International Conference on Interoperating Geographic Information Systems, 1997

CONFERENCE
December 3-4, 1997
Radisson Hotel,
Santa Barbara, California

This conference organized by:

Conference Background

The National Center for Geographic Information and Analysis and the Open GIS Consortium Inc announce an International Conference on Interoperating Geographic Information Systems, to be held in Santa Barbara December 3-4, 1997, and to be followed December 5-6 by an invitational Workshop. Topics to be addressed at the conference include the current state of research in related disciplines concerning the technical, semantic, and organizational issues of GIS interoperation; case studies of GIS interoperation; theoretical frameworks for interoperation; and evaluations of alternative approaches. The program will include invited keynote presentations and contributed papers; limited space will also be available for posters, demonstrations, and exhibits.

This conference is part of the Varenius Project's research initiative on Interoperating GIS.

Conference Scientific Program Committee

David Abel
CSIRO, Australia

Kurt Buehler
Open GIS Consortium

Max Egenhofer
University of Maine (co-chair)

Robin Fegeas
US Geological Survey

Alan Gaines
US National Science Foundation

Michael Goodchild
University of California, Santa Barbara (co-chair)

Werner Kuhn
University of Munster

Richard Muntz
UCLA

David Schell
Open GIS Consortium (co-chair)

Greg Smith
US National Imagery and Mapping Agency

Terence Smith
University of California, Santa Barbara

Andrej Vckovski
University of Zurich

John Herring
Oracle

Agnes Voisard
Free University of Berlin

Cliff Kottman
Open GIS Consortium

Maria Zemankova
US National Science Foundation

Conference Program

Wednesday, December 3rd

7:30-8:30 am - Registration, El Cabrillo Foyer

8:00-8:30 am - Continental Breakfast, El Cabrillo Room

8:30 am - Opening and Welcome

Max Egenhofer
University of Maine
Michael Goodchild
University of California, Santa Barbara
David Schell
Open GIS Consortium

8:30-10:00 am

Session 1.1 Invited keynote presentations

The U.S. Defense Vision and its Implications for GIS Technology
Annette Krygiel
National Defense University

Semantic Interoperability in Infocosm: Moving Beyond Infrastructural and Data Interoperability in Federated Information Systems
Amit Sheth
University of Georgia

10:00-10:30 Morning Break

10:30-12:00

Session 2.1 - Panel Discussion: Ongoing Activities to Promote Interoperability

David Schell, Cliff Kottmann, Kurt Buehler, and Greg Buehler
Open GIS Consortium
Alan Gaines
National Science Foundation
Robin Fegeas
U.S. Geological Survey
Greg Smith

National Imagery and Mapping Agency

12:00-1:30 - Lunch, served Poolside

1:30-3:00 - Two Concurrent Sessions

Session 3.1 - Theory of Interoperating GISs

Interoperability and Spatial Information Theory
Andrej Vckovski
University of Zurich, Switzerland

A Specification Language for Interoperable GIS
Andrew Frank
Technical University of Vienna, Austria
Werner Kuhn
University of Muenster, Germany

Session 3.2 The Institutional Context of Interoperation

Real-World Lessons in Organizational and Technological Interoperability for Geographic Information Infrastructures
John Evans
Massachusetts Institute of Technology

Execution Planning in a Spatial Internet Marketplace
Volker Gaede
CSIRO, Australia

Probing the Concept of Information Communities: A Road Towards Semantic Interoperability
Y.A. Bishr, M. Molenaar and M. Radwan
ITC, The Netherlands
H. Pundt and W. Kuhn
University of Muenster, Germany

Interoperability through Organization: The Role of Digital Libraries in Distributed Knowledge Management
Xavier Lopez
University of California, Berkeley

3:00-3:30 Afternoon Break

3:30-5:00 - Three Concurrent Sessions

Session 4.1 Semantic Interoperability

Accounting for the Semantic Differences between Various Geographic Information Systems
Mark Gahegan

Curtin University of Technology, Australia

Designing for Interoperability Overcoming Semantic Differences
Francis Harvey
EPFL-IGEO-SIRS, Lausanne, Switzerland

Development of a Global Conceptual Schema for Interoperable Geographic Information
May Yuan
University of Oklahoma

Session 4.2 Interoperation in the Transportation Domain

The Need for a Formal GIS Transportation Model
Stephen Bespalko
Sandia National Laboratories
Max Wyman
Terra Genesis, Tempe AZ
John Sutton
GIS/Trans Ltd.

Real-Time Data Exchange and Interoperability
Fred Latham and David Siegel
Viggen Corp, Knoxville TN
Demin Xiong
Oak Ridge National Laboratory

Interoperability Issues in Intelligent Transportation Systems: Testing the Cross Streets Profile
Val Noronha
University of California, Santa Barbara

Assessing Topological Similarity of Spatial Networks
John Nystuen, Andrea Frank, and Larry Frank
University of Michigan

Session 4.3 Systems Experiences I

GeoToolKit: Opening the Access to Object-Oriented Geodata Stores
Oleg Balovnev, Martin Bruenig, Armin Cremers, and Serge Shumilov
Institute of Computer Science III, University of Bonn, Germany:

Interoperability of Geographic Information: From the Spreadsheet to Virtual Environments
Pedro Pereira Gonçalves, Nelson Neves, João Silva, Joaquim Muchaxo, and António Câmara
New University of Lisbon, Portugal

A Virtual Geospatial Information Server (VGIS) Providing Transparent Access to Heterogeneous Sources
Changchu Wang, Liya Ding, and Jiankang Wu
National University of Singapore

5:00-6:30 Reception, sponsored by ESRI

Poster Presentations in the Reception Hall

Structural Design of Distributed Geographic Information Systems Based on a High Order Logic
Guillaume Koum
Universite de Bourgogne, France

Land planning through geographical information systems, remote sensing and multicriteria decision analysis
Germana Manca and Andrea De Montis
Universita' di Sassari, Italy

Characterization of Data, Queries, and Index Performance of Geographic Information Systems with Applications to Informix Geodetic DataBlade Module
Kumar Ramaiyer
Informix Software, Inc.

Thursday December 4th

8:00-8:30 am - Continental Breakfast, El Cabrillo Room

8:30-10:00

Session 5.1 Invited Keynote Presentations

Information Systems for Environmental Analysis
Jeff Dozier
University of California, Santa Barbara

From GISystems to GIServices: Spatial Computing on the Internet Marketplace
Oliver Guenther
Humboldt University, Berlin

10:00-10:30 Morning Break

10:30-12:00 Three Concurrent Sessions

Session 6.1 Conceptual Design

NCSA

Inteoperating GISs Using the Open Geospatial Datastore Interface (OGDI)
Paul Morin
J2 Geomatics, Ottawa, Ontario, Canada
Denis Gouin
Defence Research Establishment, Val-Belair, Quebec, Canada
Gilles Clement and Christian Larouche
Logiciels et Applications Scientifiques Inc, Laval, Quebec, Canada
(to be presented by **Kenn Gardels**, UC Berkeley)

GEOLIB: A Software Component for Making GIS Tools Interoperable
Donatas Kvedarauskas, Patrice Boursier
University of La Rochelle, France
Xavier Culos, Thierry Deltheil, and Sylvie Iris
SILOGIC, Toulouse, France

12:00-1:00 Lunch, served Poolside

1:00-2:30 Three Concurrent Sessions

Session 7.1 Interoperation in the Environmental Domain II

FRIEND: Framework for the Integration of Environmental and Geographical Data
Martin Brändli and Andreas Ernst
University of Zurich, Switzerland

Comparing Approaches to Cooperation between GIS and Simulation Models to Identify Criteria for Interoperation
Neil Stuart
University of Edinburgh, UK

Using Design Pattern to Define Interoperable GIS Models
F. Balaguer and S. Gordillo
UNLP, Buenos Aires, Argentina

Session 7.2 Distributed Processing

Implementation of the OGIS Simple Feature Interface
Scott Morehouse
ESRI, Redlands CA

Query Processing in Distributed Spatial Databases
Walid Aref

Panasonic Technologies Inc, Princeton NJ

A Spatio-Temporal Deductive System for Marine Ecosystem Monitoring
M.E. Carboni, F. Giannotti, and M.V. Masserotti
CNUCE-CNR, Pisa, Italy

Supporting Interoperation of GIS Objects
Silvia Nittel and Richard Muntz
University of California, Los Angeles

Session 7.3 Systems Experiences III

Hot Links as a New Way of Data Integration in a Distributed Computing Environment
Andre Hagehuelsmann
Free University of Berlin and Intergraph Germany

IRIS: A Tool to Support Data Analysis with Maps
Gennady Andrienko and Nathalia Andrienko
German National Research Center for Information Technology, Germany

Information Brokers for a Web-Based Geographic Information System
Ian Finch and Eleanor Small
University of Liverpool, UK

2:30-3:00 Afternoon Break

3:00-4:30 Four Concurrent Sessions

Session 8.1 Theory of Interoperating GISs II

A Request Specification Language for Spatial Internet Marketplaces
Volker Gaede, Kerry Taylor, and Xiaofang Zhou
CSIRO, Australia

Interoperability by Exchanging Executable Content, or What Have PostScript and Java in Common?
Andrej Vckovski
University of Zurich. Switzerland

The Use of Functional Programming in the Specification and Testing Process
Werner Kuhn
University of Muenster, Germany
Andrew Frank
Technical University of Vienna, Austria

Session 8.2 Interoperating GISs in Large Organizations

Session 8.3 Heterogeneous Environments of the Internet

Session 8.4 Systems Experiences IV

Multi-Server Internet GIS: Standardization and Practical Experiences

Carel van den Berg, Frank Tuijnman, and Tom Vijlbrief
Professional GEO Systems, Amsterdam, The Netherlands
Co Meijer, Harry Uitermark, and Peter van Oosterom
Cadastre, Apeldoorn, The Netherlands

4:30-6:00

Session 9.1 Closing Panel: Building the Research Agenda

Michael Goodchild, University of California, Santa Barbara
Max Egenhofer, University of Maine
Andrew Frank, Technical University of Vienna, Austria
Derek Reeve, University of Huddersfield, UK
Cliff Kottman, Open GIS Consortium

Conference Program

Wednesday, December 3rd

7:30-8:30 am - Registration, El Cabrillo Foyer

8:00-8:30 am - Continental Breakfast, El Cabrillo Room

8:30 am - Opening and Welcome

Max Egenhofer
University of Maine
Michael Goodchild
University of California, Santa Barbara
David Schell
Open GIS Consortium

8:30-10:00 am

Session 1.1 Invited keynote presentations

The U.S. Defense Vision and its Implications for GIS Technology
Annette Krygiel
National Defense University

Semantic Interoperability in Infocosm: Moving Beyond Infrastructural and Data Interoperability in Federated Information Systems
Amit Sheth
University of Georgia

10:00-10:30 Morning Break

10:30-12:00

Session 2.1 - Panel Discussion: Ongoing Activities to Promote Interoperability

David Schell, Cliff Kottmann, Kurt Buehler, and Greg Buehler
Open GIS Consortium
Alan Gaines
National Science Foundation
Robin Fegeas
U.S. Geological Survey
Greg Smith
National Imagery and Mapping Agency

12:00-1:30 - Lunch, served Poolside

1:30-3:00 - Two Concurrent Sessions

Session 3.1 - Theory of Interoperating GISs

Interoperability and Spatial Information Theory
Andrej Vckovski
University of Zurich, Switzerland

A Specification Language for Interoperable GIS
Andrew Frank
Technical University of Vienna, Austria
Werner Kuhn
University of Muenster, Germany

Session 3.2 The Institutional Context of Interoperation

Real-World Lessons in Organizational and Technological Interoperability for Geographic Information Infrastructures
John Evans
Massachusetts Institute of Technology

Execution Planning in a Spatial Internet Marketplace
Volker Gaede
CSIRO, Australia

Probing the Concept of Information Communities: A Road Towards Semantic Interoperability
Y.A. Bishr, M. Molenaar and M. Radwan
ITC, The Netherlands
H. Pundt and W. Kuhn
University of Muenster, Germany

Interoperability through Organization: The Role of Digital Libraries in Distributed Knowledge Management
Xavier Lopez
University of California, Berkeley

3:00-3:30 Afternoon Break

3:30-5:00 - Three Concurrent Sessions

Session 4.1 Semantic Interoperability

Accounting for the Semantic Differences between Various Geographic Information Systems
Mark Gahegan
Curtin University of Technology, Australia

Designing for Interoperability Overcoming Semantic Differences
Francis Harvey
EPFL-IGEO-SIRS, Lausanne, Switzerland

Development of a Global Conceptual Schema for Interoperable Geographic Information

May Yuan
University of Oklahoma

Session 4.2 Interoperation in the Transportation Domain

The Need for a Formal GIS Transportation Model
Stephen Bespalko
Sandia National Laboratories
Max Wyman
Terra Genesis, Tempe AZ
John Sutton
GIS/Trans Ltd.

Real-Time Data Exchange and Interoperability
Fred Latham and David Siegel
Viggen Corp, Knoxville TN
Demin Xiong
Oak Ridge National Laboratory

Interoperability Issues in Intelligent Transportation Systems: Testing the Cross Streets Profile
Val Noronha
University of California, Santa Barbara

Assessing Topological Similarity of Spatial Networks
John Nystuen, Andrea Frank, and Larry Frank
University of Michigan

Session 4.3 Systems Experiences I

GeoToolKit: Opening the Access to Object-Oriented Geodata Stores
Oleg Balovnev, Martin Bruenig, Armin Cremers, and Serge Shumilov
Institute of Computer Science III, University of Bonn, Germany:

Interoperability of Geographic Information: From the Spreadsheet to Virtual Environments
Pedro Pereira Gonçalves, Nelson Neves, João Silva, Joaquim Muchaxo, and António Câmara
New University of Lisbon, Portugal

A Virtual Geospatial Information Server (VGIS) Providing Transparent Access to Heterogeneous Sources

Changchu Wang, Liya Ding, and Jiankang Wu
National University of Singapore

5:00-6:30 Reception, sponsored by ESRI

Poster Presentations in the Reception Hall

Structural Design of Distributed Geographic Information Systems Based on a High Order Logic
Guillaume Koum
Universite de Bourgogne, France

Land planning through geographical information systems, remote sensing and multicriteria decision
analysis
Germana Manca and Andrea De Montis
Universita' di Sassari, Italy

Characterization of Data, Queries, and Index Performance of Geographic Information Systems with Applications to Informix Geodetic DataBlade Module
Kumar Ramaiyer
Informix Software, Inc.

Thursday December 4th

8:00-8:30 am - Continental Breakfast, El Cabrillo Room

8:30-10:00

Session 5.1 Invited Keynote Presentations

Information Systems for Environmental Analysis
Jeff Dozier
University of California, Santa Barbara

From GISystems to GIServices: Spatial Computing on the Internet Marketplace
Oliver Guenther
Humboldt University, Berlin

10:00-10:30 Morning Break

10:30-12:00 Three Concurrent Sessions

Session 6.1 Conceptual Design

A GIS Interoperability Approach Based on ISO RM-ODP and ISO CSMF
Arne-Jurgen Berre, Vidar Knudsen, and Jon Oldevik
SINTEF Telecom and Informatice Oslo, Norway

A Middleware for Transparent Access to Multiple Spatial Object Databases
Sang Cha, Kihong Kim, Changbin Song, Jookwan Kim, Jooyong Jun, and Yongsik Kwon
Seoul National University, Korea

Constraint-Based Interoperability of Spatiotemporal Databases
Jan Chomicki
Monmouth University
Peter Revesz
University of Nebraska, Lincoln NE

Session 6.2 Interoperation in the Environmental Domain I

Interoperable GIS Applications: Tightly Coupling Environmental Models with GISs
Hassan Karimi
NC Supercomputing Center, Research Triangle Park NC

Integrating Environmental Models and GIS in the Framework of GIS Interoperability
Ling Bian
SUNY Buffalo

Spatial Process Modelling and Interoperability
Andrew Marr, Stephen MacDonnell, Samuel Marr and George Benwell
University of Otago, New Zealand

Interoperability with the Earth Science Remote Access Tool (ESRAT)
Robert Raskin and Elaine Dobinson
Jet Propulsion Laboratory, Pasadena CA

Session 6.3 Systems Experiences II

The Geospatial Interoperability Problems: Lessons Learned from Building the GeoLens Prototype
Cliff Behrens, Chumki Basu
Bell Communications Research
Leon Shklar, Edith Au
Pencom Web Works
Nancy Yaeger
NCSA

Inteoperating GISs Using the Open Geospatial Datastore Interface (OGDI)
Paul Morin
J2 Geomatics, Ottawa, Ontario, Canada
Denis Gouin
Defence Research Establishment, Val-Belair, Quebec, Canada
Gilles Clement and Christian Larouche
Logiciels et Applications Scientifiques Inc, Laval, Quebec, Canada
(to be presented by **Kenn Gardels**, UC Berkeley)

GEOLIB: A Software Component for Making GIS Tools Interoperable
Donatas Kvedarauskas, Patrice Boursier
University of La Rochelle, France
Xavier Culos, Thierry Deltheil, and Sylvie Iris
SILOGIC, Toulouse, France

12:00-1:00 Lunch, served Poolside

1:00-2:30 Three Concurrent Sessions

Session 7.1 Interoperation in the Environmental Domain II

FRIEND: Framework for the Integration of Environmental and Geographical Data
Martin Brändli and Andreas Ernst
University of Zurich, Switzerland

Comparing Approaches to Cooperation between GIS and Simulation Models to Identify Criteria for Interoperation
Neil Stuart
University of Edinburgh, UK

Using Design Pattern to Define Interoperable GIS Models
F. Balaguer and S. Gordillo
UNLP, Buenos Aires, Argentina

Session 7.2 Distributed Processing

Implementation of the OGIS Simple Feature Interface
Scott Morehouse
ESRI, Redlands CA

Query Processing in Distributed Spatial Databases
Walid Aref
Panasonic Technologies Inc, Princeton NJ

A Spatio-Temporal Deductive System for Marine Ecosystem Monitoring
M.E. Carboni, F. Giannotti, and M.V. Masserotti
CNUCE-CNR, Pisa, Italy

Supporting Interoperation of GIS Objects
Silvia Nittel and Richard Muntz
University of California, Los Angeles

Session 7.3 Systems Experiences III

Hot Links as a New Way of Data Integration in a Distributed Computing Environment
Andre Hagehuelsmann
Free University of Berlin and Intergraph Germany

IRIS: A Tool to Support Data Analysis with Maps
Gennady Andrienko and Nathalia Andrienko
German National Research Center for Information Technology, Germany

Information Brokers for a Web-Based Geographic Information System
Ian Finch and Eleanor Small
University of Liverpool, UK

2:30-3:00 Afternoon Break

3:00-4:30 Four Concurrent Sessions

Session 8.1 Theory of Interoperating GISs II

A Request Specification Language for Spatial Internet Marketplaces
Volker Gaede, Kerry Taylor, and Xiaofang Zhou
CSIRO, Australia

Interoperability by Exchanging Executable Content, or What Have PostScript and Java in Common?
Andrej Vckovski
University of Zurich. Switzerland

The Use of Functional Programming in the Specification and Testing Process
Werner Kuhn
University of Muenster, Germany
Andrew Frank
Technical University of Vienna, Austria

Session 8.2 Interoperating GISs in Large Organizations

A Web-Based Scientific Data Server for Accessing and Distributing Earth Science Data
Liping Di, R. Suresh, K. Doan, and Doug Ilg
Hughes STX Corporation
Ken McDonald
NASA Goddard Space Flight Center, Greenbelt MD

Geospatial Modelling: A Case Study for a Statewide Land Information Strategy
David Pullar
University of Queensland
Kristin Stock
Queensland Institute of Technology, Australia

Spatial Database Design for GIS Interoperability
Lorri Peltz, Marianne August, and Rose Medina
U.S. Geological Survey

Session 8.3 Heterogeneous Environments of the Internet

A Framework for Geographical Modeling in a Heterogeneous Computing Environment
David Bennett, Raja Sengupta, and Greg Wade
Southern Illinois University:

Automated Metadata Interpretation to Assist in the Use of Unfamiliar GIS Data Sources

Brandon Plewe and Steven Johnson
Brigham Young University

Software Agent-Oriented Frameworks for the Interoperability of Geomatics Systems: From Fundamental Concepts to the SIGAL Project
Z. Maamar, B. Moulin, Y. Bedard, and G. Babin
Laval University, Quebec, Canada

Session 8.4 Systems Experiences IV

Using the Internet to Access Geographic Information: An Open GIS Interface Prototype
Frederico Torres Fonseca, and Clodoveu Augusto Davis Jr.
PRODABEL, Belo Horizonte, Brazil

Some Examples of the Usage of Internet/Intranet Technology in GIS
Wolfgang Reinhardt
Institute for Geoinformation and Land Development, Neubiberg, Germany

Multi-Server Internet GIS: Standardization and Practical Experiences
Carel van den Berg, Frank Tuijnman, and Tom Vijlbrief
Professional GEO Systems, Amsterdam, The Netherlands
Co Meijer, Harry Uitermark, and Peter van Oosterom
Cadastre, Apeldoorn, The Netherlands

4:30-6:00

Session 9.1 Closing Panel: Building the Research Agenda

Michael Goodchild, University of California, Santa Barbara
Max Egenhofer, University of Maine
Andrew Frank, Technical University of Vienna, Austria
Derek Reeve, University of Huddersfield, UK
Cliff Kottman, Open GIS Consortium

Paper search by Author's Last Name:

A B C D E F G
H I J K L M N
O P Q R S T U
V W X Y Z

Back to Conference Timeline...

A

B

Top of list

C

Top of list

D

E

F

G

Top of list

H

I

J

K

Top of list

L

M

Top of list

N

O

P

Q

Top of list

R

S

Top of list

T

U

V

Top of list

W

X

Y

Z

Top of list

-

Semantic Interoperability in Infocosm: Moving Beyond Infrastructural and Data Interoperability in Federated Information Systems

Amit Sheth
Director, Large Scale Information Systems Lab
Department of Computer Science
University of Georgia
415 Graduate Studies Research Center
Athens GA 30602-7404 USA

Abstract

This talk starts with a look at the key concerns of interoperability that researchers have addressed in developing Federated Database Systems in 80's and in the research projects of mid-90's that have integrated a broader variety of heterogeneous data. We then make a case for increasing need to address the interoperability challenges at metadata and semantic levels to enable (what I call) Infocosm-- a society characterized by information any time, any where, and in many form, for effective decision making, better development and utilization for human intellect through knowledge intensive activities, and more fun.

In the process, we will look at the interoperability concerns in the context of three perspectives: (a) distribution, heterogeneity, and autonomy, (b) data, metadata and semantic (terminological, contextual), and (c) connectivity + computation, information and knowledge. We will give examples from relevant systems we have closely observed or are developing-- DDTS/Mermaid, InfoHarness and InfoQuit. In particular, the opportunities and interoperability related challenges presented by increased accessibility of heterogeneous digital media in Internet/Intranets will be discussed.

Interoperability and Spatial Information Theory

Andrej Vckovski
Spatial Data Handling Division
Department of Geography
University of Zurich

Abstract

This proposals focuses on the relevance of spatial information theory for the development and deployment of successful interoperability strategies. Interoperability between computing infrastructures needs -- much like every information exchange -- a set of common rules and concepts which define a common understanding of the information and operations available in every cooperating system.

Standardization processes and interoperability initiatives such as OGIS try to provide an agreed-on set of such rules and concepts. Such standardization processes are often driven by market forces and vendors trying to position their particular technology and product as a common concept. However, in spatial information systems (and also other areas) the common concepts do not only include technical aspects but also fundamental questions on modeling spatial, real-world features, i.e., problems which are maybe beyond a specific technical approach.

The process of understanding a real-world phenomena and providing a "code" for the communication of their states and relationships is the development of a theory. In that sense, a theory which is agreed-on by information exchanging communities provides the previously mentioned set of common rules and concepts. Therefore, interoperability of Geographic Information Systems needs such a theory, i.e., a spatial information theory as the theoretical framework for interoperability. In the context of interoperability, the development of a spatial theory (or a set of common understandings of spatial features) needs to address several almost contradictory objectives:

- The theory needs to be at least as expressive as every "theory" or concept which is already in use.
- The theory needs to be well-defined on the one hand (e.g., to avoid misunderstandings), and flexible on the other hand to allow modifications and extensions
- The theory needs to be stable in order to be trustworthy.
- The theory needs to be accepted by a large community.
- And finally, the theory needs to be simple to provide a realistic basis for a successful deployment and implementation.

The development of a "unified spatial information theory" has been discussed in the research community for several years, and it is not at all clear whether there exists

such a thing, whether there is a conception of spatial information which is fundamentally enough to be useful for interoperability purposes and yet covers all possible application areas. However, hard- and software limitations, which have been a driving factor in the research in Geographic Information Systems for a long time, are not posing many significant barriers anymore and therefore, research will be able to focus more on representational issues beyond efficiency, and, finally, some aspects of a spatial information theory. This contribution will discuss some examples and approaches to provide a theoretical foundation for the development of interoperability strategies.

A Specification Language for Interoperable GIS

(Extended Abstract)

Andrew U. Frank
Dept. of Geoinformation
Technical University Vienna

Werner Kuhn
Institute for Geoinformatics
University of Muenster

Specifications are essential for interoperability. They must be expressed in a formal language (Frank and Kuhn 1995). This formal language description needs to be tied into the compliance testing process. This paper shows that interface specifications can be written in an executable formal language with clearly defined semantics. It presents the semantic foundations of such a specification approach. A companion paper discusses the implications for compliance testing.

1. Introduction

To achieve interoperability, precise specifications of interfaces are necessary. Elementary data types must be exactly described to allow for higher-level types and operations must be specified with their inputs and outputs. Selecting a language to write specifications with is one of the most crucial methodological problems.

2. Current State-of-the-Art

Natural language specifications are not appropriate as they require human interpretation and may thus lead to different understandings. The consequence would likely be different implementations of components which cannot cooperate. In current practice, most specifications formally describe structures for data types and the signatures of operations. They use some kind of formal syntax from the theory of formal languages (regular grammar, parsing, signatures from algebra). The interpretation of the results of operations, their semantics, is stated - faute de mieux - in natural language.

Logic-based formalisms have been explored as a means to specify semantics. However, they are generally difficult to understand and check for correctness. Also, it is difficult to express change in logic by pre- and post-conditions (Hoare 1969; Wirth 1976; Floyd 1985)

3. A Functional Language as Specification Tool

Functional languages support the expression of an operation (acting on something and yielding a result) as a mathematical function. This idea is very close to the needs of interface specifications. It allows for a functional view of the services specified, without involving procedural notions.

Purely functional languages avoid side-effects (something else being operated on at the same time) and make sure that the values of variables, once assigned, never change. This results in referential transparency (the same name always refers to the same thing) and allows for a "mathematical reading" of the code (like a collection of equations). Strictly typed functional languages come with a type theory (Milner 1978; Cardelli and Wegner 1985) which restricts the interpretation of formulae to properly typed models. This avoids some of the fundamental logical antinomies and other puzzles of mathematics.

Class-based functional languages permit an object-oriented style of coding. The best known examples are Haskell (Hudak, Peyton-Jones et al. 1992; Peterson, Hammond et al. 1996; Peterson, Hammond et al. 1997) and notably Gofer which allows classes with multiple parameters (Jones 1991; Jones 1994; Jones 1995). The underlying theory, however, is substantially simpler than that of standard object-oriented languages (Abadi and Cardelli 1996). For example, it draws a clean distinction between behavior inheritance and implementation inheritance.

4. An Example for a Specification

Running example in the full paper.

5. The Semantic Foundations of Specifications in Gofer

When selecting a language to write specifications, one must carefully assess

- its expressive power;
- its semantics and how well it is defined.

Any ambiguity in the semantics of the language will be automatically carried forward as an ambiguity in the specifications (just like the ambiguity of natural language specifications). We have found Gofer to be the language with the least semantic ambiguities in its typing system and execution procedure (Jones 1995).

Functional languages are computationally complete: any recursively computable function can be expressed. Pure functional languages like Haskell or Gofer have minimal ambiguity in their foundations. These are directly related to "denotational semantics" ((Scott 1977), more readable in (Stoy 1977)). Evaluation is done by "substitution", the expression on the left-hand side being replaced with the expression on the right-hand side of an equation.

Only the Boolean type with its operations (and, or, not) is built into these languages. It can be fully and easily tested for correctness. The finiteness of computers limits the representable integers to a finite subset. For simplicity, the languages typically use the integers and floating point numbers of the underlying hardware. This means effectively that the ANSI specifications for numbers and their implementation are imported into any specifications written with a language. Neither the finiteness of representations nor the use of (standardized) floating point approximations are a problem when writing interface specifications; they are a problem of discrete computational modeling (particularly of geometry) as a whole.

6. Executable Specifications

Gofer is a functional *programming* language and must therefore restrict expressions to executable ones. Mathematically, this means the language must be constructive: stating the existence of something without providing a method to construct it is not possible. This coerces the formal statements of (spatial) theories into computational tractability and thereby helps to make specifications implementable. At the same time, it provides a prototyping mechanism to the specifiers - something that is very badly missed in practice today.

Gofer separates the construction of a theory from that of a model for this theory. *Classes* define theories. *Data types* are the actual carriers of data from which *instances* build models for the theories. Thus, models are example implementations and can be used for prototyping.

Theories are restricted to constructive axioms. Properties which cannot be explained as constructive axioms, can be stated as testable theorems. Constructive models can be written and tested.

7. The Type Theory of Gofer

Object-oriented design is motivated by the dominance of objects in human thinking and mathematically based on universal (or multi-sorted or heterogeneous) algebra (Birkhoff 1945; MacLane and Birkhoff 1967; Birkhoff and Lipson 1970). An algebra consists of an (abstract) carrier sort (often called type), a set of operations with their signatures, and axioms defining the behavior of the operations.

Classes in Gofer are algebras and instances define particular models for these algebras. Multiple instantiations of a class can exist in the same program. For a particular data type, the behavior of multiple classes can apply. This is a clean and sound mechanism for multiple inheritance.

8. Limitations

From a programmer's point of view, it is a drawback of Gofer that its standard, freely available implementation does not offer user interface building tools. The language also has no mechanism for persistence. Both these limitations are not an issue in specification writing.

The use of monads allows for a solution to both (King and Wadler 1993; Peyton Jones and Wadler 1993) and several very powerful methods for user interface specification exist under Unix.

