the user is able to use the power of ATSQL2+Oracle Spatial to issue queries that contain spatial and temporal constraints, which perform over temporal relations and resulting, most of the times, in spatio-temporal record sets.

However, the new possibilities provided by the spatio-temporal data management system raise interesting issues regarding the visualization of spatio-temporal information directly supported by database management systems. A result from a database query may consist of a spatio-temporal record set or, in other words, may contain the spatio-temporal evolution of distinct objects, where the characterization of each object is described by a group of several records time stamped with *valid time*. Here, each record provides the description of the object for a particular interval of time.

One problem associated to the contents of a spatio-temporal record set, such as the one just described, is the visualization of its contents, since a graphical representation that provides the same highlight to all records, despite their time attributes, can be difficult to interpret. On the other hand, other approximations, based on a chronological visual representation, would focus on one instant at a time, and by doing so, they would be unable to give the observer an idea of the temporal context.

In order to overcome some of the limitations found in the visualization of spatio-temporal record sets, the current chapter proposes and describes a temporal Focus+Context visualization model. This model is based on the concept of a *Temporal Degree of Interest* (TDOI) associated to each record, which numerically expresses how much the user is interested in viewing the corresponding object. To this purpose, the TDOI of a record is based on the record *valid time* attribute and on user-defined functions which are specified according to application-specific or domain-specific goals related to data presentation, exploration and analysis. In the mapping stage of the visualization pipeline associated to this model, the temporal degree of interest is used in the control of graphic variables, such as transparency and color, or to control more complex rendering attributes, such as sketch drawing for edges or other non-photorealistic enhancement techniques (Hertzmann 2003). By carefully defining the functions that are used in the TDOI calculation, large amounts of data from distinct periods of time and from several record sets can be compressed onto the same dynamic presentation as they are depicted with distinct graphical configurations.

This visualization model has been tested with three historic-temporal datasets. An informal evaluation of results shows that the proposed approach is able to capture the relevant aspects of their domains.

According to the above described, this chapter is organized as follows: Section 5.1 presents the visualization problem of time-dependent data gathered directly from a spatio-temporal database. Section 5.2 presents the proposed F+C visualization model, while Section 5.3 addresses the Temporal Degree of Interest definition and calculation. Section 5.4 addresses the definition and application multiple TDOI functions, while section 5.5 describes how the TDOI calculated values may be used to graphically depict the temporal features of data. Next, section 5.6 describes the application of the proposed model, while section 5.7 instantiates the model into three historic-temporal datasets, used as tests scenarios. Finally,

Section 5.8 presents the main conclusions about the proposed Temporal F+C visualization model.

5.1 PROBLEM DEFINITION

Currently, temporal information systems provide temporal support that goes much beyond data time stamping: temporal SQL language implementations feature upward compatibility and temporal upward compatibility (Steiner 1998), sequenced semantics and non-sequenced semantics, as well as temporal functions and operators (Allen 1983). Among other features, state-of-the-art systems make it possible to issue queries to a temporal database using temporal restrictions which, depending on the query, may result in a spatio-temporal record set.

The work presented in chapter 3 constitutes a successful attempt to provide spatio-temporal data management, starting with a spatial DBMS and enhancing it with temporal support such as sequenced and non-sequenced semantic queries, *valid time* and *transaction time* domains, temporal functions and temporal operators. To this purpose, ATSQL2 and Oracle Spatial have been integrated onto a single STSQL language, able to provide the referred temporal support. Other important fact is that the proposed language and the corresponding prototype system provide the ability to keep legacy applications working, such as applications which require standard SQL or the Oracle Spatial extension.

Based on what has been said in chapter 3 about the adopted representation scheme (Snodgrass tuple time stamped representation scheme), in the sequel, the schema of a spatio-temporal record set is considered to have alphanumeric and spatial attributes, plus a special temporal attribute usually known as *valid time* (*vt*). The concept of *valid time* is used in temporal databases to denote the time period during which a database fact was, is, or will be valid in the modelled reality (Snodgrass, Böhlen et al. 1996). The domain of *valid time* is the set of intervals, represented by their endpoints, and the adopted notation is

$$\left[i_{start}, i_{stop}\right[, i_{start} \in N, i_{stop} \in N, i_{start} < i_{stop} \qquad (1)$$

where endpoints represent time instants from a time domain that is linear and isomorphic to the natural numbers. Moreover, the left endpoint of an interval is considered included in the interval while the right endpoint is excluded, as depicted in equation 1. This notation allows that an instant i_k can be used to specify the right endpoint of an interval and the left endpoint of another interval that immediately starts after the first interval.

Regarding the visualization of information related to spatio-temporal record sets, one of the most commonly used techniques is to continuously render the evolution of its data over the time. This can be achieved by narrowing the focus of interest to a single instant of time and by displacing the focus towards a direction (past or future), as illustrated in Figure 47.a. In

this figure, the four horizontal lines represent the *valid time* characterization of the corresponding records

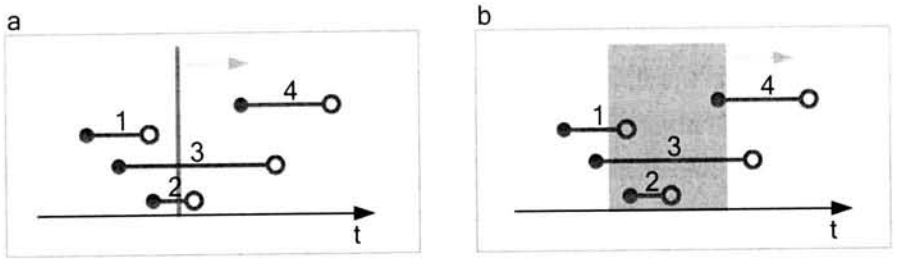


Figure 47: Instant focus and interval focus with no context.

If the focus instant, represented by the vertical line, temporally intersects the *valid time* of one record, the corresponding information is displayed; otherwise it is hidden. A direct consequence of this technique is that rendered objects appear very abruptly, with no smooth transitions, as the corresponding record temporal interval gets intersected by the focus; in the inverse direction, objects disappear abruptly as the focus gets out the corresponding record *valid time*.

The previous situation can be extended by the enlargement of the temporal focus. This situation corresponds to the user being interested on an interval of time, rather than on an instant, as shown in Figure 47.b. However, if the display criteria continue to be defined by the intersection of the temporal focus with each record *valid time*, the four records would be displayed with equal degree of interest, leading to incorrect user interpretation: records 1 and 4 get their information displayed, but mislead the user to consider that the corresponding events occur in the same interval of time. It may also lead to apparent spatio-temporal inconsistencies, such as superimposing two distinct objects that exist at non-overlapping periods being superimposed, or the depiction of the same object in several distinct places. Also, the abrupt show/hide as the *valid time* of records intersects/does not intersect the temporal focus continues to be a problem. The above technique concentrate on displaying information intersecting the temporal focus and give no clue about the temporal context, such as what is ceasing to exist or what is starting to exist, considering a temporal focus.

Hence, to provide effective support for the observation, interactive exploration, analysis and presentation of spatio-temporal information, directly gathered from a spatio-temporal database, better visualization techniques are required. In order to do so, the current work proposes the Temporal Focus+Context (F+C) Visualization Model, where the basic idea is to provide visual clues about time in questions such as those suggested by MacEachren (Maceachren 2004), regarding temporal existence, temporal location, temporal interval, rate of change, sequence and synchronization. In this work, it has been considered that it is possible to account for the rich complex relations between the temporal features of records and the focus of interest defined for the visualization. For instance, in Figure 47.b, a record

could be rendered with distinct degrees of interest by taking into consideration how much of its *valid time* intersects the temporal focus.

5.2 THE TEMPORAL F+C VISUALIZATION MODEL

As justified above, a visualization technique for time-dependent data should not limit itself to show and hide data from records. Instead, it must be able to render the data differently, according to some "degree of interest". The visual representation of the data from a record can be further enhanced with clues about time, by controlling graphical variables used in the rendering of such data, according to the record calculated temporal degree of interest. These visual clues are intended to help in the identification of temporal features, such as:

- The time span of a data element;
- Whether a data element starts or ends to exist at a given time;
- The data elements that co-exist in a selected time period.

In order to quantify these visual clues, we propose a model that builds on ideas from Information Visualization such as the Fisheye Concept (Furnas 1986), an information distortion visualization technique. According to Tory and Möller (Tory and Möller 2004) fish-eye lenses magnify the center of the field of view, with a continuous fall-off in magnification, towards the edges of the image. Here, degree of interest values determine the level of detail to be displayed for each item and are assigned through user interaction.

Card *et al.* classified Fisheye as a Focus and Context (F+C) visualization technique (Card, Mackinlay et al. 1999), based on three premises:

1. The user needs both, overview (context) and detail information (focus);
2. Information needed in the overview may be different from that needed in detail;
3. The two kinds of information, overview and detail, can be combined within a single dynamic display, much as it is perceived by the human vision system.

The approach adopted in the Temporal Focus+Context Visualization Model is based on the same concepts, but transposes them from the spatial to the temporal domain: the focus is the time period which has the user attention, while the context consists of two periods, immediately before and after the focus period, extending towards the past and the future respectively. Focus and context, together with specific user requirements, are quantified in a TDOI calculated for each record. This is further transformed, through mapping functions, into values that control graphical properties. These mapping functions can emphasize the representation of data on the temporal focus, while maintaining visible data on the temporal context. Graphic properties such as overall color, tint color, texture, transparency, scale or sketch drawing, can be used to render 3D geospatial objects in such a way that they improve the user understanding of the temporal properties of the record set.

5.3 TEMPORAL DEGREE OF INTEREST

The Temporal Degree of Interest (TDOI) for a record *r* of a temporal record set is expressed in the following function:

$$TDOI = f(I, vt, FVT, FI) \tag{2}$$

where

- *I* is the overall time period under user analysis; it is represented by an interval, defined by its endpoints $[i_{start}, i_{stop}[$.
- *vt* is the *valid time* of record *r*;
- *FVT* is a user-defined function that returns the *filtered valid time* (*fvt*) for record *r*;
- *FI* is a user-defined function that describes the temporal degree of interest for every instant in time period *I*;

In the TDOI definition, *I* and *vt* provide the means to determine if the TDOI for record *r* is nonzero. Period *I* reflects the period the user is interested in and depends on the application domain and the user context. *FVT* is a selectable function which allows the user to specify what feature of *vt* should be considered in the TDOI. For record *r* this function consistently returns *vt* or a subset of it. For instance, the following expression represents an *FVT* function, *start*, which returns an interval containing the start instant of *vt* (and taking into account the domain of time instants defined in section 5.1).

$$start(vt) = [i_{r.start}, i_{r.start} + 1[\tag{3}$$

By selecting this *start* function, the TDOI calculation for record *r* is based on the start instant of the record *valid time*, disregarding other instants in this attribute.

Finally, the *FI* function allows the user to quantitatively define the degree of interest for every time instant contained in *I*. For instance, a constant function must be used if the user gives the same degree of interest to all the instants contained in *I*: in another way, an increasing function must be used if more degree of interest is given to the starting or ending instants of *I*. More complex functions may be defined to better represent the user needs in terms of degree of interest distribution over *I*.

The computation of TDOI for a record *r* starts with the calculation of the *fvt* interval based on the selected *FVT* function and the *vt* attribute; next, the result is used to calculate an effective Period of Interest (*PI*) for the record *r*. *PI* corresponds to the temporal intersection ($\cap$) of the previously determined interval *fvt* and the user time period of interest, *I*:

$$PI = fvt \cap I$$

Assuming that *PI* and *I* are non-empty intervals, the TDOI value for record *r* is obtained as:

$$TDOI = \begin{cases} FI(PI_{r.start}), PI_{r.stop} - PI_{r.start} = 1 \\ \dfrac{\int_{PI_{r.start}}^{PI_{r.stop}} FI(t).dt}{\int_{I_{start}}^{I_{stop}} FI(t).dt}, PI_{r.stop} - PI_{r.start} > 1 \end{cases} \quad (4)$$

otherwise TDOI gets value 0. In expression (4), the resulting TDOI corresponds to a ratio between the integral of function *FI* for period *PI* and the integral of function *FI* for period *I*.

Figure *48* gives an informal graphical representation of this.

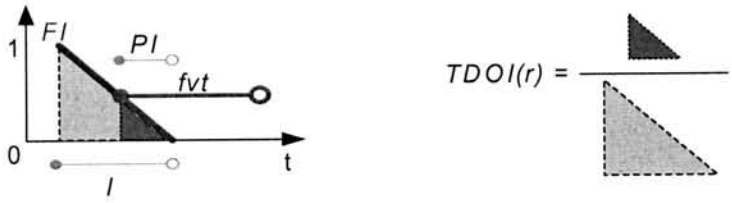


Figure *48*: Illustration of the TDOI calculation.

Therefore, the smallest TDOI (value zero) corresponds to records for which *PI* is empty. On the other hand, the largest TDOI (value one) corresponds to records whose *PI* is matched by the user's time period of interest *I*.

5.4 Multiple TDOI Functions

With this model, the user can define more complex behaviours to express the degree of interest. This can be achieved by defining multiple TDOI functions, *FI*, each function being defined for a distinct non-overlapping *I*, as illustrated in Figure 49.

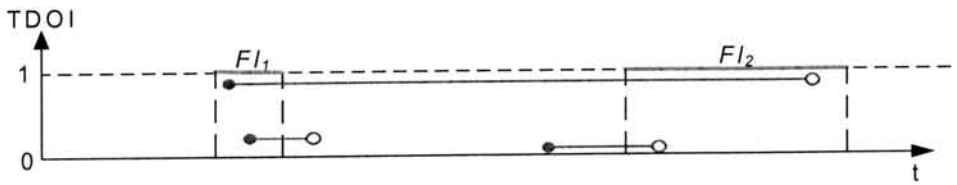


Figure 49: Two constant functions used in the TDOI calculation for three records.