9. Scalability

Gofer (in its current implementation) has some space limitations. However, the limitations of human thinking appear to be more serious: we simply cannot comprehend the interaction of several pages of specification. A design of a database for storing and accessing Gofer code has been completed at TU Vienna and is currently being tested.

Our experience shows that "clean" parts of a specification can be combined without restrictions on depth (i.e. layer can be put onto layer). Any "unclean" part will surface several layers higher up and confuse issues profoundly. This is most likely not linked to language specifics, but to a general problem of specification writing; a sharp tool like Gofer only makes it painfully visible.

10. Availability

Gofer is a public domain software and implementations for UNIX, Macintosh, DOS and Windows are available over the Internet.

11. Conclusions

The use of a purely functional language with a class-based type theory is highly recommended for writing specifications of OpenGIS interfaces. The semantics which come with such a language are mathematically sound and provide a good foundation for building specifications. The constructive nature and executability of the code ensures "no-nonsense" specifications and offers prototyping capability.

12. References

Abadi, M. and L. Cardelli (1996). *A Theory of Objects*. New York, Springer-Verlag.

Birkhoff, G. (1945). *Universal Algebra*. First Canadian Math. Congress, Toronto University Press.

Birkhoff, G. and J. D. Lipson (1970). "Heterogeneous Algebras." *Journal of Combinatorial Theory* **8**: 115-133.

Cardelli, L. and P. Wegner (1985). "On Understanding Types, Data Abstraction, and Polymorphism." *ACM Computing Surveys* **17**(4): 471 - 522.

Floyd, C. (1985). On the Relevance of Formal Methods to Software Development.

Formal Methods and Software Development. C. Floyd, H. Ehrig, M. Nivat, J. Thatcher (Eds.). Springer-Verlag, Lecture Notes in Computer Science**:** 1-11.

Floyd, R. W. (1979). *The Paradigms of Programming*. ACM Communications 22 (8), August 1979.

Frank, A.U. and W. Kuhn (1995). Specifying Open GIS with Functional Languages *Advances in Spatial Databases (SSD'95),* ed. Max J. Egenhofer, and John R. Herring. Springer-Verlag, Lecture Notes in Computer Science Vol. 951: 184-195.

Hoare, C. A. R. (1969). "An Axiomatic Basis for Computer Programming." *ACM Communications* **12**(10): 576-580.

Hudak, P., S. L. Peyton Jones, et al. (1992). "Report on the functional programming language Haskell, Version 1.2." *SIGPLAN Notices* **27**.

Jones, M. P. (1991). An Introduction to Gofer, Yale University.

Jones, M. P. (1994). The Implementation of the Gofer Functional Programming System, Yale University.

Jones, M. P. (1995). Functional Programming with Overloading and Higher-Order Polymorphism. *Advanced Functional Programming*. J. Jeuring and E. Mejer. Berlin, Springer. **925:** 331.

King, D. J. and P. Wadler (1993). Combining Monads. *Functional Programming - Workshop Glasgow 1992*. S. Ayr, J. Launchbury and P. M. Sansom. Berlin, Springer Verlag.

MacLane, S. and G. Birkhoff (1967). *Algebra*. New York, Macmillan.

Milner, R. (1978). A Theory of Type Polymorphism in Programming. *Journal of Computer and System Sciences* **17**: 348-375.

Peterson, J., K. Hammond, et al. (1996). Report on the functional programming language Haskell, Version 1.3. *http://haskell.cs.yale.edu/haskell-report/haskell-report.html*

Peterson, J., K. Hammond, et al. (1997). "The Haskell 1.4 Report." *http://haskell.org/report/index.html*.

Peyton Jones, S. L. and P. Wadler (1993). *Imperative functional programming*. ACM Symposium on Principles of Programming Languaes (POPL), Charleston, ACM.

Scott, D. S. (1977). *Logic and Programming Languages*. ACM Communications 20 (9), September 1977: 634 - 641.

Stoy, J. (1977). *Denotational Semantics*. Cambridge MA, MIT Press.

Wirth, N. (1976). *Algorithms + Data Structures = Programs*. Prentice Hall, Englewood Cliffs, N.J.

Real-world Lessons in Organizational and Technological Interoperability for Geographic Information Infrastructures

John Evans, Ph.D
MIT Dept. of Urban Studies and Planning

What does it take to share geographic information? It's unusual for planners or public managers to share information with each other across organizational boundaries. It's rare despite the high cost of geographic information and its potential for widespread re-use; despite the advent of ubiquitous data networks; and despite ecological relationships across jurisdictions, industries, and hierarchies. One difficulty seems to be a lack of technological and organizational infrastructures to help people make use of each other's information resources in their work. Building such infrastructures presents a number of organizational and technological interoperability challenges, as I found by (i) conducting an organizational case study among natural-resource agencies in the U.S. and (ii) building a networked orthophoto data service on the National Spatial Data Infrastructure. These two research thrusts together provide insights that are both grounded in real-world contexts, and sensitive to coming technological changes, to help public agencies articulate strategic choices of technology, organizations, and policy that will reap the promise of interoperability for geographic information sharing and improved ecosystem management. Further details on both of these research programs, and their synthesis, may be found in my doctoral dissertation (May 1997), available on the Web at http://mit.edu/jdevans/thesis.html.

First, the case study compared the technological and organizational characteristics, impacts, and growth patterns of three regional inter-agency infrastructures: the Great Lakes Information Network, the Gulf of Maine Environmental Data and Information Management System, and the Pacific Northwest StreamNet and its predecessors. The three cases highlight the importance, in launching geographic information infrastructures, of a convergence between shared norms, resources, and people to articulate these norms and leverage the resources. Once launched, the cases show, inter-agency infrastructures geographic information risk getting stuck at an experimental, "scaffolding" stage of development, with few tangible impacts on planning and policy. At these and other choice points, they need someone to integrate participants' many views of information sharing, so as to grow the organizational and technological complexity needed to affect real decisions, and to sustain this complexity over the long term. Indeed, in these cases the choice of a growth path and the unfolding of other decisions over time were generally more important than a priori blueprints or success factors. Nonetheless, a "laissez-faire" approach was inadequate: some evolving standard (a geographic reference system, or functional standards such as metadata or queries) was important to build convergence among participants. Finally, the three cases suggest that deeper-than-expected organizational changes are needed to capitalize on an online data services model of information sharing, in which data management and communications are merged, and interdependence and teamwork govern a complex "ecosystem" of government agencies.

The second thrust of the research instantiated the data services model for geographic information sharing. We used simple, freely-available software components to build a networked orthophoto browser that provides an efficient, interactive, multi-resolution service for widely-useful geographic information, in a way that easily integrates with client-side mapping software. This service, accessible on the Web at http://ortho.mit.edu, has opened the use of previously quite arcane orthophoto data to a much wider audience, and encourages convergence of geographic data among different sources. It also suggests an expanded conceptualization of the National Spatial Data Infrastructure (NSDI) as a collection of networked services and not just static datasets, and foretells a need to shift the standards focus from static data formats and structures to newer functional standards that will govern the interactions between information systems. Building the browser, and "spin-off" browsers for other orthophoto series, also highlighted the many design choices implied by a loose coupling between clients and servers, and the ephemeral nature of these choices given ongoing technological changes. In particular, this experience suggests that designers of interoperable geographic data services will often face the following challenges: providing useful features vs. reaching a wide audience; building for a widely diverse set of users and uses; tuning the service for current server hardware, networking bandwidths, and client software; and adapting these choices to a changing context.* The orthophoto browser also provides a tangible view of the organizational changes implied for the three cases and other similar contexts, as agencies using each other's data services share responsibilities for information collection and management.

Together, these two sets of findings show that in building geographic information infrastructures, a standard is not just a rule restricting the kinds of data, interfaces, or languages that the infrastructure will support, but also a resource that enables certain kinds of joint work through information sharing. Thus, particularly in the geographical arena, choosing what parts of the infrastructure "stack" to standardize, and what parts can remain heterogeneous, is a complex, dynamic decision that defines the constituency of the information sharing mechanism (who can take part in it) vs. its performance (what they can do with it). These findings also show that new organizational structures and relationships ("shadow organizations") are both a requirement and a consequence of effective geographic infrastructures, and that agencies face the dual challenge of redistributing information responsibilities and balancing incremental vs. radical change.

---

* In particular, J. Ferreira's I-20 position paper presents several emerging interoperability shifts that represent near-future directions for the orthophoto browser.

Planning in Spatial Internet Marketplaces

Volker Gaede
CSIRO Mathematical and Information Sciences
GPO Box 664
Canberra, ACT 2601, Australia

Notes: slides from this conference presentation can be found at http://www.wiwi.hu-berlin.de/~gaede/planning.slides.ps.gz

Recently, the idea of Internet Marketplaces has been proposed as a new form of making data and computational services available to a broad public. Like in a traditional marketplace these services are made available by providers and can be purchased by customers. In contrast to traditional marketplaces, however, customers do not receive a personal copy but pay for the use of the service. The Internet is hereby used as a communication medium for both requesting and delivering the service. This is a very important difference to on-line shopping over the Internet or electronic commerce, where the Internet is typically only used to initiate a service request and the actually delivery is by, for example, surface mail.

Since its first proposal, the idea of Internet Marketplaces has been adapted by various application domains including decision support systems, mathematical software and spatial information systems. Compared to other Internet market proposals, Spatial Internet Marketplace seem have a greater market potential since they encompass a large number of communities sharing a common interest in spatially referenced data and the associated computational services. With respect to concrete implementation of such Spatial Internet Marketplaces numerous problems have to be resolved and the SMART project currently conducted at CSIRO Mathematical and Information Sciences is a first step this direction. Within the scope of the SMART project an object-oriented design model along with a list of required actions of a Spatial Internet Marketplace has been proposed. One of the major differences of the SMART proposal in comparison with other proposals is that Spatial Internet Marketplaces should cater for some support of planning. That is, designated services accept a declarative request specification and generate an executable plan which materializes the request by combining distributed data and computational services. In other words, the system determines automatically from which site to acquire the data and where to perform certain operations. This planning procedure involves identification of suitable services, generating the corresponding requests for each of them and combining the generated results.

The motivation behind having such planning services in an Internet Marketplace is similar to database systems, where users ideally should be able submit requests without knowing any internal details and the query optimizer has to generate by means of some information an executable program. At first glance, one is tempted to think that planning in a Spatial Internet Marketplace is no different to query optimization in federated database systems, but as a closer investigations shows this is not true. In the paper, we provide a detailed comparison how planning in Internet Marketplaces differs to planning in federated database systems and discuss planning with respect to the unique requirements imposed by Spatial Internet Marketplaces. For example, potential customers in a Spatial Internet Marketplace are expected to have no or only a limited knowledge of the services available and how to access them. Instead of referring to particular services, Spatial Internet Marketplace customers simply describe what kind of services they are interested. For example, a customer should be able to express requests as the following one: Intersect the Australian crop map from 1996 with the Australian soil map. Because requests like this are expected to be the rule rather than the exception, the planning paradigm of federated database systems needs to be extended by a number of additional steps such a resource discovery step which aims at identifying providers and services (ie., resources) useful for answering a given request. In summary, Spatial Internet Marketplaces planning consists of the following steps:

- Resource discovery and classification step. The planning service accepts a customer's request and by means of the information stored in the registry or made available by the services performs a first selection of qualifying services, ie., identifies classes of equivalent services.
- Estimation and integration step. In a next step the planner generates for each of these equivalence classes requests and contacts the various services within each class to acquire further information (eg., cost estimates, precision). If two service classes are incompatible, the planning services tries to identify a transformation service to perform the corresponding transformation, ie., conducts another resource discovery step. Clearly, the cost induced by such a transformation service have to he added to the overall cost.
- Selection step. On the basis of the returned information, the planning services rank eligible services and generates a set of alternative plans.

However, in order to generate such a plan, the planning service must have some information about the existing services and the functions provided by them. The availability and the quality of this information is the key to successful planning, which in turn requires the dynamic solution of various issues to guarantee the interoperability of the services involved in the plan execution. It is not immediately clear what kind of information is necessary for planning and how this information should be made accessible. In a constantly changing environment such as the Internet, it is to be expected that services and their interface definitions also change. In contrast to distributed spatial applications, however, providers in a Spatial Internet Marketplace are less likely to inform planning services about changes. Furthermore, once generated plans might be come invalid because of the unavailability of a certain service at run-time or the change of an interface. In such cases it is important to perform dynamic service substitution by sending the request to an equivalent service. Basically, there are two possible ways to approach the above outlined problems. In the paper we present and compare these approaches with respect to their suitability and their impact on the overall design of Spatial Internet Marketplaces.

Probing the Concept of Information Communities - A Road Towards Semantic Interoperability

Y. A. Bishr[1)], H. Pundt[2)] , W. Kuhn[2)], M. Molenaar[1)], M. Radwan[1)]

1)International Institute for Aerospace Survey and Earth Sciences (ITC), Enschede, The Netherlands

2)University of Muenster, Germany
Institute for Geoinformatics (IFGI)

Summary

The information communities model is an important part within the framework of the Open GIS interoperability specification (OGIS). The paper discusses the idea of information communities, raising issues like:

- the assumption, that semantic mapping problems can occur both within and accross information communities boundaries
- the identification of problems of semantic interoperability
- the research on semantic translators and the development of prototypes to support semantics mapping between information communities

Extended Abstract

The complexity of geodata is one main factor which makes interoperability between different data sources and software systems so difficult to achieve. The complexity is caused by various factors, such as the underlying digital formats imposed by a particular software application or acquisition method and the complexity of higher level descriptions, conventions, and rules imposed by individuals, organizations, and disciplines using the software (Buehler & McKee 1996). The notion of interoperability has different meanings depending on whether it is used by network designers, operating systems designers, or application software engineers. As shown in figure 1, interoperability can be viewed at six different levels, where network protocols are at the lowest level and the information community interoperability is at the highest level. Today's desktop and enterprise environments lack interoperability between geographic information systems. The lower four levels of interoperability provide a distributed computing platform where interoperable GIS can be built on. By GIS interoperability we mean that users can transparently access and share remote geospatial databases and services regardless of their underlying GIS platform. Different applications have different world views, different representations, different schemas and hence different semantics.

Heterogenity at the information community level is a semantic problem, which is due to the differences in the interpretation of the spatial data encoded in the database. The problem of heterogenity at the application level has to be tackled, considering application interoperability and semantic interoperability as synonyms.

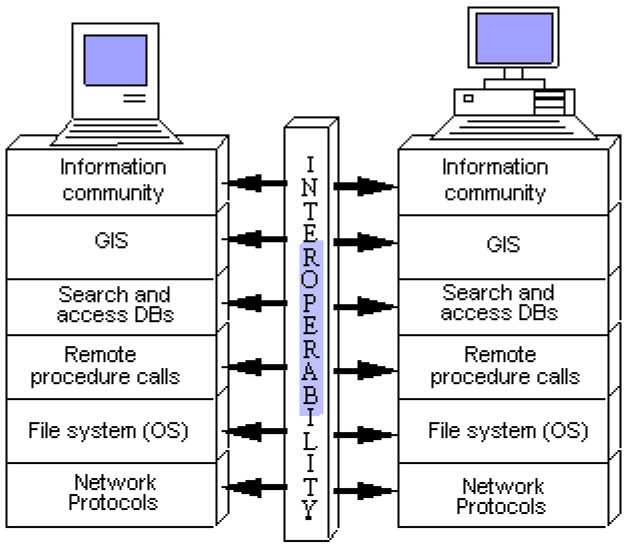


Figure 1: Levels of interoperability

In response to the need for interoperability the Open GIS Consortium (OGC) is working on the Open GIS interoperability specification (OGIS). The OGIS framework includes three parts (Buehler et al., 1996), the Open Geodata Model, OGIS Services Model, and the Information Communities Model.

The proposed paper will present a refined model of information communities and semantics mapping. The first chapter will describe the current view of information communities briefly. In addition to this view it will be shown, that problems of semantics mapping can occur both within and accross information communities boundaries. This view is schematically shown in figure 2. The assumption that semantic mapping is essential for data sharing between and within information communities requires case studies of specific applic ation areas.

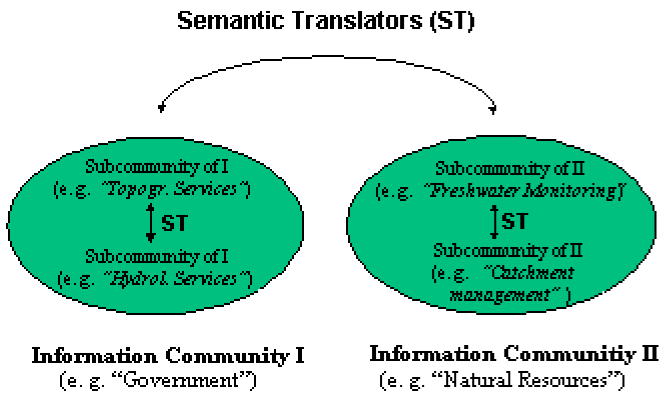


Figure 2: Application Interoperability between and within Information Communities and the role of Semantic Translators

In the second chapter we will present some examplary problems of geodata sharing which can occur in freshwater ecology, watershed management and transportation. These problems include the use of:

- same terms for different concepts
- different terms for the same concepts
- semantically similar attributes which have different meanings in their domains
- attributes which have different generalization and aggregation levels

  same attributes, but different data quality requirements, e. g. accuracy

A formal notion of mapping semantics between different data sources is essential for data sharing between information communities. It must include concepts to answer the open question, which role data quality plays within the framework of semantics mapping.

The next section will describe an approach how data sharing between different information communities can be realized without casualities in semantics. In the real world semantic interoperability is achieved by providing metadata about the related data set. Notwithstanding the ability of the metadata to provide an insight in related data sets, users are required to map the retrieved data from the domain of the provider to their own domain. The availability of semantic translators is required to support this task. Semantic translators are middleware components which allow heterogeneous applications to communicate and share data:

> A semantic translator is a middleware which can map among spatial database schemas while preserving their semantics. The concept of semantic translators will be outlined and a prototype will show, that semantic translators are an adequate means to achieve application interoperability.

Finally the aspects discussed in the paper lead to the definition of important issues of research on semantics interoperability. Some of these issues are:

- formal description of application semantics
- formal models of data quality requirements
- implementable forms of semantic mappings
- automatic detection of semantic similarities of attributes
- development of common ontologies for different application domains
- research on semantic translators for different geospatial applications

Interoperability Through Organization: The Role of Digital Libraries in Distributed Knowledge Management

Xavier R. Lopez, Ph.D.
School of Information Management and Systems (SIMS)
University of California, Berkeley

The rapid development of communication and computing technology is changing the way scientific information is created, disseminated, managed, and used. A new scientific information infrastructure is emerging, one that enables electronic peer to peer communication and unprecedented access to distributed information resources. Geographic information scientists are likely to be at the forefront of this new infrastructure with the development of globally integrated geospatial digital libraries. These geo libraries promise to boost scientific innovation, productivity, and returns on investment. They also pose immediate challenges to data and organizational interoperability. This paper examines the organizational dimension of interoperability for geographic information. In particular it examines how digital libraries can promote interoperability for geospatial information and other forms of distributed knowledge. Recent developments from operations research and organizational science are highlighted to illustrate how organizational innovations resolve interoperability challenges, create new opportunities for virtual product development and service delivery. This paper focuses on:

1. Interoperability challenges in geoprocessing
2. Organizational approaches to interoperability
3. Digital libraries as interoperable organization systems

1. Interoperability Challenges in Geoprocessing

The interoperability issues of the geographic information community are both technical and organizational in nature. As defined by Litwin (1990), and paraphrased by the UCGIS: "interoperability generally refers to a bottom-up integration of pre-existing systems and applications that were not intended to be integrated but are systematically combined to address problems that require multiple DBMS and application programs" (UCGIS 1996, p. 1). Effective communication and transfer of geographic information requires that organizations resolve interoperability of data models and components across organizational boundaries and applications. Organizations have evolved their own systems, legacy databases, and applications to serve internal needs. This has resulted in data models and applications uniquely tailored to meet specific internal requirements. However, as the importance of sharing information across organizational computing environments is recognized, data interoperability becomes paramount (ibid.).

The interoperability of geographic information across systems and platforms is also an organizational issue. Traditionally, government geospatial data suppliers have operated under centralized and hierarchical organizational structures to serve bounded communities of users with unique semantic and conceptual requirements (e.g. military, federal agencies, transportation agencies). This bureaucratic framework has resulted in closed, proprietary, and centralized geoprocessing services. Increasingly, however, there is an urgent need to access distributed information from many organizations to address boundary-spanning problems such as: disaster relief, environmental monitoring, interagency coordination, joint force deployment, and provision of integrated geospatial mapping services over the Internet. The need to access information resources across bureaucratic and hierarchical boundaries calls for new organizational processes that permit open network exchanges.

2. Organizational Approaches to Interoperability

Access to information across organizational boundaries is enabled by distributed computing. But distributed computing, alone, cannot support the complex supply chain of interactions that will be increasingly needed. Interoperability between organizations requires organizational planning. Interorganizational alliances and partnerships will establish de jur and/orde factos standards. The adoption of interoperable data models and modular software components, along open network standards and protocols, permit concurrent and autonomous development of applications (OGIS 1996). Process and component design can be enhanced by embedding coordination between distributed component activities and other activities required to fully develop a new product, eliminating the need for much, if not all, of the overt managerial coordination of development activities.

Geospatial data supply-chain integration, necessary for enhanced product development and service delivery, can be streamlined in an electronic environment, increasing organizational efficiency and effectiveness. Embedding coordination and transportable computation to remotely linked resources and competencies make supply-chain interactions efficient, effective, and scalable. In this manner, interoperability and distributed computing can even lead to new modular forms of organization that incorporate firm-specific and firm-addressable resources along a value-added chain. The challenge of exploiting new technological opportunities, therefore, lies with a complementary organizational structure to guide implementation. Moreover, the resulting structures need to be flexible enough to accommodate change. Digital libraries are emerging as an organizational form that are responsive to managing scientific information in a digital age.

3. Digital Libraries as Interoperable Organization Systems

Digital libraries provide a meaningful framework for integrating information resources and competencies from multiple organizations to deliver a synergistic service that is greater than its parts (Lopez 1997). They can play an instrumental role in overcoming current impediments to interoperability, by harmonizing the transfer of open geospatial data. The concept of "digital library," however, must be clarified before being used further. A digital library is defined as a coordinated set of heterogeneous actors/organizations which interact along an electronic and communication network to develop, add-value, disseminate, and archive electronic information and related specialist services. It is characterized by flexibility, decentralized planning and control, and lateral ties to other organizations. The chief structural characteristic of a digital library is a high degree of data integration across formal boundaries.

Organizational relations are crucial to a digital library. In a sense, the organizational challenge of maintaining peer to peer communication and relationships is analogous to the technical challenges of transferring geographic information across computing platforms. Open data models can reduce transaction costs, stimulate component generation, and provide a standard platform for new components and applications. Contractual arrangements and hierarchical rules also facilitate data interchange between the geo library and suppliers and the geo library and clients. However, static agreements and protocols alone, may not provide needed flexibility to respond to changing internal and external requirements and opportunities. Increasingly, human relations are necessary to (re)define common objectives and establish consistent work processes. Such relations are reinforced through interorganizational norms, consensus, and trust. Fortunately, digital libraries can effectively respond to both technical and organizational interoperability by coordinating resource integration across institutional boundaries and geographic space. In doing so, the network self-organizes, responding to client/user demands. It is also responsive to supply-side opportunities that can further enable service delivery.

REFERENCES

Buehler, Kurt and McKee, Lance. (editors) 1996. The Open GIS Guide, The OGIS Project Technical Committee, Open GIS Consortium, Inc.

Lopez, Xavier R. 1997. "The Network as Organization: Digital Libraries for Spatial Information." Proceedings of the First Assembly and Retreat of the University Consortium for Geographic Information Science (UCGIS) held in Bar Harbor, ME June 15-20.

UCGIS. 1996. Interoperability of Geographic Information, Research Initiative of the University Consortium for Geographic Information Science (UCGIS). URL: http://www.ucgis.org/

Accounting for the semantic differences between various Geographic Information Systems

Mark Gahegan
Geographic Information Science
Curtin University of Technology
PO BOX U 1987
Perth 6001, WESTERN AUSTRALIA

Geographic Information Systems (GIS) employ distinct conceptual models of geographic space (Goodchild, 1992), often as a reflection of the origins of the software (e.g. CAD and image processing). Some of these models are radically different, such as the images employed by Idrisi(compared to the object coverages used by Arc/Info(. Others are more subtly different, such as a topologically oriented coverage compared to the 'spaghetti' polygons used by many 'desktop' GIS. The meaning of spatial data is not the same within these models, and translation that is based solely on the geometry can lead to logical inconsistencies within the translated data. Whilst a good deal of very useful progress has been made by the likes of ISO TC211 and the Open Geodata Interoperability Specification (and related models), as yet these standards fall somewhat short in addressing the semantics of the underlying geographic models. In earlier work (Gahegan, 1996) a semantic notation was developed to describe the various transformations that occur as data is operated on or changed from one conceptual model to another. It is based on a data communication protocol described by Pascoe & Penny (1995) which has been extended to encompass certain key geographic properties and both a conceptual and physical data model. The notation describes a 'before' and 'after' state for a given transformation and is useful for communicating the likely effects of a specific transformation in terms of the data properties that may change as a consequence. In turn this can highlight any changes in the underlying conceptual model that occur and furthermore can show where assumptions regarding the meaning of the data are invalid or need to be made explicit. More recently, further additions have allowed the specification of uncertainty characteristics within the data (Gahegan & Ehlers, 1997).

This paper proposes some extensions to the notation to help describe the (sometimes subtle) differences between the data models used by different GIS and thus to aid in the interoperability process by providing a concise and symbolic description of geographic Perth data, specifying its semantic content as opposed to relying on the geometry to imply a meaning. This description, termed a 'transformation expression' can be equally applied to both datasets and operations. A dataset contains meaning which is imposed as a consequence of the conceptual model of the GIS under which it was gathered. This is represented by an expression of the form:

```
((abstract properties(, (geographic model(, (physical data structures(,
(system details(),
```

where:

`abstract properties` describe the data as the user perceives it (equivalent to an external view). `geographic model` describes the implications and limitations of the geographic model of space under which the data exists. `Physical data structures` describes how the data is physically encoded on the storage device, and is necessary since the choice of data structure can have an affect on other data properties. `system details` describes the actual package and platform that the data resides in. In practice, each of these components is further broken down into a number of distinct parts. additional components may also be added, to fully embrace interoperability standards such as the Open Systems Environment (OSE).

Transformations require expressions with both a left and right side and show the changes imposed on the data:

$$(before\_state) \xrightarrow{transformation} (after\_state)$$

where the states are described according to form given above. The after state contains a revised expression where any properties that have changed are flagged. Thus it is straightforward to build a taxonomy of transformation consequences in terms of the properties of the data that change. A useful high level grouping is:

- Transformations changing only the abstract data properties (no changes in the physical data structures or geographic model).
- Transformations causing the geographic model to change.
- Transformations causing the physical encoding of the data to change.
- Transformations moving the data to another system.

For example, using A, G, P, and S to represent the dataset properties respectively, a transformation which moves data from one system to another but using the same geographic model and data structures is given by:

$$(A, G, P, S) \longrightarrow (A, G, P, S')$$

When considering interoperability, the transformation will often be made up of several components: first moving the data to a new system, then operating on it, them possibly moving it back again:

$$(A, G, P, S) \xrightarrow{export} (A, G', P', S) \xrightarrow{import} (A, G'', P'', S') \xrightarrow{operation} (A', G'', P'', S') \ldots$$

The export transformation moves the data into the interoperability format from the host system, changing its physical structure and (possibly) its geographic model. From there it is imported into the internal format of the new system, again changing its physical structure and (possibly) its geographic model. Next, some operation is carried out (here shown as only affecting the abstract data properties) after which it may be passed back again to the original host. For simplicity, only the highest level properties are shown above, with the introduction of further properties, the transformation expressions become can quite specific in identifying exactly what has changed.

It is a relatively straightforward task to move from the symbolic description a set of automated rules and constraints that can determine if some interoperation is likely to cause difficulties; by comparing a semantic description of a chosen operation in one GIS with a description of a chosen dataset within another. Any semantic differences between the description of the dataset and the left side of the transformation expression indicate a potential conflict in meaning that may require resolution. Mismatches can be graded according to their severity, ranging from warnings to outright conflicts. In some cases, it may be possible to carry out any required conversion in an automated fashion; in others, some form of user intervention might be necessary. In either case, warnings can be issued and the mismatch documented.

The work is motivated by research into interoperability and data translation in regard to a new three dimensional geo-information system being developed by CSIRO (Australia) to support the needs of a wide range of geoscientists, including geologists. The aim is to make this system a semantically rich environment by ensuring that objects are ascribed meaning based on their modelling role, as opposed to their geometry. Interoperability issues are not restricted to the more 'standard' GIS, but also include many of the available geological and exploration packages such as Surpac(and Vulcan(. These provide a wealth of further spatial primitives beyond the standard points, lines, regions and surfaces; including volumes and profiles.

References

Gahegan, M. N. (1996), Specifying the transformations within and between geographic data models. Transactions in GIS, Vol. 1, No. 2, pp. 137-152.

Gahegan, M. N and Ehlers, M. (1997). A framework for the modelling of uncertainty in an integrated geographic information system. Proc. ISPRS International Workshop on Dynamic and Multi-Dimensional GIS, Hong Kong.

Goodchild, M. F. (1992), Geographical data modeling. Computers and Geosciences, Vol. 18, No. 4, pp. 401-408.

Pascoe, R. T. and Penny, J. P. (1995), Constructing Interfaces between (and within) Geographical Information Systems. International Journal of Geographical Information Systems, Vol. 9, No. 3, pp. 275-291.

Designing for Interoperability Overcoming Semantic Differences

Francis Harvey
EPFL-IGEO-SIRS
CH-1015 Lausanne

Interoperability can be understood in a number of ways. In a minimal sense, even the capability to transfer data from one computer system to another without transformation loss can be identified as interoperability. In a broader sense, interoperability can be taken to suggest the ability of different applications to interact dynamically, facilitating the smooth interface of multiple information sources. This paper examines interoperability in this second sense, specifically the role of semantics in facilitating the exchange of information.

Various technical solutions are known and are being worked on to organize the basic technical infrastructure for exchanging data. OpenGIS documents, for example, describe standardized data formats, protocols, and transfer mechanisms that in the near future will permit a technically smooth exchange of data. This level of interoperability will change the way people interact with geographic information technologies. This technical foundation is crucial, but wide-spread success will require further methods to preserve geographic and attribute meaning.

While, on one hand, issues connected to the technical transfer of data are resolved, the sharing of information still requires examination of the underlying semantic issues (Kuhn, 1994). The exchange of information and its inherent meaning, although implicated in the discussions of data exchange mechanisms, does not find the same lucid technical solutions, nor discussion. In the OpenGIS abstract specification the authors acknowledge that information exchange between "World Views" requires application specific knowledge beyond the existing technical description of interoperability and the Geographic Information Community (Open GIS Consortium, 1996).

Clearly, there is a need to develop mechanisms to facilitate the exchange of information between different Information Communities. Semantics plays a crucial role in linking the different conceptual worlds. The research I present here investigates an approach for considering semantic issues in the design of interoperable GIS systems. By articulating the semantics of each conceptual world in a rigorous manner, the technical prowess of interoperability can be harnessed to help resolve deeper data sharing issues. This approach relies on the construction of 'portals' between the different realms. In particular, this work builds on a concept coming from ethnographic studies of science and technology, called boundary objects.

Boundary objects, in many ways analogous to boundary markers in geographic space, are points of reference for multiple actors who have different disciplinary, institutional, and/or social perspectives on phenomena. A boundary object, for example the wetland layer in a county GIS, connects multiple perspectives through its arbitration of differences, but, at the same time, maintains distinctions between the actors. Taking the example of a theoretical wetland layer in a county GIS, an attribute COUNTY_WET may be used for areas accepted by the county administrator as wetlands following the Cowardin schema, another attribute FEMA_WET may delineate wetlands adjudicated by the Corps of Engineers, and the attribute RIP_WET could be used by ecologists to designate environmentally sensitive wetlands protected by the county's riparian habitat preservation ordinance.

In such a case, the different mandates and legal frameworks of these three agencies necessitate the differentiation of unique wetland inventories. Perhaps COUNTY_WET overrides the others for general county purposes, but cases may well arise when it becomes necessary to reconcile the different inventories. At present, the only methods we have for performing this activity is the cumbersome process of negotiation. Because of existing contentions and political wrangling, these processes frequently end up in "data wars" (King & Kraemer, 1993).

Although trenchant institutional differences can often require solutions outsides the realm of GIS, the costs and long-term damage they incur is ultimately to no one's advantage. Instead of getting caught up in long-term struggles, a design method for interoperability that nips these problems in the bud and creates a stable foundation for stable solutions is obviously preferable.

Clearly, this requires overcoming semantic differences. This can be done in multiple ways. I will describe a method under consideration at the Canton Vaud (Switzerland) that builds on the boundary object concept for purposes of design. Through an iterative process of focused group meetings to lay out differences and commonalities, each actor's semantics are laid bare and the foundation built up for constructing means of transforming different models. Called 'portals', 'gates', or 'paths' they connect semantic differences and commonalities providing means of transformation. This broadening of the discussions surrounding design are the basis for constructing more robust data transfer mechanisms that preserve the semantic integrity of the transferred information. This modeling approach is suitable for formalization and integration with existing technical frameworks, applying specifications and outlining behavioral concepts for interoperability. Chains, actions, threads, entities are behavioral concepts for the formalization of the requisite morphisms. Instead of wearisome negotiations, this approach opens ways of rigorously describing and formalizing interoperability operations, thereby enriching the exchange of geographic information.

Clearly, this approach is limited in this form to 'closed-settings.' These are administrative or corporate environments with a distinct number of actors and distinct purposes. 'Open distributed processing', such as the provision and exchange of global-change data for research, with highly variable uses, will require more advanced solutions that can build on this preliminary research.

In summary, the approach I describe here suggests a solution to enriching interoperability by bringing fundamental semantic concerns into the primary design process. Beyond the technical differences that interoperability has addressed so successfully, the broadening of considerations to include the semantics of interoperability is a crucial next step.

REFERENCES

King, J. L., & Kraemer, K. L. (1993). Models, facts, and the policy process: The political ecology of estimated truth. In M. Goodchild, B. O. Parks, & L. T. Stayaert (Eds.), Environmental Modeling with GIS (pp. 353-360). New York: Oxford University Press.

Kuhn, W. (1994). Defining semantics for spatial data transfers. In T. C. Waugh & R. G. Healey (Ed.), Sixth International Symposium on Spatial Data Handling (SDH '94), 2 (pp. 973-987). Edinburgh, Scotland, UK: IGU/AGI.

Open GIS Consortium (1996). The OpenGIS Abstract Specification: An Object Model for Interoperable Geoprocessing, Revision 1 (Project Document No. 96-015R1). Open GIS Consortium.

Development of a Global Conceptual Schema for Interoperable Geographic Information

May Yuan
Department of Geography
University of Oklahoma

1. Defining information interoperability in a GIS context

Interoperability enables sharing and exchange of information and processes in heterogeneous, autonomous, and distributed computing environments. The idea aims at a cost effective and user friendly means to maximize the usefulness of information computing resources across multiple platforms and institutions. It facilitates access to needed information resource that can be used independently of a computing environment. This is particularly important in the field of GIS since collection and editing of geospatial data often involves labor intensive and time-consuming tasks. To achieve information interoperation for applications and end users, a wide variety of approaches has been taken, including using distributed object technology (Paepcke *et al.* 1996), query languages (Gingras *et al.* 1997), interface standardization (Wegner 1996), and interface bridging (Clement *et al*. 1997). However, interoperability presents a much greater challenge in GIS than in other fields of information science because the greater complexity of geographic information adheres to ways that acquire, represent, and operate geospatial data

The complexity of geographic information and processing raises the fundamental issues related to the incompatibility of representations, structures, and semantics that need to be addressed to achieve geographic information interoperation. There are three aspects of information interoperation; each of which emphasizes resolving either syntactic, semantic, or software incompatibility. The syntactic approach enforces standards for encoding and interpreting geospatial information to allow one system capable of understanding the meaning of data from another system. Syntactic interoperability can be achieved by standardizing meta-data and meta-information regarding data formats and definitions to allow the data to be processed in remote environments (UCGIS 1997). From the syntactic perspective of common data descriptions, the long-term goal of research in interoperability is to develop automatic methods that extract and update essential meta-data and meta-information. A syntactic approach can ease data transformation among different systems but has limitations to overcome the barriers that result from semantic gaps between communities of different cultures and histories to share geospatial information because of the distinct variations in conceptualizations and interpretations of geographic worlds (Buehler and McKee 1996). Regional planners, farmers, and hydrologists possess unparalleled soil classification schemes, but an ideal soil database should allow data interoperable by different user groups. In addition to the syntactic and semantic propositions to resolve interoperation of geographic information, considerations are also taken to promote software interoperability which aims at developing hardware-independent modules and mobile codes executable at remote systems. Instead of exchanging data or information, this approach achieves interoperability by transmitting processes across heterogeneous distributed systems. It is appealing but limited to intranet applications for system security.

The position paper aims to stress the importance of semantics to the enhancement geographic information interoperability by developing a generic GIS conceptual model. The conceptual model is used to define common information elements (or ideas used to communicate the needed geospatial data) that underpin the sharing of common data among islands of software-specific GIS applications. It is not to dismiss the significance of syntactic and software interoperability but to focus on semantic compatibility that pertains to fundamental issues and profound implications in GIS representations and data modeling. Moreover, a generic conceptual information model can serve as a precursor to software interoperability (Singh and Weston 1996), and the idea of using contextual knowledge to achieve information interoperability has been suggested superior to using instructions about data structures or formats (Laplante 1996).

2. Use of a global conceptual schema to enhance geographic information interoperability

Date (1994) suggested three levels of information modeling: internal schema, external schema, and global schema. An internal schema refers to data structures and is usually software dependent, and an external schema relates to information needs for individual applications. In contrast, a global schema outlines concepts and attributes, which can be defined in a generic reference model based on conceptual views of geographic information. That is, the design of conceptual schema will encounter different ways in which humans perceive the world and communicate their perceptions. In doing so, it is important to make semantics explicit to retain common interpretation of the relationships among data items (Gingras *et al.* 1996). Thus, one of the primary barriers to achieving data interoperability attributes to the lack of such a common framework to signify the content and geospatial information. A global conceptual schema representing constructs of geographic information will be useful to overcome the barrier.

One of the primary concerns in the design of a global conceptual schema for interoperating geospatial information is to support data relativism, *i.e.* multiple perspectives on the same underlying data set, to enable information interoperable among users and systems. Semantic modeling is one way to achieve data relativism by encapsulating the structural aspects of data (such as data types, file structures, constraints, and relationships) to allow users to focus directly on abstract objects corresponding to concepts or things in their applications (Hull and King 1987). Likewise, the global conceptual schema needs to represent geospatial information in ways that users can refer to "geographic things" as of abstract concepts or real beings, including themes, states, locations, events, or processes. Thus, users can navigate through the schema by applying attributes directly to the thing of interest. In doing so, data exchange between two systems is carried out via geographic things rather than data records as in a relational database, for example. Unlike relational databases, this approach releases users or client systems from constraints of any pre-defined data and file structures that the user needs to learned before traversing from one relation to another. Instead, use of abstract concepts or real beings can ease communication for data exchange by modularizing the details of data structures to enable the user (or the client system) to access data at different levels of abstraction.

As a result, the primary objective of the global conceptual schema for interoperable geospatial information is to provide a coherent family of constructs representing abstract objects in a structural manner and to encapsulate the structure in these constructs. The global conceptual schema acts as a mediate system to which a system (or the user) poses requests for information, and the other system responds via abstract objects (Figure 1). Both systems will incorporate a global conceptual schema that is used for information communication and data exchange from one system to another. Data management and computation are performed according to data models specific to software application and computing environments. The global conceptual schema tags and structures information components for communication, while data in each system remain in their native forms.

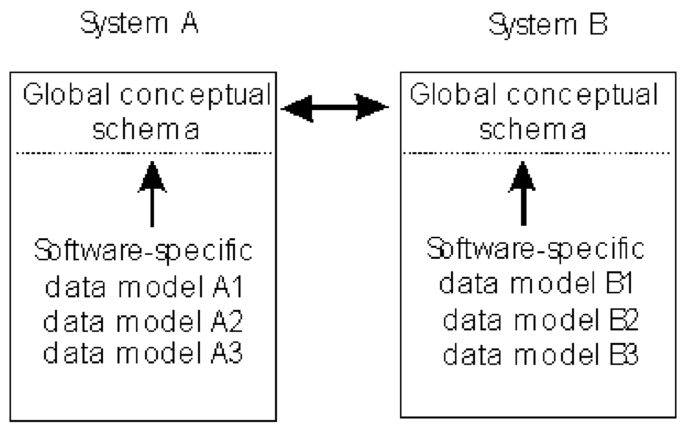


Figure 1: Use of a global conceptual schema to facilitate information communication among data models.

The emphasis of a global conceptual schema is to facilitate communicating information of meanings (rather than data structures) among systems and users. With certain modifications, many semantic data models can be applicable to the development of a global conceptual schema for geospatial information. Each semantic model is centered on attribute, relationship, or concepts (abstract objects) in its data organization. The ER model (Chen 1976) is perhaps the most used semantic model in GIS applications with data organization centered on relationships of attribute sets. Its applications have been tightly linked to relational databases. In contrast, the functional data model (FDM, Shipman 1981) is designed with emphases on attributes in that it connects data objects directly with attributes without the use of intermediate constructs such as a table to aggregate attributes of the same set. The third type of semantic data models stresses the importance of concepts, entities, events, states, and processes by representing them as individual constructs with encapsulated attributes, behaviors, and structures. In addition, relationships among composed attributes are represented explicitly as part of construct definitions so that this information can be accessed in a direct manner without searching for references (or keys). The model of conceptual graphs (Sowa 1984) is an example of the third type of semantic data models. Conceptual graphs provide logical frameworks that mimic mental models of human knowledge to represent abstract concepts, real beings and their relationships. Rooted in cognitive evidence of information processing, conceptual graphs provide mappings between extensional objects and intensional concepts and formalisms to describe relationships. Among the three primary approaches in semantic data modeling, the model of conceptual graphs is, perhaps, the one that can offer the most valuable foundation to the development of a global conceptual schema to enhance the interoperability of geospatial information. The following section, thus, describes a design that adopts the ideas from conceptual graphs to geospatial information modeling.

3. Design a global conceptual schema for interoperable geospatial information

A global conceptual schema of geospatial information should consist of three elementary information components (geographic semantics, space, and time) and support four primary user views of geography (states, entities, events, and processes). The state view suggests geography comprise static properties at locations, including a snapshot of a field or individual geographic features. From this perspective, geospatial information is recorded according to locations that have been identified and represented by spatial objects (points, lines, polygons, or raster cells). The entity view stresses geospatial information describing properties of a geographic entity as a unity, which may or may not have homogeneous, contiguous attributes in space or time. Hence, geographic entities of interest need to be determined prior to association of proper geospatial information. The event view defines space and time according to the incidence of one or multiple events to relate geospatial information before, after, or during the events. More often than not, geographic attributes triggering or influenced by the events are emphasized in an event-based analysis rather than the attributes of the events themselves. On the other hand, the process view interprets space as evolving composites of geographic attributes through time. Recognition of the four primary user views and three elementary information components is the first step in developing a global conceptual schema to facilitate geographic information interoperability.

Subsequently, it is necessary to structure a global conceptual schema that corresponds to the four user views based upon the three geospatial information components. The idea aims at utilizing the global conceptual schema as a data translator among databases and between users and databases. With the global conceptual schema, communication among users and databases can be performed by inquiring the content and semantic structures of geospatial information instead of low-level data formats or data structures. A global conceptual schema can be constructed by three domains of geographic information; each of the domains maps to geographic semantics, space, and time. The domain of geographic semantics consists of information about geographic attributes, entities, and events that correspond to abstract concepts or physical objects in the real world. The spatial domain constitutes spatial objects in one-, two-, and three-dimensional geometry and coordinates, and the temporal domain is composed of temporal objects as points (instants) or lines (intervals). There are links among the three domains to present the four views of geographic information. Communication among geospatial databases is, therefore, to resolve tags for geographic semantics, spatial extents, and temporal ranges from the source database (or users) and then to rebuild them in the target database (or users). That is, the source and the target databases need little or no knowledge about the data structures embedded in their counterpart. Geospatial data are interoperable because it is not data but geospatial information about geographic semantics, space, and time being transmitted. Necessary procedures are later performed to transform the transmitted information to embedded data structures for a particular database. Use of the three information domains, the four primary user views (states, entities, events, and processes) can be supported in a geospatial data set through ordered links of semantic, spatial, and temporal objects. Examples are given in Figure 2.

Communication through information appears to provide a more effective means than through data with software or hardware dependent formats (Laplante 1996). The global conceptual schema provides ways to share geospatial data among user views, not to specify data structures or data formats but to identify information components from the source database and link them in ways appropriate to the target database. For example, a wildfire data set can be used for fire spread simulation by associating a fire (geographic semantics) and locations (of burns) to time (of burns). It can also be used for fire history modeling by associating locations (of burns) and time (of burns) to geographic semantics (fire burns). The global conceptual schema needs only to parse and tag the three elements of geospatial information, and it is up to the user or database to restructure them in the way that suits their purposes.

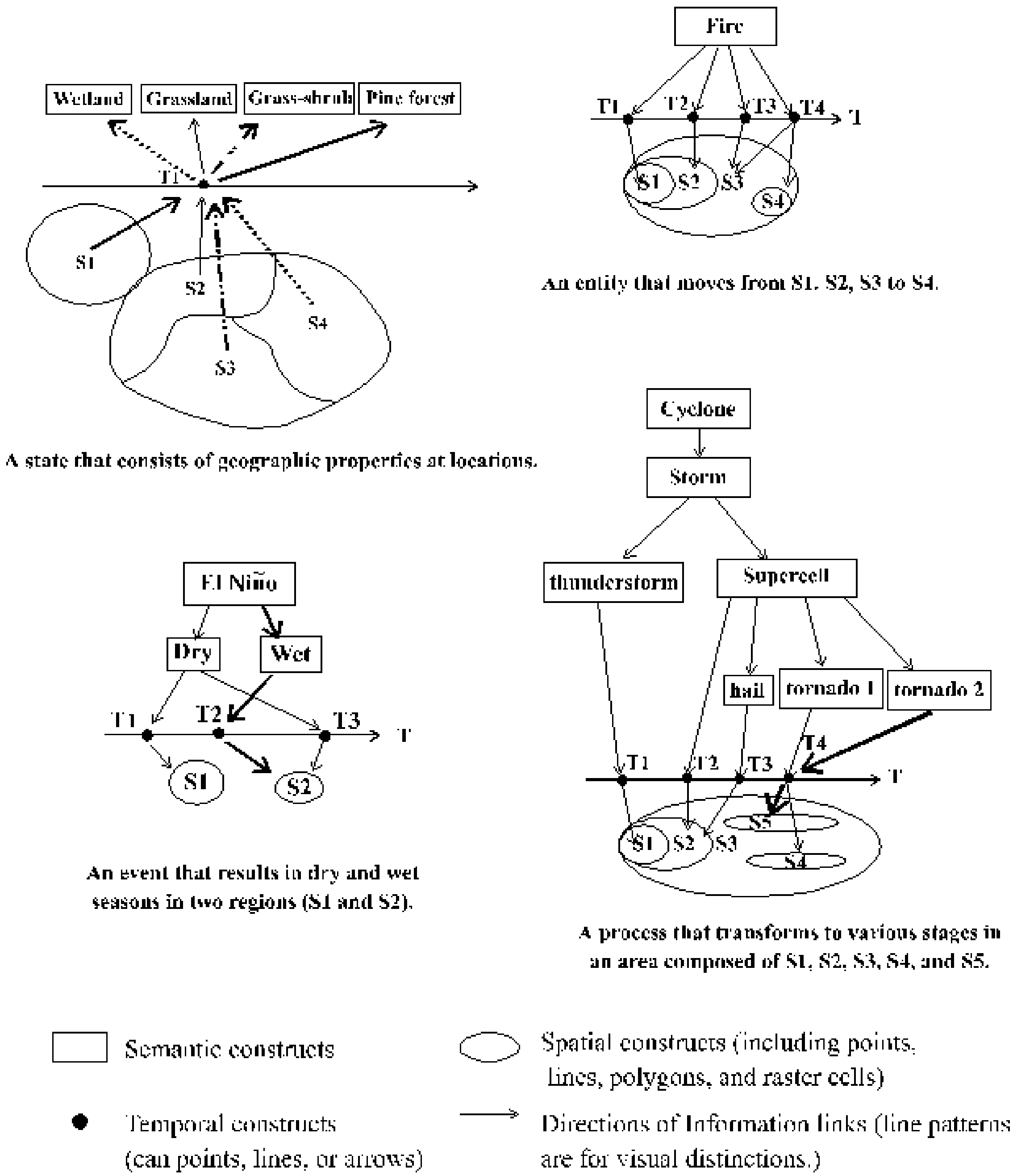


Figure 2: Examples of using semantic, spatial, and temporal constructs to represent information interpretations from four primary geographic information perspectives.

4. Concluding remarks and research directions

Recent development in interoperability has provided implications to geospatial information interoperation, but fundamental issues of interoperability in GIS cannot be fully addressed without a thorough understanding of the essence of geospatial information. Syntactic and software interoperability alone may be inadequate due to the complexity and diversity of geospatial information sources and interpretations. This position paper propounds the idea of using a global conceptual schema to achieve semantic interoperability of geospatial information.

The basic idea stems from the argument that external and internal schemata in database modeling are either software or application dependent, but a global schema can provide a common conceptual framework to support information interoperability. The proposed framework to provide such a global conceptual schema is rooted at three domains of geographic information in semantics, space, and time. The framework applies information constructs from the three domains to structure geospatial representations from four basic views of geographic states, entities, events and processes.

There are other efforts that appear to follow ideas similar to the global conceptual schema, especially the Spatial Data Transfer Standard (SDTS) by the Federal Geographic Data Committee (FGDC, 1991) and the Open Geodata Model proposed by OpenGIS™ Consortium (Buehler and McKee 1996). However, most approaches attempt to provide rigid definitions for geospatial objects or ways to encode attributes. Consequently, they tend to result in volumetric specifications and definitions or another strict data model. On the other hand, the idea of a global conceptual schema fosters a flexible but robust framework to incorporate user views by communicating through geospatial information instead of transmitting data. Unarguably, the efforts on standardization of data and meta-data as well as formalization of geodata models are very important to the eventual realization of interoperable geospatial information. Alternatively, the global conceptual schema for geospatial information communication provides an effective approach to information interoperation from a perspective of higher abstraction. This extended abstract only outlines the key ideas in the design of the global conceptual schema. Detailed discussions and case studies are to be elaborated in a follow-up essay.

References:

Buehler, K. and McKee, L. ed., 1996. *The Open GIS™ Guide: Introduction to Interoperable Geoprocessing*. OpenGIS Consortium, Inc. (Wayland, Massachusetts).

Chen, P. P., 1976. The entity-relationship model: towards a unified view of data. *ACM Transactions in Database Systems*. 1 (1): 9-36.

Clement, G. Larouche, C., Gouin, D., Morin, P. and Kucera, H., 1997. OGDI: Toward interoperability among geospatial databases. *SIGMOD Record*, 26(3): 18-23.

Date, C. J., 1994. *An Introduction to Database Systems*, Chapter 2. 6th edition. Addison-Wesley (Reading, MA).

FGDC, 1991. *Spatial Data Transfer Standard*. Wachington, DC: Department of the Interior.

Gingras, F. Lakshmanan, L. V. S., Subramanian, I. N., Papoulis, and D., Shiri, N., 1997. Languages for multi-database interoperability. *SIGMOD 97*, Pp. 536-538.

Laplante, M. 1996. Information Interoperability. *Inform*. Pp. 16-18.

Hull, R. and King, R., 1987. Semantic database modeling: survey, applications, and research issues, ACM Computing Surveys, 19(3): 201-260.

Shipman, *D. W. The Functional Data Model and the Language DAPLEX*. ACM TODS, 6(1). ACM.

Singh, V. and Weston, R. H., 1996. Information models: a precursor to software interoperability. *Production Planning & Control*, 7(3): 242-257.

Sowa, J., 1984. Conceptual Structures: Information Processing in Mind and Machines. Addison-Wesley Publishing Inc. Reading, MA.

Paepcke, A., Cousins, S.B., Garcia-Molina, H., Hassan, S. W., Ketchpel, S. P., Roscheisen, M., Winograd, T., 1996. Using Distributed objects for Digital Library interoperability. (includes related article on Stanford's Digital Library)(Digital Library Initiative). *Computer*, 29(5): 61-68.

Wegner, P., 1996. Interoperability. ACM Computing Surveys, 28(1): 285-287.

The Need for a Formal GIS Transportation Model

Stephen J. Beskpalko
Sandia National Laboratories
PO Box 5800 MS 0977
Albuquerque NM 87185-0977

Max M. Wyman, Ph.D.
Terra Genesis, Earth Modeling & Systems Restructuring Group 1268 East McNair Drive
Tempe AZ 85283

John C. Sutton, Ph.D.
GIS/Trans, Ltd.
2801 Business Center Drive, Suite 145
Irvine, CA 92715

ACKNOWLEDGMENT AND DISCLAIMER

This work was partially supported by the United States Department of Energy and was performed at Sandia National Laboratories under Contract DE-AC04-94AL85000.

The opinions expressed in this document are those of the authors, and do not necessarily reflect the opinions or positions of their employers, other individuals, or other organizations. Mention of commercial products does not constitute endorsement by any person or organization.

POSITION

GIS for Transportation (GIS-T) is an interesting subset of the overall suite of technologies referred to as GIS. The requirements of GIS-T are intermediate between the purely cartographic applications and the more demanding topographic requirements of civil engineers. None-the-less, the GIS-T is an important component of the GIS community, and one where change is needed (Vander Veer, 1997).

Proposed herein is a new direction in GIS for Transportation (GIS-T) infrastructure modeling which is potentially free of the pathologies associated with current LRS data models. A model is proposed which builds towards a 3-D GIS based on location references provided by GPS and its World Geodetic Spheroid 1984 (WGS84).

Current GIS-T is built up from three layers:

- A fundamental arc-node (point, line, and polygon) layer as derived from traditional cartographic systems (vector or raster). This model is frequently linked to tabular details through a relational database manager.
- A 1-D offset measurement technique known as a linear reference system (LRS).
- Dynamic segmentation as an enabling tool for assigning multiple attribute sets over a single linear event.

Our position is that this architecture is not sufficient to accurately or economically model transportation infrastructure, as evidenced by Sutton's (1995, 1996) ever growing list of pathological transportation segments. A pathology is, in short, a transportation element that can not be directly represented using current GIS technology. Given the number of pathologies that have now been identified, it is not productive to ask whether or not a particular technique (usually referred to as coding) embedded in the classical GIS can be imagined to circumvent a particular pathology. Rather, researchers should be exploring whether the classical arc-node-datum model is the best and most economical method for representing complex spatial (or transportation) constructs. We believe not. New technologies including the Global Positioning System (Brown, 1989, Kaplan, 1996), GPS, provides the means for solving these problems. What is missing is the information technology to integrate the GPS with a suitable transportation model incorporating three-dimensional data. In short, what is missing is a formal model of the transportation system, and a subsequent implementation of the model.

HISTORICAL PERSPECTIVE

Many of the current constraints found in GIS stem from decisions that made sense at the time they were made, but are no longer valid. The 2-D cartographic map was an acceptable representation when overpasses were rare and various modes of transportation were largely disjoint. During these earlier days, LRS was vital because few alternatives could record absolute field locations easily. A field accuracy of 0.5 km was acceptable because other than the distance measuring instrument (DMI) there was no alternative.

Dynamic segmentation was also the only alternative for representing more than one event along a linear feature. Because classical GIS-T techniques required the use of multiple local reference frames, data sharing between agencies was difficult or impossible. Problems multiplied as systems grew in complexity and temporary fixes were added to overcome the multiple datum requirement. Our current work leads us to conclude that even common national datum would not suffice to solve the problem.

Multiple datum LRS and color coding of complex infrastructure objects often result in misleading or incorrect computation, identified by Sutton (1996) as network pathologies. The incorrigible nature of these pathologies suggests the need for a radical change, rather than evolutionary changes to either traditional LRS data models or national datum (see Fletcher 1995, 1996, Vonderohe, 1995, 1996, Dueker and Butler, 1997).

The fundamental thesis of our position is that these convoluted historical constraints are no longer valid. There are clear indications that the layered LRS/multiple datum architecture is incapable of representing contemporary transportation features. Because GPS service is now ubiquitous, it is possible to build a fully three-dimensional and topologically correct model for transportation infrastructure.

The GPS provides the common origin needed to make data interchange fast, easy, and error-free, just as a geometrically correct 3-D GIS-T model would eliminate the pathological errors bound within the limits of LRS. Mutually incompatible datum and complex dynamic segmentation coding techniques would be replaced by a common language, and the gap between precision drafting methodologies and connectivity-bound topologies like GIS could be closed forever. The potential of 3-D GIS-T data storage should be viewed as profound and ready to meet the needs of local, state and federal agencies forced to accomplish more with less.

PROPOSED THREE DIMENSIONAL GIS MODEL

We propose to outline the basic elements of a new model based on a singular worldwide reference. Further, the model we propose must be lane-based for transportation, and likely something other than object-oriented (at least in terms of the current state of object technology). The implementation we propose would be based on a new form of database, which is constructed from loosely coupled schemas, as opposed to the traditional monolithic database architecture. This work on loosely coupled schemas funded by DARPA and is currently under way at the University of Colorado.

RESEARCH TOPICS FOR THE FUTURE

There is both short-term and long-term work that must be undertaken to transform the GIS from the link-node based technology of current systems to the accurate fundamental technology needed for the future.

1. The urban canyon problem must be addressed. There are promising mathematical techniques that need to be developed to reduce the likelihood of inadequate or reflected signals in some parts of the country.
2. A formal lane-based GIS model must be developed to insure interoperability between other GIS technologies and GIS-T.
3. Fundamental distance and reference algorithms must be developed and tested

to demonstrate the appropriate accuracy is achieved with the new GPS technology.

4. Appropriate foundation work must be done to move the field away from the practice of incorporating ever more complicated patches on the link-node-datum models. The maximizing possibility of a single unified worldwide and highly accurate referencing system should be the goal of future spatial data research.
5. It must also be demonstrated that the three-dimensional lane-based and topologically correct model does indeed eliminate the possibility of pathological cases, as we suspect.

REFERENCES

1. Brown, A. Extended differential GPS. Navigation 36(3): 265-285, 1989.
2. Dueker and Butler. GIS-T Enterprise Data Model with Suggested Implementation Choices. The Center for Urban Studies, Portland State University, 1996.
3. Fletcher, et al, Geographic Information Systems - Transportation ISTEA Management Systems Server-Net Prototype Pooled Funds Study. FHWA, 1995.
4. Kaplan, E., et al. Understanding GPS, Principles and Applications. Artech House Publishers, 1996.
5. Fletcher, et al, The Case for a Unified Linear Reference System. Alliance For Transportation Research, 1996.
6. Sutton, J. Network Pathologies Phase 1 Report. Sandia National Laboratories, Project AH-2266, November 1995.
7. Sutton, J. Network Pathologies Phase 2 Report. Sandia National Laboratories, Project AH-2266, March 1996.
8. Vander Veer and Bespalko. GIS: Reaching the Third Dimension. Society of Women's Engineers, 1997 National Convention, 1997. Also available as Sandia National Laboratories Technical Report SAND97-1616a.
9. Vonerohe, A. Results of a Workshop on A Generic Model for Linear Referencing Systems, University of Wisconsin - Madison, 1995.
10. Vonderohe, A. A Methodology for Design of a Linear Referencing System for Surface Transportation. Sandia National Laboratories, Project AT-4567, 1996.

Real-time Data Exchange and Interoperability

Fred E. Latham
Viggen Corp.
Knoxville, TN

David Siegel
Viggen Corp.
Knoxville, TN

Demin Xiong
Oak Ridge National Laboratory
Oak Ridge, TN

Real-time data exchange is a concern of more than a little importance to ITS interoperability. With the rapid advances in computational technology and the technology used for data communications among distributed computational platforms, a computational framework for spatial data interoperability will become essential in ITS applications. Within this framework, real-time data exchange, along with other types of data sharing, process sharing and multiple services access will be possible. To advance such a framework, an interoperability standard will need to be developed from the foundation of a comprehensive ITS data transfer standard; a comprehensive standard that will provide specifications for spatial feature catalogues, spatial objects, attributes and other basic elements.