Multiple TDOI functions provide an effective mean to compress, in a single visualization, spatio-temporal data from distinct non-overlapping periods of time. For instance, it is possible to simultaneously display the urban scenario for the same location at two or more distinct periods of time and, therefore, to allow the user, at each time period, to analyse what exists, what changes, and what remains.

However, results are better understood if the visual representation of each record accounts for which of the multiple functions actually contribute to the final TDOI value of a record. Here, graphical properties related to the visual appearance of the contents of each record can be controlled either by the overall TDOI (the sum of the TDOI resulting from every defined function) or by one specific TDOI function.

An useful possibility resulting from multiple TDOI functions can be found in interactive contexts, where the user slides the time periods *I*. As time period *I* starts to intersect the *fvt* of a record, the corresponding TDOI increases, reaches a maximum, and finally decreases as the intersection of *I* and *fvt* gets reduced. Having this behaviour in mind, the increase and decrease behaviour of TDOI can be better specified by defining two or more TDOI functions. For instance, in Figure 50, as *I* is slided by the user, I_1 and I_3 intersect a record *fvt* and, consequently, FI_1 and FI_3 permit a more fine-grained control over the increase/decrease behaviour of TDOI for the record.

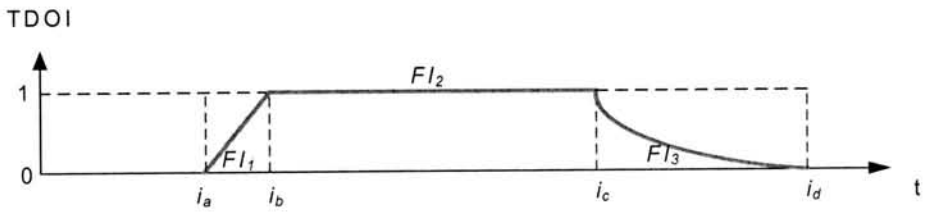


Figure 50: Multiple TDOI functions.

Expression (4) may also be used to compute the TDOI based on several functions. However, it must be taken into account the span (also known as length or duration of an interval) of *I* for each TDOI function, $\#(I)$ and, consequently, to obtain the corresponding accumulated span, *A*, given by the following expression:

$$A = \sum_{j=1}^{n} \#(I_j) \qquad (5)$$

The following expression defines the TDOI value for record *r* based on *n* TDOI functions:

$$TDOI = \sum_{j=1}^{n} \left(f\left(I_j, vt, FVT, FI_j\right) \times \frac{\#(I_j)}{A} \right) \qquad (6)$$

From expression (6), every parcel *j* that contributes to the TDOI value of record *r* gets its value weighted by the relative weight of its span, $\#(I_j)$, with respect to the accumulated span *A*.

In this way, while expression (4) takes into account how a TDOI function relates to the *fvt* of a record, expression (6) ensures that the contribution of each TDOI parcel to the overall TDOI is balanced. This also means that even with multiple TDOI functions, the TDOI calculated values are still kept within the interval [0;1].

5.5 Mapping of the Temporal Degree of Interest

The previous sections has described the model that is responsible for the calculation of a numeric value, for every record in a spatio-temporal record set, which expresses the temporal degree of interest according to the user visualization requirements. The corresponding set of operations happen at the filtering stage, considering the visualization pipeline depicted in Figure 51. This figure illustrates the visualization pipeline associated to the proposed Temporal Focus+Context Visualization Model along with the user participation (shown in brackets) in the visualization process.

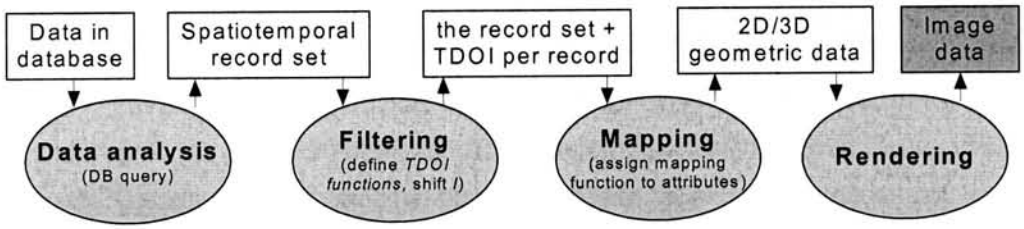


Figure 51: The Temporal F+C visualization pipeline.

After the filtering stage, the user can assign a mapping function to one or more attributes that are part of the spatio-temporal record set schema. Generally speaking, a mapping function is considered a previously defined operation that takes a particular content (alphanumeric, geometric, etc.), retrieved from an attribute of a record, and transforms it according to the record calculated TDOI.

It is important to note that the goal in the development of a mapping function is the clear and unambiguous display of one or more temporal features regarding the data being rendered. To improve this goal, the development of a mapping function should take into account the feature type being rendered, as well as specific application domain constrains and usages.

In order to display the temporal features, the TDOI can be used in the control of the graphic variables associated to the visual representation of the contents. The more illustrative examples of these graphic variables are the degree of transparency and the color of an object. For instance, a mapping function can use the TDOI associated to a record to change the degree of transparency of the 2D/3D representation of an object. In this way, the TDOI can be directly mapped to the degree of transparency, having objects with maximum TDOI displayed as opaque and objects with minimum TDOI represented as transparent. Similarly, the color used in the representation of an object can be manipulated in order to reflect the TDOI. For instance, objects with maximum TDOI can be displayed with their original color while objects with intermediate values of TDOI can be displayed with a combination of a pseudo-color that encodes the degree of interest. Furthermore, the use of pseudo-color can also provide visual results that allow the identification of the temporal location of an object

regarding the focus of interest: if the valid time corresponding to an object intersects the context period regarding past or the context period regarding future.

Hence, more elaborate mapping functions can be developed that simultaneously control several graphic variables such as transparency and pseudo-color, or more elaborated variables such as sketch drawing, edge enhancement, blur effect, texture modification as part of possibilities can be provided by non-photorealistic rendering.

The study of the effectiveness on the usage of one or more of these variables can be regarded as a possibility of future work that should take into account the characteristics of the human visual system and the application of visual perception models.

5.6 Application of the Temporal Degree of Interest

The proposed temporal F+C visualization model has been thought in the context of spatio-temporal information visualization based on spatio-temporal record sets that result from queries to a database. In this context, the user formulates one or more database queries, using a spatio-temporal SQL, which are then evaluated by the STDBMS. For each database query, the corresponding results are retrieved, as a record set, most of the times containing the alphanumeric and the spatial representation of a set of objects across a time period. Here, the evolution of each object in a record set is represented by a set of records where each record contains the *valid time* for which the facts in the record are considered to hold. After the query results are retrieved, it is the user's responsibility:

- For each record set, to select which of the attributes in the schema that should be displayed, as part of the visualization results. This selection is performed through the association of a mapping function to an attribute. The user can also define one of the attributes in the schema, not necessarily being displayed in the visualization results, as the key to the record set. In this situation, objects can be identified as being represented though groups of records sharing the same key value and each group being the representation of the temporal evolution of an object.
- According to the visualization requirements, to define one or more TDOI functions. These will be used in the calculation of the TDOI for each record.

The visualization alternatives provided by the proposed model are considerably large, but mainly depend on the user's visualization requirements, expressed through *I*, *FVT*, *FI* and the selected mapping functions. Furthermore, the model has also been thought to permit dynamic visualizations where, resulting from user interaction, the time period of interest *I* is sequentially or randomly shifted across the time.

For dynamic visualization of results, the TDOI values are recalculated in the filtering stage for every record whose *fvt* intersects *I* (*PI* contains at least one instant), each time *I* is shifted. The variation of TDOI (ΔTDOI) that results from this recalculation can also constitute a

valuable piece of information, since it describes if the TDOI of a record is increasing, not changing or decreasing, regarding a particular *I*.

In the mapping stage, the calculated TDOIs are then remapped into graphic variables used in the representation of the record data, in order to depict dynamic variables regarding spatio-temporal features such as sequence, duration and temporal co-existence.

The proposed model is also able to provide results for simpler visualization scenarios. For instance, the visualization of temporal evolution showing one instant at a time is achieved by setting the span of *I* to a single instant and by shifting *I*. The displacements may be constant or not, depending on the user needs for a linear or non-linear travel throughout the time scale. In the visualization of this kind of temporal evolutions, TDOI values alternate between the maximum and minimum values of the TDOI scale, without assuming intermediate values. The reason is that the *fvt* of a record set either does contain the instant *I* or it does not.

Another interesting configuration is where the user defines a longer time period *I*, allowing the compression of information temporally intersecting *I*, to make it visually coexist. Although coexisting in the same defined time period, the degree of interest of the distinct records depends on how *I* touches *fvt*, on the span of *I* and on the span of *fvt*.

For instance, consider the example illustrated in Figure 52, where the value of TDOI is represented for a record with *valid time vt, FVT* is defined as returning the whole *valid time* (*fvt=vt)* and the time period of interest, *I*, is shifting towards the past. *I* is depicted as the gray rectangle and *FI* is the constant function depicted on the top of it. The constant function *FI=1* is used in this example to provide a constant upper limit for the TDOI.

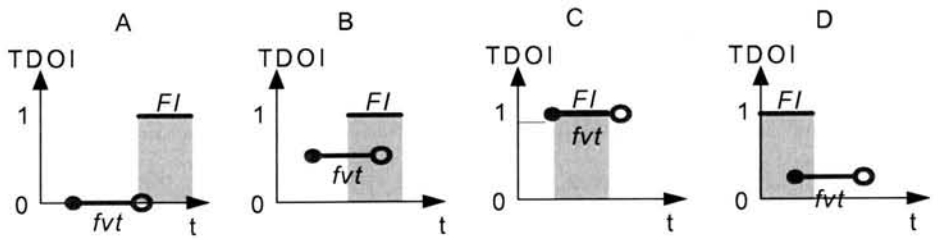


Figure 52: TDOI calculation for a record where *I* is sliding from future to past.

In situation A, the *filtered valid time* for the record (*fvt*) is disjoint from the period of interest *I*, and therefore TDOI=0. As the period of interest is moved to the past, *I* overlaps *fvt* and the TDOI increases (situation B, span *fvt* is getting higher levels in the figure) eventually reaching a maximum when *I* is overlapped by *fvt* (situation C). If the period of interest goes further to the past, the TDOI decreases (situation D).

5.7 Test Scenarios

The Temporal F+C visualization model has been prototyped into an application which, at the level of data management and query, provides the means for the user to issue STSQL

statements to a STDBMS and retrieve results, such as temporal record sets. The goal is to have a workbench, able to support the research and validation tasks.

Figure 53 shows the time window of the prototype where an *FI* function (depicted as a red line) has been defined with a span of one century (*I*, depicted as a shadowed rectangle). In this figure, each green line corresponds to a record *valid time* and the height at which it is displayed corresponds to the calculated TDOI; *FVT* has been selected to return the entire scope of each record *valid time*; yellow segments displayed over the time axis correspond to records that do not intersect I and so their TDOI equals 0.

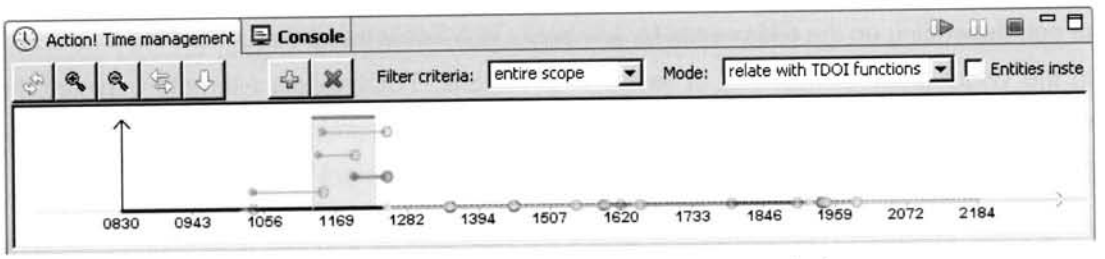


Figure 53: The temporal F+C configuration window.

For testing purposes, only discrete representations have been addressed, in the scope of urban and historical information management and analysis, involving three different scenarios. Quite often, the analysis of data over time requires the display of temporal evolution, the individual display of distinct non-overlapping times to depict features holding for these times, the simultaneous display of two non-overlapping times to infer spatial relations between different *valid time* objects and the visualization of a period of time, altogether with the immediate past and future, rendered with distinct degrees of interest. The following three scenarios try to show the appropriateness of the presented model to these different types of visualization requirements.

The first test scenario concerns the political history of Europe and has been chosen due to the richness of data evolution, as included nations have been born, conquered, annexed, divided and rejoined. For this scenario, the facts are represented in a database table with attributes for the name of the country, its capital city, the 2D geometry of the political boundary and the *valid time* for which the fact is considered to hold. The evolution of a country takes several records depending on the number of individual states actually represented. Through a spatio-temporal SQL query, the evolution of all represented countries has been retrieved and a mapping function has been assigned the political boundary attribute. Figure 54 illustrates a visualization of this scenario, between the years 1132 and 1232, with three countries represented: kingdom of Portugal, kingdom of Poland and the Czech kingdom. The corresponding data and TDOI functions have already been illustrated in Figure 53.

The mapping function that was developed specifically to support this test scenario controls the transparency of shapes and the style of drawing of edges to display the context

information where TDOI < 1; while on the other hand, records with TDOI = 1 have their information shown opaque with no effect on the edges. Through the control of transparency and edge drawing, this mapping function allows objects that temporally intersect the temporal context to be rendered by depicting a transition state that is proportional to the calculated degree of interest and which graphically should have less emphasis than the objects contained in the focus.

Furthermore, in interactive usage scenarios, as the user shifts *I*, TDOI values change. Between two shifts of *I*, it is possible to determine if the TDOI of a record is increasing or decreasing. To encompass this feature, the developed mapping function also makes a correspondence of the TDOI variation to color, showing TDOI negative variations in red tint and TDOI positive variation in green tint.

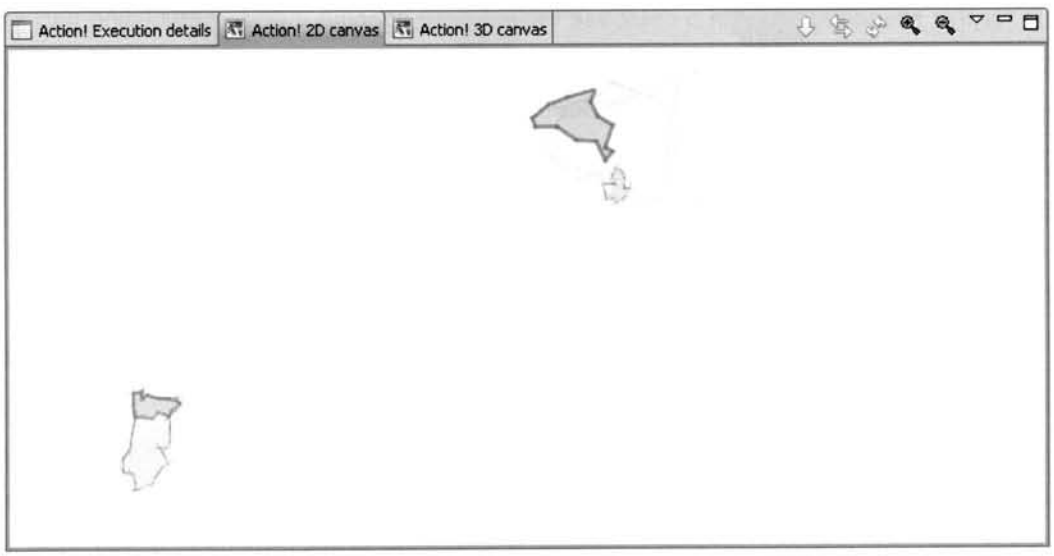


Figure 54: View of the political history of Europe dataset.

Figure 54 visually captures the situation where the user was sliding *I* into to the future. It is possible to depict that:

- Regarding the current temporal focus, Portugal is expanding south. There are two states depicting Portugal, and both have positive TDOI variations depicted by green. However, the state that geographically extends towards the south is more transparent which indicates it still temporally intersects the context information.
- The first kingdom of Poland has ceased to exist and, in its place, a new kingdom has been born. The first is depicted in red, since the TDOI is getting reduced as the object is leaving the context information towards the past. The new kingdom is displayed with strong edges, illustrating that it temporally intersects the focus. Its green-gray color indicates that the variation of TDOI, although still increasing, is becoming very small.
- Another new kingdom (Czech Kingdom, smaller polygon) is also being born in a slightly later future which is contained in the context information. The representation of this

country is depicted with high amount of transparency (it is context information) and in green, indicating that the variation of TDOI is positive, since the focus period is moving towards this country *valid time*.

The edge lines drawn are an attempt to use sketch drawing states for TDOI values lower than 1: sketch, as well as transparency, is progressively reduced as TDOI increases from 0 to 1. This test scenario shows that, with the proposed model and according to the user definition of *I* and *FI*, it is possible to display data with distinct visual emphasis depending on the records intersection with *I* and on the direction the user shifts *I*.

The second test scenario addresses an area of urban data management, more specifically, real estate cadastre. A building presents several distinct states across its existence, resulting from partial construction, destruction or modification. Spatio-temporal data for this scenario is kept in a database table whose schema includes a building part *id*, a blob for keeping its 3D geometrical representation and the *valid time* for which the fact is considered to be true. The images in Figure 55 illustrate the temporal evolution of the building that is represented by several records kept in the table. The mapping function developed to support this test scenario is applied to the blob attribute contents (3D geometry) and, again, it maps the calculated TDOI value into transparency, for drawing faces, and wireframe amount for rendering edges. Furthermore, the TDOI value is mapped into color saturation, by combining the original color of the 3D object with white: saturation increases (less white) as TDOI goes from 0 to 1.

To develop this mapping function, multi-pass rendering has been applied and no graphical distinction of the TDOI variation is used as the user shifts *I*. For this reason, the results do not provide enough visual clues to distinguish objects appearing from those that are disappearing. However, it is still possible to understand the temporal aspects associated to the building: the main building was the first to be constructed (left-top) followed by a left wing; when both were fully built, the construction of the left tower was started and, later, after it was fully built, the left wing of the building was removed; finally, the construction of the right tower was started. The last bottom-right image represents the state of the building immediately before the right tower was completed (wireframe rendering is minimum but still visible).

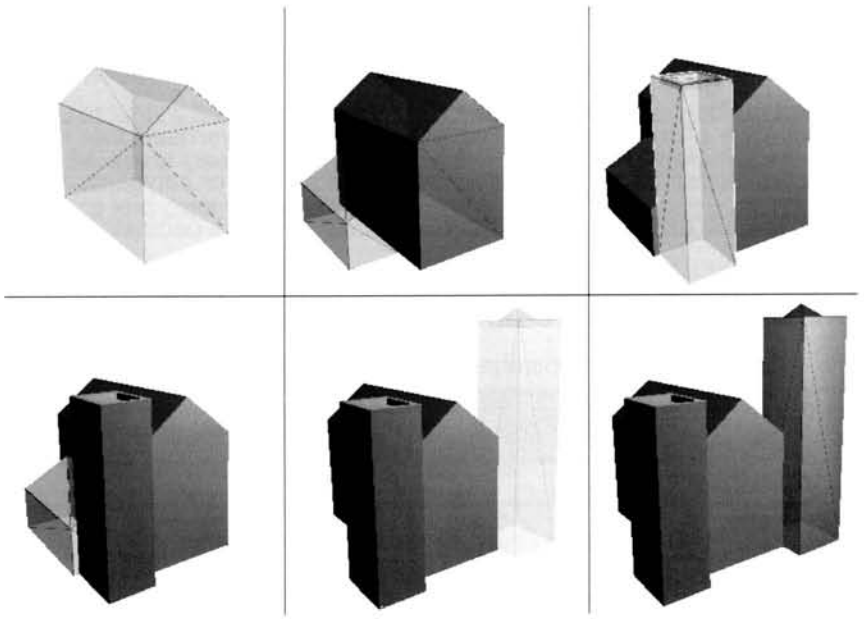


Figure 55: States of a building evolution across time.

The third test scenario is more elaborate and corresponds to the construction of one complex building, from 532 to 539, and it is assumed the building still exists in the present (details in Figure 57). As no real data is digitally available and the manual development of this data is a time-consuming task, the construction process was simulated by using a pre-built 3D model (from Google SketchUp) which was cut into horizontal slices (Figure 56).

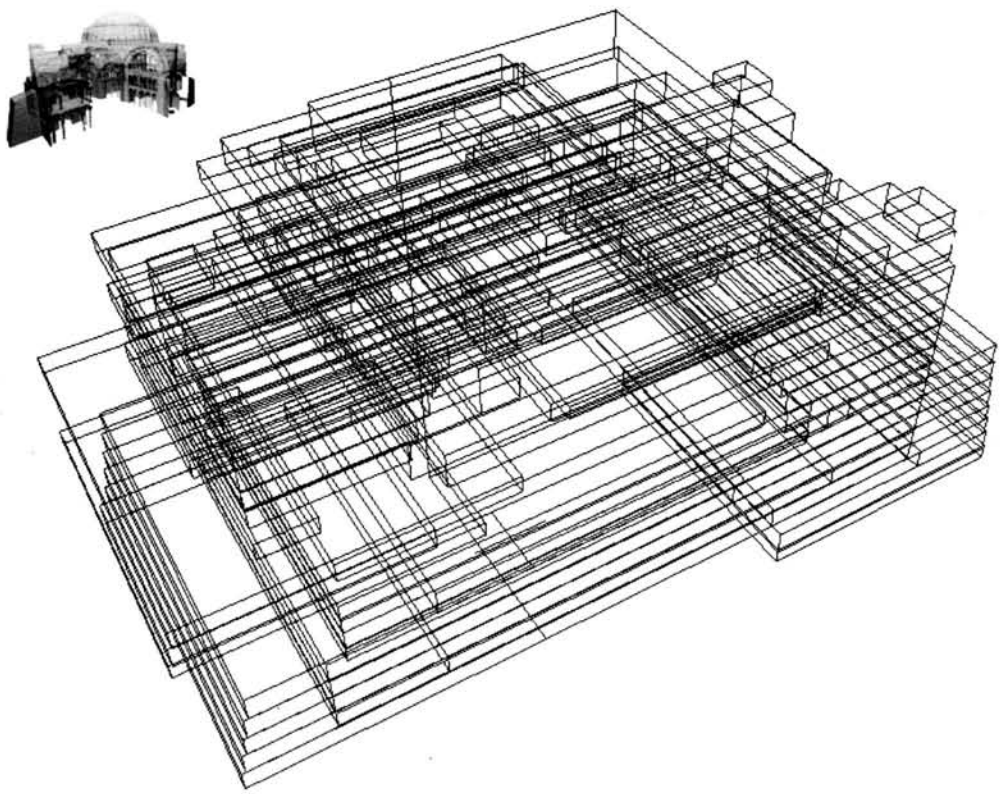


Figure 56: Bounding box view of 3D geometry in blocks.

Each slice was divided into four to six blocks and each block was tagged with an *id*. Figure 56 illustrates the block structure that resulted from this operation.

One disadvantage of this approach lies in the geometry overhead that results from the slicing process, as large polygons are sliced into several smaller polygons. Furthermore, as the adopted 3D engine exclusively requires triangles to describe geometry, further geometry is generated from the triangulation of polygons. However, this assembly serves to demonstrate that the suggested model is able to provide good results even when the source spatio-temporal data is clearly not optimized, such as the geometry used here.

The representation of the construction process in the spatio-temporal database is performed in two stages:

- SQL commands insert records into a table describing, for each block, the *valid time* start of construction. The time stamping ensured that the start of construction for each block was confined to [532-539] and that, for starting a block, all the underlying blocks would already exist;
- Record blobs are filled with 3D geometry based on the *id* match between the record and the block.

Through the developed prototype, the user can query the database for construction data, assign a mapping function to the blob attribute and define a TDOI function to render the data.



Figure 57: Spatio-temporal building construction evolution.

Figure 57 illustrates one of these user scenarios, obtained by defining the scope of *I* to loosely correspond to the scope of the construction period, and the *FI* function to correspond

to a descending ramp with its maximum assigned to the beginning of *I* and the minimum assigned to the end instant of *I* (see Figure 58, which is the configuration corresponding to left-bottom state of Figure 59). The mapping function applied to the blob attribute of the record set schema is the same used in the previous test scenario (controls transparency for face, wireframe for edge and color for both).

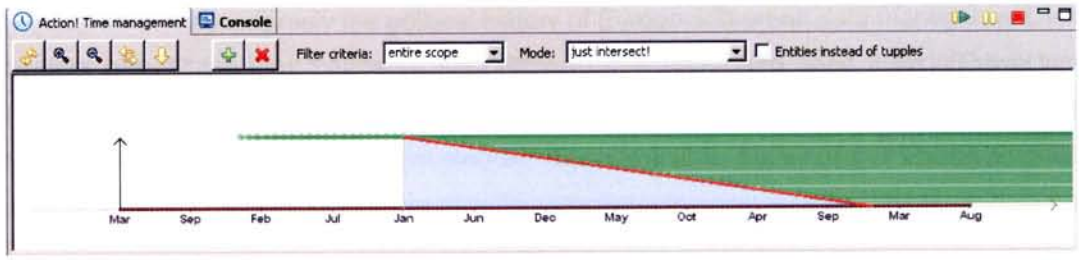


Figure 58: Illustration of function FI where I=[535-538].

The spatio-temporal visualization result illustrated in Figure 57 clearly shows that the bottom part of the building corresponds to blocks that get higher levels of TDOI: less transparency with more saturation and less wireframe rendering. The lowest TDOI values are found at the topmost blocks of the building: high transparency, high level of tint color and wireframe rendering. The geometry overhead resulting from slicing and triangulation is visible, especially in blocks with small TDOI.

The use of the proposed temporal F+C visualization model in the dynamic presentation of this test scenario allows the smooth visualization of the construction evolution. Figure 59 illustrates four distinct periods of a dynamic presentation, where the user shifts the defined *I*, starting in year 532 and ending in 539.