Many advanced ITS user services require real-time spatial data exchange and communications. Numerous real-time data can be formatted into messages that can be transmitted among distributed ITS application components. Other real-time data, due to their volume, must reside in an on-line database, which will be retrieved upon requested.

Performance, or speed, should not be a criterion of a transfer standard, but when a standard must be used in an environment where performance is critical, it is a factor must be examined carefully. Performance is a concern in this type of transfer standard because a standard is intended to provide flexible and meaningful data transfer. This type of transfer requires complex information engineering, which in return will impact the computational performance or transfer speed. Also a transfer standard is usually used to transfer a complete data set, even when a user needs only part of the data. The user must then extract the necessary data using other programs. Three tactics for addressing the issues are:

Classify those data likely to be transferred in real-time as a special category. When mapping this special category of data into a transfer, a performance measurement can be used to test choices and alternatives. Adopt special options to achieve an efficient design. These options can take advantages of the specifications of the standard, but will be tailored to fulfill the real-time requirements. Develop an independent profile to be compatible with the ITS spatial transfer standard for the purpose of real-time data exchange. This profile will build appropriate formats for the data so as to reduce transfer overhead and allow partial data transfer.

An interoperability standard must be distinguished from a transfer standard, and clarification made as to what a transfer standard can do and can not do with respect to interoperability. A transfer standard must limit the scope of the data to be transferred. By its nature, a transfer standard is designed to transfer large volumes of complete sets of data. In contrast, data exchange in the interoperability framework will take place at all levels of data organization. A transfer standard should focus on the specification of the data. Interoperability however, requires the specification on both the data and the processes that operate on the data. Notably, the process of transferring data through a transfer standard is uni-directional. In the interoperability framework, communication between the data server and the data receiver must be allowed.

Although an interoperability standard and a transfer standard are different, they are highly correlated, especially concerning data representation. It is possible and desirable that the specifications of data models, data structures and data format in a transfer standard will be directly utilized in an interoperability standard. In this respect, the development of a comprehensive ITS data transfer standard should consider the constraints imposed by interoperability requirements; otherwise the transfer standard may become an obstacle rather than a facilitator to spatial data interoperations implementation.

Interoperability Issues in Intelligent Transportation Systems: Testing the Cross Streets Profile

Val Noronha
Vehicle Intelligence Testing & Analysis Laboratory
NCGIA - University of California
Santa Barbara CA 93106-4060, USA

Intelligent Transportation Systems (ITS) is an emerging suite of technologies that promises to make highways safer and more efficient, by offering motorists tools to interact with other vehicles, and with ground-based information service providers. Some of the potential application areas are emergency assistance, real-time routing based on up-to-date freeway congestion measurements, traveller facility directories (e.g. motels and ATMs) and collision avoidance systems.

Vehicle and incident location are central data items in ITS. In a world of competing vendors of street network data - and inevitable discrepancies and errors in position, street naming, addressing and classification - the success of ITS hinges on the ability to communicate a location message unambiguously across dissimilar map bases. Currently this is not achievable within loosely specified industry-acceptable error tolerances. To be successful and acceptable to commercial navigation system vendors, a solution to this problem would need to minimize impact on vendor data bases and practices. For example, vendors should not be required to re-survey their national databases to more stringent standards; similarly, conflation to one vendor's database would imply that the other vendors' data were inferior.

Potential solutions to the location referencing and messaging problem are being proposed by Standards Development Organizations (SDOs) at the national and international levels. These solutions need to be field tested in simulated and real application conditions. Two current proposals under consideration are the Location Reference Messaging Specification (LRMS) and the ITS Datum (ITSD) from Oak Ridge National Laboratories.

The Vehicle Intelligence Testing & Analysis Laboratory (VITAL) at UC Santa Barbara is a testbed for interoperability issues related to ITS. Funded by the state and federal governments, we are pursuing a research agenda driven partly by the industrial needs of SDOs, and partly by academic questions. The initial task of the lab was to set up a testing infrastructure, consisting of a client system in a vehicle and a ground-based server. The client has a moving map display, differential GPS vehicle locator, dynamic routing and real-time wireless message exchange with a server. One of the first research tasks of the lab is to test the LRMS Cross-Streets Profile; this is the focus of this paper.

There are several ways to communicate a location on a network, e.g. coordinates, street addresses, grid references, routes-and-offsets. Each method has its advantages and drawbacks. Unfortunately it is not practical to use multiple methods for a single message, because message length is a constraint in wireless communications systems.

The Cross-Streets Profile is one of the message specifications in the LRMS, and currently the one most favoured by vendors. It specifies a location in the form of an offset along a street segment, where the segment is defined by a street name and bounding cross-streets. For example, "2473 metres down Hollister Avenue between Patterson Road and Walnut Street." Consider the potential for lack of interoperability when sending a message in this form between two data bases:

- (a) Spelling errors and differences in vendor practices: Hollister is spelt Holister; Avenue is abbreviated Av in one database, Ave in another.
- (b) Gross differences in street names: El Camino Real in database A, Highway 101 in database B.
- (c) Topological and inclusion errors. A street that appears in database A does not appear in database B. Therefore an intersection referenced in the source database is not identifiable in the target.
- (d) Positional differences at intersection nodes and shape points

In addition, there are problems inherent in the method itself, in particular the case where a street intersects another several times. Some of the above problems can be corrected using fuzzy search algorithms and reinforcement of clues. To test the effectiveness of this profile, we first define application scenarios, testing criteria, metrics and tolerances. The problem is broken down into components, e.g. (a) the probability of finding an accurate lexical match using simple and complex search techniques; (b) the mean positional error when relaying an offset from one database to another.

This paper discusses the ITS interoperability problem in general, and presents experimental design and results of work to date using popular commercial databases for the County of Santa Barbara, California.

Assessing Topological Similarity of Spatial Networks

John D. Nystuen, Andrea E. Frank, and Larry Frank
University of Michigan

Problem Statement

A common need in spatial operations and analyses is the ability to transfer data between digital databases. The requirements f or creating such capacity have received a great deal of attention, for example, through the efforts to develop Spatial Data Transfer St andards (SDTS) (USGS, 1990). However, more is involved than the need to agree on standard definitions for data from different sources to be effectively combined; although the standards issue remains of paramount importance. Many technical, semantic and organizational issues arise when attempting to transfer data between databases. Our concern is with how to compare or combine databases that characte rize spatial networks such as road and highways systems.

Digital maps of road networks play an important role in many transportation related applications including motor vehicle naviga tion, traffic advisories, route scheduling, and emergency vehicle routing. Many different agencies, both public and private, are creat ing databases of road networks and the need to coordinate outputs often arises. These needs are receiving attention but a great deal o f theoretical, technical, and institutional effort is required before transportation applications achieve widespread use (Goodwin, 1996).

In another allied public field, TIGER files have been shown to have reasonably accurate address ranges and effectively associat e street addresses with census polygons. However, at municipal street map scales, streets known to be straight lines appear noticeably crooked on printouts of TIGER files. It would be useful if the TIGER line file could be linked to the typically more positionally accu rate municipal street map. The municipal street maps are often tied to land parcel maps and merging them with the TIGER street map woul d create a link between the census polygon database and the land parcel polygon database.

A third general reason for developing means to compare networks is to be able to make comparisons of a network over different t ime periods. Overlaying a digital database of a network in one time period with the database in another time period would allow the di fferences to be highlighted. Change detection is often of interest. In general, being able to merge or match different databases char acterizing the same spatial network would be useful in many contexts.

Approach

The key to comparing two spatial networks is to associate nodes in one coverage to corresponding nodes in another coverage. Th is is not a simple task. In the first place, two maps of a network such as a highway system, are unlikely to have exactly the same num ber of nodes. Databases will especially not have the same number of pseudo-nodes because pseudo-nodes are generated for a variety of s pecial functions not likely to be universal[1]. However, the topology or connectiveness of different representations of the same real ne twork should be similar. Acting on this assumption the approach adopted is to identify for each node in a first coverage a nearest nod e in the second coverage which also has the identical topological properties. As the coverages may have many nodes, the matching proce dure is automated using standard GIS software. The automated processing results in some but not all nodes in a coverage being matched with corresponding nodes in the second coverage. Unmatched nodes are identified and an interactive process is initiated in which human judgment is used for finding additional matches. Initial proximity parameters are relaxed and additional information referenced to in crease the number of matched pairs identified. The interactive procedure ends with a summary statement of the degree to which the netw orks match. Once a clean subset of paired nodes is created it can be used to assess displacement discrepancies using standard error ellipses.

This work is derivative of research we have undertaken to evaluate the quality of digital map databases for Intelligent Transpo rtation Systems (ITS) applications (Aggarwala, et al, 1997). In that research we developed methods for identifying and separating diff erent types of errors in digital map databases intended for use in motor vehicle navigation. Error detection was accomplished by compa ring segments of a vendor map with the corresponding portion of a reference map of known greater positional accuracy. Errors of displa cement, omission and commission are identified by these procedures. The procedures we have developed are appropriate for addressing wi der issues involved in analysis of spatial networks.

Research Concerns

We have encountered a number of items that appear to be interesting research topics. A few a listed below.

- Scale effects. When databases are derived from sources of greatly different resolution or accuracy standards, map generalizati on problems arise that are similar to those encountered in analytic cartography. A potential approach might be to perform transformati ons on the databases to bring them to the same level of generalization before comparing them. This is an open research task.
- Definitional discrepancies. Spatial data standards are not fully complied with and even with compliance many definitional disc repancies occur in representing the same features in databases originating from different sources. These differences confound quality evaluation and some means for addressing them needs to be developed.
- Proprietary databases. Spatial databases made for motor vehicle navigation or for business location applications sometimes hav e proprietary elements in the

data structure that prevent access for the general user. Some accommodation between research and proprie tary interests is needed.

- Pattern recognition. Human spatial pattern recognition skill remain quite superior to algorithms used for this purpose. Networ ks mutually far enough out of line to negate proximity/topological matching can be modified by interactive processing. More automated means of recognition is a desirable research goal.

References

Aggarwala, Raj, John D. Nystuen, Andrea Frank, Jyothi Palathinkara, "Digital Map Database Quality Evaluation Issues for ITS Applications," Paper Presented at UCGIS Annual Assembly and Summer Retreat, Bar Harbor, Maine, June 17, 1997

Goodwin, Cecil W. H., 1996, "Location Referencing for ITS," White Paper prepared for Oak Ridge National Laboratory and the National ITS Architecture Program, April, 1997, 18 pages.

U. S. Geological Survey, (1990) National Mapping Division: Spatial Data Transfer Standards, U. S. Department of the Interior, Washington DC.

---

[1] Nodes are topologically distinguished if they have degree other than two. A node with two incident arcs is termed a pseudo-node and is not topologically distinguishable. Shape points are not topologically distinguishable.

GeoToolKit: Opening the Access to Object-Oriented Geodata Stores

Oleg Balovnev, Martin Breunig, Armin B. Cremers and Serge Shumilov
Institute of Computer Science IIIUniversity of Bonn
Roemerstr. 16453117, Bonn
Germany

We report on our experience with GeoToolKit [BBC 1997] - a software which is intended to facilitate the development of 3D/4D geo-applications. It addresses primarily the efficient maintenance of 3D-spatial objects within a database. To achieve this GeoToolKit is tightly coupled with the object-oriented DBMS ObjectStore®. GeoToolKit is not a closed GIS-in-a-box package - it is rather a library of C++ classes that allows the incorporation of spatial functionality within an application under development. Thus it is primarily oriented on soft-ware engineers with the C++ experience involved in the development of special-purpose geo-applications which can be hardly modeled within standard GISs. Currently GeoToolKit offers classes for the representation and manipulation with simple (point, segment, triangle, tetrahe-dron) and complex (curve, surface, solid) spatial objects. Complex objects are approximated and represented as homogenous collections of topological simplexes of the same dimension (e.g. triangle networks). Following the object-oriented modeling technique, not only abstract geometric primitives, such as points, curves, and surfaces but real world entities such as drilling wells, geological sections and strata, can be modeled and maintained. Applications developed with GeoToolKit simply inherit geometric functionality from GeoToolKit, extend-ing it with the application-specific semantics. Spatial objects are collected into a special con-tainer class, which provides an efficient spatial retrieval using multidimensional indexes supplied within GeoToolKit.

GeoToolKit was successfully used for the re-implementation of GeoStore, an information system for the management of geologically defined geometries [BBC 1994] as well as for the development of a range of new applications. Due to GeoToolKit an application developer could focus on the application semantics instead of such "creative" tasks as optimal assem-bling of spatial objects from multiple tables or the implementation of routine geometric al-gorithms. This resulted in the considerable reduction of the code written and made the sources more understandable. Data earlier hidden within a particular application became available for all other applications built on top of GeoToolKit, that encouraged a consistent and non-redundant maintenance of data.

However, a direct database client-server communication with GeoToolKit-based data was not always possible because of the extreme heterogeneity of already existing applications and hardware platforms we have to deal with. Apart from this a database communication level did not provide relevant network navigation facilities. To implement a remote access to data for an external application, intrinsically not compliant with GeoToolKit class hierarchy, we had to use low-level UNIX-sockets libraries. This approach turned out to be very efficient for the direct communication between two particular applications. However, it lacks generality. To make external applications aware of internal conventions we needed a kind of standard dis-tributed object computing platform. Taking into account the object-oriented nature of data the most suitable solution is Object Management Architecture [OMG97] which promises to be-come a world-wide standard. Using this approach any other CORBA-compliant application can get an open access to GeoToolKit-based data.Following the standard way for the integrating an external library into a CORBA environ-ment we developed a wrapper. It encapsulates access to all GeoToolKit classes, providing run-time control of their instances, e.g. objects creation, caching and garbage collection. We re-produced GeoToolKit^Òs class hierarchy in CORBA^Òs Interface Definition Language (IDL). The IDL specification of GeoToolKit classes conforms the representations of geometries proposed within the Object Geodata Model [OGIS96], thus enabling their convergence in future.

Clients interact with objects of the encapsulated classes through special stub and skeleton parts generated in the compile time from IDL specifications. However, this approach does not work for classes not included in the original IDL class hierarchy because it requires a com-plete code recompilation in the both the client and server sides. An incremental extension of the static core shared by different applications is the most challenging problem in the wrapper development. We propose a mechanism that provides a dynamic creation of new user-defined classes without server recompilation. Clients interact with instances of dynamic classes through a specialized Meta-Object Protocol (MOP) implemented with the use of the Dynamic Invocation Interface (DII) on the client side and Dynamic Skeleton Interface (DSI) on the server side. However, operating through MOP is extremely annoying for the user. Special mediator classes generated on the client side hide the MOP from the user and allow the inter-action with the server according to the standard C++ interface.

To deal with persistency we use a method based on the substitution of the Basic Object Adapter (BOA) by a special Object Database Adapter (ODA). We chose Orbix® as an imple-mentation platform because it provided a core functionality necessary for the ObjectStore ODA development. Connections between implementation objects and their proxies are organ-ized according to the TIE binding technique, i.e. only implementation objects are stored in the database. All relationships between CORBA objects in a database are actually relationships between corresponding implementation objects. For the unique objects identification ODA maintains an internal table of associations between CORBA object references and implementation objects.

In the project started recently GeoToolKit is used as a platform for an open distributed envi-ronment that will provide an open access to the GeoStore database for various remote geo-services involving the 3D-modeling tool GOCAD® located in Bonn and the geophysical modeling tool IGMAS® resided in Berlin. A free data exchange via a common database gives for geo-scientists an opportunity to perform a cooperative adjustment of geophysical density and geological stratigraphic models.

References

[BBC97] O. Balovnev, M. Breunig, A.B. Cremers. From GeoStore to GeoToolKit: The second step. In: Proceedings of the 5th International Symposium on Spatial Databases, LNCS, Vol. 1262, Springer, 1997, pp. 223-237.

[BBC94] Th. Bode, M. Breunig, A.B. Cremers. First Experiences with GeoStore, an Information Sys-tem for Geologically Defined Geometries. In Proceedings IGIS^Ò94, LNCS, Vol. 884, Springer, 1994, pp. 35-44.

[OMG97] Object Management Group. CORBA 2.0/IIOP Specification. OMG technical document r-mal/97-02-25. http://www.omg.org/corba/corbiiop.htm

[OGIS96] The OpenGIS Guide. Open GIS Consortium Inc. OGIS TC Document 96-001. http://ogis.org/guide/guide1.htm

This work is funded by the German Research Foundation (DFG) within the collaborative research centre 350 "Interactions between and Modeling of Continental Geosystems" at Bonn University. This work is funded by the German Research Foundation (DFG) as part of the program "Interoperable Geoscientific Information Systems".

Interoperability of Geographic Information: From the spreadsheet to virtual environments

Pedro Pereira Gonçalves, Nelson Neves, João P. Silva, Joaquim Muchaxo and António Câmara
Grupo de Análise de Sistemas Ambientais
Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa

Introduction

Environmental modelling, statistical data analysis and GIS are three examples of computational activities that are usually relegated to separate, large and sophisticated computer systems. Usually each one of them has more functionality than is required by the user, whose needs exceed the range of a single one (e.g. both GIS and statistical data analysis). In the first two, the data organisation benefits from a GIS perspective and some very simple issues like the quality of spatial coverage of the data can be resolved by a simple geographical display. Likewise, the quality of the time-dimension over a geographical region can be combined with a geographical display by enhanced visualisation. For these two examples (spatial and temporal coverage of data), for interfacing models over a large range of possible input parameters, and for many other circumstances, it is beneficial to have a level of integration which is much greater than a simple file transfer mode.

One of the most often critic to Geographic Information Systems (GIS) is that has failed to give adequate attention to principles of cartographic design, or for regarding the map as a simple store of information rather than a tool for communication [Goodchild, 1990]. If the database could be considered as truth, then the produced map would be no more than a store device, since would exist a simple correspondence between objects on the database map and objects on the map. But, usually, the database is an approximation of the geographical truth and, since it can affect the user's view of the world, the design of the output display is critical. The actual of electronic display go far beyond conventional cartography.

The advent of the Open is Geodata Interoperability Specification (OGIS) opens for the first time a real opportunity to develop data structure-independent GIS applications [Buehler, 1994], [OGIS 96]. The Open GIS Consortium (OGC) does not attempt to define high level operations. Its specifications are restricted to low level database (SQL-like) and topological operations based on the work by Egenhofer [Egenhofer and Franzosa, 1991] and [Egenhofer *et al.*, 1993]. This is achieved through the use of a common language for sharing geodata and standardised definitions of interfaces to functions that operate on geographic information. While their initial work is focusing on traditional geoprocessing, such as spatial selection, thematic overlay, measurement, and distance analyses, other services which access geographic information, such as hydrodynamic models, seismic prediction, and allocation functions, will also be able to directly access geodata stores (as well as other geoprocessing functions).

But the effects of the development of object-oriented databases and object-oriented programming systems in computer science are likely to be much more profound. The argument that has to be done is that GIS software does not have to be visible to the end-user. By the contrary, the GIS as to be understood as another variable available to the end-user. The GIS community needs to transform all the GIS concepts, and functions, into a variable as simple to define an *integer*, *byte* or *float*. In this context the NovaGIS project [NovaGIS, 1997] pretend to establish the concept of an "invisible GIS". Adding to the computer system the variables necessary to work with Geodata.

System Description

The NovaGIS software is an OLE server that creates an interface to *Geodata*. One of the main goals is to transform *Geodata* into a normal variable in common macro language. Using this object traditional geoprocessing services can be added to our software. At the same time, the GIS application remains invisible to the user. When the variable is define the user works directly in the data structure and functions.

For example, a simple Basic programme can do the query of a map:

```
Public alt as Object
Sub PointXY_Example()
Dim ZZ as Single
Set alt = CreateObject("NovaGIS.geodata")
alt.FileName = "c:\users\pedro\prj\teste\temp_alt"
Rem and then do some operations ....
ZZ = alt.pointxy(10, 10)
End Sub
```

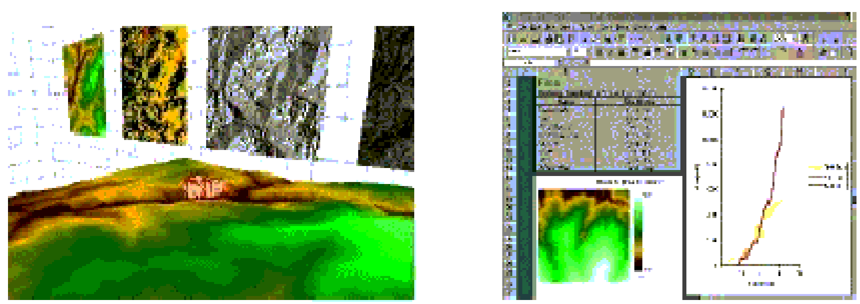
The simplicity of defining our map, study area or set of data is given at the same time by the potential of many macro languages present in common office applications. At the same time there is no need to invent or rebuild new script or macro to work with this structure. The full potential of software dedicated to environmental modelling, statistical data analysis simulation, or even data visualisation can enhanced by the add-on of GIS concepts and functions.

The NovaGIS server can be used in any OLE compliant program like Excel, Visual Basic, Delphi or any other OLE-compliant software. All these software products have the capability or to run macros or even to compile the source code. The system functions are well separated, but to the end-user the system works as one. This interconnection is achieved by means of the Object Linking and Embedding (OLE) from Microsoft. The foundation of OLE is the Component Object Model (COM), that dictates how OLE applications behave and interact, and provides mechanism that lets one application connect to the OLE interfaces that another supports.

Application

The case study chosen was forest fire modelling, using the model FireGIS [Gonçalves and Diogo, 1994], and to prove the feasibility of the systems, two intertwined applications were built. In the first, the forest fire model interface was build in a spreadsheet of *Microsoft Excel, with all the data input, model parameters, map and charts visualisation (Fig.1b). At the same time the outputs of the model are seen in a Virtual Environment (VE) using the VirtualGIS system [Nelson et al.*, 1997].

The VirtualGIS application updates the Digital Terrain Model (DTM), where a correspondence established between cell sizes in the virtual environment and the simulation environment that is established when the model starts running. As new simulation steps are computed the information about the cells that changed is sent to the VE, reporting the cell location (x,y) and the new cell value. At the same time the user can operate the data inputs and model results in a common spreadsheet program. In Fig. 1a a snapshot of the visualisation of the outputs of a forest fires simulation model in the VE are shown.



(a) (b)

Fig. 3 - Visualising a forest fire in the Virtual GIS Room (a) and in a spreadsheet (b).

So far all the tests were done in a single computer by means of OLE. In the case of several computers will be used the distributed version of OLE. The Distributed COM (DCOM) is a technology that extends the local capabilities of COM to cross network communications between objects on different computers. This will enable the GIS and VE system in different computers and even in completely different locations. The use of a virtual environment system to visualise and interact, with spatial information and associated simulation models, allows the user to explore information recreating their usual interaction in a real GIS room.

The objects created can be used in different applications, allowing their use in a diverse range of applications, from word processors, spreadsheets, programming tools (like *Visual capabilities wazzu Basic*, *Delphi* or *C++*). This structure fits as well to the development of different versions with access through HTTP protocol. The interoperability of the system allows the design of different user interfaces according to the different needs as well a more integrated analysis of the results.

Acknowledgements

This work was partially supported by JNICT under research contract PEAM/C/FGI/633/95. The PRAXIS grant 3/3.2/AMB/04/94 supports the work of Pedro Pereira Gonçalves.

References

[Buehler, 1994] Buehler, K., 1994. OGIS Project Document 94-019. OGIS Geodata Model Overview. The Open GIS Foundation, Cambridge, MA.

[Buttenfield and Ganter, 1990] Buttenfield, B.P. and Ganter, J. H., 1990. Visualization and GIS: What Shoul we see? What might we miss? Proceedings of the 4th International Symposium on Spatial Data Handling, Zurich, Switzerland, Vol.1, pp. 307-316.

[Clapham, 1992] Clapham, S., 1992. A formal approach to the visualization of spatial data quality, in Proceedings of GIS/LIS'92, San Jose, CA, Vol.1, pp.138-148.

[Egenhofer and Franzosa, 1991] Egenhofer, M. and R. Franzosa, 1991. Point-Set Topological Spatial Relations. International Journal of Geographical Information Systems 5(2), pp. 161-174.

[Egenhofer et al., 1993] Egenhofer, M., Sharma, J. and D. Mark, 1993. A Critical Comparison of the 4-Intersection and the 9-Intersection Models for Spatial Relations: Formal Analysis. Proceedings, Eleventh International Symposium on Computer-Assisted Cartography (Auto Carto 11), pp. 1-11.

[Gonçalves and Diogo, 1994] Gonçalves, P. and Diogo, P., 1994. Geographic Information Systems and Cellular Automata: A New Approach to Forest Fire Simulation in Proc. of The European Conference on Geographical Information Systems (EGIS 94), Paris - France, 1994.

[Goodchild, 1990] Goodchild, Michael F., 1990. Keynote Address: Spatial Information Science. Proceedings of the 4th International Symposium on Spatial Data Handling, Zurich, Switzerland, Vol.1, pp. 3-12.

[Goodchild, 1992] Goodchild, M., 1992, Geographical Information Science, International Journal of Geographic Information Systems, 6(1), pp.31-45.

[NovaGIS, 1997] NovaGIS homepage. http://virtual.fct.unl.pt/~pedro/NovaGIS/

[Nelson *et al.*, 1997] Neves, N, Silva, J.P., Gonçalves, P., Muchaxo, J., Silva, J. and Câmara, A., 1997. "Cognitive Spaces and Metaphors: A solution for Interacting with Spatial Data", Computers & Geosciences, Vol. 23, No.4, pp.483-488.

[OGIS 96] OGIS. Open GIS Consortium. http://www.ogis.org/

A Virtual Geospatial Information Server (VGIS) Providing Transparent Access to Heterogeneous Sources

Chuanchu Wang, Liya Ding, Jiankang Wu
Institute of Systems Science
National University of Singapore
Singapore 119597

ABSTRACT

Geodata sharing and interoperability are the main goals of our virtual geospatial information server, VGIS. VGIS is originally designed to provide one step service for public users who want to access geographical data sources provided by some government departments. Through VGIS, the user can access, view, browse, print and order the data they are interested in. VGIS has an open system architecture. It provides access to common gis systems such as Intergraph, Arc/Info. It can be smoothly extended to include other spatial data sources.

VGIS consists of three sub-systems. One is the client. It consists of a graphic user interface (GUI), used by Internet-user to input their query commands, to view and browse geodata. It is implemented using Java. So it is platform independent. User can run it by a Java-enabled Internet browser. VGIS also provide Java applet to display multimedia information such as text, image, audio and video. This is useful in many applications where rich information type is necessary.

The other two sub-systems are server type. The first one is the catalog server which manages metadata about the location, function, identification, catalog, spatial domain, schema and other information of all geospatial data sources which can be accessed through VGIS in the network environment. Data providers who want their data sources to be open to public need to register with the catalog server by providing all necessary information the catalog server requires. All the metadata information is managed by an database. The primary function of the catalog server is to help to locate the address (URL) of the geospatial data sources. It also helps the user to construct SQL query conditions with client GUI. The information in this server can also be requested when data format conversion is needed.

The second server is called Query/Geoprocessing server. It executes query commands and performs certain operations, such as data format conversion, spatial relationship verification. The Interface of this server is an extended SQL, referred to as Spatial SQL (SSQL). SSQL commands will be parsed and decomposed to sub-query SSQL commands according to the metadata information from the catalog server. The sub-query commands will be sent to correspondent database drivers to retrieve geodata. The returned result from these database drivers will be further processed according to SSQL conditions and integrated as the final result to be sent to the corresponding client.

One feature of VGIS is the transparency of the locations of geospatial data to the end users. User can issue a query command without knowing the destination of the data he wants. The query server can locate the Internet address of the data store by the information available from the catalog server, and the client can receive data from the query server. However, the user can also obtain the information from the catalog server directly about all the services available on the Internet, information on a particular data source. If the user is only interested in the geodata stored in one data store, and he also knows the address of the correspondent GIS system (maybe by the information requested from the directory server), he can communicate to the database driver directly, and the result will be sent directly from DB driver to the end-user, just as in the systems such as GRASSLAND.

To provide very friendly interface, we have developed an icon based visual spatial SQL. The visual SSQL consists of spatial object icons and spatial relationship icons. It is self-explainable. User can use spatial object icons and spatial relationship icons to construct query commands. This exempts the user from understanding the complex syntax of SSQL, which may otherwise easily cause syntax errors.

STRUCTURAL DESIGN OF DISTRIBUTED GEOGRAPHICS INFORMATION SYSTEMS BASED ON A HIGH ORDER LOGIC

Guillaume Koum
CRID-Universite de Bourgogne-Faculte des Sciences
Mirande-Allee Alain Savary. B.P 400-21011 Dijon
cedex-France

One of the major technics while distributing relational databases lies on the partitioning of the relations which is followed by allocation issues. The partitioning is computed through a query based approach organizing data vertically and/or horizontally. Even object oriented distributed databases are built in general on these principles. The aim of this paper is to show that as far as GIS are concerned, a structural modeling of objects may be obtained without taking care of queries asked or methods applied to the databases . The mechanism used requires a high order logic statements and processes a set of sites which ares components of a distributed GIS interacting in a global intelligent environment. The results are tested out in Urban planning sector.

Land planning through geographical information systems, remote sensing and multicriteria decision analysis

Germana Manca
CRS4
Via Nazario Sauro,10
09123 Cagliari
Italy

Andrea De Montis
Dipartimento di Ing. del Territorio
Universita' di Sassari
Via E. di Nicola,2- 07100 Sassari
Italy

During the last few years researchers and environment planners have carried on their preliminary studies, according to a multidisciplinary approach. One of the main issue in this approach is the sustainability of the economic development. As a matter of fact it is recognized the importance of the study related to the distribution and to the limits of natural resources; this can be accomplished easier than before by the use of extremely powerful hardware and software. The purpose of this contribution is to demonstrate the usefulness of an integrated approach to planning of GIS and RS in a complex problems of decision making. Many GIS applications are suited for remote sensing input and the integration of the techniques are becoming accepted in the landscape planning. By means of GIS and RS, environmental information will be integrated with administrative, political, social and economic data. GIS, RS and fuzzy analysis will be used to assist in environmental compatibility studies and decision making. As a case of study, an area located in the center part of Sardinia will be analyzed. This area is especially interesting because of the presence of industrial and agricultural resources and infrastructures.