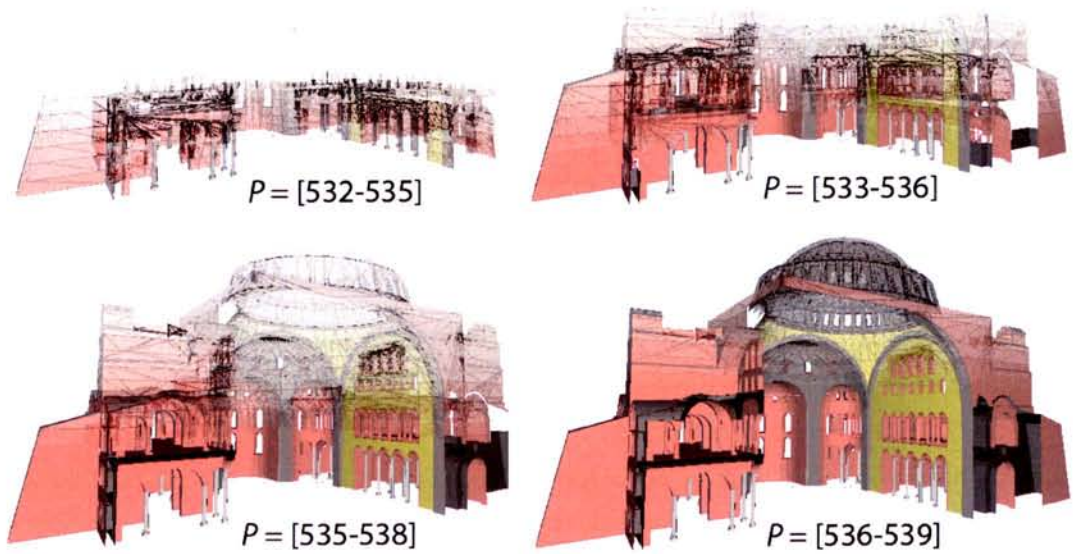


Figure 59: Evolution of the construction.

In the left topmost state of Figure 59, the base of the building is displayed, with small TDOI and the top areas are hidden, since they are out of the scope of *I*. As period *I* is moved towards the future, displayed bottom blocks get higher TDOI (they get less transparent) and hidden blocks start to be visible, with a small TDOI. In the last illustrated period, all the blocks with start of construction up to 536 are rendered with maximum TDOI and the remaining blocks are still rendered with high TDOI.

The tests have been performed with discrete data regarding both spatial and temporal dimensions: spatial data has been represented through discrete objects, temporally time stamped with intervals where it is assumed the data is unchanged. Further tests are still required regarding representation of continuous change over space and over time.

According to Döellner and Hinrichs (Döllner and Hinrichs 1997), the behavioural modelling is considered as important as geometric modelling, because the specification of time-varying and event-dependent properties is the key for animation and interaction control. Hence, regarding mapping functions development, it would be adequate to test an object-oriented framework aimed at modelling behaviours and geometry at the same level, such as VRS (Döllner and Hinrichs 1998).

5.8 Conclusions

The changes in the paradigm of spatio-temporal information management, through database systems, provide the challenges for developing new visualization techniques, aimed at a more effective representation of dynamic variables about temporal features of spatio-temporal data. Traditional visualization techniques such as ignoring time, showing only one instant or non-interactive animation provide limited support for visually exploring and analysing the answers to questions about the temporal aspects of data. In order to overcome these limitations, the temporal F+C visualization model has been proposed, based on the concept of a Temporal Degree of Interest (TDOI) associated to each record in a record set. A TDOI is defined based on user requirements, and tailored to a specific application domain, allowing the specification of a distinct degree of interest to the temporal Focus and to the temporal Context. The TDOI values are used in mapping functions that control graphical properties that visually enhance the temporal features of the data the user is interested in, while keeping temporal context data still visible but with less emphasis. These mapping functions can be defined according to the application domain needs.

To show results of this work, two mapping functions were developed, that transform TDOI into transparency, color and wireframe rendering of edges. In interactive scenarios, the variation of TDOI between shifts of $\Delta TDOI$ can also be used along with TDOI values to express the rate of change. One of the presented mapping functions also uses the record TDOI variation to graphically represent the increase/decrease of TDOI, by tinting the displayed entities with pseudo-color (green and red).

The proposed temporal F+C visualization model subsumes more traditional visualizations, while allowing other, more manageable visualization modes: by carefully configuring the TDOI functions, more data can be compressed onto the same visualization, while maintaining readability.

The current model has been tested with three test scenarios that involve historical-temporal information, namely the political history of Europe and urban data management, with the data repositories in a full-fledged STDBMS (Carvalho, Ribeiro et al. 2006). These tests provide preliminary results about the model capability to calculate and provide good visualization results for spatial object-model temporal data, meaningfully relating the data temporal characteristics with user visualization requirements.

Finally, it is believed that this visualization model is very promising for a large range of application domains requiring visualization methods for the temporal aspects of discrete 2D/3D temporal multivariate data.

Current work is addressing the study of the best techniques that provide the definition of mapping functions for concrete application domains. Ongoing work has also started to address the use of the proposed visualization model with distinct representation models of time. For instance, one of these models concerns objects moving continuously over space and time (Moreira, Ribeiro et al. 2008).

6 CONCLUSIONS AND FUTURE WORK

The work presented on this document addresses two research areas: spatio-temporal data management and spatio-temporal information visualization. It has been the aim to research methodologies and solutions for dealing with spatio-temporal data, from the representation and storage, through the retrieval, to the visualization, in what concerns visual and exploratory data analysis.

The motivation for undertaking this work began on the increasing desire to start managing spatio-temporal data through efficient tools. Such is the case of several domains of human activity, especially those involving or relying on Geographic Information Systems. In this application domain, the last decade has demonstrated that database management systems perform a critical role in information storage and retrieval. However, these systems still lack adequate support for efficient temporal data management and retrieval, despite the will to start managing and querying spatio-temporal datasets. Consequently, due to the lack of efficient solutions when it comes to manage and query these datasets, complex SQL statements must be written, usually embedded within the application code, which is then issued to a spatial database management system. Quite often, this increase in complexity disencourages the formulation of more complex questions involving spatio-temporal datasets.

Spatio-temporal information visualization aimed at exploratory data analysis is one of the fields that can benefit from proper spatio-temporal data management. If spatio-temporal datasets become easily and efficiently retrievable, through a spatio-temporal query language, more complex and rich datasets become available to be visualized and explored. In this context, the user can focus his attention into the interactive visual exploration of spatio-temporal data, instead of addressing his efforts into the formulation of complex spatio-temporal queries, through conventional SQL. However, the visualization of spatio-temporal data introduces new problems that are normally not present in the visualization of spatial data, where time is not used or is filtered to a single instant. For example, assuming that objects in a dataset are described through a set of tuples, the visualization of these objects cannot be accumulated onto static representations by displaying a single stage or by displaying all the stages at the same time, since, without any additional visual clues, neither of these solutions may provide easy to interpret visual results.

Hence, dynamic and interactive visual representations of spatio-temporal datasets, which provide tools for the exploratory navigation across time and space, can be regarded as one of the best possible ways to provide visual representations for this type of data.

Moreover, on one hand, these visualizations should enhance the meaningfulness of the displayed results in what concerns the temporal features of data and, on the other hand, should provide the user with the proper means to specify what his interests are, regarding the visualization of data across the time. To fulfill these two requisites, additional parameters

must be set to provide the visualization system with more precise information on what the user is really interested to observe. Through these parameters the visualization system is able to visually enhance the information items in which the user is interested, while keeping other information items still visible with less or no emphasis, in order to provide a suitable information visualization context or overview.

In the following, and considering what has been presented until this moment, the most important conclusions of this work are summarized as well as the topics that are worth researching.

6.1 SPATIAL DATABASE MANAGEMENT SYSTEMS

The research and development in spatial database management systems has been greatly motivated by GIS. However, GIS has not always relied on spatial DBMS to provide efficient support and management for spatial data. Before the use of spatial DBMS, common problems in GIS data management focused on the representation of the structure of spatial information, the management of concurrent access to resources, the efficient retrieval of spatial data, the fault tolerance and the preservation of semantic integrity.

Most spatial DBMS still provide spatial data management by extending a relational or object-relational database management systems with spatial data management features, being the most important the representation for spatial data types of a spatial algebra, the representation for spatial data types within a data definition and query language, the availability of algorithms for spatial functions and operators (such as spatial join) and, finally, spatial indexing mechanisms.

The first developments of spatial DBMS were considered complex and time consuming tasks due to the complex structure of data representation, the dynamic behavior of spatial objects and the increase in computational requirements that resulted from the calculations required by spatial operators and functions, as compared to alphanumeric data.

This work has reported five categories of spatial support provided by a spatial DBMS, namely, through natural languages, extensions of SQL, tabular languages, graphical languages, and finally, through visual languages. Among these, extensions to SQL are considered to be the best formal support to manage and retrieve spatial data with the disadvantage that it can only be used by people having some knowledge in databases.

Several known spatial DBMS have been presented in section 2.2.2, along with a characterization of the spatial support provided. Most of the current solutions provide support for spatial data management through an extension to SQL and try to comply with standards such as OGC Simple Features Specifications for SQL or the ISO SQL/MM Part 3, among other standards, thus contributing to more efficient spatial data interoperability and to the development of distributed web GIS solutions.

6.2 SPATIO-TEMPORAL REQUIREMENTS FOR DBMS

Concerning temporal data management, current DBMS still provide limited support. Usually, this support is limited to temporal data types and to a small set of temporal functions. The immediate consequence is that, over the past years, due to the lack of proper temporal data management in DBMS, non-temporal and semi-temporal approaches were widely adopted in a wide range of application domains. The non-temporal approach concerns the management of the most recent state of facts, deleting the facts regarding the past, while the semi-temporal approach concerns the time stamping of spatial data through attributes of date/time data type.

Even the semi-temporal approach has proved to be limited in application domains that involve complex data management of spatial and temporal features, such as it happens in GIS. Among other features required to provide efficient temporal data management is the availability of temporal data types (such as *instant*, *period*, *interval*, *span*) and a query language that supports the *valid time* and *transaction time* dimensions, sequenced and non-sequenced semantic statements, temporal functions and operators, such as temporal join and coalescing, and finally indexing mechanisms for temporal data. Other features are also considered very important, such as temporal upward compatibility, snapshot-reducibility, interval preservation and non-restrictiveness. Applications can manage distinct states of facts more easily and efficiently, such as the temporal evolution of a set of spatial features, if they use these features through a query language that is provided by the DBMS level. Hence, temporal data management built-in the database management system, available through a query language, seems to be one of the most adequate approaches to efficiently manage time-varying spatial data in application domains that rely in GIS technology and for which time is critical.

One of the reasons that justified the lack of the described temporal data management concerned some previous lack of consensus, in the research community addressing temporal data management, about a reference data representation scheme and query language. But perhaps the most important reason is the disagreement inside the ISO committee as to where temporal support in SQL should go. This disagreement has prevented the evolution of working draft, "Part 7, SQL/Temporal" until the present time. Important contributions on temporal data management have been proposed since the 1980s which provided, not only distinct perspectives on the representation of time in databases, but also a wide set of temporal data models and query languages (more than 40 temporal data models have been proposed). The reason for such a high number of contributions has to do with the fact that adding time to a data model has been considered a very complex task, since an ideal temporal data model encompasses too many requirements, some of which are mutually incompatible. Furthermore, some of the desired temporal characterization lacked for efficient implementation.

An important contribution was made to unify these data models based on the Bitemporal Conceptual Data Model. This conceptual data model captures the essential semantics of time-varying relations without having strong concerns on efficiency regarding presentation, storage and retrieval. Hence, based on the concept of informational content, equivalence mappings between the BCDM and several data representation schemes can be exploited in order to use these schemes to support representation and storage of temporal data. A demonstration of the mappings between the BCDM and the following representational schemes has already been performed: Ben-Zvi's Tuple Timestamped Representation Scheme, Jensen's Back-log Based Representation Scheme, McKenzie's Attribute Value Timestamped Representation Scheme, Snodgrass's Tuple Timestamped Representation Scheme and Bhargava's Attribute Value Timestamped Representation Scheme.

Concerning the temporal SQL that is required to take advantage of the underlying temporal data model, there have been a considerable number of proposals, mostly temporal extensions to SQL. A temporal query language which provides all the temporal features and fulfills all the requirements of temporal data management is not yet available. However, TSQL2 can be considered one of the most suitable temporal query languages: in *Road Map of Additions to SQL/Temporal*, presented to the international committee (ISO/IEC JTC 1/SC 21/WG 3 DBL) Snodgrass *et al.* justifies why TSQL2 should be the way. Shortly, this author states that "*TSQL2 is by far the most comprehensive, well-documented, consensual temporal query language. It has become the de facto standard in the temporal database community*". It is important to say that the startup effort to develop TSQL2 was performed by 18 temporal database experts, distinguished authors of several data models and query languages.

Regarding the development of temporal DBMS, three architectural approaches have been studied, namely, the monolithic, the extensible and the layered architectural approaches. These architectural approaches differ between each other in where temporal data management support is provided, regarding the proximity to the DBMS kernel. The monolithic architectural approach is considered the best under almost all criteria, since temporal data management support is directly developed on the DBMS kernel and so takes full advantage from this proximity in terms of efficiency. However, this approach is considered complex and time consuming. On the other hand, the layered approach provides fast development, but the temporal data management support is placed outside the DBMS and, for this reason, may not be as efficient as the monolithic approach. The extensible approach can be regarded as an intermediate approach, since temporal data management is developed as an extension to the DBMS kernel.

Several temporal database management systems have been studied in this work, namely the CONCERT, DEDALE, SECONDO, TIGER, TEMPOS, TRIPOD, Oracle and TimeDB. These Temporal DBMS use several architectural approaches, although the extensible approach is the most common. Among these, TimeDB and its query language – ATSQL2 – was concluded to be a well suited solution for temporal data management.

6.3 ACTION! – A STDBMS

Chapter 3 addressed the contribution for a new STDBMS – *Action!* - that is able to provide spatio-temporal data management by adopting Snodgrass Tuple Timestamping Representation Scheme and by providing spatio-temporal data management through a query language that integrates ATSQL2 with the Oracle Spatial extension, and which allows the formulation of spatio-temporal data definition and modification statements.

In fact, the support provided by integrating the Oracle spatial extension into ATSQL2 is more than the grouping of features from spatial and temporal support. Concerning the data definition statements, temporal relations can include spatial attributes in order to represent a wide range of spatial objects. Furthermore, data modification statements, such as queries, can use spatial functions and spatial operators in the projection list as well as in the selection list, as spatial selection criteria, of the query statement. These functions and operators can also be used within sequenced and non-sequenced semantic queries. Moreover, temporal operators, such as coalescing, are also able to properly manage spatial attributes in what concerns attribute value comparison between two tuples from a relation being coalesced.

The resulting query language is still able to maintain important temporal features from ATSQL2 such as upward compatibility and temporal upward compatibility, meaning that, through *Action!*, legacy applications using SQL are kept operational: concerning upward compatibility, it is still possible to execute SQL statements that only require spatial support or which require no spatio-temporal support at all. In what concerns temporal upward compatibility, snapshot queries containing spatial projections and selections can be performed over temporal relations. As in ATSQL2, these queries are evaluated concerning only the tuples that regard the current state of facts, regarding a temporal relation.

By fully implementing the proposed SQL, the *Action!* prototype system works as the proof of concept to prove the first hypothesis presented in this document (Section 1.1, Problem definition), that "*storage and retrieval of spatio-temporal information in a GIS can be provided through a conventional spatial DBMS, by extending it with temporal support for both valid time and transaction time and by preserving important features in temporal DBMS such as upward compatibility, temporal upward compatibility, sequenced semantics and non sequenced-semantics*".

In order to develop *Action!*, the layered approach and the extensible approach were combined, dividing *Action!* into a client-side and a server-side parts. The first corresponds to a module which integrates third-party applications. Through an API, this module allows an application to formulate spatio-temporal SQL statements and to receive the corresponding results. Hence, the client-side module is responsible for the management and translation of each spatio-temporal statement into a set of snapshot-equivalent statements; these use conventional relational operators that are able to be successfully evaluated and executed by Oracle and its spatial extension. The server-side part of *Action!* is Oracle Spatial itself and it is

responsible for the evaluation of the statements delivered by the client-side part and to return the corresponding results.

Concerning the client-side and more specifically the translation of spatio-temporal SQL into SQL compatible with Oracle Spatial, nine distinct temporal scenarios of temporal compatibility between the statements and their respective relations have been identified. For each of these scenarios, distinct translation of the spatio-temporal SQL statement is performed in order to translate it into a set of snapshot-equivalent statements which can then be issued to Oracle. In this translation process it has been confirmed that the spatial support can be managed orthogonally to the temporal support, during the translation process (Snodgrass had already generalized that non-temporal attributes could be managed orthogonally to time). Hence, in the core of the translation process, and regarding snapshot-equivalent mappings, the spatial attributes, spatial functions and spatial operators are managed in a similar way to any other non-temporal attributes, functions and operators. Consequently, each snapshot-equivalent SQL statement resulting from translation can contain references to spatial attributes, spatial functions and spatial operators.

It is not straightforward to provide the management of the spatial support inside the translation algorithm, since a considerable amount of features need to be incorporated in it. In order to accommodate spatial and spatio-temporal support, the translation algorithm requires support for the spatial schema (data types, tables, indexes functions and operators) and, perhaps more important, that all the mappings of temporal relational algebra operators require adequate value-comparison when spatial attributes are at stake.

The correctness of the features provided by *Action!* (snapshot, spatial, temporal and spatio-temporal data management) has been shown (section 3.5), based on a data set that was developed specifically for test purposes. It was shown that, by enhancing a spatially-enabled conventional database management system with temporal support, a STDBMS can be achieved that provides spatial data management for *valid time* and for *transaction time* and that preserves important temporal features such as upward compatibility, temporal upward compatibility, sequenced semantics and non sequenced-semantics.

It can be concluded that the followed spatio-temporal data management approach can be considered suitable to perform the role of representation, storage and retrieval of data, evolving discretely over time and space. This kind of support is appropriate for application domains where considerable amounts of resources have been invested in GIS, in the past, and for which the need to manage the temporal aspects of data is starting to emerge.

However, it has been verified that *Action!* has limitations that are a consequence of the architectural approach: more robust spatio-temporal data management would be available if the spatio-temporal query language could include additional features defined in SQL3 concerning assertion definition, cursor expressions and fetch statements. Nonetheless, it is

believed that the development of these and other spatio-temporal features may require that spatio-temporal support must be totally developed on the server-side.

6.4 VISUALIZATION OF INFORMATION

Action! also constitutes a suitable tool to support the research on time-dependent information visualization: it has been verified that the research community on spatio-temporal information visualization is still managing test data through conventional DBMS or through file-based repositories. By exploring the potential of a STDBMS such as *Action!*, spatio-temporal datasets become easily manageable and retrievable and, in this way, researchers can more easily focus in research areas that require spatio-temporal data as input. In this context, the spatio-temporal information visualization directly retrieved from a STDBMS can pose interesting questions, for instance, how to visually represent information about a temporal evolution of facts where data from two distinct time intervals share the same spatial location.

To this purpose, the multi-disciplinary research domain of Information Visualization has been surveyed (Chapter 4). The purpose here was to identify techniques that offload the cognitive process to the visual system, by visually representing large amounts of data in a well suited manner, allowing to effectively discover and to provide explanations about patterns and relations on individual or groups of information items. In order to achieve this, most of the information visualization techniques go through a sequence of stages, designated by the visualization pipeline, which transforms raw data into manageable, meaningful and easy to interpret visual representations, making no distortions of the represented values in what concerns the original data. However, the accomplishment of such task can be considered complex, especially in the presence of large datasets and for scenarios where the user is allowed to have detailed information of a particular item while keeping available an overview representation of the informational context.

A wide range of information visualization techniques has been presented and grouped according to an adapted version of Shneiderman's taxonomy for information visualization techniques: techniques for displaying linear, hierarchical, networked and multidimensional structures. Two groups have been added, namely time-dependent structures and geovisualization. The reason why time-dependent information has been detached from the group of techniques focusing on multidimensional structures is that the term multidimensional refers to the number of the independent variables, while the term multivariate refers to the number of the dependent variables of a data set. For the same reason, geovisualization has been separated from the other groups. Although the spatial nature of data is predominant, geovisualization can encompass other groups of techniques, for instance, those regarding time-dependent structures, in order to visually represent spatio-temporal data.

Among the visited techniques, the current work has focused into micro/macro readings designated by Focus+Context, which allow the presentation of large amounts of complex information, with high levels of detail and organization, through multiple and frequently hierarchical levels of contextual readings. Among these, a distorted view technique designated by *Fisheye Concept*, proposed by Furnas has particular relevance due to the underlying idea of a mental map created by the user about the data of interest, according to subjective perception of the information being displayed. In this technique, each information item being presented is assigned a number resulting from a degree of interest function chosen by the user. The function provides the expression of the user's interest for each information item. During the visualization of data items, a filtering process takes place when the user defines a numerical threshold from which items will be rendered. Calculations take place at the pre-processing stage and no assumptions are made regarding the mapping stage, meaning that the mapping strategy is independent of the model, in what concerns how to map an information item based on its calculated degree of interest. However, drawbacks have been identified in Focus+Context approaches. The two most relevant are the occurrence of problems in focus when a transition is taking place between views and the potential user disorientation in case of excessive distortion.

The current work has also focused on time-dependent information visualization, where groups of techniques have been presented and grouped into static and dynamic visual representations. The most relevant features that can be identified in static visual representations are the benefit from having the ability to allow the conclusion of quantitative facts as well as explore special data features, data values, time steps and positions, according to the underlying time scales, without limitations. However, the application of these techniques has most often been restricted to depict one variable or a limited number of variables, whereas the analysis of multivariate correlations is reduced. On the other hand, dynamic visual representations of time-dependent data are regarded as making a more direct use of time to depict the data. This can be done by mapping the temporal aspects of the data directly onto the time control of a dynamic representation and by mapping data values into representations that change over time and which control variables such as size, shape, color and texture.

In the context of geovisualization, the use of animation techniques for displaying time-dependent data has already been studied. On one hand, animation has been considered a well-suited visual tool for qualitative analysis and also to provide the ability to represent change over time and thus facilitate an understanding of process rather than state; it could also reveal subtle space-time patterns that are not evident in static representations, even to expert users who are highly familiar with the data. On the other hand, several authors consider that the success on using animation techniques depends on the interaction and data-exploration tools that are available to the user, in order to explore the data. Still, it is

considered that further studies are required on the effectiveness of these representations for learning, knowledge discovery and knowledge construction.

6.5 INTERACTIVE VISUALIZATION OF SPATIO-TEMPORAL INFORMATION

The most important contribution of the current work is towards the interactive visualization of spatio-temporal data retrieved from a STDBMS. The goal here is to prove the second hypothesis presented in this document (Section 1.1, Problem definition), namely: *Focus+Context visualization techniques provide appropriate solutions for the visualization of spatio-temporal information directly gathered from STDBMS in the context of exploratory data analysis.* To achieve such goal, a visualization model has been defined that adopts several of the presented concepts in information visualization namely, Focus+Context distorted view techniques, the *Fisheye Concept* and the dynamic representation of time-dependent information considering a linear discrete time model.

The underlying motivation concerned the changes in the paradigm of spatio-temporal information management through a STDBMS. Here, new challenges arise from the availability of data through spatio-temporal queries, resulting in spatio-temporal record sets which can be viewed independently or combined onto a single visualization. On the other hand, the motivation has resulted from the fact that traditional visualization techniques such as displaying the entire dataset (by ignoring the time), showing only one instant of time or displaying the temporal evolution of data through non-interactive animation may lead to lack or excess of information being represented and may also provide limited support for visually exploring and analyzing the temporal aspects of data.

In order to overcome these limitations, a research work was performed on a new visualization model that provides distinct representations for data items according to their temporal features and according to the user visualization interests. Hence, data items that comply with the user visualization interests have their representation enhanced while other data items are represented with less visual emphasis. This representation model allows visual analysis of large datasets by simultaneously providing focus to some data items while displaying other items as an overview.

In the core of the proposed model, designated by temporal F+C visualization, is the concept of Temporal Degree of Interest (TDOI), which is calculated for each data item of a dataset. The TDOI values result from one or more TDOI functions that are defined by the user to describe the visualization interests about time. The set of TDOI functions allows the characterization of a complex degree of interest, where the user can define distinct degrees of interest for one or more temporal focuses and for one or more temporal contexts. The TDOI value calculated for each data item can then be used in mapping functions to control graphical variables used to render the data; this allows the visual enhancement of the

temporal features corresponding to the data the user is interested, while keeping the temporal context data still visible but with less emphasis. It should be noticed that the fish eye concept has previously been applied in other Information visualization techniques; however, previous uses have addressed space, while the TDOI visualization model of this work addresses time. It should also be noticed that there is a clear separation of the TDOI model from the mapping functions that use the TDOI calculated values. The possibilities of using TDOI values in mapping functions are wide. Thus, these functions should be developed according to particular visualization problems, to permit the most adequate visualization techniques to be used.

In order to demonstrate the work on Temporal Focus+Context visualization, two test mapping functions have been developed, that control transparency, color and wireframe rendering of edges. In interactive scenarios, the variation of TDOI (triggered by shifts on the period of interest in TDOI functions) $\Delta TDOI$ can also be used along with the TDOI values to visually express the rate of change. One of the test mapping functions presented uses the TDOI variation to graphically represent the increase/decrease of TDOI by blending color into the displayed data-items.

The Temporal Focus+Context visualization model has been tested with three test scenarios that involve historical-temporal information managed by the STDBMS prototype, namely the political history of Europe and two scenarios addressing urban data. These test scenarios provide results that allow the conclusion on the model capability to calculate and provide good visualization results for spatio-temporal data, meaningfully relating the data temporal characteristics with the user visualization requirements. The model showed that it is able to provide results similar to more conventional representations, such as snapshot and animations, as well as to provide representations for distinct time periods, showing for each period what is happening in the immediate past and immediate future.

This visualization model seems to be very promising for a wide range of application domains that require visual analysis methods for displaying the temporal aspects of discrete 2D/3D temporal multivariate data, the temporal relation between sets of data items and the identification of temporal patterns. However, the research work concerning the visualization model can be improved with further developments.

Further work can be performed regarding the identification of cross-application domain visualization problems. For each of these visualization problems, further study can be performed about which graphic variables (not limiting to those addressed in this work) can be used to graphically depict the corresponding answer. Furthermore, after the development of the mapping functions according to the previous studies, a formal evaluation can be carried where a test scenario is handled by a large group of users that interact with a prototype in order to answer a set of carefully thought questions about the effectiveness of visual results. Only after, the gathered results can be analyzed in order to fine-tune the mapping functions,

allowing the improvement of their effectiveness. Moreover, these conclusions can also be an aid to improve the calculation model of the temporal degree of interest.

6.6 FUTURE WORK

Spatio-temporal data management and visualization has several research streams that can be further developed in future work. Regarding spatio-temporal data management, the method proposed in this work can be further researched in the future, especially in what concerns the possibility to represent objects which continuously change their spatial attributes, such as the location (for instance, for moving objects), and the shape (for instance, for climate phenomena). By enhancing the current representation data model and provided a set of data types, functions and operators, other domains of urban activity related to moving points could be encompassed, such as traffic management. However, more elaborate work seems to be required for objects that change over the time, by not only changing its shape but also by splitting or by merging. Here, not only a more sophisticated data model is required but also the development of spatio-temporal operators and functions that efficiently allow the retrieval of results.

In the particular context of efficiency, it would also be desirable to study for multidimensional access methods, such as spatio-temporal indexing mechanisms, since the current solution only benefits from these indexing mechanisms for spatial data. However, the development of such access methods may require a distinct architectural approach, where the whole spatio-temporal data management is directly supported as an extension developed inside the DBMS kernel.