Characterization of Data, Queries, and Index Performance of Geographic Information Systems with Applications to Informix Geodetic DataBlade Module

Kumar Ramaiyer
Informix Software, Inc.

Abstract

This paper discusses various types of geometry data on which GIS applications build index, and also presents a collection of queries that tests for various characteristics of the index. The proposed merits include the size of the index and the number of I/O's performed during the query. The goal is to come up with a standard collection of data and queries that will help us to characterize the performance of various systems. The paper proposes to show that it is important to implement various indexing schemes and also allow for various tunable parameters to optimize the performance for given data.

As a case study, we apply these standard data and queries to evaluate and characterize the performance of the Informix Geodetic DataBlade, a module that allows for storage and retrieval of geographic data collected on or near the surface of our Earth.

Introduction

Geographic Information Systems (GIS) are useful tools for storing, retrieving, and visualizing geographic data. They are widely used in various applications areas like, utilities management, real-estate management, weather modeling, monitoring of natural habitats like, wet lands, bird migration, rain forest, etc., navigation, map generation, and scheduling routing for various services like fire personnel, security, postal, etc. The requirements of GIS include support for:

Data Modeling: This is the process of representing the information content of the applications, like roads, pipelines, housing development, lakes, forest, etc., in some internal form, which store the connectivity as well as the geometry of the geographic data or objects.

Data Storage and Spatial Index: The geographic data is typically stored in some database. Database provides an integrated environment for storage and retrieval of information. Database also allows for indexing of the underlying geographic data. Several strategies for indexing geographic or spatial data have been developed over the years.

Data Retrieval: Applications retrieve the information in several ways, but they can be abstracted into a finite collection of queries. These queries make use of the underlying spatial index built on the data to retrieve the data in an efficient manner.

To compare the performance of the various GIS, we need a common set of data and queries that represents the needs of various applications that use GIS.

Motivation for Collection of Data and Queries

The performance of a GIS depends on the efficiency of the underlying database to store, index, and retrieve the geographic data. But the indexing strategy is the key factor in deciding the performance. There are several indexing strategies for the spatial or geographic data, but there are two fundamental approaches:

- Space Partition: This approach involves partitioning the underlying domain space occupied by the spatial data into different cells. Spatial indexing, then involves associating the data with different cells and vice versa. During queries, one first finds the cells occupied by the query region, and then finds all the data associated with these resulting cells. The efficiency is then proportional to the space required to store the data in different cells and the number of data-cells searched during the query.
- Quad-tree indexing method and its several variations use this approach. This approach uses a partition technique which is independent of the data and also subdivides the underlying space at fixed positions. Therefore this allows for faster-direct indexing, but the space and query performance varies a lot with data distributions.
- Data Partition: This approach involves partitioning the underlying data into different groups based on their geometric properties. Typically this approach recursively partitions the data into groups till the size of the group reduces to a small constant. The groups are then organized in a hierarchical fashion in multiple levels, and in each level the information of union of space occupied by all the data below that level is stored. The efficiency is then proportional to the number of levels in the hierarchy, the partitioning strategy, and the number of levels and groups in each level to be search during the query.

k-d-tree, R-Tree, and their variations use this approach. Since this partition technique is data dependent, the space requirement is proportional to size of the data and is better than other approach. But the query performance again varies with the data characteristics, and direct indexing is not possible here.

The different characteristics of these two approaches, their several variations, as well as the subtleties and anomalies of the various implementations stress the need for a common data set and queries, which will enable us to characterize the behavior of a GIS. The characterization will then enable us to choose appropriate indexing scheme for a give type of data.

The goal of this study is therefore to first come with a data collection that reflect different data distribution in the underlying domain, different types of connectivity and geometry of the data, and different sizes of the individual data--which are either artificially generated or reflect some natural data in our universe. Once the data is collected, which is a separate process common to all the GIS applications, various indexes can be built and evaluated for their size. To test the spatial selectivity of the index, one needs to generate a collection of queries that reflect the needs of the various applications of GIS.

In the following sections, we address these issues in data collection and queries, which we propose to use as benchmark for comparing the performance of the GIS, and for characterization of the efficiency of the different indexing schemes.

Data Collection

In this section we describe the goals of the data collection. Each application deals with a unique collection of objects that vary in their connectivity, geometry, distribution, size, etc. The benchmark that tests the efficacy of a GIS should include data to represent all these parameters.

Distribution Properties

The distribution properties of the objects we would like to consider for the benchmark are:

- Uniform distribution: The objects are spread uniformly over the domain (see Fig.~\ref{fig:uniform}). The uniform distribution should enable the indexing strategies to achieve good space utilization as well a balanced indexing structure, thereby achieving good performance. This type of data is therefore fundamental for any type of benchmark data and it tests the quality of the implementation.
- Non-Uniform distribution: The objects are spread non-uniformly over the domain, and hence there are regions where the concentration is more than that of others (see Fig.~\ref{fig:nonuniform}). This type of data is very common in GIS applications. It significantly affects the space utilization as well as query performance of most indexing strategies.

Coupling Property

We define the coupling as a property that determines the extent to which one object in the domain interacts with the other objects i.e., whether it can intersect with any of the other objects or whether it can be inside any of the other objects and so on. The data collection should support the following coupling properties between the objects:

- Disjoint: All the objects in the domain have no intersection and hence are disjoint (see Fig.~\ref{fig:disjoint}). The disjointness property of the data should allow for good spatial partitioning when building index structures. As a result queries should have better selectivity and as a result perform better on data with this property.
- Overlap: There are objects in the domain that overlap or intersect with each other (see Fig.~\ref{fig:overlap}). The inherent errors associated with the data collection process sometimes leads to overlapping geometric data. Also during aerial photographic scanning overlapping is allowed so as not to miss the details. The overlap causes problems in terms of reduced spatial selectivity

because it requires larger bounding boxes, and also complicates the index selectivity by either forcing the objects to be referenced multiple times or causes splitting of the objects into smaller disjoint parts as in quad-tree partitioning. A good performance of an index on data with this property is very critical for GIS applications.

- Nested: There are objects in the domain that are nested within each other (see Fig.~\ref{fig:nested}). The nested objects arise naturally in GIS applications as objects with forbidden regions. For example, lakhs and oceans with islands, road ways, plan of a campus without details of buildings, etc. The forbidden region poses challenges in terms of selectivity during index scan, as they are also included in the bounding-box representation of the object typically used by the index. This causes serious problem when the area/volume of the forbidden region is very large compared to that of the object itself. A good performance of an index on data with this property is thus very important and desirable for GIS applications.
- Random: There are no defined coupling properties among the objects. They may or may not have any of the above three properties (see Fig.~\ref{fig:random}). A good performance of the index on data with this property shows the robustness of the selectivity of the index. The randomness in coupling as well as distribution will test the weakness in the spatial selectivity of the index, if any assumptions were made about the properties of the data in terms of coupling or distribution.

Size Property

We define the size of a object as the storage requirement imposed on the GIS. Some of the objects require compact representation and hence require lesser storage than others. The size properties of the objects we would like to include in the benchmark are:

- Uniform or Variation is bounded by a constant: All the objects are either of equal size or their sizes do not vary much (see Fig.~\ref{fig:uniformsize}). Some applications deal with the objects that are either simplicial like square, triangle, quadrilateral, etc., or complex and still require only as many as vertices as other objects in the domain to describe. The size variation generally causes problems with data storage as not all pages of the secondary storage of the index will contain equal number of objects and also may increase the overall size of the index as some of the large objects may require new pages. Algorithmically the size variation adds additional constraint as the index-building algorithm needs to not only achieve good spatial selectivity, but also need to fit the objects in as few pages as possible so as not to increase the overall size of the index. Therefore, the almost-constant-size property of the data should help GIS build good index, and hence a good data to be part of the benchmark.
- Variable: There are no constraints on the individual sizes of the objects (see Fig.~\ref{fig:varsize}). The Fig.~\ref{fig:varsize} shows two objects, one

requiring four vertices and other requiring many vertices (about 20) for description. The data set consists of large objects that may require special handling since the data base server generally puts a limit on the individual size of objects. The special handling may affect the performance of the index, and hence the data set with this property is very important to be part of the benchmark.

Area/Volume Property

The objects stored in a GIS come in various aspect ratios and various shapes. Accordingly they occupy different area/volume in the underlying Universe. Therefore the volume or area properties of the objects we would like to include in the benchmark are:

- Zero Area/Volume Objects: The point data have no area or volume, and there are number of applications that deal with point data to abstract information like position of landmarks, data collection sensors, etc.
- Variable: There are no constraints on the area or volume occupied by the individual objects (see Fig.~\ref{fig:varshape}). The relatively larger objects or ``whole Earth objects'' cause serious problems during index generation and also during scan, as their bounding boxes occupy large portion of the underlying domain, thereby reducing the spatial selectivity. The smaller objects will fall inside the region covered by the bounding-box of the larger objects, thereby rendering them ineffective during the index-scan. Therefore approaches other than traditional bounding-box generation or some variation of it is called when the domain contains larger and variable shaped objects.
- Uniform or variation is bounded by a constant: The area/volume occupied by the different objects are all either same or the difference is bounded by a constant (see Fig.~\ref{fig:uniformshape}). This similar-shaped data property will help index to achieve better spatial selectivity as the bounding box of the individual objects tend not to interfere with each other.

These various parameters and their combinations reflect the properties of the data used in most of the GIS applications, and hence are good candidates to be part of a benchmark.

Data Formats and Different Standards

The major hurdle in coming up with a standard benchmark is the various data formats used by the different vendors of GIS. There are efforts to unify the data representation and to come up with a common data model. But there are multiple unifying efforts and thus we have several standards available like OGIS, SDTS, etc. As a result, there is a proliferation of software that converts the data from one format to another.

Queries

Each application that uses a GIS to retrieve geographic data has a unique set of queries. Also the application differ in the manner in which they perform the accesses. Some of the queries access same or adjacent regions in the underlying domain repeatedly. Other queries access different regions that are not proximal, but the number of different accesses performed over a given interval of time is fixed.
Though the queries and their patterns look vastly different they can be abstracted into a finite collection of queries as follows:

- Range Searching: This is the most general form of queries used in the application, and it involves searching the underlying domain to retrieve the objects that fit the search criteria specified by the query range. The Fig.~\ref{fig:intersect} illustrates the information retrieval through range searching. This query can be further classified as:
- Range Intersection: Retrieve all the objects in the underlying domain that intersect or overlap the region specified by the query range. In Fig.~\ref{fig:intersect}, range intersection query retrieves the objects A through F.
- Range Containment: Retrieve all the objects in the underlying domain that are completely inside or contained within the region specified by the query range. In Fig.~\ref{fig:intersect}, the range containment query retrieves the object D, which is the only object inside the query range.
- Range Outside: Retrieve all the objects in the underlying domain that are completely outside the region specified by the query range. In Fig.~\ref{fig:intersect}, this query retrieves all the objects in the domain, except the objects A through F.
- Range Searching with Tolerance: Retrieve all the objects in the underlying domain that either intersect the query region or lies within certain distance from the boundary of the query region. The tolerance is specified in terms of distance from the query range. In Fig.~\ref{fig:tolerance}, this query retrieves all the objects in the domain that either intersect the query region or lies within the distance specified (shown as thick lines surrounding the query region).

The shape of the ranges also vary from the typical recti-linear ranges to other simplicial objects like triangle, quadrangle, etc., to other objects like circle, sphere, convex objects, simple polygons, complex polygons like polygons with holes, etc.

- Proximity Searching: This form of query involves computing distances between objects, and hence is applicable only to those domains, which contain objects that allow for distance metric to be applied on them. This query can be further classified as:
- Within a Given Distance: Retrieve all the objects that are within certain distance, specified as parameter, from the query object (also specified as parameter). In Fig.~\ref{fig:within}, the query retrieves all the points within the circle, with radius equals to the distance parameter.

- Beyond a Given Distance: Retrieve all the objects that are beyond the distance, specified as parameter, from the query object (also specified as parameter). In Fig.~\ref{fig:within}, the query retrieves all the points that are outside the circle, with radius equals to the distance parameter.
- Post-Office Problem or Nearest-Neighbor Queries: This classical query involves retrieving the object from the domain which is at the closest distance to the query object specified as a parameter (see Fig.~\ref{fig:postoffice}).

Other variations of proximity searching include ray shooting i.e., finding closest object to a query object along a given direction, closest pair i.e., finding the pair of objects in the domain that are closest to each other, etc.

Apart from the above abstract queries, the query patterns are unique for GIS applications. The queries used in GIS applications do not uniformly sample the entire domain, but are rather non-uniform and adaptive in nature. Typical GIS queries have spatial coherence and temporal coherence properties:

- Spatial Coherence: This is the property that same regions are accessed more frequently than other regions. For example, service-dispatch GIS query high-population regions more often than low-population regions.
- Temporal Coherence: This is the property that between any two consecutive queries to the same region, the set of queries, which we refer to as the working set, is fairly small. For example, in a data collection GIS, sensors are finite in number, widely distributed, and are tracked periodically to collect the data.

It is easy to assume that virtual-memory properties of the underlying system would automatically achieve good performance for queries with above properties. But there is no performance guarantee, and it is important that the indexing structures themselves adapt their organization to guarantee performance.

Informix Geodetic DataBlade Module

In this section, we discuss the application of ideas discussed in the previous to characterize the performance of the Informix Geodetic DataBlade module. The Geodetic DataBlade module extends the data types supported by the Object-Relational DBMS Informix Universal Server to represent geometric objects such as points, polygon, circle, ellipse, lines, line-segments, etc., on the earth's ellipsoidal surface. The module supports true geodetic coordinate reference system i.e., geodetic latitude and geodetic longitude to describe the coordinates on the surface of the Earth. For each object, the DataBlade supports storing an altitude, which is measured in meters above or below the surface of the Earth, and also supports time-range, which helps to associate time with the data. The Geodetic DataBlade module measures the distance between two points along a geodesic, which is the shortest path between two points on the surface of an ellipsoid.

The Geodetic DataBlade module also provides:

- SQL support for defining columns in a tablecorresponding to the geodetic types.
- SQL support for inserting data into these columns.
- Number of useful SQL-level data manipulation functions like find the coordinates of a spatial-object column, find the center of a circle-column, find the major and minor axis of an ellipse-column, and setting and updating values of a spatial-object column.
- Number of SQL-level boolean operators for performing queries:
    - Intersect: Test whether two geodetic objects intersect.
    - Outside: Test whether two geodetic objects do not intersect.
    - Inside: Test whether one geodetic object is inside another.
    - Within: Test whether two geodetic objects are within certain distance.
    - Beyond: Test whether two geodetic objects are not within certain distance.
- Support for building spatial R-Tree index on a column of any geodetic type. It also supports different heuristics for organizing the information in the internal pages of the R-Tree.

The GIS vendors use Geodetic DataBlade to build applications on top of SQL support layer, to store and retrieve global geographic data.

Performance of the Geodetic Blade

The performance of the Geodetic DataBlade on the data collection and queries discussed in previous sections will be presented in the final version of the paper.

Conclusions

The indexing methods available in the literature have different characteristics, have different variations, allow for several tunable parameters, and their implementations invariably have certain limitations. The geographic data have various properties in terms of their distribution, size, coupling, and volume. The queries used by GIS applications are of various types.

Therefore to optimize the performance of a Geographic Information System for a particular application, one needs to first understand the behavior of the system for the data used by the application. For example, for an application dealing with data with several overlap, R-Tree indexing will be better, and for an application dealing with data containing ``whole Earth objects'', bounding-box based methods will not work properly, and so on.

Therefore it is very important for a GIS to have support for multiple indexing schemes, and allow for tunable parameters.

Acknowledgements

I would like to sincerely thank all the members of Geodetic DataBlade team, Robert Uleman, Toni Guttman, David Ashkenas, Jim Vaudagna, and Ed Katibah, for all the interesting discussions about the topics in this paper.

Next: References

From GISystems to GIServices: Spatial Computing on the Internet Marketplace

Oliver Günther
Institut für Wirtschaftsinformatik
Humboldt-Universität zu Berlin

Question: Why do people buy a GIS? *Answer:* Because their neighbor has one. Richard Newell of Smallworld Systems told this joke during his keynote speech at SSD'97 - and he did not only refer to Smallworld customers. The truth behind his joke is that GIS are often greatly underutilized. Many customers use only a small fraction of the functionalities offered by their GIS. Some of them are aware of that: they simply do not care about the remaining features. Others are not: they may thus miss functionalities that are actually there and use complicated ways to reimplement them with the features they know. Yet other users may not use their GIS at all: they bought it because they thought it may help them with their problems but then found out that it does not. Some customers may not even have bothered to look: they bought the GIS and left it in the package.

To be fair, this can be said not only about GIS but also about many other types of software. Microsoft, for example, estimates that 90% of Excel's functionalities are used by only 10% of its users worldwide. A large part of requests for new functionalities received by Microsoft each day can be answered simply by telling the customer that the requested functionality already exists. What makes the situation somewhat different for GIS, however, is the relatively high price of a GIS license. GIS come in relatively *large packages*: a single license often costs 2,000 US$ or more, it requires powerful hardware to run on, and it takes considerable training on the customer's part to use the software in a productive way. For most commercial GIS, potential customers face an all-or-nothing choice. Either they invest a relatively large amount to get the license, the required hardware, and some training - or they do not, in which case they get nothing.

We claim that a large number of potential customers in the second category could well become faithful users if they could do so at a smaller entry cost. Of course, the lower ticket price would not buy them the whole license indefinitely. But rather than putting a time limit on the license, as is typically done, vendors should try to tailor their offerings to the specific user requirements. This could mean in particular that the GIS vendor does not sell a classical system license but a ``service'' to perform a set of GIS-typical tasks. Typical services are, for example, a data conversion, a map overlay, some special-purpose spatial analysis, or simply the retrieval of a specific data set.

The service could be performed either at the site of the customer (client-side computing) or at some site run by the vendor or a third party (server-side computing). In the first case, the vendor software would have to be installed at the customer's site for the time of the computation. Hardware requirements remain basically unchanged compared to the traditional licensing process. Training requirements could possibly be reduced depending on the task in question. Nevertheless, the only substantial difference to traditional licensing is the duration and possibly the scope of the license. In the second case, however, customers would simply make their data available to the vendor software and pick up the results once the computation has been performed. No special hardware or training is required on the user's part.

Payment schemes would follow this service-oriented approach. Users just pay for a particular usage of the vendor's software. This would most likely result in a larger number of customers with a lower per capita revenue than in the case of the classical license business. Depending on the application, however, overall revenue could well increase considerably.

A direct consequence of such a shift from Geographic Information *Systems* to Geographic Information *Services* would be the rise of an Internet marketplace [6] for spatial data and services. Anybody with Internet access could act as both a provider and a consumer of related goods.

We recently presented our MMM (Method ManageMent) system, a distributed computing infrastructure that supports the business model and electronic marketplace described above [5]. MMM is a collection of middleware services that facilitate Web-based access to software modules. The idea is that it should be equally easy to post a software module on the Internet as it is to post a Web page. Similarly, it should be equally easy to use such a software module as it is to read a Web page. Some key features of MMM are

- its implementation of stateful services to support the often exploratory nature of spatial data analysis,
- its use of middleware services to enable interoperability between proprietary computing packages, and
- its publishing support facilities to help software authors with their interface definition.

A prototype is available on the Internet at http://mmm.wiwi.hu-berlin.de. A CORBA-based reimplementation is currently in progress [7].

MMM is cooperating closely with two other projects that have related objectives. *DecisionNet* [4][3] is an organized electronic market for decision support technologies. The market infrastructure consists of agents that support consumers and providers in transactions. The decision technologies themselves reside on provider machines distributed across the Internet. DecisionNet is accessible via the World Wide Web, at http://dnet.sm.nps.navy.mil/. *SMART* [2][1] is another Internet marketplace model with an emphasis on spatial data and related algorithms. Like MMM and DecisionNet, SMART is based on the asynchronous communication between service providers and consumers. It offers *query services* to obtain data from a provider, *function services* to model computational tasks (such as conversion between representations), *planning services* to combine and coordinate different tasks, and *execution services* to execute a plan on behalf of a customer.

All of these approaches represent important steps toward an open marketplace for computational services. However, there are still several critical issues to resolve before similar schemes will become commonplace. First, the development of appropriate licensing and payment schemes is still work in progress. It is crucial for the success of Internet marketplaces that service providers can be sure to collect fees from all customers that use their services (directly or indirectly). Second, one needs sophisticated algorithms to encrypt the input data for an algorithm without compromising the results of the computation. I.e., the service provider should be able to perform the service without necessarily having access to the data in unencrypted form. This is not always possible but one should know whether it is the case in a given application. Third, there is the issue of data volume. Large data sets are hard to ship and encrypt; sometimes it may be easier to port the algorithm to the machine where the data resides. Fourth, the usage and configuration of services should not be too complicated. Many GIS vendors pride themselves on the turnkey nature of their systems: setup efforts are minimal, and one can start using the system shortly after purchase. This may not always be the case in a digital marketplace type of situation, where users have to select and combine the services they need.

In summary, many of the functions performed by GIS seem to be amenable to a business model that is fundamentally different from the one we see today. At present, GIS users typically own the hardware and software they use. They pay license and maintenance fees to various vendors and they have to train their staff in using the system. The alternative would be a service-oriented approach where users make their input data available to some GIS service center that performs the necessary computations remotely and sends the results back to the user. Customers pay only for that particular usage of the GIS technology - without having to own a GIS. Our MMM system is one example of a communication infrastructure to support this business model.

- References
- *About this*
- *document ...*

A GIS Interoperability Approach based on ISO RM-ODP and ISO CSMF

Arne-Jurgen Berre, Vidar Knudsen, Jon Oldevik
SINTEF Telecom and Informatics P.O.Box 124, Blindern, N-0314 Oslo, Norway
{ Arne.J.Berre | Vidar Knudsen }@informatics.sintef.no

Abstract

The paper will present and discuss an approach to GIS Interoperability, based on use of the ISO RM-ODP model and the ISO Conceptual Schema Modeling Facility. The practical work on this is being done in the Europan DISGIS projects. The objective of the DISGIS project, running from 7/96 until 12/98 is to provide models, methods, tools and frameworks for the development of interoperable distributed systems in general and interoperable distributed Geographical Information Systems in particular.

1. Integration and Interoperability Problems and Solutions

Basic work in systems integration, such as the ECMA/NIST "Toaster" reference model for integrated environments, provide a good basis for identifying the dif ferent technology areas that needs to be addressed in an integrated system. The "Toaster" model distinguishes 6 different areas: data, processing, communicatio n, presentation, workflow and management. In the context of interoperability g etting two or more systems to work together the notions of technical and semant ic interoperability can be discussed for each area.

The ISO reference model for Open Distributed Processing (RM-ODP) also provides a basis for discussing various aspects of interoperability, as various emphasis is given in its 5 different description viewpoints.

2. Open Distributed Processing (ISO RM-ODP)

The ISO reference model for open distributed processing ISO RM ODP is an int ernational standard that describes an architecture within which support for dist ribution, interoperability and portability can be integrated. ODP standardisati on considers distributed systems spanning many organisations and technological b oundaries. These typically lack any central point of control, and therefore show additional characteristics, such as heterogeneity, autonomy, evolution and mobi lity. In order to deal with these characteristics ODP standardisation aims to en able the building of systems with the following properties: openness, integratio n, interoperability, flexibility, modularity, federation, manageability, provisi on of QoS (Quality of Service), security and transparency.

ODP is defined based on five viewpoints: enterprise, information, computatio nal, engineering and technology. Each viewpoint is an abstraction of the whole s ystem focusing on a specific area of concern. The enterprise viewpoint is concer ned with the purpose, scope and policies of the enterprise. The information view point is concerned with the semantics of information and information processing. The computational viewpoint is concerned with the interaction patterns between the components (objects) of the system. The engineering viewpoint is concerned w ith the design of distribution-oriented aspects, i.e., the infrastructure requir ed to support distribution. Finally, the technology viewpoint is concerned with the provision of an underlying infrastructure.

3. Metadata Interoperability and the Enterprise viewpoint

We will in this section describe meta data interoperability in the context o f federations and different organisational units and information communities, ba sed on the concepts in the ODP enterprise viewpoint.

4. Data Interoperability and the Information viewpoint

We will in this section describe data interoperability based on the concept s in the ODP enterprise viewpoint, and ISO CSMF. Support for semantic data inte roperability is being addressed in a separate position statement being submitted to the workshop associated with the conference.

The purpose of the information viewpoint is to describe the information tha t flows in the system and is processed by the system. This can be captured by se mantic descriptions if objects, their properties, and their relationships. The i nformation viewpoint focuses on the structuring of semantic information, typical ly the information that will be stored in databases and communicated between sys tem components. Traditionally, this modelling have been done according to a thre e-layer schema-architecture, where a schema language (e.g. OMT) is used to descr ibe a class-model which can be instantiated. The ISO CSMF (CD ISO/IEC 14481) hav e introduced a 4-layer schema architecture.

The ISO CSMF (CD ISO/IEC 14481) defines the applicable constructs that shal l be contained in any modelling facility that is used to create a formal concept ual description of various aspects of an enterprise. The purpose of the standard is to provide a mechanism for end users and for information system analysts, de signers and constructors to communicate with each other in a formal way to agree about contents of a conceptual schema.

We will show how the ISO CSMF approach can support both a dynamic API-based specification of schema types (OGC/OpenGIS), as well as a conceptual schema lan guage based approach (ISO/TC211 and CEN/TC287).

5. Processing and Service Interoperability and the Computational viewpoint

We will in this section describe a GIS reference architecture and geoproces sing interoperability, based on concepts in the ISO RM-ODP computational viewpo int.

6. Infrastructure Interoperability and the Technology viewpoint

We will in this section describe infrastructure a interoperability, such as mapping to and interworking between different Distributed Computing Platforms (D CPs).

The intention of the RM-ODP approach is that the information and computatio nal viewpoint can focus on models which are not cluttered by details only necess ary for the mapping to a particular underlying infrastructure, such as CORBA, CO M/OLE, SQL/ODBC, Internet or others. They can thus serve as a common basis for different implementation mappings.

7. GI Standard Interoperability

There is currently many approaches to standards in the GI domain, e.g ISO/TC 211, CEN/TC287, OGC/OpenGIS and others. It is actually an area of concern how t o ensure that these standards might be "interoperable" with each other. We will argue that a common interoperability mapping soon should be established in the a reas of modeling, architecture and terminology.

With the ISO CSMF approach in the information viewpoint we will in particula r show how it is possible to ensure interoperability between the ISO/TC211 and C EN/TC287 approach of using a conceptual schema language, and the OGC/OpenGIS app roach of using an API interface for creating feature types.

8. Conclusions and future work

In this paper we will discuss an approach to GIS Interoperability based on t he use of ISO RM-ODP, including use of ISO CSMF in the information viewpoint.

A practical experiment with this approach is being undertaken in the Europea n DISGIS project. A first pilot case demonstrating this will be finished in Dece mber 1997.

References

[1] "DISGIS - Distributed Geographical Information Systems - White Paper", 3rd EC GIS workshop, Leuven, June 1997 and EUROGI GIS Interoperability workshop, October 1996

[2] A.-J. Berre, J. X. Aagedal, and A. R. Silva, "SIMOD - An ODP-extended Ro le-Modeling Methodology for Distributed Objects," presented at Thirtieth Annual Hawaii International Conference on System Sciences, Wailea, Hawaii, 1997.

[3] "An Object-Oriented Framework for Systems Integration and Interoperability", A.J. Berre, Phd-thesis, University of Trondheim, August >>93

A Middleware for Transparent Access to Multiple Spatial Object Databases

Sang K. Cha, Kihong Kim, Changbin Song, Jookwan Kim, Jooyong Jun and Yongsik Kwon
Knowledge and Data Engineering Lab.
School of Electrical Engineering
Seoul National University
San 56-1 Shinrim-dong Kwanak-ku Seoul 151-742 KOREA

Extended Abstract

The need for accessing multiple datab ases arises frequently in the geograph ic information processing domain becau se a single database may not contain a ll the desired information in the righ t level of detail, precision, and corr ectness. For example, in planning the construction of underground utilities such as gas and electric power lines, it is necessary to access databases of existing and planned utility networks , and such databases are usually maint ained independently by the individual companies. It is also common that th ese databases maintain much of geograp hic information redundantly with diffe rent levels of abstraction, completene ss, precision, and correctness. Many important decision making processes ca n take advantage of such redundancy by combining the content of one database with those of others to match the des ired level of detail, completeness, pr ecision, and correctness.

This paper describes an on-going effo rt of developing SDBC(Spatial DataBase Connectivity) as a middleware for sup porting multiple spatial database acce ss in the client/server environment. A single application programming mode l based on the spatial extension of th e ODMG object database standard provid es application clients with transparen t access to multiple spatial object-or iented database management systems(OOD BMS). To free the programmers from t he burden of knowing the details of in dividual OODBMS implementation archite ctures, SDBC also defines a server lay er that encapsulates the difference an d commonality among OODBMSs. In addi tion, the global transaction managemen t scheme of SDBC ensures the consisten cy of SDBC transactions accessing mult iple databases. Finally, a framework for spatial integrity constraint defi nition and checking is being developed as a part of SDBC to ensure the quali ty of spatial databases. Sponsored by the Korean government as a project of the NGIS(National GIS) D atabase tool development initiative, S DBC has been under development since t he December of 1996 together with SDBX (Spatial DB eXtension), which provides a set of spatial object classes, oper ators, indexing and clustering schemes on top of commercial OODBMSs. Two o ther projects under this initiative ar e devoted to the development of a dedi cated spatial OODBMS. Figure 1 shows the relationship among SDBC, SDBX, and the spatial OODBMS being developed. It also shows how an existing commerci al GIS system can be incorporated on t op of

SDBC.