In what concerns the visualization of spatio-temporal information through the proposed Temporal Focus+Context visualization model, this work has identified that a major research stream consists in the visualization of data with two associated temporal dimensions, namely *transaction time* and *valid time*, since the current work has only addressed the *valid time* dimension. This stream has not been addressed in this work because the visualization of data time stamped with just *transaction time* seems to be similar to dealing with *valid time*, provided some differences due to the semantics of *transaction time*. Furthermore, a large number of application domains are mostly requiring appropriate visualization tools to deal with *valid time* data and, currently, are not yet interested in the visualization *transaction time*. Hence, on one hand, the research effort to provide efficient visualization tools for *valid* and *transaction time* simultaneously seems to be considerable and, on the other hand, the visualization of *valid time* data still presents a considerable amount of interesting challenges.

One of these challenges is, for instance, the efficient use of the TDOI values in the mapping stage of the visualization pipeline. By efficient use it is meant that the controlled graphic variables clearly and undoubtedly express the temporal feature that is being analyzed. One particular problem consists in how to display two objects that share the same

location at distinct times. Possible solutions should encompass contributions from other fields of knowledge, from computer graphics to semiology.

Furthermore, the availability of efficient methods for representing and querying continuous change provide fertile grounds to evaluate the possibility of using or enhancing the proposed visualization model to represent continuous change in spatial attributes. The current work has only addressed discrete changes on non-moving objects.

Finally, since the primary motivation of this work in what concerns its application area has been the urban data management, it would be suitable to study a systematization of questions that can be answered though the proposed visualization model for areas such as urban data management. The mapping functions that control graphical variables based on TDOI calculations have been developed to test the current visualization model. Further study can analyze more graphical variables and explore other possibilities, particularly relating human perception to visualization problems on spatio-temporal data, from specific application domains.

REFERENCES

Acevedo, W. and P. Masuoka (1997). "Time-series animation techniques for visualizing urban growth." Computers & Geosciences **23**(4): 423-435.

Ahlberg, C. and E. Wistrand (1995). "IVEE: an Information Visualization and Exploration Environment." INFOVIS '95: Proceedings of the 1995 IEEE Symposium on Information Visualization: 66.

Aigner, W., S. Miksch, et al. (2007). "Visualizing time-oriented data - A systematic view." Computer Graphics **31**(3): 401-409.

Allen, J. F. (1983). "Maintaining knowledge about temporal intervals." Communications ACM **26**(11): 832-843.

Alpern, B. and L. Carter (1991). Hyperbox. IEEE Visualization '91, San Diego, California.

Amagasa, T., M. Yoshikawa, et al. (2000). A Data Model for Temporal XML Documents. Proceedings of the 11th International Conference on Database and Expert Systems Applications, Springer-Verlag.

Andrews, K., V. Sabol, et al. (2001). Search Result Visualisation with xFIND. Second International Workshop on User Interfaces to Data Intensive Systems (UIDIS'01), Zurich, Switzerland.

ANSI.X3H2-91-183 (1991). Database Language SQL2/SQL3. Gaithersburg, MD, National Institute of Standards and Technology.

Aufaure-Portier, M.-A. and C. Trépied (1996). A Survey of Query Languages for Geographic Information Systems. 3rd International Workshop on Interfaces to Databases, Napier University, Edinburgh, Springer.

Aufaure-Portier, M. A. (1992). CIGALES : un langage graphique d'interrogation de Systèmes d'Information Géographiques. Paris, University of Paris VI. **Phd**.

Ball, T. and S. G. Eick (1996). "Software Visualization in the Large." Computer **29**(4): 33-43.

Balovnev, O. T., M. Breunig, et al. (1997). From GeoStore to GeoToolKit: The Second Step. Proceedings of the 5th International Symposium on Advances in Spatial Databases, Springer-Verlag.

Bartram, L., A. Ho, et al. (1995). "The continuous zoom: a constrained fisheye technique for viewing and navigating large information spaces." UIST '95: Proceedings of the 8th annual ACM symposium on User interface and software technology: 207-215.

Becker, R. A. and W. S. Cleveland (1987). "Brushing scatterplots." Technometrics **29**: 127–142.

Ben-Zvi, J. (1982). The time relational model, University of California, Los Angeles: 286.

Bergeron, R. D., W. F. Cody, et al. (1993). "Database Issues for Data Visualization: Developing a Data Model." Proceedings of the IEEE Visualization '93 Workshop on Database Issues for Data Visualization: 3-15.

Bertin, J. (1967). Sémiologie Graphique. Paris, Editions de l'Ecole des Hautes Etudes en Sciences Sociales.

Bertin, J. (1981). "Graphics and graphic information processing." 62-65.

Bertin, J. (1999). "Graphics and graphic information processing." 62-65.

Beshers, C. and S. Feiner (1993). "Autovisual: rule-based design of interactive multivariate visualizations." IEEE Computer Graphics and Applications **13**(4): 41–49.

Bhargava, G. and S. K. Gadia (1989). Achieving zero information-loss in a classical database environment. Proceedings of the 15th international conference on Very large data bases. Amsterdam, The Netherlands, Morgan Kaufmann Publishers Inc.

Bhargava, G. and S. K. Gadia (1993). "Relational Database Systems with Zero Information Loss." IEEE Trans. on Knowl. and Data Eng. **5**(1): 76-87.

Blair, J., C. S. Jensen, et al. (1997). "Notions of upward compatibility of temporal query languages." Business Information **39**(1): 25-34.

Blott, S., L. Relly, et al. (1996). An open abstract-object storage system. Proceedings of the 1996 ACM SIGMOD international conference on Management of data. Montreal, Quebec, Canada, ACM.

Böhlen, M. (1994). Managing Temporal Knowledge in Deductive Databases. Departement Informatik,, ETH Zürich. **PhD**.

Böhlen, M., R. T. Snodgrass, et al. (1996). Coalescing in Temporal Databases. Proceedings of 22nd International VLDB.

Böhlen, M. H. (1995). "Temporal database system implementations " ACM SIGMOD Record **24**(4): 53-60.

Böhlen, M. H., C.S.Jensen, et al. (1998). Spatio-temporal Database Support for Legacy Applications. SIGMOD'98.

Böhlen, M. H., C. S. Jensen, et al. (2000). "Temporal statement modifiers." ACM Trans. Database Syst. **25**(4): 407-456.

Breunig, M., C. Türker, et al. (2003). Architectures and Implementations of Spatio-temporal Database Management Systems. Spatio-Temporal Databases: The CHOROCHRONOS Approach. M. Koubarakis, T. K. Sellis, A. U. Franket al, Springer. **2520:** 263-318.

Brewer, C. (1994). Color Use Guidelines for Mapping and Visualization. Visualization in Modern Cartography. A. M. MacEachren and D. R. F. Taylor. Tarrytown, NY**:** 123-147.

Brian, F. and J. Pritchard (1997). Visualisation of historical events using Lexis pencils. Case Studies of Visualization in the Social Sciences.

Bryce, D. and R. Hull (1986). SNAP : a Graphics-Based Schema Manager. 2nd IEEE International Conference on Data Engineering, New-York, USA.

Buja, A. and D. Asimov (1986). Grand tour methods: An outline. 18th Symposium on the Interface, American Statistical Association.

Buja, A., J. A. McDonald, et al. (1991). "Interactive data visualization using focusing and linking." VIS '91: Proceedings of the 2nd conference on Visualization '91: 156-163.

Campo, M., R. Orosco, et al. (1997). Automatic abstraction management in information visualization systems. IEEE Conference on Information Visualisation, Washington, DC, USA, IEEE Computer Society.

Card, S. K. and J. Mackinlay (1997). "The structure of the information visualization design space." INFOVIS '97: Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis '97): 92.

Card, S. K., J. D. Mackinlay, et al. (1999). Readings in information visualization: using vision to think. San Francisco, CA, USA, Morgan Kaufmann Publishers Inc.

Carlis, J. V. and J. A. Konstan (1998). "Interactive visualization of serial periodic data." UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology: 29-38.

Carmo, M. B. and J. D. Cunha (1998). Filtragem e Escolha de Representação na Visualização de Informação. 8º Encontro Português de Computação Gráfica, Coimbra, Portugal, Revista VIRtual.

Carvalho, A., C. Ribeiro, et al. (2006). A Spatio-Temporal Database System Based on TimeDB and Oracle Spatial. IFIP International Federation for Information Processing. Springer. **205:** 11-20.

Carvalho, A., C. Ribeiro, et al. (2008). A Temporal Focus + Context Visualization Model for Handling Valid-time Spatial Information. Workshop on Movement and Change at the AGILE 2008 Conference, Girona, Spain.

Carvalho, A., C. Ribeiro, et al. (2008). A temporal focus + context visualization model for handling valid-time spatial information. Information Visualization - Special Issue of the ICA Commission on Geovisualization on Geovisualization of Dynamics, Movement and Change. G. Andrienko, N. Andrienko, Jason Dykes, S. I. Fabrikant and M. Wachowicz, Palgrave-McMillan. **7:** 265–274.

Carvalho, A., C. Ribeiro, et al. (2006). A Spatio-Temporal Database System Based on TimeDB and Oracle Spatial. Research and Practical Issues of Enterprise Information Systems. I. I. F. f. I. Processing. New York, Springer **205:** 11-20.

Cattell, R. G. G., D. K. Barry, et al. (2000). The Object Database Standard: ODMG 3.0, Morgan Kaufmann.

Chalmers, M. and P. Chitson (1992). "Bead: explorations in information visualization." 15th annual international ACM SIGIR conference on Research and development in information retrieval: 330-337.

Chi, E. H. (2000). "A Taxonomy of Visualization Techniques Using the Data State Reference Model." INFOVIS: 69-76.

Chuah, M. C. and S. F. Roth (1996). "On the semantics of interactive visualizations." INFOVIS '96: Proceedings of the 1996 IEEE Symposium on Information Visualization (INFOVIS '96): 29.

Cleveland, W. S. (1993). Visualizing Data. New Jersey.

Cleveland, W. S., M. E. McGill, et al. (1993). "The shape parameter of a two-variable graph." Journal of American Statistical Association **38**: 289–300.

Clifford, J., C. Dyreson, et al. (1997). "On the semantics of \“now\” in databases." ACM Trans. Database Syst. **22**(2): 171-214.

Collins, B. M. (1993). Data visualization – Has it all been seen before? Animation and Scientific Visualization – Tools and Applications. R. A. Earnshaw and D. Watson, Academic Press. **1**.

Costagliola, G., G. Tortora, et al. (1995). GISQL - A Query Language Interpreter for Geographical Information Systems. IFIP Third Working Conference on Visual Database Systems, Lausanne, Switzerland.

Cowen, D. J. (1988). "GIS versus CAD dbms: what are the differences?" Photogrammetric Engineering and Remote Sensing **54**: 1551-1554.

Coyne, B. and R. Sproat (2001). "WordsEye: an automatic text-to-scene conversion system." SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques: 487-496.

Croft, W. B. and H. R. Turtle. (1992). Text retrieval and inference. Text-based intelligent systems: current research and practice in information extraction and retrieval. P. S. Jacobs. Mahwah, NJ, USA, Lawrence Erlbaum Associates, Inc.

Date, C. (2000). Temporal Databases (Chaper 22). An Introduction to Database Systems, Addison-Wesley.

DiBiase, D., A. MacEachren, et al. (1992). "Animation and the Role of Map Design in Scientific Visualization." Cartography and Geographic Information Systems **19**(4): 201-214.

DiBiase, D. and A. M. MacEachren (1992). "Animation and the role of map design in scientific visualization." Cartography and Geographic Information Systems **19**(4): 201-214.

Dieker, S. and R. H. Güting (2000). Plug and Play with Query Algebras: SECONDO-A Generic DBMS Development Environment. Proceedings of the 2000 International Symposium on Database Engineering \& Applications, IEEE Computer Society.

Döllner, J. (2000). Dynamic 3D maps as visual interfaces for spatio-temporal data. ACM Workshop on Advances in Geographic Information Systems.

Döllner, J. and K. Hinrichs (1997). "Object-oriented 3D modeling, animation, and interaction." Journal of Visualization and Computer Animation **8**(1): 33-64.

Döllner, J. and K. Hinrichs (1998). Support of Explicit Time and Event Flows in the Object-Oriented Visualization Toolkit MAM/VRS. Heidelberg, Springer Verlag.

Dorling, D. and S. Openshaw (1992). "Using computer animation to visualize space - time patterns." Environment and Planning B: Planning and Design **19**(9): 639-650.

Dragicevic, P. and S. Huot (2002). "SpiraClock: a continuous and non-intrusive display for upcoming events." CHI '02: CHI '02 extended abstracts on Human factors in computing systems: 604-605.

Dueker, K. J. (1979). Land Resource Information Systems: a review of 15 years experience. Geo-Processing.

Dyreson, C., F. Grandi, et al. (1994). "A consensus glossary of temporal database concepts." SIGMOD Rec. **23**(1): 52-64.

E.Andonoff, C.Mendibourne, et al. (1993). "OHQL : un langage graphique pour l'interrogation d'une base de données." Ingénierie des systèmes d'information **1**(2): 203-227.

Edsall, R. and D. Peuquet (1997). A graphical user interface for the integration of time into GIS. ACSM/ASPRS, Seattle, Washington.

Edsall, R. M., G. Andrienko, et al. (2008). Interactive maps for exploring spatiotemporal data. ASPRS Manual of Geographic Information Science. M. Madden. Bethesda, MD: American Society of Photogrammetry and Remote Sensing.

Egenhofer, M. (1994). SPATIAL SQL: A Query and Presentation Language. IEEE Transactions on Knowledge and Data Engineering.