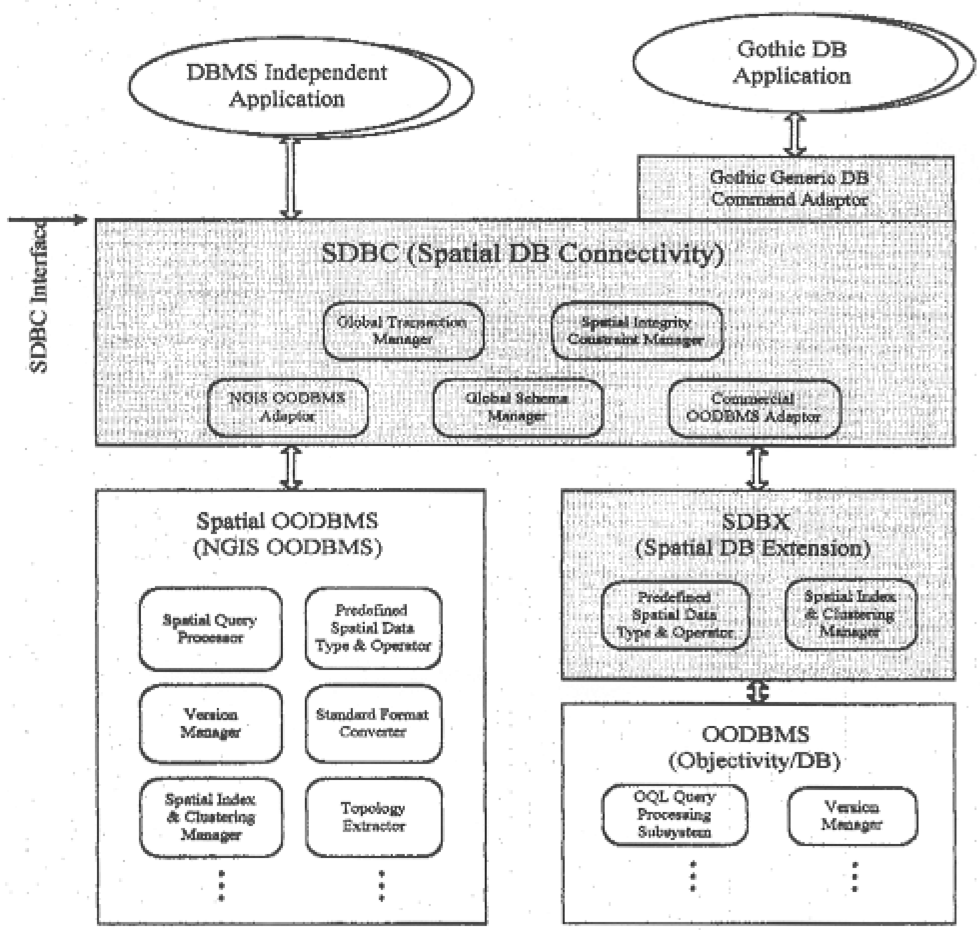


Figure 1. SDBC, SDBX, and Spatial OODBMS in Korean NGIS DB Tool development

Figure 2 shows the layered SDBC archi tecture in the current implementation. It includes the OMG Object Request B roker(ORB) layer for the transparent n etwork-level access. Thus SDBC server s are CORBA object implementations reg istered in the ORB, and the ORB direct s the client requests to these servers . However, the client application pr ogrammer does not have to know the exi stence of ORB. The SDBC client API hides the ORB layer. In addition to SDBC servers, the current SDBC impleme ntation also includes the SDBC coordin ator for maintaining the global schema information and coordinating the glob al transaction management.

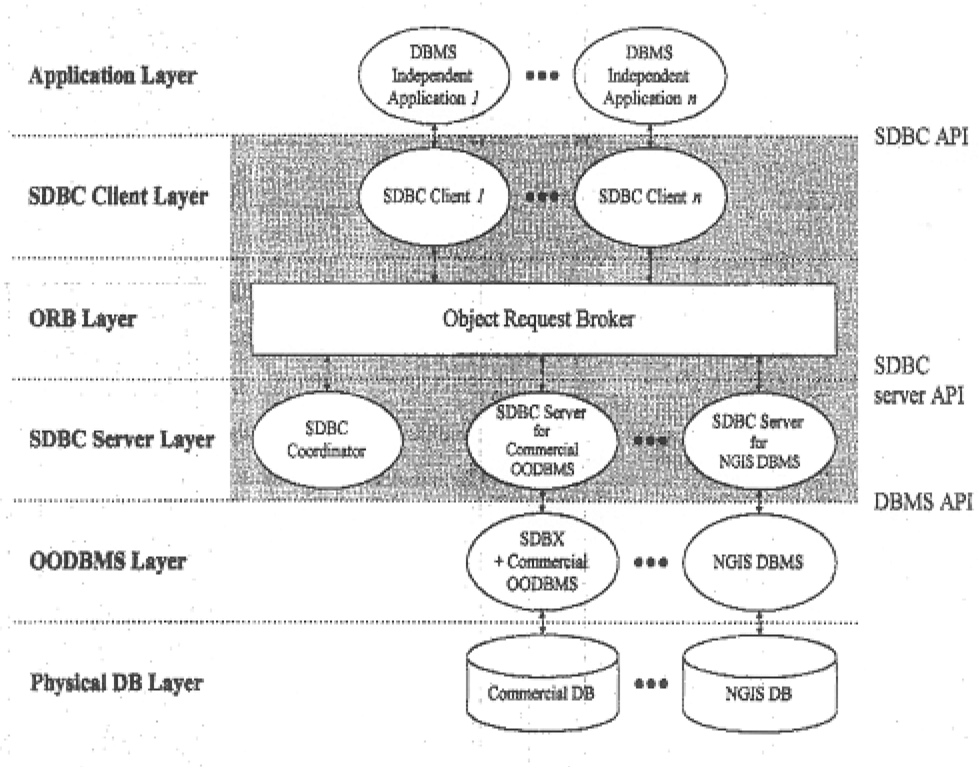


Figure 2. Layered Architecture of SDBC

Although the name SDBC reads similar to ODBC and JDBC, it differs significa ntly from the latter two in the level of abstraction provided to the applica tion programmer. In summary, it prov ides the user with an object-oriented view of spatial databases, global tran saction management, and integrity cons traint checking. At the same time, it shields the user from the burden of k nowing the details of target OODBMS ar chitectural details.

An SDBC prototype is currently being implemented at Seoul National Universi ty in the SUN UltraSPARC environment w ith C++, ORBIX, and SDBX on top of Obj ectivity/DB. A demo application util izing multiple spatial object database s for finding the optimal vehicle navi gation path is also planned to prove t he utility of SDBC.

Constraint-Based Interoperability of Spatiotemporal Databases

Jan Chomicki
Dept. of Computer Science
Monmouth University

Peter Z. Revesz
Dept. of Computer Science
University of Nebraska - Lincoln

Very large temporal, spatial and spatiotemporal databases are a common occurrence nowadays. Although they are usually created with a specific application in mind, they often contain data of potentially broader interest, e.g., historical records or geographical data. By database interoperability we mean the problem of making the data from one database usable to the users of another. Data sharing between different applications and different sites is often the preferable mode of interoperation But sharing of data (and application programs developed around it), facilitated by the advances in network technology, is hampered by the incompatibility of different data models and formats used at different sites. Semantically identical data may be structured in different ways. Also, the expressive power of some data models is limited.

Temporal and spatial databases share a common characteristic: they contain interpreted data, associated with uninterpreted data in a systematic way. For example, a temporal database may contain the historical record of all the property deeds in a city. A spatial database may contain the information about property boundaries. Moreover, as this example shows, spatial and temporal data are often mixed in a single application.

In this research, we propose that constraint databases (Kanellakis et al. 1995) be used as a common language layer that makes the interoperability of different temporal, spatial and spatiotemporal databases possible. Constraint databases generalize the classical relational model of data by introducing generalized tuples: quantifier-free formulas in an appropriate constraint theory. For example, the formula 1950 <= t <= 1970 describes the interval between 1950 and 1970, and the formula ((0 <= x <= 2) AND (0 <= y <= 2)) describes the square area with corners (0,0), (0,2), (2,2), and (2,0). The constraint database technology makes it possible to finitely represent infinite sets of points, which are common in temporal and spatial database applications. We list below some further advantages of using the constraint database technology:

1. Wide spectrum of data models. By varying the constraint theory, one can accommodate a variety of different data models. By syntactically restricting constraints and generalized tuples, one can precisely capture the expressiveness of different models.

2. Broad range of available query languages. Relational algebra and calculus, Datalog and its extensions are all applicable to constraint databases. Those languages have well-studied formal semantics and computational properties, and are thus natural vehicles for expressing translations between different data models. Also, constraint query languages may be able to express queries inexpressible in the query languages of the interoperated data models, augmenting in this way the expressive power of the latter. (This is more a practical than a theoretical contribution. We simply mean that if, for instance, we have a TQuel database, then translation to a constraint database with dense order constraints allows querying by Datalog, a query language which is more expressive than TQuel. Similar comments apply to several other spatial and temporal data models in use.)
3. Decomposability. The problem of translating between two arbitrary data models, which is hard, is decomposed into a pair of simpler problems: translating one data model to a class C of constraint databases, and then translating C to the other data model. Also, by using a common constraint basis, we need to write only 2n instead of n(n-1)/2 number of translations for n different data models.
4. Combination and interaction of spatial and temporal data within a single framework. This is an issue of considerable recent interest, for example in the ESPRIT Chorochronos project.

In this paper we address the issue of application-independent interoperability of spatiotemporal databases. We show that the translations between different data models can be defined independently of any specific application that uses those models. We distinguish between data and query interoperability. For the former, it is the data that is translated to a different data model, while the latter concerns the translation of queries. The constraint database paradigm is helpful in both tasks. For data interoperability, constraint databases serve as a mediating layer and translations between different data models are expressed using constraint queries. For query interoperability, it is the constraint query languages themselves that serve as the intermediate layer. In an actual implementation, the presence of a mediating constraint layer may be completely hidden from the user.

We show below two scenarios in which data interoperability may be useful in practice.

SCENARIO 1: The user of a data model Mod2 wants to query a database D1 developed under a data model Mod1. He translates D1 to a Mod2-database D2 (using constraint databases as an intermediate layer) that he can subsequently query using the query language of Mod2. (As a practical matter, if a user is interested in a query Q2 in Mod2, then only the part of the database that is relevant to the query needs to be translated.)

SCENARIO 2: The user of a data model Mod1 wants to augment the power of the query language of Mod1. For example, this language may be unable to express recursive queries. However, such queries can be formulated in an appropriate constraint query language. Thus whenever the user wants to run such a query on a database D1, he first translates D1 to a constraint database, runs the query in the constraint query language on it (using a constraint query engine), and translates the result back to Mod1. (N.b., interoperating query results is an often neglected aspect of database interoperability.)

We report here on the preliminary results of this NSF-funded research project. We have studied the interoperability between the two-dimensional spaghetti spatial data model (which we believe to be representative of a large class of spatial data models) and linear arithmetic constraint databases. The move to spatiotemporal databases has turned out to be tricky: we are still in the process of defining an appropriate temporal extension of the spaghetti data model. While constraint databases are clearly an appropriate formalism for specifying the translations between different data models, current constraint database engines are too slow to compute the translations. This suggests the need for developing efficient algorithms for data translations, whose correctness can then be checked against the constraint-based specifications.

Interoperable GIS Applications: Tightly Coupling Environmental Models with GISs

Hassan A. Karimi
North Carolina Supercomputing Center
3021 Cornwallis Road
Research Triangle Park, NC 27709

ABSTRACT

The two possible strategies for integrating environmental models with GISs are the loosely-coupled approach and the tightly-coupled approach. The loose coupling of environmental models with a GIS, which relies on the transfer of data files between the GIS and the external modeling programs, though it is the simplest of the two approaches, is inefficient. The tight coupling of environmental models with a GIS, which facilitates means of building the environmental programs within the GIS or building the GIS within the environmental programs, is currently very difficult. Taking a tightly-coupled approach is highly desired by environmental modelers who intend to take full advantage of GISs. This is because through tight coupling, modelers can improve their efforts in building models and in applying their models to real-world problems. Taking a tightly-coupled approach requires that environmental modelers develop their models using GIS programming languages. Programming in a GIS language consists primarily of scripting GIS functions and commands in series. In some instances these GIS languages in GISs can simplify code requirements, but often offer no real advantage to the modeler from a modeling point of view. The effort in developing environmental models generally focuses on limiting the required GIS knowledge of the model user; and on creating a customized, menu driven display environment. Unlike the loosely-coupled approach, in tightly-coupled models no file conversions or intermediate file editing are necessary to run the model. The two primary inhibiting factors in developing tightly-coupled models within GISs are a) translating existing modeling codes into (generally) more restrictive GIS languages and b) the inability of GIS languages to support the same capabilities in more traditional programming languages to accomplish processing tasks. As modeling needs change or expand, modelers may require access to different types of solutions. If new or separate models are needed, the tightly-coupled approach requires the additional investment of writing and developing each new model. Existing numerical routines in environmental models often rely on algorithms designed for the structure of parent languages and are incompatible with GIS languages. Attempts to perform complex environmental modeling have met with little success due to the inability of GIS languages to handle complex algorithms and iterative processes. Despite many advances in programming languages (e.g., C and C++) Fortran remains the preferred programming language by many modelers to develop numerically-oriented models. Compared with the Fortran routine which could

process hundreds of iterations almost instantaneously, the GIS routine takes longer to process each iteration. One reason for such performance discrepancy is that GIS programming languages are interpreters. A compiled GIS programming language in lieu of the slower interpreted programming language could improve processing efficiency. Environmental modelers should have the freedom to develop models in whatever language they feel most comfortable and are most skilled. GIS programmers should be able to take working statistical or numerical codes and incorporate them into systems for environmental modeling without translation. For modelers to integrate with a GIS the integration must be open to multiple data types and formats. Using the GIS as the pre- and post- processor on the model can improve efficiency in developing the integration method and in managing the data . The custom file formatting capabilities in a GIS are typically not capable of handling the necessary preparation so modelers are left supporting a third platform to facilitate the integration. Still, programmers can create GIS applications which run models with the appearance of never leaving the GIS. Data in the GIS must be readily converted into required formats for model input and incorporated easily with non-spatial information. This can currently only be easily accomplished with external programming and only if the model uses ASCII file input. In order to fully integrate with GISs, modelers should be able to build model input data sets and perform simple file editing tasks without leaving the GIS environment. Interoperable GISs can overcome these problems and allow tightly coupling of environmental models with GIS functionalities. Interoperable GISs should support such features as high-speed of data transfer, non-GIS-specificity, and high-level of integration. Speed of data transfer is important as typically many diverse sets of data need to be transferred between GIS functions and models. GIS-specificity refers to the dependency of the integrated outcome on a specific GIS; GIS-specific means that the strategy can work only with a specific GIS, whereas non-GIS-specific means that the integrated outcome can work with different GISs. Level of integration is the degree of synergism between all modules, GIS functions and models. The primary goals of interoperable GISs are to facilitate application development (environmental modeling) activities by providing easy-to-use and flexible tools and to support similar capabilities that are available with programming languages familiar to modelers. For example, the availability of an embedded multi-language, perhaps within an object-oriented environment, will greatly enhance model development efforts. Interoperable GISs will help modelers to better utilize GISs and optimize modeling effort.

Intergrating Environmental Models and GIS in the Framework of GIS Interoperability

Ling Bian
Department of Geography
SUNY, Buffalo 14261-0023

Environmental models, especially process models, simulate the natural process of energy and material exchange over space and time (Maidment, 1993; Kemp, 1993). Connected from a spatial perspective, the integration of environmental modeling with GIS has b een a necessary step in the development of both disciplines. The progress made in the past decade has greatly benefited GIS as an emerging science and environmental modeling as an established discipline. Despite the apparent success, many issues of integr ation remain as obstacles for further, in-depth integration between GIS and environmental modeling. Many of these issues are fundamental, especially at the dawn of Open-GIS era. This paper attempts to address several of these issues under a greater framew ork of geographic information interoperability. As an important user of GIS data and geoprocessing functions, environmental modeling is an inseparable element of the interoperability.

Integrating GIS and Environmental Models

When environmental modeling embraced GIS, it took great advantages of spatial data and spatial analytical tools offered by GIS. With the benefits there came a series of integration issues because GIS and environmental modeling have different scientific f oci and each took unique development route in different evolution time frame. Research in the past has dealt with integration issues at several levels, from simple data translation to more integrated coupling. Various conceptual integration models have be en proposed and implemented (Abel et al. 1994; Chou and Ding, 1992; Nyerges, 1993, Bian et al., 1996). However primitive or sophisticated, these integration efforts focus on sharing data without addressing the incompatibility of the basis of GIS and the m odels.

The incompatibility between the two systems is beyond the issue of data format. Because the development of environmental modeling peaked prior to the development of GIS, environmental model development had to cope with the lack of effective spatial means by taking simplified assumptions of spatial variation. In the GIS era, many such models inevitably under-used the rich content and capability of GIS because the limitations in using spatial data. However, a more critical issue lies in the simplified spat ial assumptions the models use. The spatial assumptions are intertwined with other assumptions such as temporal assumptions. Conceptually, it is possible to create a more flexible basis to adjust the incompatibility between the model assumption and the sp atial data. Practically, overcoming this incompatibility is not an easy undertaking partially due to the structure of most process models.

Environmental Models

Similar to monolithic GIS packages, many environmental models are closed, stand-alone systems. They require fixed input/output data format with virtually no capability to interface with other systems. Many of these models are still in batch mode without modularized structure. Modifications to a model are often handled as patches added to the main body of code developed decades ago. Such a tightly wrapped structure may have been the primary cause that limited the integration to the level of "coupling", in stead of a full integration.

In contrast to the weak spatial component, process models are well developed in simulating physical processes and handling temporal variation. Process models focus on clearly identifiable entities or phenomena. The processes taken by or posed upon the en tities and phenomena are represented as mathematical functions. The state of the entities and phenomena, as initial conditions and especially the results of the processes, is the ultimate interest of the modeling. The dynamic nature of the processes is im plemented by the state change of the entities and phenomena over a series of explicit time steps. Spatial locations of the entities and phenomena, however, may not play an active role in the dynamic process.

Current GIS Systems

Unlike process models, geographic locations are explicitly represented in GISs and they form the conceptual and structural basis of GIS data system. All other properties of the entities and phenomena are attached to the locations. However, such a system is largely static with virtually no capability to handle temporal changes. This spatial-temporal incompatibility between GIS and process models requires a more revolutionary change in the basis of both systems.

Further from the issue of spatial-temporal incompatibility and closer to the core of process modeling is the dynamic nature of the processes. In addition to change in time, entities move (wildlife) and so do many phenomena (precipitation, at a proper sca le). Neither current GISs nor the process models can explicitly represent the spatial dynamics of the processes. While process models have a weak spatial component, current GISs are too rigid to accommodate frequent location changes.

Object-Orientation

In searching for a more effective model to represent the dynamic world, Peuquet and Duan (1995) proposed a system that is based on time. Changes to the location or other properties are attached to the framework of a time-line. Among many proposed systems , the approach that is repeatedly proposed as a better solution for modeling the dynamic world is the object-oriented design, Tang et al. (1996) proposed a system based on geographic features, in which the semantic feature objects form the basis of the sy stem. Geographic locations are properties of the geometric objects that are encapsulated by the semantic features. A similar design philosophy was presented by Takeyama and Couclelis (1997) although applied to a cellular automata system.

Raper and Livingstone (1995) outlined another design for modeling natural processes. The design bases the representation of real world on form, process, and material objects. Geographic location and time are treated as properties of the objects. Both fea ture-based or time-based designs allow easier handling of spatial and temporal dynamics of the entities or phenomena. These designs, and especially the ones that focus on features in dynamic progress is particularly appalling to modeling processes. This d esign is consistent with the vision of Open-GIS (Buehler and McKee, 1996).

Object-orientation is perhaps the most effective framework that can house both GIS and process models in one compatible system. With this framework, the entities and phenomena of interest to process models form the essential objects. The objects are rela ted through associations. Geographic location and time are the properties of the objects. The resultant easy update of location and time allows an effective simulation of spatial and temporal dynamics. The processes can be explicitly implemented as events that lead to state change of an object. Issues such as incompatibility in data resolution, spatial-temporal handling, and dynamic simulation can be adjusted with flexibility within this framework.

From an implementation perspective, object-orientation supports re-use of object libraries, effective spatial-temporal query, easy interfacing with visualization, and flexible customization. These technical advantages support the realization of component ware, a concept and practice that is foreseen as the future form of Open-GIS, as well as for future environmental modeling.

References

Abel, D.J., and Kilby, P.J. (1994) The systems integration problem. International Journal of Geographical Information Systems, 8(1):1-12.

Bian, L., Sun, H., Blodgett, C.F., Egbert, S.L., Li, W., Ran, L. & Koussis, A.D. (1996) An integrated interface system to couple the SWAT model and ARC/INFO. Proceedings, the Third International Conference on Integrating GIS and Environmental Modeling, Sa nta Fe, New Mexico (January 21-25, 1996) World Wide Web and CD ROM publications.

Buehler K. and McKee L. (1996) The OpenGIS Guide. http://www.opengis.org, The Open GIS Consortium, Inc.

Chou, H.-C., and Ding, Y. (1992) Methodology of integrating spatial analysis/modeling and GIS. Proceedings, 5th International Symposium on Spatial Data handling, Charleston, South Carolina, 514-523.

Kemp, K. (1993) Environmental Modeling with GIS: A Strategy for Dealing with Spatial Continuity. Technical Report 93-3, National Center for Geographic Information and Analysis, Santa Barbara, USA.

Maidment, D. R. (1993) GIS and hydrologic modeling. in Environmental Modeling with GIS, Goodchild, M. F., B. O. Parks, and L. T. Steyaert (ed.) Oxford University Press, New York.

Nyerges, T. (1993) Understanding the scope of GIS: its relationship to environmental modeling. in Environmental Modeling with GIS, Goodchild, M. F., B. O. Parks, and L. T. Steyaert (ed.) Oxford University Press, New York.

Peuquet, D.J. and Duan, N. (1995) An event-based spatialtemporal data model (ESTDM) for temporal analysis of geographical data. International Journal of Geographical Information Science, 9: 7-21.

Raper, J., and Livingstone, D. (1995) Development of a geomorphological spatial model using object-oriented design. International Journal of Geographical Information Systems, 9(4): 359-383.

Takeyama, M. and Couclelis, H. (1997) Map dynamics: integrating cellular automata and GIS through geo-algebra. International Journal of Geographical Information Science, 11: 73-91.

Tang, Y., T.M. Adams, and E.L. Usery (1996) A spatial data model design for feature-based geographical information systems. International Journal of Geographical Information Science, 10(5): 643-659.

Spatial Process Modelling and Interoperability

Andrew J. Marr, Stephen MacDonell, and George L. Benwell
Spatial Information Research Centre,
Department of Computer and Information Science,
University of Otago
PO Box 56
Dunedin, New Zealand

Recent developments in GIS interoperability, particularly those emerging from the work of the Open GIS Consortium, are likely to result in significant improvements in the utilisation of spatial data resources. Current research into the development of OpenGIS interfaces which enable interoperability among heterogeneous spatial databases will offer fundamental improvements over traditional methods and techniques.

Parallel to the development of interoperability, is the increased demand for research into Spatial Process Modelling Systems (SPMS). SPMS represent the integration of GIS analysis with the functionality found in non-spatial simulation modelling tools (e.g. Stellar and ExtendOE). This is sometimes referred to as Geographical Modelling Systems (GMS). The integration of the two systems is difficult to achieve, and example applications to-date are typically too specific to the component subsystems and narrow in focus. A set of important emerging research themes in this area focus on the desire for generic and interactive visual construction tools that facilitate collaborative process model development. Further work is also required into modular hierarchical structures, and modelbase management systems that permit successful embedding of model structures in the form of composite model components.

For a flexible SPMS, there is an inherent requirement that the simulation of spatial modelling events occur using a variety of data sources, usually derived from a diversity of heterogeneous data formats. With interoperability, the restrictions caused by fixed proprietary vendor data formats are removed. In addition the capability is developed for querying of desired data subsets for retrieval. These attributes of interoperability highlight the virtual dependence of generic SPMS on such technology in a collaborative environment.

A fully integrated SPMS is a system that encompasses existing GIS principles and techniques but also includes extensions for additional spatial process modelling functionality. These additional simulation modelling requirements include consideration of feed back systems, support for temporal dynamic modes, equation generation, and model calibration and validation. In addition the complexity of some process model definitions may become too overwhelming. Some authors have asserted the need for the incorporation of AI and agents to guide users through the processing steps.

In this paper, a conceptual framework for the collaborative sharing of models describing spatial process structures is presented. The proposed functionally independent modules in the context of SPMS are constructed in the form of reusable services. These services are: graphical model design, model interpretation and implementation, data transformation, spatial operations, and visualisation. This set of services are viewed as the principle components required for basic system operation. It is assumed that other specialist components will also be included as required (e.g. DTM and network analysis). Each service is designed to be replaced as new technology isintroduced with minimal impact on the existing system.

The graphical model design service is composed of a graphical interface in which the user may visually construct the required spatial process model. The model is then written to a process modelling language design file. This generated file may then be exchanged and integrated as sub-components in other designs. Once a process model is complete it will be sent to the interpretation and implementation service. This service will then act on the design file and call supporting services such as data transformation and spatial operations as required. The current proposal for data transformation is to use the feature manipulation engine (FME) software which facilitates powerful interoperable functionality between diverse systems. While this software is suited to back-end mapping operations, there are limitations in the types of data that may be utilised and in the processing capabilities. When the objectives of the Open GIS project are realised, it is hoped that this service may be enhanced. To enable cross-platform portability, the intended system and graphical interfaces are being designed in Java.

The aim of the project is to develop a tool for integrated spatial process modelling, that is in-line with the with the identified research goals in this area. The proximity of objectives between Interoperable GIS and SPM are noted and seen as mutually supportive. The objectives of both concepts stress the desire to develop distributed `plug and play' tools that manipulate spatial data of variable formats. The incorporating of interoperability into SPMS is crucial if maximum flexibility and functionality of the system is to be achieved. In addition,the development of this system is also seen as providing methods for collaborative research.

Interoperability with the Earth Science Remote Access Tool (ESRAT)

Robert Raskin and Elaine Dobinson
Physical Oceanography Distributed Active Archive Center (DAAC)
Jet Propulsion Laboratory
Pasadena, CA 91109

The Earth Science Remote Access Tool (ESRAT) is an http-based client-server application that facilitates internet access to distributed Earth Science raster data. ESRAT addresses the variety of data formats used in the Earth Sciences by providing a common data model with translators for the models inherent in many standard formats. It also enables applications normally using local data in a particular format to access remote datasets in that or other supported formats.

The client side of ESRAT features a Java applet GUI that allows the user to specify spatial/temporal regions of interest. A query to the server returns a summary of datasets satisfying the search criteria, displayed visually as layers that can be interactively selected. Selected data subsets are retrieved by the server and loaded directly into an application package (MATLAB, in this case) by invoking a helper application in a web browser. Spatial/temporal subsetting is carried out on the server side prior to data transmission, reducing bandwidth requirements. Remote data from multiple sources and in multiple formats can be combined readily into a single MATLAB script.

The server side includes master directory, dataset catalog, and data access servers, implemented as C++ CGI programs. The master directory contains a list of dataset holdings at local or remote sites and their associated URLs. The dataset catalogs contain the spatial/temporal bounds of each data subset; currently, data in swath, grid, and point network models are supported. For swath data, each cross-track is individually cataloged in space and time. When a user requests a spatial/temporal subset of swath data, the server concatenates any contiguous cross-tracks satisfying the selection criteria, and returns a swath polygon to the client.

ESRAT currently is built on the Distributed Oceanographic Data System (DODS) developed at the University of Rhode Island and the Massachusetts Institute of Technology [1]. DODS provides the capability to:

- ingest data residing in one of several supported formats (HDF, netCDF, MATLAB, DSP and JGOFS)
- translate from the data model of the supported format to the DODS data model consisting of arrays, grids, tables, and structures.
- establish a network connection to carry out an http-based data transfer
- translate back to a data model and format expected by the client.

To add a new supported format, translation between the format's data model and DODS's data model must be provided. One of the lessons learned in the development of ESRAT was that 100% interoperability between all formats is non-trivial. However, partial interoperability is sufficient for most Earth Science applications, as certain translation combinations are not likely, in practice.

In the near future, we expect to convert the entire tool to Java. Java provides an unprecedented level of interoperability because both its bytecode and its internal representation of numbers are platform independent. This feature permits Java's data model (in terms of arrays, classes, and streams) to be used as the intermediate data model (when enhanced with class libraries developed for spatial data types). This approach will increase the number of supported clients and servers by leveraging on the work of software vendors developing Java translators/interfaces. Java has a further advantage of allowing users to package executable code as metadata that can accompany the dataset. The code might be used to properly geolocate, subset, interpret, or analyze the data.

We also plan to convert our catalog databases to a commercial object/relational database system (ORDBMS) with spatial data type extensions. This will provide greater GIS functionality, including better support for spatial operations on swaths. A global quadtree representation of the catalog entries is being designed to provide efficient global searches when the system is expanded to include large volumes of data.

This work is being funded by the EOSDIS Prototype Office and is ongoing.

Reference:

[1] Gallagher, J. and G. Milkowski, "Data Transport Within the Distributed Oceanographic Data System", 4th International World Wide Web Conference Proceedings, 691-700, 1995.

The Geospatial Interoperability Problem: Lessons Learned from Building the *GeoLens* Prototype

Clifford Behrens
Bellcore, MCC 1J-314R
445 South Street
Morristown, NJ, USA 07960-6438

Leon Shklar
Pencom Web Works
40 Fulton St.
New York, NY, USA 10038

Chumki Basu
Bellcore, MCC 1E-322B
445 South Street
Morristown, NJ, USA 07960-6438

Nancy Yeager
NCSA
Univ. of Illinois
Urbana-Champaign, IL, USA 61820

Edith Au
Pencom Web Works
40 Fulton St.
New York, NY, USA 10038

1. Introduction

In 1994 NASA issued a Cooperative Agreement Notice to support new research on digital library technology that would enable broader public use of its earth science data over the Internet. As a response to this CAN, the Universal Spatial Data Access Consortium (USDAC) was formed and it proposed to prototype the *GeoLens System* that would not only give broader public access to NASA's earth observation data, but also made these data interoperate with other geospatial data served by the Federal government. Part of the challenge of the *GeoLens Project* has been to decompose the larger "geospatial interoperability problem" into constituent issues. This paper will address these issues and describe solutions implemented in our *GeoLens* prototype. The purpose of this exercise is to present one possible end-to-end use case, beginning with geospatial data discovery and ending with conflation of geospatial data extracts from extremely heterogeneous sources. The larger goal is to demonstrate the manner in which this use case might be better supported by new information processing standards and innovative digital library technology.

At the moment there exists growing demand for robust geospatial information services, ones capable of federating massive repositories of distributed heterogeneous data and metadata, and of integrating data extracted from these repositories. For example, recent BAAs issued by the National Imagery and Mapping Agency (NIMA) have sought commercial catalog services for federating geospatial data in its archives (e.g., DMA 1996, NIMA 1997). Also, the Open GIS Consortium (OGC) recently approved a set of implementation specifications to support access of simple geospatial features, and issued an RFI, concerning geospatial catalog services, as part of its process to develop a consensus-based interoperability specification that can be implemented by commercial GIS vendors and others (OGC 1996). Evidence of this demand can also be found in the increasing activity within Federal (e.g., FGDC and NASA EOSDIS) and International (e.g., ISO TC211) standards groups to define geospatial metadata content standards as part of mission-critical information infrastructure needed to implement National and Global Geospatial Information Infrastructure frameworks.

While it would seem desirable to have everyone adopt a single metadata standard to catalog their image and map data holdings, this is not realistic. In fact, already the FGDC metadata content standard (currently in Version 2) has undergone several revisions in response to comments from user communities (FGDC 1994, 1997), NIMA has proposed its own metadata standard which extends the FGDC and CIO-SPIA (CIO 1995) schema to better support its users (NIMA 1996), and Hughes has published a revision (Release B) of its EOSDIS Core System metadata standard which also extends the NASA Global Change Master Directory's (GCMD) Directory Interchange Format (DIF) (NASA 1997) and FGDC schema for NASA earth science data users (Hughes 1996). There exist other similar schema and extensions (e.g., ISO 1997 and the US National Biological Service 1997). Hence, we assert that (1) *data schema evolution is inevitable and even "healthy"* and, therefore, *(2) federation-enabling geospatial information infrastructure must be extremely flexible and adaptive!*

The core capability of the *GeoLens System* is to provide a viable solution to federating distributed, heterogeneous geospatial data and metadata by supporting multiple dynamic standards. This approach is entirely consistent with the notion of "Information Communities", as introduced in the *OpenGIS User's Guide - Abstract Specification*, and the desirability of supporting each community's "standardized" view on data while also providing translations between these views (Buehler and McKee 1996). As an illustration, this paper will examine the manner in which the current *GeoLens* prototype implements the FGDC and EOSDIS metadata content standards, the FGDC Spatial Data Transfer Standard - Vector Profile (SDTS-VP) and Hierarchical Data Format-EOSDIS Core System (HDF-EOS) data archive types, and other de facto and incipient data and information processing standards (e.g., the Open Geospatial Interoperability Specification) to achieve greater interoperability. We will further discuss how our implementation of these standards serves to achieve greater interoperability.

2. Use Case Scenario

Let us suppose that a Planning Engineer in the Morris County Park Commission wants to create a USGS Level I land use/land cover classification (see Anderson et al. 1976) for Morris County, NJ. Currently, the Park Commission has only historical black-and-white aerial photography for the county, so this engineer wishes to locate all available multispectral satellite imagery that covers Morris County. Since the county contains areas of relatively steep relief in the northwest and southeast, this engineer believes that digital elevation data for the same area could be used to improve the classification of image pixels obtained from an unsupervised classification. This engineer has ISDN access to the Internet and a workstation with a WWW browser, and so wants to use the Web to obtain the satellite imagery and topographic data. Since Morris County forms part of the New York Metropolitan region and is heavily trafficked, he also wants to acquire county boundary and transportation features to overlay on his land use/land cover layer. Once these data are obtained, the engineer can better determine the amount of the forested park land managed by the MCPC that is located in mountainous areas. His search and retrieval of data proceeds as follows:

(1) Because the engineer doesn't have handy a map from which to determine coordinates, he begins his search for useful data by drawing a bounding box around a boundary line graph of Morris County presented in his browser. Since he is also interested in identifying data objects whose content relates to "New Jersey" and the purpose of "land cover classification," he enters these character strings in a form provided by his client. The bounding box is used to query the footprint metadata attributes, the placename is used to query placename keyword metadata attributes, and the purpose character string is for querying full-text indices, of <u>all</u> geospatial catalogs in the earth science data federation.

(2) An *Object ID* is returned for each geospatial data object whose footprint intersects the one drawn by the engineer <u>or</u> whose content relates to New Jersey <u>and</u> the purpose of land cover classification. In this scenario, four *OIDs* are listed: two Landsat ETM+ (Enhanced Thematic Mapper) objects, a USGS 1:250K DEM (Digital Elevation Model) object, and a USGS 1:100K DLG (Digital Line Graph) Transportation object.

(3) The engineer browses metadata for one of the ETM+ objects and discovers that it contains 90% cloud cover. While browsing metadata for the other, he learns that only 20% of it is covered by clouds. He inspects a browse image for this second ETM+ object and finds that most of the clouds are located outside the area containing Morris County.

(4) Satisfied that the second ETM+ image is useful for making his classification, he creates an order to extract bands 3, 4, 5 and PAN but <u>only</u> those rasters needed to fill his bounding box. This ETM+ data access order is stored in a "shopping cart" object.

(5) In a similar manner, the engineer browses metadata for the DEM and DLG objects returned from his catalog query to confirm that they also cover his area of interest. In the case of DLG objects, the user is also presented with a list of DLG-3 features (US DOI 1987, 1990) that are supported by data in the object, and he selects those features for which he wants data. Then he creates orders to extract data from these DEM and DLG objects, again storing with each order his bounding box coordinates, and in the case of the DLG object a list of DLG-3 feature codes, so that he obtains only the data required for his application. The DEM and

DLG data access orders are also added to the shopping cart.

(6) The ETM+, DEM and DLG data access information stored in the shopping cart is sent to networked Data Access Servers. The three extracts are retrieved and stored locally for use by the engineer's image classifier and GIS.

This scenario provides several distinguishing features. A user is provided with a consistent, unified view over a federation of distributed, heterogeneous geospatial catalogs. This user need not be aware of the location (or even the existence) of individual catalogs in the federation; they merely appear as a single local source of metadata. The catalog services make available both content-descriptive and access-descriptive metadata. The latter references data access services, which are needed either to acquire additional metadata or to extract information from data objects referenced by the catalogs. User queries are recursively applied over individual catalogs that are linked within a collection tree. Furthermore, a single query is distributed over all (or some subset) of catalogs in the federation without the user having to be cognizant of the metadata standard used to organize the catalog, and consequently of which queries are appropriate. To support this type of seamless query, an infrastructure is required that maintains multiple, dynamic metadata content standards within a single catalog hierarchy, along with translation or mapping services between catalogs at query and data extraction time. In addition to attribute-based SQL queries, full-text indices on specific attributes within a catalog may also be queried and then the combined results of these two query methods presented to the user.

3. The Geospatial Interoperability Problem

What makes geospatial data difficult to use? The use case scenario above exposes numerous issues that contribute to the overall "geospatial interoperability problem." These issues and their solutions in the *GeoLens System* are represented in the pyramid shown in Fig. 1.

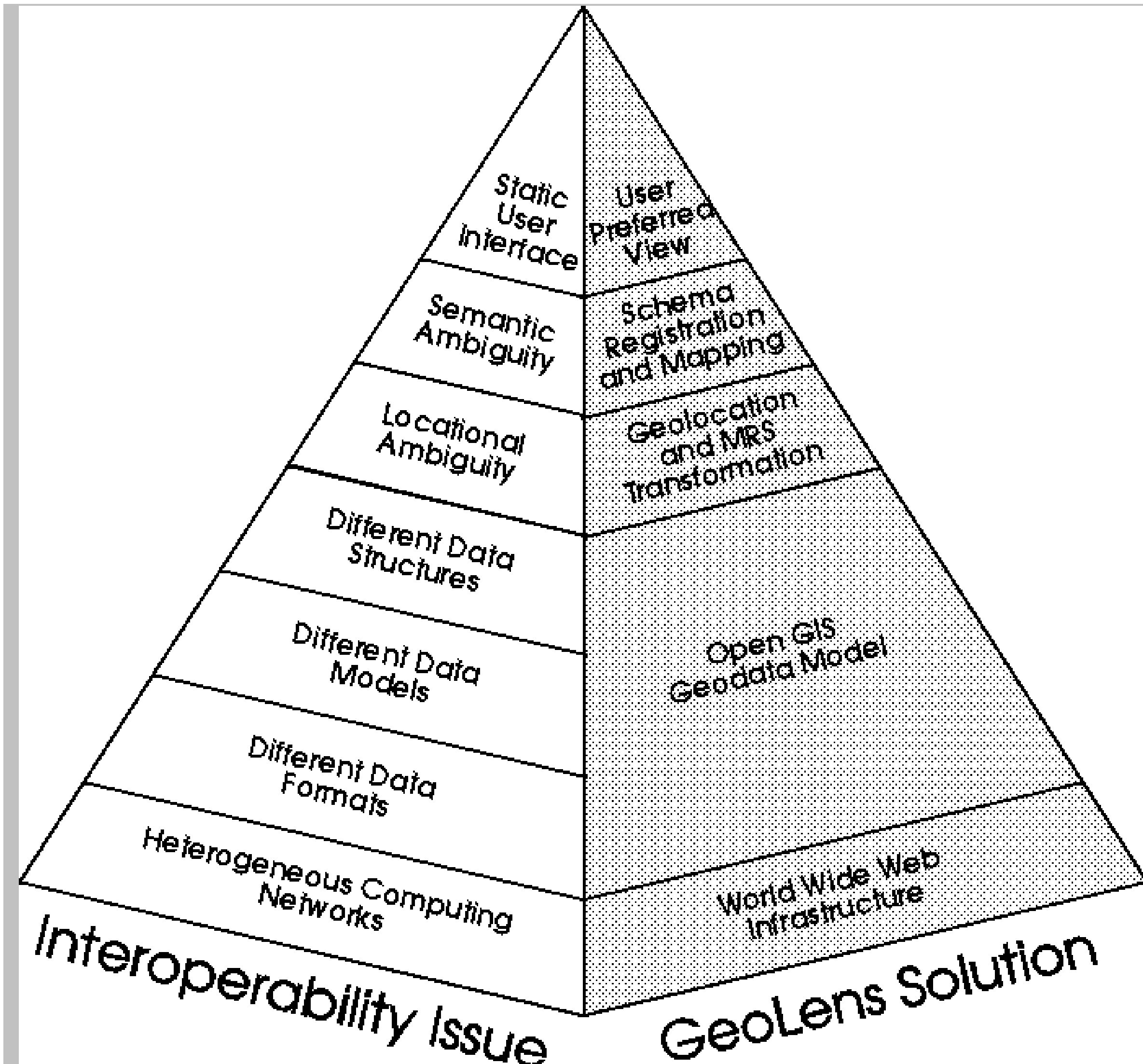


Fig. 1. Geospatial interoperability issues and their solutions as implemented in the *GeoLens* prototype.

3.1 <u>Heterogeneous computing environment</u>

Geospatial data are created and exist in extremely heterogeneous computing environments. For example, a variety of hardware and software platforms are employed to serve geospatial data by federal and commercial providers; and users have their own hardware and software to access and process these data. To neutralize the effect of this heterogeneity, we have implemented our *GeoLens System* on Web-Internet infrastructure, principally exploiting *http* (Hypertext Transfer Protocol) for client-server communications.

3.2 <u>Heterogeneous data (and metadata)</u>

Data are often acquired in *different formats* which may make them inaccessible to an application. For example, absence of *CR/LF* as end-of-record delimiters in DEM data is a common problem for those GIS applications that require these delimiters. Other incompatibilities may be attributed to the use of *different models* for representing a geospatial theme (Burrough 1986, Clarke 1990). For example, elevation data may be modeled as grids, e.g., a DEM, or vectors, e.g., a DTM (Digital Terrain Model). To complicate matters further, the same model may store data in *different structures*. For example, if the data are of type vector, then are they stored as TIGER, SDTS-VP, ARC or some other structure? These problems are exemplified in the use case above. The ETM+ imagery is of type raster stored in HDF-EOS containers (Fingerman 1997), the DEM data are type grid stored in DEM format (as flat ASCII files), and the DLG data are of type vector stored in SDTF-VP (Spatial Data Transfer Standard-Vector Profile) distributions (US DOI 1995). We address these issues in *GeoLens Data Access Services* by implementing OGIS-like interfaces which wrap the native file formats and access libraries as OGIS-like objects and translate data to OGIS-like formats.

Support for multiple schema standards introduces other heterogeneity in metadata. Existing metadata content standards have different topologies for their attribute representations, and these may even be extended with the introduction of new standards. To account for metadata evolution in a uniform manner, the *GeoLens* encapsulates metadata describing content into hierarchies of attributes and their values. These attributes are further grouped according to whether they describe individual data objects or collections of data objects.

3.3 Locational ambiguity

Many uses, such as the county-level planning application alluded to in the use case, require only a subset of a geospatial data object, e.g., a subscene of a Landsat image, to provide coverage for an area of interest. So there is also the need to efficiently discover all of the current image, topographic and cartographic data available for an area, browse the metadata for these data to determine their usefulness, and retrieve only the highest quality data required to cover the study area but in a form, and with sufficiently accurate georegistration, so that they can be used together by a classifier.

Thus, another serious problem that can make it difficult for geospatial data to interoperate is *locational ambiguity*, due to poor (or no) georegistration. This is often the case with imagery whose areal extent or "footprint" may be crudely approximated to support discovery, but whose registration may be so inaccurate as to make its use with other georectified data inappropriate. A less severe problem exists when geospatial data are registered to different Map Reference Systems, e.g., the 1 Degree DEMs mentioned in the use case are registered in Geographic Coordinates while the ETM+ imagery is registered to the Universal Transverse Mercator grid. *GeoLens Data Access Services* solve this problem by requiring geolocated data, but also provide translations between Map Reference Systems, when necessary (DMA 1983, Snyder 1987).

3.4 Semantic ambiguity

Semantic ambiguity exists when different meanings are associated with the same term (*polysemy*), or when different terms mean the same thing (*synonymy*). For example, "floodplain" may have different meanings to a civil engineer who views this feature as an area that may need protection from inundation, and an insurance claims adjuster whose notion may only include the area of financial liability. Terms used to attribute meaning to geospatial objects, and the semantic relationships between terms, may be expressed formally as schema (MacEachren 1995, Medyckyj-Scott and Hearnshaw 1993). In the use case above, the FGDC and ECS metadata content standards as well as the DLG-3 feature schema provide useful examples.

The use case scenario requires satellite imagery, topographic information and other cartographic data. New Landsat 7 Enhanced Thematic Mapper (ETM+) data will soon become available from the USGS EROS Data Center, as are USGS 1 degree and 7.5 minute Digital Elevation Models (DEMs), and USGS 1:100K Transportation DLGs. However, the catalogued metadata for these data types implement different standardized metadata schema. The ETM+ collection is managed by the EOSDIS Core System (ECS) which organizes its metadata compliant with the ECS Metadata Standard; while both the USGS DEM and DLG metadata are catalogued compliant with the FGDC Content Standard for Digital Geospatial Metadata. In addition, the DLG data are attributed with the DLG-3 feature schema.

The design of the *GeoLens System* is driven by the need for semantic interoperability through better metadata support. We recognize that (1) metadata schema represent particular views and capture the unique semantics of different groups, or "Information Communities" (cf. Buehler and McGee 1996) of geospatial data users, (2) multiple schema exist for describing the content of geospatial data, and (3) these schema will evolve over time as geospatial data are increasingly distributed and as users become more specialized and sophisticated in their applications of these data. Much of the support for multiple schema and schema translation resides in the *GeoLens Catalog Server*.

3.5 Static, non-tailored information presentation

Finally, the manner in which information about the content of geospatial data is presented to a user can significantly affect their understanding of how it might be used (Hearnshaw and Unwin 1994, Medyckyj-Scott and Hearnshaw 1993). There currently exist many map and image browsers on the Web that allow a user to browse geospatial metadata. However, most of these present a static view of metadata; the labeling and formatting of information about a geospatial data object does not adapt well to data content or the preferred view of the user. By supporting multiple schema, we can exploit a variety of "views" to drive an interactive presentation by the *GeoLens Client*, i.e., metadata may be organized and tagged consistent with a different schema than was employed by the data provider who catalogued these metadata.

4. GeoLens Solutions

Figure 2 illustrates the distributed architecture of the *GeoLens System*. It includes a *GeoLens Client*, a *GeoLens Catalog Server*, a *Schema Mapping Server*, *Data Access Servers* and, potentially, other servers to process geospatial data or their metadata (Shklar et al. 1997).

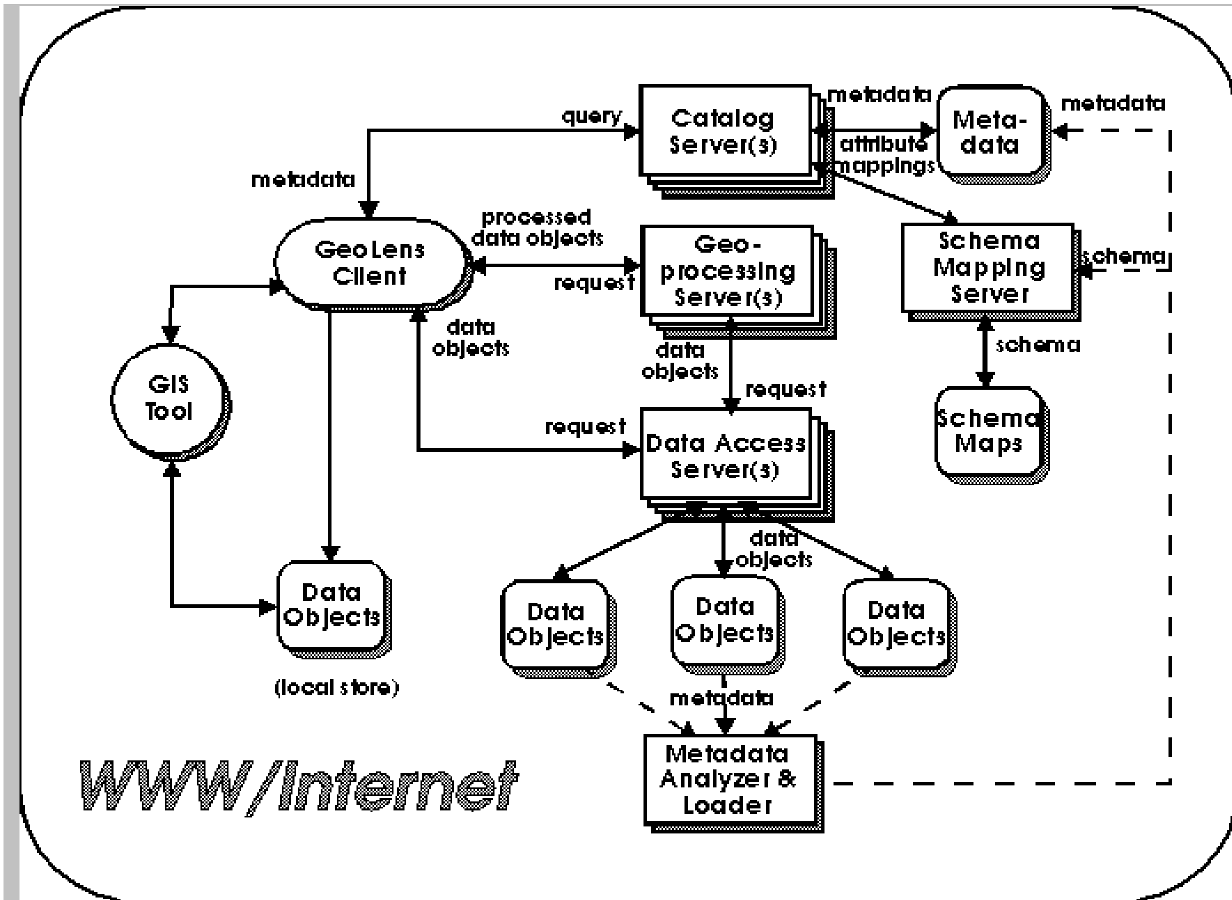


Fig. 2. *GeoLens* system architecture.

4.1 Client and Graphical User Interface

The *GeoLens Client* is implemented as a framework of powerful Java applets that exploit the unique full-feature capabilities of *GeoLens Catalog Services*. Using the server side support of multiple schema to drive the browser side client, makes it possible to achieve greater customization of the presentation. The overall effect is a graphical presentation of metadata that preserves both the structure and semantics of a user's preferred schema or metadata content standard. Queries of *GeoLens Catalogs* may be spatial or formed by combining any of a query schema's attributes with logical operators. Specific features of our client are described below in greater detail.

Fig. 3. Collection tree representation of *GeoLens* federation.

Soon after a *GeoLens Catalog* is accessed, a user is presented with a geospatial data *Collection Tree* by the *GeoLens Client*. As shown in Fig. 3, the tree, represented in indented-outline form, displays classes of data objects, e.g., "Digital Elevation Data Collection" or "Satellite Imagery and Aerial Photography," and subclasses of these, e.g., AVHRR and Landsat for the latter class. A user can follow down branches of this tree to collections of data objects, e.g., Landsat Thematic Mapper or Multispectral Scanner, by clicking on nodes in the tree to open their subtrees, eventually reaching leaves containing these collections. Having selected either a collection or leaf-level object, the user can browse the collection-level or inventory-level metadata, which may include encapsulated data ranging from plain text to a browse image, if one exists for the target data object, illustrated in Fig. 4. The *GeoLens Catalog Service* processes requests based on encapsulation attributes, and so the same text or images may be presented differently depending on their encapsulation type. Thus, the *GeoLens Catalog Service* does not perform any format conversions of the original information. Instead, metadata are passed directly to the *GeoLens Client*, which either presents them directly, or uses them to retrieve the information.

Fig. 4. *GeoLens* metadata browser.

Special facilities exist to easily query *GeoLens Catalogs* spatially with a user-supplied bounding rectangle. The coordinates for this rectangle may be defined in several ways: interacting graphically with a map, a dialog box or search on a placename with the USGS Geographical Name Information System Gazetteer service. The last method also nicely demonstrates the manner in which the *GeoLens Client's* framework easily provides plug-in support for third-party, external services. Attribute schema may also be queried to aid users in defining queries. A user can profile a preferred schema and all or just some of its attributes for building queries of *GeoLens Catalogs*. At the moment, a simple dialog box displays a list of a query schema's attributes from which a user may select member attributes and appropriate conditions to form complex queries that may include conditions for searching full-text indices.

Other facilities are provided in the *GeoLens Client* to help a user navigate through the *Collection Tree*. A presentation history is cached so that a user may easily browse collection-level and inventory-level metadata by traversing the *Collection Tree*.

4.2 Catalog Services

The *GeoLens Client* locates geospatial data on the Internet through a particular *GeoLens Catalog Server*, which also serves as a transparent proxy for other *GeoLens Catalog Servers*.

Metadata loading and analysis routines extract metadata descriptions and build the catalog. These metadata are analyzed for their schema and their properties are verified for compliance to that schema, by a *Schema Mapping Service*. Finally, the metadata are added to a catalog which is currently implemented as an *O2* database (O2 Technology 1995). (Our design makes it very easy to substitute other commercial products). A user launches a search for geospatial data by querying the contents of metadata catalogs with the help of one or more *GeoLens Catalog Servers*, even though the use of multiple catalogs is completely transparent to the user. Catalogs may be queried for the schema used to structure and document metadata for a collection of data objects, or for any of the metadata attributes used to catalog data objects. Queries may be spatial, temporal or composed of conditions on arbitrary metadata attributes used to catalog data objects. Since a query may be issued to one or more catalogs, each with metadata that might be organized consistent with different metadata schema, the query processor may contact the *Schema Mapping Service* to translate between schema with different attribute names and structures. This feature enables a recursive search of all (or some) of the catalogs known to *GeoLens Catalog Servers* in the federation.

4.3 Data Access Services

Once a list of candidate data objects has been returned to the *GeoLens Client*, a user can select objects from this list for data retrieval. The *GeoLens Client* sends a message to a *Data Access Server* with the ID of a data object and a user-provided Minimal Bounding Rectangle applied by the *Data Access Server* to clip the data object. The data extract is remodeled as an *OpenGIS-like Well-known Structure* (WKS) and returned to the *GeoLens Client* where it may be stored locally, or immediately exploited by an application, e.g., a commercial GIS, that implements *OpenGIS* interfaces. In addition, the *GeoLens* architecture allows for integration of other services, e.g., geoprocessing or map production, that may be requested by the *GeoLens Client* or *Data Access Servers.*

5. Lessons Learned

As might be expected in any prototyping effort, numerous technical obstacles arose during the project, particulary ones involving the integration of geospatial information processing standards. Table 1 lists the standards leveraged in the *GeoLens* system, their type and application. This section will discuss the manner in which some of the most severe difficulties in implementing these standards were resolved in the design and implementation of *GeoLens*.

Standard	**Type**	***GeoLens* Implementation**
FGDC CSDGM	metadata content	DEM & DLG Catalog Servers
ECS Metadata Standard	metadata content	MSS Catalog Server
DLG-3	metadata content	DLG Catalog and Data Access Servers
HDF-EOS	data archive format	MSS Data Access Server
DEM	data archive format	DEM Data Access Server
SDTS-VP	data archive format	DLG Data Access Server

http	messaging	client-server communications
Common Gateway Interface	messaging	client-server communications
Java Development Kit 1.0.2	development language	distributed software applications
gif	image format	browse image metadata
jpeg	image format	browse image metadata
OGIS-like *Grid Coverage*	well-known structure	DEM & MSS Data Access Servers
OGIS-like *Simple Features*	well-known structure	DLG Data Access Server
RDF/XML (see text)	schema representation	Schema Mapping Service (planned)

Table 1. Standards integration in the GeoLens prototype.

5.1 Catalog Federation and Schema Integration

The approach to federating heterogeneous catalogs within *GeoLens Catalog Services* assumes that metadata are just like actual physical data and need not be stored together in the same physical repository. Using the same mechanism that enables the *GeoLens Catalog Server* to encapsulate physical location and processing information for different data objects, information about the network location of catalogs may be stored as *access-descriptive attributes*. In this way, catalogs in part (or in their entirety) may be referenced by other catalogs, each residing on separate servers. This not only demonstrates the flexibility of catalog access and presentation, but also the generalizability of the *GeoLens* encapsulation mechanism.

As the use case revealed, not only should the physical existence of individual catalogs be hidden from a user, true federation of heterogeneous catalogs implies that users are able to navigate through different catalogs without being burdened with the complexities of querying separately each of these catalogs for pertinent information. Such "transparency" touches on the key issue of *semantic ambiguity*. It is not reasonable to assume that any single metadata schema can capture the full extent of meaning held by the name of any given metadata attribute. Therefore, some mechanism is needed to resolve ambiguous situations. Because there are many different types of these situations, it has been useful to elucidate cases when schema translations are required, from most simple to most difficult.

Perhaps the simplest situation requiring cross-schema mapping of metadata attributes is one-to-one attribute naming translations. For example, translation of FGDC bounding box coordinates to their counterparts in the ECS schema follows below:

```
        Identification_Information:
                Spatial_Domain:
                        Bounding_Coordinates:
```

```
                                East_Bounding_Coordinate=
```

maps onto

```
        SingleTypeCollection:
                Spatial:
                        SpatialDomainContainer:
                                HorizontalSpatialDomainContainer:
                                        BoundingRectangle:
                                                EastBoundingCoordinate=
```

In situations like these, it is sufficient to know which attribute names in two schema carry the same meaning.

The next level of complexity might involve one-to-many attribute translations. A slightly contrived example of this type might be:

```
        Spatial_Domain:
                Bounding_Coordinates:
                        Southeast_Pair:
```

maps onto

```
        SingleTypeCollection:
                Spatial:
                        SpatialDomainContainer:
                                HorizontalSpatialDomainContainer:
                                        BoundingRectangle:
                                                SouthBoundingCoordinate=

        AND

        SingleTypeCollection:
                Spatial:
                        SpatialDomainContainer:
                                HorizontalSpatialDomainContainer:
                                        BoundingRectangle:
                                                EastBoundingCoordinate=
```

In these situations not only are attribute names different between two schema, but the values assigned to several attributes in one schema may correspond to only a single attribute in another schema.

Still a third level of complexity would involve translations of meaning for different attribute values, rather than just the attribute names themselves. Some of these may only require simple kinds of conversions, e.g., between coordinates expressed as DD.MM.SS (Degrees, Minutes and Seconds) and DD.DD (Decimal Degrees), or Geographic Coordinate to UTM Grid conversions. While other kinds of translation can be extremely difficult such as negotiation between a property owner's notion of the term "parcel" and the one held by a county tax assessor.

Current approaches to dealing with semantic interoperability of heterogeneous schema usually rely on some form of human input to resolve conflicts. The *GeoLens* approach exploits key attribute mapping information from schema providers. We further define concepts that are primed for specific meaning (e.g., consider the concept of a "bounding box"), and the constituent elements necessary to define a concept (e.g., what minimal amount of information describes a bounding box). This core information is made available to the *Catalog Server* and is used at query time both to retrieve permissible mappings between schema standards, and to fill the necessary "slots" associated with a given concept.

In the course of our work, we have come to appreciate the complexity of this problem, especially with regard to achieving a general solution. The derivation of new concepts requires an understanding of the domain in question and each concept entails additional knowledge which would need to be "prebuilt." Even simple one-to-one attribute name mappings raise interesting questions. For instance, are the mappings transitive? In other words, suppose that two schema standards share no registered or standardized mappings, but there is a third schema which maps to both of them. At the present, this kind of transitivity is not implemented in the *GeoLens* system. However, as the analysis of Semantic Translation issues reveals, there are different levels of complexity to semantic translation, and much can be accomplished by at least solving the simplest situations as we have begun to do in the *GeoLens*.

5.2 Representation of Schema Standards

An important objective of *GeoLens* was to provide a proof of concept for supporting multiple schema standards instead of trying to enforce a single one. An additional complication is a *lack of uniformity* in representing standards. Until recently, the standard-setting activities concentrated exclusively on defining syntax and semantics of individual standards but not on defining a common representation for these standards. As part of the project, we have defined the most important characteristics of such a representation, where a standard is composed of attribute specifications that include *data types*, *applicability*, *generality*, *topology* and *extensibility* (Shklar et al. 1997). Attribute data types determine processing of query conditions and serve to support schema verification (type mismatch is likely to represent an inconsistency). Examples of data types include strings, integers, and geospatial coordinates.