Eick, S. G., J. L. Steffen, et al. (1992). "Seesoft-A Tool for Visualizing Line Oriented Software Statistics." IEEE Trans. Softw. Eng. **18**(11): 957-968.

ESRI. (2004). "ArcGIS: The Complete Enterprise GIS." from http://www.esri.com/software/arcgis/index.html.

F.A.Cardenas, I.T.Ieong, et al. (1995). "The Knowledge-Based Object-Oriented PICQUERY+ Language." IEEE Transactions on Knowledge and Data Engineering **5**(4): 644-657.

F.Allen, J. (1983). Maintaining knowledge about temporal intervals. Communications of the ACM, ACM Press.

F.Staes, L.Tarantino, et al. (1991). A Graphical Query Language for Object Oriented Databases. IEEE Workshop on Visual Languages, Kobe, Japan.

Fabrikant, S. and K. Goldsberry (2005). Thematic relevance and perceptual salience of dynamic geovisualization displays. Proceedings of the 22th ICA/ACI International Cartographic Conference, A Coruña, Spain.

Fabrikant, S. and D. Josselin (2003). "La «cartactive», cartographie en mouvement: un nouveau domaine de recherche pluridisciplinaire ou un plan de la géomatique?" Revue Internationale de Géomatique, Numéro spécial «cartographie animée et interactive» **13**(1): 7-14.

Fairchild, K. M. (1993). Information Management Using Virtual Reality-Based Visualizations. Virtual Reality: Applications and Explorations. A. Wexelblat, Academic Press Professional, Cambridge, MA: 45-74.

Fairchild, K. M., S. E. Poltrock, et al. (1988). SemNet: Three-Dimensional Graphic Representations of Large Knowledge Bases. Cognitive Science and its Applications for Human-Computer Interaction. R. Guindon. Hillsdale NJ, Lawrence Erlbaum.

Feiner, S. K. and C. Beshers (1990). "Worlds within worlds: metaphors for exploring n-dimensional virtual worlds." UIST '90: Proceedings of the 3rd annual ACM SIGGRAPH symposium on User interface software and technology: 76-83.

Ferguson, S. (1991). The 1753 carte chronographique of Jacques Barbeu-Dubourg. Princeton University Library Chronicle. **52:** 190-230.

Franck, G. and C. Ware (1994). "Representing nodes and arcs in 3D networks." IEEE Symposium on Visual Languages, 1994. Proceedings: 189-190.

Frank, A. U. and S. Timpf (1994). "Multiple representations for cartographic objects in a multi-scale tree - An intelligent graphical zoom." Computers & Graphics **18**(6): 823-829.

Freeman, E. and S. Fertig. (1995). "Lifestreams: Organizing your electronic life."

Furnas, G. W. (1986). Generalized fisheye views. Proceedings of the SIGCHI conference on Human factors in computing systems. Boston, Massachusetts, United States, ACM.

Gad, A. (1986). "A temporally oriented data model." ACM Trans. Database Syst. **11**(4): 499-527.

Gadia, S. K. (1986). Weak temporal relations 5th ACM SIGACT-SIGMOD symposium on Principles of database systems Cambridge, Massachusetts, United States ACM Press.

GeoServer. (2001). "GeoServer." Retrieved 2008-05-28, 2008, from http://geoserver.org/display/GEOS/Welcome.

GEOTOOLS. (2008). "GeoTools, The Open Source Java GIS Toolkit." Retrieved 2008/09/28, 2008, from http://geotools.codehaus.org/.

Gersmehl, P. J. (1990). "Choosing Tools: Nine Metaphors of Four-Dimensional Cartography." Cartographic Perspectives **5**: 3-17.

Gould, P., D. DiBiase, et al. (1990). "Le SIDA: la carte animee comme rhetorique cartographique appliquee." Mappe Monde **90**: 21-26.

Granitzer, M., W. Kienreich, et al. (2004). "Evaluating a System for Interactive Exploration of Large, Hierarchically Structured Document Repositories." INFOVIS '04: Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04): 127-134.

GRASS. (1982). "The GRASS Project." from http://grass.itc.it/.

Grumbach, S., M. Koubarakis, et al. (2003). Spatio-temporal Models and Languages: An approach based on constraints. Spatio-Temporal Databases: The CHOROCHRONOS Approach. M. Koubarakis, T. K. Sellis, A. U. Franket al, Springer. **2520:** 177-201.

Grumbach, S., P. Rigaux, et al. (2000). The DEDALE prototype system, Springer-Verlag.

Grumbach, S., P. Rigaux, et al. (1998). "The DEDALE system for complex spatial queries." SIGMOD Rec. **27**(2): 213-224.

Grumbach, S., P. Rigaux, et al. (2000). Manipulating Interpolated Data is Easier Than You Thought. VLDB 2000, Cairo, Egypt.

Günter, O. (1998). Environmental Information Systems, Springer.

Güting, R. H. (1993). "Second-order signature: a tool for specifying data models, query processing, and optimization." SIGMOD Rec. **22**(2): 277-286.

Gütting, R. H. (1994). "An Introduction to Spatial Database Systems." VLDB Journal, Special issue on spatial database systems **3**(4): 357-399.

Gütting, R. H. (1994). "An Introduction to Spatial Database Systems." VLDB Journal **3**(4): 357-399.

Gütting, R. H. and M. Schneider (1995). "Realm-Based Spatial Data Types: The ROSE Algebra." VLDB

Journal **4**(2): 243-286.

Harris, R. L. (1999). Information Graphics: A Comprehensive Illustrated Reference. New York, NY, USA, Oxford University Press, Inc.

Harrower, M. (2002). "Visualizing Change: Using Cartographic Animation to Explore Remotely-Sensed Data." Cartographic Perspectives **39**: 30-42.

Harrower, M. (2007). "The cognitive limits of animated maps." Cartographica **42**(4): 269-277.

Harrower, M. and S. Fabrikant, Eds. (2008). The role of map animation in geographic visualization. Geographic Visualization. Chichester UK, Wiley and Sons.

Harrower, M. and B. Sheesley (2007). "Utterly lost: Methods for reducing disorientation in 3-D fly-over maps." Cartography and Geographic Information Science **34**(1): 19-28.

Havre, S., B. Hetzler, et al. (2000). "ThemeRiver: Visualizing Theme Changes over Time." INFOVIS '00: Proceedings of the IEEE Symposium on Information Vizualization 2000: 115.

Healey, C. G., K. S. Booth, et al. (1993). High-Speed Visual Estimation Using Preattentive Processing. Vancouver, BC, Canada, Canada, University of British Columbia.

Hearst, M. A. (1995). "TileBars: visualization of term distribution information in full text information access." CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems: 59-66.

Hendley, R. J., N. S. Drew, et al. (1995). "Case study: Narcissus: visualising information." INFOVIS '95: Proceedings of the 1995 IEEE Symposium on Information Visualization: 90.

Hertzmann, A. (2003). "A Survey of Stroke-Based Rendering." IEEE Computer Graphics & Applications, Special Issue on Non-Photorealistic Rendering **23**(4): 70-81.

IBM (2008). IBM Spatial Solutions.

Inselberg, A. (1997). "Multidimensional detective." INFOVIS '97: Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis '97): 100.

Inselberg, A., M. Reif, et al. (1987). "Convexity algorithms in parallel coordinates." Journal of ACM **34**(4): 765–801.

J.Chae and S.Lee (1995). Natural Language Query Processing in Korean Interface for Object-Oriented

Databases. 1st Workshop on Applications of Natural Language to Data Bases, Versailles, France.

J.R.Carter (1989). On defining the Geographic Information System. Fundamental of Geographic Information System: a compendium. W.J.Ripple. Falls Church, VA, ACPRS/ACSM: 3-7.

Jayaraman, S. and C. North (2002). "A radial focus+context visualization for multi-dimensional functions." VIS '02: Proceedings of the conference on Visualization '02: 443-450.

Jensen, C. S. (2000). Temporal data management. **PhD**.

Jensen, C. S., L. Mark, et al. (1999). Using caching, cache indexing and differential techniques to efficiently support transaction time, University of Maryland at College Park: 27.

Jensen, C. S. and R. T. Snodgrass (1999). "Temporal data management." IEEE Knowledge and Data Engineering **11**(1): 36-44.

Jensen, C. S., M. D. Soo, et al. (1994). "Unifying temporal data models via a conceptual model." Inf. Syst. **19**(7): 513-547.

Jihong, G. (2005). "GQL: Extending Xquery to query GML documents." Geo-Spatial Information Science **9**(2): 118-126.

Johnson, B. and B. Shneiderman (1991). "Tree-Maps: a space-filling approach to the visualization of hierarchical information structures." VIS '91: Proceedings of the 2nd conference on Visualization '91: 284-291.

Johnson, I. (2004). "Putting Time on the Map - Using TimeMap™ for Map Animation and Web Delivery." GeoInformatics **July/August**.

Johnson, I. (2005). "Indexing and Delivery of Historical Maps online using TIMEMAP." National Library of Australia Magazine.

K. Torp, C. J., Michael Böhlen (1997). Layered Implementation of Temporal DMBSs – Concepts and Techniques. TimeCenter Technical Report.

Karlsson, A. (2004). "GIS and Spatial Extensions with MySQL." Retrieved 2008-09-28, 2008, from http://dev.mysql.com/tech-resources/articles/4.1/gis-with-mysql.html.

Keahey, T. A. (2003). Network Visualization Course, Indiana University.

Keiding, M. (1990). Statistical inference in the lexis diagram. Philosophical transactions of the Royal Society of London.

Keller, P. R. and M. M. Keller (1994). Visual Cues: Practical Data Visualization. Los Alamitos, CA, USA, IEEE Computer Society Press.

Kosara, R., S. Miksch, et al. (2002). "Useful properties of Semantic Depth of Field for better F+C visualization." VISSYM '02: Proceedings of the symposium on Data Visualisation 2002: 205-210.

Koubarakis, M., T. K. Sellis, et al. (2003). Spatio-Temporal Databases: The CHOROCHRONOS Approach, Springer-Verlag.

Koussoulakou, A. and M. J. Kraak (1992). "Spatio-temporal maps and cartographic communication." The Cartographic Journal **29**(2): 101-108.

Kraak, M. J. (1994). Interactive modelling environment for three-dimensional maps: Functionality and interface issues. Visualization in modern cartography. A. M. MacEachren and D. R. F. Taylor. Oxford, Pergamon Press: 269-285.

Lamping, J. and R. Rao (1994). "Laying out and visualizing large trees using a hyperbolic space." UIST '94: Proceedings of the 7th annual ACM symposium on User interface software and technology: 13-14.

Larue, T., D. Pastre, et al. (1993). Strong Integration of Spatial Domains and Operators in a Relational Database System. 3rd Symposium on Spatial Databases, Singapour.

LeBlanc, J., M. O.Ward, et al. (1990). Exploring n-dimensional databases. IEEE Visualization '90, San Francisco, California.

Leslie Edwin Mckenzie, Jr. (1988). An algebraic language for query and update of temporal databases, The University of North Carolina at Chapel Hill: 326.

Leung, Y. K. and M. D. Apperley (1994). "A Review and Taxonomy of Distortion-Oriented Presentation Techniques." ACM Trans. Comput.-Hum. Interact. **1**(2): 126-160.

Lorentzos, N. A. and Y. G. Mitsopoulos (1997). "SQL Extension for Interval Data." IEEE Trans. on Knowl. and Data Eng. **9**(3): 480-499.

M. Böhlen, C. J., R. Snodgrass (1995). Evaluating the completeness of TSQL2, Recent Advances in Temporal Databases. J. C. a. A.Tuzhilin.

M. Dumas, C. F., P. Scholl (2000). TEMPOS: A Temporal Database Model Seamless Extending ODMG. LSR-IMAG. Grenoble, University of Grenoble.

M.H. Böhlen, C. S. J., B.Skjellaug (1998). Spatio-temporal Database Support for Legacy Applications. SIGMOD'98.

MacEachren, A. M. (1994). How Maps Work: Representation, Visualization, and Design, The Guilford Press.

Maceachren, A. M. (2004). How Maps Work: Representation, Visualization, and Design, The Guilford Press.

MacEachren, A. M., F. P. Boscoe, et al. (1998). Geographic Visualization: Designing Manipulable Maps for Exploring Temporally Varying Georeferenced Statistics. Proceedings of the 1998 IEEE Symposium on Information Visualization. North Carolina, IEEE Computer Society.

MacEachren, A. M. and M.-J. Kraak (2000). Research Challenges in Geovisualization. C. I. V. a. V. Environments, International Cartographic Association.

Mackinlay, J. (1986). "Automating the design of graphical presentations of relational information." ACM Trans. Graph. **5**(2): 110-141.

Mackinlay, J. D., G. G. Robertson, et al. (1991). "The perspective wall: detail and context smoothly integrated." CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems: 173-176.

Maguire, D. J., M. F. Goodchild, et al. (1991). Geographical Information Systems: Principles and Applications.

Maletic, J. I., A. Marcus, et al. (2002). ATask Oriented View of Software Visualization. Proceedings of the 1st International Workshop on Visualizing Software for Understanding and Analysis, IEEE Computer Society.

Marcus, A., L. Feng, et al. (2003). 3D Representations for Software Visualization. ACM symposium on Software visualization, San Diego, California, ACM New York, NY, USA

Marey, E. J. (1885). La Méthode Graphique. Paris.

Matkovic, K. i., H. Hauser, et al. (2002). "Process Visualization with Levels of Detail." INFOVIS '02: Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02): 67.

Matos, J. L. d. (2001). Fundamentos de Informação Geográfica. Lisboa, Portugal.

Melton, J. and A. R. Simon, Eds. (1993). Understanding the New SQL: A complete guide, Morgan Kaufmann.

Microsoft (2008). Microsoft SQL Server - product documentation.

Mihalisin, T., E. Gawlinski, et al. Visualizing a scalar field on an ndimensional lattice. IEEE Visualization '90, San Francisco, California,.

Monmonier, M. (1990). Strategies for the visualization of geographic time-series data. Cartographica. **1:** 30-45.

Moreira, J., C. Ribeiro, et al. (2008). A model of approximations for representing time-varying multidimensional data Data and Services Management in Mobile Environments (DS2ME) Workshop, IEEE 24th International Conference on Data Engineering.

Morse, E., M. Lewis, et al. (2002). "Testing visual information retrieval methodologies case study: comparative analysis of textual, icon, graphical, and "spring" displays." J. Am. Soc. Inf. Sci. Technol. **53**(1): 28-40.

Müller, W. and H. Schumann (2003). Visualization methods for time-dependent data-an overview. Winter simulation conference, New Orleans, USA.

Munzner, T. and P. Burchard (1995). "Visualizing the structure of the World Wide Web in 3D hyperbolic space." VRML '95: Proceedings of the first symposium on Virtual reality modeling language: 33-38.

Nation, D. A., C. Plaisant, et al. (1997). "Visualizing Web Sites using a Hierarchical Table of Contents Browser: WebToc." Proceedings of the third conference on Human Factors and the Web.

Navathe, S. B. and R. Ahmed (1989). "A temporal relational model and a query language." Inf. Sci. **49**(1-3): 147-175.

OGC04-095 (2005). OpenGIS® Filter Encoding Implementation Specification. P. A. Vretanos.

OGC (2000). OpenGIS® Geography Markup Language Specification.

OGC (2000). OpenGIS® Web Map Server Interface Implementation Specification.

OGC (2002). Web Feature Service Implementation Specification.

OGC. (2007). "OGC Glossary of Terms." Retrieved October 2007, from http://www.opengeospatial.org/ogc/glossary.

OGC (2008). OpenGIS® City Geography Markup Language (CityGML) Encoding Standard

OLIVE. (1997). "OLIVE: Online Library of Information Visualization Environments." from http://otal.umd.edu/Olive/.

Oliveira, M. A. (2007). Acesso Distribuído e Interoperável à Informação Geográfica para Suporte à Gestão de Infra-estruturas Críticas. Dept. de Sistemas de Informação. Guimarães, Universidade do Minho. **MSc.:** 150.

Olsen, K. A., R. R. Korfhage, et al. (1993). "Visualization of a document collection: the vibe system." Inf. Process. Manage. **29**(1): 69-81.

OpenGIS (1999). OpenGIS Simple Features Implementation Specification for SQL.

Oracle Corporation (2001). Oracle Spatial 9.0.1 - Technical white paper. O. Corporation.

Oracle Corporation (2007). Oracle Database 11g Workspace Manager White Paper. An Oracle White paper.

Oracle Corporation (2008). Oracle Cartdriges - product information.

Özsoyoğlu, G. and R. T. Snodgrass (1995). "Temporal and Real-Time Databases: A Survey." IEEE Trans. Knowledge and Data Enginnering **7**(4): 513-532.

Patton, D. K. and R. G. Cammack (1996). An examination of the effects of task type and map complexity on sequenced and static choropleth maps. Cartographic Design: Theoretical and Practical Perspectives. W. a. Keller. New York, John Wiley and Sons: 237-252.

Perera, N., A. Goodman, et al. (2007). "Preattentive processing: using low-level vision psychology to encode information in visualisations." SAC '07: Proceedings of the 2007 ACM symposium on Applied computing: 1090-1091.

Pickett, R. M. and G. G. Grinstein (1988). Iconographics displays for visualizing multidimensional data. IEEE Conference on Systems, Man, and Cybernetics, Beijing and Shenyang.

Plaisant, C. (2004). The challenge of information visualization evaluation. Proceedings of the working conference on Advanced visual interfaces. Gallipoli, Italy, ACM.

Pomaska, G. (2004). Between photo-realism and non-photo realistic rendering - modeling urban areas for realtime VR.

PostGIS. (2008). "PostGIS spatial database extension for PostgreSQL." Retrieved 2008-09-28, 2008, from http://postgis.refractions.net.

Priestley, J. (1765). A Chart of Biography. London: BL: 611.I.19.

Rao, R. and S. K. Card (1995). "Exploring large tables with the table lens." CHI '95: Conference companion on Human factors in computing systems: 403-404.

Refractions Research Inc. (2001). "uDIG: User-friendly desktop Internet GIS." from http://udig.refractions.net/.

Rekimoto, J. and M. Green. (1993). "The information cube: Using transparency in 3d information visualization."

Relly, L. and U. Röhm (1999). Plug and Play: Interoperability in CONCERT. DBLP INTEROP'99.

Revesz, P. Z. (1998). Constraint Databases: A Survey. Selected Papers from a Workshop on Semantics in Databases, Springer-Verlag.

Rhyne, T.-M., A. MacEachren, et al. (2006). "Guest Editors' Introduction: Exploring Geovisualization." IEEE Computer Graphics and Applications **26**(4): 20-21.

Rhyne, T. M., W. Ivey, et al. (1994). Visualization and Geographic Information System integration: what are the needs

and the requirements, if any?? IEEE Computer Society Visualization 94, Washington, IEEE Computer Society Press, Los Alamitos, California.

Robertson, G. G. and J. D. Mackinlay (1993). "The document lens." UIST '93: Proceedings of the 6th annual ACM symposium on User interface software and technology: 101-108.

Robertson, G. G., J. D. Mackinlay, et al. (1991). "Cone Trees: animated 3D visualizations of hierarchical information." CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems: 189-194.

Rocha, A. and A. V. d. Carvalho (2006). Temporal WFS XML e Tecnologias Associadas, Portalegre, Portugal.

Rowe, L. and M. Stonebraker (1987). The Postgres Data Model. 13th International Conference on Very Large Data Bases, Brighton, UK.

S. Grumbach, P. R., L. Segoufin (2000). Manipulating Interpolated Data is Easier Than You Thought. VLDB 2000, Cairo, Egypt.

Santos, S. and K. Brodlie (2004). "Gaining understanding of multivariate and multidimensional data through visualization." Computers & Graphics **28**(3): 311-325.

Sarda, N. L. (1990). "Extensions to SQL for Historical Databases." IEEE Trans. on Knowl. and Data Eng. **2**(2): 220-230.

Sarkar, M. and M. H. Brown (1992). "Graphical Fisheye Views of Graphs." Human Factors in Computing Systems, CHI'92 Conference Proceedings: Striking A Balance: 83-91.

Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. Proceedings of the 1996 IEEE Symposium on Visual Languages, IEEE Computer Society.

Snodgrass, R., M. Böhlen, et al. (1996). Adding Transaction time to SQL/Temporal, SQL/Temporal Change Proposal. I. I. J. S. W. D. M.-r. ANSI X3H2-96-502r2.

Snodgrass, R. T. (1987). "The temporal query language TQuel." ACM Trans. Database Syst. **12**(2): 247-298.

Snodgrass, R. T. (1993). "An overview of TQUEL." ACM Transactions on Information Systems **2**(2).

Snodgrass, R. T. (1996). A Road Map of Additions to SQL/Temporal. A. X3H2-96-013.

Snodgrass, R. T. (1998) "Temporal Support in Standard SQL." **Volume**, DOI:

Snodgrass, R. T. (2006). "TSQL2 and SQL3 Interactions." Retrieved 2008-09-11, 2008, from http://www.cs.arizona.edu/~rts/sql3.html.

Snodgrass, R. T., I. Ahn, et al. (1994). TSQL2 Language Specification.

Snodgrass, R. T., M. H. Böhlen, et al. (1996). Adding Valid Time to SQL/Temporal. ANSI Expert's Contribution. I. I. J. S. W. D. M.-r. ANSI X3H2-96-501r2, ANSI.

Snodgrass, R. T., C. Dyreson, et al. (2008). "Validating quicksand: Temporal schema versioning in \&\#964;XSchema." Data Knowl. Eng. **65**(2): 223-242.

Spoerri, A. (1993). "InfoCrystal: a visual tool for information retrieval & management." CIKM '93: Proceedings of the second international conference on Information and knowledge management: 11-20.

Steiner, A. (1998). A Generalization Approach to temporal Data Models and Their Implementations. Zurich. **Phd thesis**.

Stonebraker, M., J. Chen, et al. (1993). "Tioga: a database-oriented visualization tool." VIS '93: Proceedings of the 4th conference on Visualization '93: 86-93.

Storey, M.-A. D. and H. A. Muller (1995). "Manipulating and documenting software structures using SHriMP views." ICSM '95: Proceedings of the International Conference on Software Maintenance: 275.

Stuart K. Card, J. D. M., Ben Shneiderman, Ed. (1999). Readings in information visualization: using vision to think, Morgan Kaufmann Publishers Inc.

Tansel, A., J. Clifford, et al. (1994). Temporal Databases: Theory, Design and Implementation, Benjamin/Cummings.

Thrower, N. J. W. (1961). "Animated cartography in the United States." International Yearbook of Cartography(1): 20-30.

Tobler, W. R. (1970). "A Computer Movie Simulating Urban Growth in the Detroit Region." Economic Geography **46**(2): 234-240.

Tominski, C., J. Abello, et al. (2004). "Axes-based visualizations with radial layouts." SAC '04: Proceedings of the 2004 ACM symposium on Applied computing: 1242-1247.

Tominski, C., P. Schulze-Wollgast, et al. (2005). 3D information visualization for time dependent data on maps.

Tony Griffiths, A. A. A. F., NormanW. Paton, Keith T. Mason, Bo Huang, Mike Worboys, Chris Johnson, John Stell (2001). TRIPOD : A Comprehensive System for the Management of Spatial and Aspatial Historical Objects. 20th International Conference on Conceptual Modeling: Conceptual Modeling.

Tory, M., M. Campo, et al. (1997). "Automatic abstraction management in information visualization systems." IV '97: Proceedings of the IEEE Conference on Information Visualisation: 50.

Tory, M. and T. Möller (2004). "Human Factors in Visualization Research." IEEE Transactions on Visualization and Computer Graphics **10**(1): 72-84.

Treisman, A. (1985). "Preattentive processing in vision." Comput. Vision Graph. Image Process. **31**(2): 156-177.

Tryfona, N., R. Price, et al. (2003). Conceptual Models for Spatio-temporal Applications. Spatio-Temporal Databases: The CHOROCHRONOS Approach. M. Koubarakis, T. K. Sellis, A. U. Franket al, Springer. **2520:** 79-116.

Tufte, E. R. (2001). The Visual Display of Quantitative Information, Graphics Press.

Tukey, J. W. (1977). Exploratory Data Analysis, Addison-Wesley.

V.M.Ozemoy, D. R. Smith, et al. (1981). "Evaluating Computerized Geographic Information Systems Using Decision Analysis." Interfaces **11**: 92-98.

van Dam, A. A. (2000). "Immersive VR for scientific visualization: a progress report." IEEE computer graphics and applications **20**(6): 26-27.

van Wijk, J. J. and R. van Liere (1993). HyperSlice: visualization of scalar functions of many variables. VIS '93: Proceedings of the 4th conference on Visualization '93, San Jose, California.

van Wijk, J. J. and E. R. van Selow (1999). "Cluster and Calendar Based Visualization of Time Series Data." INFOVIS '99: Proceedings of the 1999 IEEE Symposium on Information Visualization: 4.

Vassilakis, C., P. Georgiadis, et al. (1996). A Comparative Study of Temporal DBMS Architectures. DEXA Workshop.

Vimercati, S. D. C. d. (2002). An authorization model for temporal XML documents. Proceedings of the 2002 ACM symposium on Applied computing. Madrid, Spain, ACM.

Voigt, R. (2002). An Extended Scatterplot Matrix and Case Studies in Information Visualization. Fachbereich Elektrotechnik. Magdeburg, Germany, Hochschule Magdeburg-Stendal. **Master's thesis**.

Ward, M. O. (1994). XmdvTool: Integrating multiple methods for visualizing multivariate data. IEEE Visualization '94, Washington, DC.

Wattenberg, M. (1999). "Visualizing the stock market." CHI '99: CHI '99 extended abstracts on Human factors in computing systems: 188-189.

Wijk, J. J. V. and H. v. d. Wetering (1999). Cushion Treemaps: Visualization of Hierarchical Information. IEEE Symposium on Information Visualization, IEEE Computer Society.

Wise, J. A., J. J. Thomas, et al. (1995). "Visualizing the non-visual: spatial analysis and interaction with information from text documents." INFOVIS '95: Proceedings of the 1995 IEEE Symposium on Information Visualization: 51.

Wong, P. C. and R. D. Bergeron (1997). "30 Years of Multidimensional Multivariate Visualization." Scientific Visualization, Overviews, Methodologies, and Techniques: 3-33.

Wood, A., R. Beale, et al. (1995). "HyperSpace: A World-Wide Web Visualiser and its Implications for Collaborative Browsing and Software Agents."

Wu, Y., S. Jajodia, et al. (1998). Temporal Database Bibliography Update. Temporal Databases: Research and Practice. S. J. O. Etzion, and S. Sripada, Springer-Verlag: 338–366.

York, J., S. Bohn, et al. (1995). Clustering and Dimensionality Reduction in SPIRE. Symp. on Advanced Intelligence Processing and Analysis.

Zaniolo, C., S. Ceri, et al. (1997). Overview of temporal databases. Advanced Database Systems. M. K. P. Inc.

Zloof, M. M. (1977). "Query-by-Example : A Database Language." IBM Systems Journal **16**(4): 324-343.



Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, s/n 4200-465 Porto PORTUGAL
www.fe.up.pt

FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO
BIBLIOTECA
0000101289