Attribute applicability determines whether an attribute is mandatory, mandatory-if-applicable, or optional, while attribute generality determines whether it belongs to the common part of a standard or to a named extension. If an attribute is defined as optional for the common part of a standard, it may still be defined as mandatory or mandatory-if-applicable for one or more named extensions. If the attribute is not defined for the common part of a standard, its specifications for different named extensions don't have to match. A mandatory-if-applicable attribute may be further characterized by a list of other attributes, the presence of which would change its status to mandatory.

Attribute topology is defined by specifying the component-of relationships with parent attributes. Extensibility is only defined for composite attributes and determines whether all their child attributes (or components) are already known. This is important because we are considering two sources of information for constructing a schema standard: *schema standard specifications* and *incoming metadata entities*. If a composite attribute is marked as non-extensible, an encounter of its unknown components is considered an error.

In the absence of a common approach, we have invented a proprietary standard representation syntax, but we consider it only an interim solution. Our hopes are with the *World Wide Web Consortium* which has initiated several standard-setting activities around the so-called *Resource Description Framework* (RDF). We are strongly encouraged by the working documents that emerged from this body and stand ready to comply to RDF specifications when they stabilize (Lassila and Swick 1997). Already the richness of the RDF model seems sufficient for expressing geospatial schema. RDF specifications utilize the XML syntax, which is of course a promising common idiom for such specifications.

5.3 Geoprocessing Services

The initial design of the cooperation between the *GeoLens Metadata Browser* and an external service, such as the *Data Access Server,* is rather simple -- the *GeoLens Metadata Browser* invokes an external service based on the encapsulated information of a single data object. However, there are situations where a more complex higher-level layer is required to process multiple data objects. For example, a user may want to exploit network services to create a visualization by overlaying a DLG object on a DEM object, each extracted by different data servers. Alternatively, this user may want to extract data and store them locally for use by his image classifier and GIS. These operations require the *GeoLens Metadata Browser* to send multiple data objects to an external service (e.g., a visualization service). While not currently implemented, our new design will utilize a "shopping cart" to collect and send multiple data objects to external geoprocessing services capabile of handling multiple requests.

5.4 Distributed Applications

A Java-enabled Web browser is by far the most common "platform" used to access the Internet. It accommodates the *GeoLens Metadata Browser,* which is implemented as a framework of Java applets using JDK (Java Development Kit) 1.0.2, to deliver geospatial metadata and data to users. Although Java applets can run across platforms and web browsers, we found instances of inconsistencies. Our experience with JDK 1.0.2 can be described as "write once, test everywhere! (and repeat)" Moreover, the AWT (Abstract Window Toolkit) provided by JDK 1.0.2 is not powerful and flexible enough to implement the desired GUI design. We are currently evaluating the impact of migrating the *GeoLens Metadata Browser* to JDK 1.2, which includes an enhanced GUI widget set called JFC (Java Foundation Classes) for developers to create professional looking applications. We anticipate that Java will become more mature and stable, and hope for the standardization of Java Virtual Machines utilized by HTTP browsers.

5.5 Security

Two obstacles made it difficult to implement the *GeoLens* as a seamless, distributed system: firewalls and the Java security "sandbox."

Firewall. As mentioned earlier, the *GeoLens* is a distributed system. *Catalog Server*, *Metadata Browser* (the client), and other external *Data Access Servers* are working together on an open network. Currently, communications between the *Metadata Browser* and the *Catalog Service* are socket based. A socket based connection may fail depending on the severity of firewall restrictions that exist between the client and the server. Unfortunately, Internet firewalls are getting more restrictive by the day. To address this problem, the next version of our system will provide an *http*-based connection to support users who wish to access a *GeoLens Catalog Server* from behind restrictive firewalls.

Java Security "Sandbox." Java applets loading across an open network are considered "untrusted," and should only be running inside a restricted environment known as a "sandbox." While the Java sandbox model protects a client machine from being attacked by malicious applets, it is also highly restrictive. For example, it cannot access any system resources (e.g., the file system) or communicate with any other machines except the one sending the applet. Although these restrictions are vital, they have introduced obstacles to implementing our design. The *GeoLens Metadata Browser* cannot directly invoke a *Data Access Service* unless it is running on the same machine that is serving our browser. While it is possible to host both the metadata browser and the data access service on the same machine, it is not desirable. Such a solution would limit our system to exploit only data access services on a particular machine. Currently, we are using a combination of light-weight CGI (Common Gateway Interface) gateways and proxy programs to work around the sandbox restrictions. The proxy program is running on the machine where the Metadata Browser originates, and acts as an intermediary between the browser and a service located on another machine. This is not a preferred solution although it works out well in our prototype. We are currently looking into solving this problem with signed applets, a Java feature introduced in JDK 1.1. An authenticated signed applet may run in a less restricted environment where it is permitted to communicate with external services directly.

5.6 Data Model Implementation

A key objective in the *GeoLens Project* has been to develop proof-of-concepts for ideas expressed in OpenGIS documents and Working Groups. While we have taken primarily a breadth-first approach to support the end-to-end scenario above, nonetheless we have learned some important lessons related to current limitations of the OGIS data model when processing and retrieving very large geospatial data objects. For example, while the OGIS data model provides API's (Application Programming Interfaces) for requesting data, the model stops short of suggesting how data should be returned to an application. This leaves open the possibility for many different implementations of the model. Moreover, in a distributed (and potentially low-bandwidth) environment such as that provided by the Internet, it is not feasible for a client application to invoke fine-grained operations (such as those specified for OGIS features) on an object that resides on a remote data server. Instead, a transfer syntax is required for transmitting a representation of the object (or part of it) to the client, allowing the client applications to operate directly on a copy of the object. At the moment, *GeoLens Data Access Servers* return only small data extracts to the client in the form of *gifs* and flat *ascii* files for demonstration purposes. But "industrial-strength" data access services will need to do a better job of packaging data extracts for Internet transfer.

Other implementation issues exist regarding the location and support of massive collection-level metadata such as feature schema. While one might conclude on logical grounds that the distinction between collection and inventory-metadata is arbitrary, such a distinction may be critical to make for performance reasons. For example, the DLG-3 feature schema is needed to support both data discovery and access; yet it is too large to transfer between catalog servers, clients and data access servers to support these functions. Moreover, we discovered that only a small subset of features were actually supported by data in any particular DLG data object, and the size of this subset actually varied from object to object. Our manner of addressing this issue in the *DLG Data Access Server* was to store the DLG-3 feature schema as an *Entity_and_Attribute_Information.Detailed_Description* attribute in FGDC collection-level metadata (though one could alternatively store a reference to a DLG-3 feature schema service). The DLG-3 features actually supported by each DLG data object were precomputed and stored as other inventory-level metadata on the *Data Access Server*. This information is used to build an interface for users so that they select for extraction only those relevant features in a data object discovered by the *GeoLens Catalog Server*. This approach provides the user with copies of the feature schema for reference and access purposes, but minimizes the need for transporting feature schema information. It also enables a data provider to update their DLG repository in a timely manner without having to also update feature schema entries in the DLG catalog.

This last point touches on another significant accomplishment of the *GeoLens Project*, i.e., developing a relatively inexpensive approach to federating data repositories, but one that also permits maximum autonomy among data providers and encourages their participation. Our system allows independent data providers to participate in the federation with very few "hooks." There is no requirement that their metadata need map to any single preexisting metadata content standard. This allows data providers to make available as many (or as few) mappings as desired. While the advantage of having a common schema "interlingua" permits multiple standards to "communicate" with one another, deriving and enforcing widespread adherence to such an "interlingua" is currently neither practical nor desirable. Instead, the understanding supported here is that the data provider has the flexibility to choose those mappings which are more readily accessible and meaningful, and these would be supported by *GeoLens*.

6. Conclusions

As this paper has shown, the geospatial interoperability problem actually consists of many smaller, but in themselves, extremely complex problems. We have used the *GeoLens Project* to better expose these problems, analyze them in detail, and experiment with different software approaches to solving them. Among the most significant lessons that we have learned over the last three years is that data modeling is an approach to federation that offers both the power and flexibility required to ensure the autonomy desired by data providers, the seamless accessibility to data required by users and the potential for technology to evolve with changes in the requirements of these communities. Finally, we believe that our work has demonstrated the importance of geospatial information processing standards to solving geospatial interoperability issues, and the manner in which one might design and implement sophisticated catalog and data access services on the Internet to more effectively support multiple, dynamic geospatial data standards.

7. References

Anderson, J. R., E. Hardy, J. Roach, and R. Witmer. 1976. A land use and land cover classification system for use with remote sensor data. *U. S. Geological Survey Progessional Paper 964*.

Buehler, K., and L. McKee (eds.). 1996. *The OpenGIS Guide: Introduction to Interoperable Geoprocessing. Open Geodata Interoperability Specification (OGIS), Part I.* Wayland, MA: Open GIS Consortium, Inc.

Burrough, P. A. 1986. Principles of geographical information systems for land resources assessment. *Monographs on Soil and Resources Survey, No. 12*. Oxford: Clarendon Press.

Central Imagery Office. 1995. *Standards Profile for Imagery Access, Version 2 (December 8)*. CIO-2020. Vienna, VA: Central Imagery Office.

Clarke, K. C. 1990. *Analytical and Computer Cartography*. Englewood Cliffs, NJ: Prentice Hall.

Defense Mapping Agency (DMA). 1983. *Geodesy for the Layman (DMA TR 80-003)*. Washington, D.C.: Defense Mapping Agency.

Defense Mapping Agency (DMA). 1996. Global Geospatial Information & Services (GGI&S) and Data Architecture and Gateway Services (DAGS). Solicitation released on June 27, 1996.

Federal Geographic Data Committee. 1994. *Content Standards for Digital Geospatial Metadata (June 8)*. Federal Geographic Data Committee. Washington, D.C.

Federal Geographic Data Committee. 1997. *Content Standards for Digital Geospatial Metadata, Version 2.0 (Revised April)*. Federal Geographic Data Committee. Washington, D.C.

Fingerman, P. W. 1997. *HDF-EOS 2.00 Version Description Document (VDD) for the ECS Project, Version 1.00 (814-RD-009-001)*. Upper Marlboro, MD: Hughes Information Technology Systems.

Hernshaw, H. M., and D. J. Unwin. 1994. *Visualization in Geographical Information Systems*. New York: John Wiley and Sons.

Hughes Information Technology Corp. 1996. *Release-B Science Data Processing Segment (SDPS) Database Design and Database Schema Specifications for the ECS Project*. Document No. 311-CD-008-001.

International Standards Organization (ISO). 1997. *Geographic Information - Metadata - Version 2.0*. ISO/TC211 Working Group 3. Working Document No. 1997-01-20.

Lassila, O., and R. R. Swick (eds.). 1997. *Resource Description Framework (RDF) Model and Syntax*. WWWC Draft Specification (WD-rdf-syntax-971103). http://www.w3.org/Metadata/RDF/Group/WD-rdf-syntax/.

MacEachren, A. M. 1995. *How Maps Work: Representation, Visualization, and Design*. New York: Guilford Press.

Medyckyj-Scott, D., and J. M. Hearnshaw (eds.). 1993. *Human Factors in Geographical Information Systems*. London: Belhaven Press.

National Aeronautics and Space Administration. 1997. *Directory Interchange Format (DIF) Writer's Guide, Version 5.0a*. NASA GSFC, Global Change Data Center, Code 902.

National Biological Service. 1997. *National Biological Information Infrastructure Metadata*

Standard. National Biological Service. Washington, D. C.

National Imagery and Mapping Agency (NIMA). 1996. *Geospatial Metadata: DoD Geospatial Data Standardization Project Report, Vol. 3*. (Sept. 16). National Imagery and Mapping Agency.

National Imagery and Mapping Agency (NIMA). 1997. Geospatial Information Integrated Product Team (GI IPT) Geospatial Information Infrastructure GII 97 Requirement - SOL BAA. *Commerce Business Daily*, January 17-23, pages A-6 to A-8.

O2 Technology, Inc. 1995. *Technical Overview of the O2 System*. Technology Technical Report No. 9. Palo Alto, CA: O2 Technology, Inc.

Open GIS Consortium. 1996. Request for Information: OGIS Catalog Service Interfaces. OGC Request 3, Open GIS Services Working Group - August 30, 1996. Wayland, MA: Open GIS Consortium.

Shklar, L, C. Behrens, C. Basu, N. Yeager, and E. Au. 1997. *New Approaches to Cataloging, Querying and Browsing Geospatial Metadata*. Paper presented at the 2nd IEEE Metadata Conference. NOAA, Silver Springs, Maryland, Sept. 16-17, 1997.

Snyder, P. 1987. Map Projections, A Working Manual. *U. S. Geological Survey, Geological Survey Professional Paper 1395*. Washington, D. C.: U. S. Government Printing Office.

U.S. Department of the Interior, U. S. Geological Survey. 1987. *Digital Line Graphs from 1:100,000-Scale Maps --Data Users Guide 2*. Reston, VA.

U.S. Department of the Interior, U. S. Geological Survey. 1990. *Standards for Digital Line Graphs, Part 3: Attribute Codes*. Reston, VA.

U.S. Department of the Interior, U. S. Geological Survey. 1997. *DLG-3 SDTS Transfer Description, Draft (May 23, 1995)*. ftp://sdts.er.usgs.gov/pub/sdts/datasets/tvp/dlg3/.

ACKNOWLEDGEMENTS

This work was funded by NASA's Information Infrastructure Technology and Applications (IITA) initiative of the High Performance Computing and Communications (HPCC) program, Cooperative Agreement NCC5-102. The authors would like to thank Nand Lal, Mike Lesk, Ed Runnion, Ronald Underwood, Kent Wittenburg, Eric Sigman, Gomer Thomas, Charles St. Charles and Tim Muradian for their contributions to the GeoLens project.

INTEROPERATING GEOGRAPHIC INFORMATION SYSTEMS USING THE OPEN GEOSPATIAL DATASTORE INTERFACE (OGDI)

Paul Morin
J2 Geomatics
615 Booth Street, Ottawa, Ontario, K1A 0K2

Denis Gouin
Defence Research Establishment Valcartier (DREV)
2459 Pie-XI Blvd. North, Val-Bélair, Quebec, G3J 1X5

Gilles Clément and Christian Larouche
Logiciels et Applications Scientifiques (L.A.S.) Inc.
1600, Boul. St-Martin Est, Tour A, bureau 650, Laval, Quebec, Canada, H7G 4R8

ABSTRACT

Interoperability of spatial systems is a goal sought by many and, by most measures, attained by none. If systems were truly interoperable, it would be straightforward to share geographic information across software platforms and between databases collected for different purposes by different organizations. In contrast, a variety of technical, institutional, and physical barriers inhibit the fluid exchange of geographic information.

One of the main problems in today's spatial or geographic information management framework is geospatial data conversion and integration. Very often, GIS developers and users need to import geospatial data from different sources. This task has proven to be difficult and time consuming. Industry experts believe that 60% to 85% of the total cost of implementing a GIS is data conversion. Geospatial data products are offered in a large variety of different and incompatible formats, possibly in different coordinate systems or cartographic projections. Typically, GIS software developers have developed their own proprietary geospatial data format and use ethnocentric translators to convert foreign geospatial data formats into their own format. Consequently, data suppliers have to develop versions of geospatial data products for several software packages.

Geospatial data format standardization is one solution to this problem. However, it is very unlikely that the industry will move to a single standard. At least half-dozen important standards can be expected besides all the proprietary commercial data products already gaining momentum in the marketplace. This means that

standardization alone will not solve the geospatial data conversion and integration problem by itself.

This paper proposes a solution, the Open Geospatial Datastore Interface (OGDI), a dynamically loadable C language application programming interface (API) that provides a standardized method through which a GIS software package can access a variety of geospatial data products. OGDI uses a client/server architecture to facilitate the dissemination of geospatial data products both locally and over any TCP/IP network, and a driver-oriented approach to provide access to many geospatial data products and formats.

OGDI provides tools to convert various formats into a uniform transient data structure, to adjust coordinate systems, cartographic projections and platform-dependent data representations, and to retrieve geometric and attribute data, all "on the fly." It can access a growing number of geospatial data products and formats, and it is tailored to use the Internet as a medium to distribute geospatial data products.
The transient data structure supports both geometric and attribute data. Geometric data is divided into vector (area, line, point) and raster data (line or tile access). Metadata used includes geographic regions and coverage, cartographic projection and sources.

Drivers are used to access various geospatial data formats, one for each format. A driver is also a dynamically loadable library that processes C language API requests for a specific datastore. Once a driver is loaded, it receives requests, fetches information from the datastore, translates it into the uniform transient data structure and returns the results to the application. All the APIs are available for UNIX operating systems (Solaris and Linux) and for Microsoft's Windows NT and Windows 95 operating systems.

Drivers can be accessed either directly, for local datastores, or remotely, for external datastores. To improve the exchange of geospatial data over the Internet, a new stateful protocol GLTP (Geographic Library Transfer Protocol) has been developed to replace the stateless HTTP (Hyper Text Transfer Protocol). With stateless protocols, such as HTTP, each query (or call to a method) is processed independently such that only one server is sufficient to communicate with several users. For OGDI, a protocol with a "persistent memory" was necessary to handle successive related queries or geospatial transfers. In contrast to HTTP, there is a server process for each client.

This new protocol has been integrated into a small utility program called gltpd (geographic library transfer protocol deamon) that mimics the behavior of the C language API on a remote computer. The combination of the gltpd process and a specific driver becomes a server to the client (i.e., the application's connection). For a programmer using OGDI, there is no difference between a local driver and a remote one. The gltpd process takes care of the communication protocol transparently and automatically transforms data between incompatible processor architectures. In the current implementation, the gltpd process is based on the ONC RPC 4.0 protocol.

OGDI drivers are already available for several geospatial formats such as VRF/VPF, ADRG, CADRG, CIB, DTED, CCOGIF, CEOS, Oracle SDO, GRASS, PAMAP, Arc/Info, DGN, DWG, DXF, USGS DLG-3 and USGS DOQ. Drivers will soon be developed for other geospatial formats such as GeoTIFF, Mapinfo MID/MIF and Imagery (TIFF, GIF, PCX, BMP, etc.).

To interact with a datastore, a simple application goes through a sequence of steps. First, it establishes a connection or creates a "client," the term used to describe each instance of a connection. Second, it selects a geographic region. Then it selects a map layer and extracts objects from it, either sequentially or randomly. It processes the results and finally terminates the connection.

Each connection between the application (a client) and a driver (a server) is defined by an ASCII string similar to the Uniform Resource Locator (URL) used by the World Wide Web. Each string is prefixed by "gltp" (analogous to the URL prefixes "http" or "ftp"). The prefix is followed by a host name for remote driver access, a driver descriptor and then a file path-name that indicates the location of the datastore. The general form is:

```
gltp:[//<hostname>]/<name of driver>/<pathname> .
```

The hostname is not used when accessing a local datastore.

The use of OGDI in the GRASSLAND GIS software package is demonstrated, highlighting direct access to several geospatial data formats, including GRASS, VRF, ADRG and DTED. Considerations about future OGDI development such as writing new drivers, applying spatial analysis services to geospatial data accessed through OGDI in its native format, improving remote communications, adding encryption facilities and authentication services are discussed.

Finally, the strong interest shown in OGDI from two important international committees: the ISO/TC 211 (International Standards Organization/Technical Committee 211) and the OGC (Open GIS Consortium) is briefly outlined.

OGDI and its source code are freeware in order to improve the interoperability of GIS software.

GEOLIB :a software component for making GIS tools interoperable

Donatas KVEDARAUSKAS, Patrice BOURSIER
Laboratoire d'Informatique et d'Imagerie Industrielle (L3I) Universite de La Rochelle
Avenue de Marillac
17042 La Rochelle - FRANCE

Xavier CULOS, Thierry DELTHEIL, and Sylvie IRIS
SILOGIC
Toulouse, France

Introduction

The aim of the GEOLIB project was to develop a library of geographic functionalities, portable and interoperable with application development environments, database management systems, multimedia/hypermedia "authoring systems" as well as external GIS, and also usable in an Internet/Intranet environment.

This software component gives the possibility to integrate more easily the spatial dimension within existing interactive multimedia applications, as well as office or management applications. They allow to manipulate maps, itineraries, but also to address spatial queries to existing spatial databases, without having to know how to use GIS tools.

GEOLIB has been developed in the framework of a contract funded by the french agency for technology transfer between research and industry1. It has been developed in collaboration with the Computer Science Research Laboratory at the University of Paris-Sud (LRI).

Functionalities

A first release of the GEOLIB component has been launched in September 1997, which contains the following set of functionalities :

Data acquisition and handling

- Import of a vector or raster map layer and associated descriptive non-spatial data,
- Creation of a vector map layer from a drawing layer,
- Creation of a map from the superposition of raster and/or vector layers,
- Map updating (insertion, deletion, layers reorganization),
- Conversion vector <-> raster,

- Map projection and coordinate systems.

Exploitation and querying

- Vector layer indexing,
- Non spatial queries,
- Simple spatial queries, based on distance, adjacency, intersection and inclusion.

Display and visualization

- 2D map display,
- Physical zoom on raster or vector layer,
- Physical zoom on raster or vector layer (access to more detailed maps),
- Panning,
- Management of links between spatial and multimedia data (creation, modification, deletion).

Geographic data processing

- Map overlay,
- Length and area measurement (lines, polygons),
- Distance between objects,
- Buffer zone management.

Architecture

GEOLIB is made of several modules, in order to be easily portable and interfacable with other software tools, and usable in a client-server (Internet/Intranet) environment :

- the Gateway module, whose functions are the following :
  - import existing geographic data files,
  - load data within object structures in main memory, in order to be processed by the other modules of GEOLIB.
- the Dispatcher module :
  - analyse and pre-process queries,
  - determine the optimal strategy for answering the query, depending on the processing and result of previous queries :
    - access data stored on the client,
    - access data stored on the server,
    - extract data from files stored on the server,
  - retrieve data,
  - process data and prepare the answer.
- the Visualization module :
  - translate user interaction into queries,

address the query to the Dispatcher module,
    - display results on the client.
- the Core module :
    - reception of queries addressed by the Dispatcher module,
    - execution of basic geographic functionalities,
    - transmission of results to the Dispatcher module.
- the Extensions module :
    - reception of queries addressed by the Dispatcher module,
    - execution of specific geographic functions,
    - transmission of results to the Dispatcher module.

Query processing

Three different scenarii have been identified and considered for query processing :

1. *Query processed on the client*

The simplest way of processing a query is the local mode :

1. following a user interaction, the Visualization module on the client creates a query object that is sent to the Dispatcher module,
2. the Dispatcher module analyses the query and concludes that it can be processed locally. Consequently, the query is sent to the Core module on the client,
3. the Core module takes charge of the query processing,
4. once the query has been processed locally by the Core module, the answer is sent to the Visualization module for being displayed.

2. *Query processed on the server*

The processing of a query may require access to the GEOLIB server. The execution scenario is then the following :

1. the Visualization module translates a user interaction into a query object, and sends it to the Dispatcher module,
2. the Dispatcher module on the client analyses the query and concludes it must be sent to the server for further proocessing, either because all required data are not present on the client, or because functions required to process the query are not available on the client. Consequently, it initiates a communication between the client and the server, and the query is sent to the server for being processed,
3. the GEOLIB server receives the query and restructures it, before sending it to the Dispatcher module on the server,
4. the Dispatcher module analyses the query and decides which processing module must be called, either the Core module is the query is simple enough, or the Extension module if a more specific function is required.

5. once the answer is elaborated, the GEOLIB server sends it back to the client,
6. the Dispatcher module on the client receives the answer and reformats it before it is displayed,
7. the Visualization module can display the result to the user.

3. *Query addressed to an external GIS*

When the query concerns external data (not already available, neither on the client, nor on the server), query processing is a little more complex :

1. the Visualization module translates a user interaction into a query object, and sends it to the Dispatcher module on the client,
2. the Dispatcher module on the client analyses the query and concludes it must be sent to the server for further proocessing, either because all required data are not present on the client, or because functions required to process the query are not available on the client. Consequently, it initiates a communication between the client and the server, and the query is sent to the server for being processed,
3. the GEOLIB server receives the query and restructures it, before sending it to the Dispatcher module on the server,
4. the Dispatcher module notices that the query corresponds to a script that concerns an external GIS. The execution of this script is then activated,
5. a communication process is established between the GEOLIB server and the external GIS. Data are being formatted and exchanged,
6. once this operation is terminated, the GEOLIB server sends the answer back to the client,
7. the Dispatcher module on the client builds the result and can display it to the user.

The GEOLIB software component has already been used for different kinds of application, either in a standalone mode, or in a client-server environment. It already supports ARC/INFO shapefiles and MAPINFO MIF/MID formats. It is currently being extended in order to support other GIS formats. New functionalities are also being added, in order to be able to deal with more complex applications.

1 ANVAR : Agence Nationale de Valorisation de la Recherche.

FRIEND - FRamework for the Integration of ENvironmental and geographical Data

Martin Brändli and Andreas Ernst
Spatial Data Handling Division
Department of Geography
University of Zurich
Switzerland

Different concepts and developments of managing persistent data (flat files, relational and object-oriented databases) have led to significant obstacles for the development of interoperable systems. This fact is especially true for the domain of geographical information systems (GIS). Currently available GIS are very similar with respect to their architecture and functionality. Nevertheless, they are not compatible with each other concerning the data they manage. Most systems distinguish between the management of geometrical and descriptive data. For performance reasons, geographical data are mainly stored in proprietary databases. Consequently, the interoperation between different GIS is very difficult, since many different interfaces have to be supported. FRIEND (Framework for the integration of environmental and geographical data) is an interdisciplinary project in the area of GIS. Its main objective is the integration of heterogeneous, space-related data repositories under a common roof. In particular, it aims at solving (or at least decreasing) the integration problem posed by the necessity to develop and maintain many interfaces. This paper intents to outline the integration approach which is used in FRIEND. Starting from a framework that approaches data integration from a database technology view in general, it specifically describes the integration of geodata components using an approach based on OGIS (Open Geodata Interoperability Specification), implemented with the Java programming language.

Typically, there are four choices to bring together data from different sources: migration (physical transfer of data from different systems to one new system), data transfer between two systems, use of common data catalogs, and federated database management systems (FDBMS). FRIEND is based on the last option. It allows the integration of data without migration and is thus called "logical integration". Within a FDBMS, local database management systems (components of the federation) keep their autonomy. Some of the issues of the architectural design of this logical integration will be discussed throughout the paper. FRIEND aims at developing a generic solution for the integration problem. This generic solution is implemented with a framework that is adjustable to particular situations. It utilizes a set of interoperable objects following an object-oriented approach, and is implemented as a layer connected to an object-oriented database management system which handles data that have to be stored on the global level of the federation. The data model used in the framework is an ODMG-compatible (Oject

Database Management Group) object-oriented model.

In order to integrate GIS-components on the logical level, the object-oriented data model has to be extended by geographical data types and methods. Since one of the goals of the FRIEND-project is to reach high conformity with international standards, the integration of and access to GIS-components is based on OGIS interfaces. As a consequence, the object-oriented data model used for the integration layer has to be completed with OGIS-compliant data types and concepts. Even if the integration layer is developed as generic as possible, we start the implementation of the integration of GIS-components and geographical data from a real world integration problem which exists at the municipality of the city of Zurich (Switzerland), and which may be similar or comparable to many other urban administrations. During the past thirty years the different institutions of the municipality (surveyors's office, water and energy supply utilities, etc.) developed their own spatial data handling solutions leading to a heterogeneity of systems and data models in conjunction with multiple and inconsistent data collections of the same objects. Today, this situation is not acceptable anymore and the joint use of the data is required. The most important demand on data integration concern the supply utilities in conjunction with the surveyors's office, since the supply utilities heavily depend on consistent survey data. The complexity of the present data as well as the apprehension of loosing autonomy make a real world integration application difficult at the time being. However, the knowledge of the situation acquired by an analysis of the current state of the involved institutions motivated us to implement a so-called "miniworld", which simulates the actual situation by substantially reducing the complexity and size of the data and the number of involved institutions.

This miniworld is designed in the following manner: The federated system consists of two geodata components including the surveyors's office exposing objects such as parcels, buildings, and landmarks on the one hand, and the water supply utility exposing features such as reservoirs, pipes, pumps, consumer sockets, and sleeves on the other hand. In order to install a heterogeneous environment, the surveyors office is implemented with an object-oriented, the water supply utility with a relational database system. The realization of the miniworld involves four steps, each of it coming closer to the real world situation. It corresponds to the five-level schema to describe the architecture of a FDBMS proposed by Sheth and Larson (1990):

1. The two components (survey and water supply) are implemented with the programming language Java storing the data in flat files. They will act as servers of geographical features. Since both components are expressed in an object-oriented data model, there is no difference between the data model of the two components at this state. Consequently, local and component schemata are identical.
2. An export schema is generated for each component, that is, each object intended to be exported is extended by an OGIS-interface.
3. Implementation of a viewer, which is able to access geographical objects of

both components via the OGIS-interfaces. This viewer is the client part of the miniworld.

4. The local components are differentiated. The water supply component will store its data in a relational environment, whereas the survey component keeps on being based on an object-oriented data model. This differentiation requires an additional schema translation for the water supply component, since the local schema and the component schema are not identical any more. The main benefit of this incremental and pragmatical approach is that the access mechanisms via OGIS-interfaces used in the miniworld implementation will finally be embodied in the integration layer described above. The current research concentrates on technical issues concerning the Java-based communication between different platforms, and modeling issues that arise from the mapping of the miniworld data types to the open geodata model types of OGIS. For both concerns, the proposed miniworld serves as an ideal testbed in order to verify the completeness and adequacy of the concepts and components proposed by OGIS. Since the miniworld is characterized by reduced complexity and amount of data, future work has to prove the adequacy of the chosen approach by using real world data. In addition, the miniworld will act as an application to explore geographical consistency requirements which arise at the global level of federated database management systems.

Sheth, A. P., and J. A. Larson (1990): Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys, 22 (3), 183-236.

Comparing approaches to co-operation between GIS and simulation models to identify criteria for inter-operation.

Neil Stuart
Department of Geography
University of Edinburgh
Edinburgh EH8 9XP
Scotland, UK

ABSTRACT

The present research effort to build a basis for different, physically distributed GIS to commonly communicate, should be extended to allow complimentary technologies such as animation, spatial analysis, terrain analysis and visualisation to co-operate more closely with GIS. This paper analyses recent work to increase the co-operation between GIS and simulation models. By observing mismatches found in the representation of data and processes between systems, one may begin to identify appropriate extensions to geographical data models. The analysis also confirms the need for common descriptions of operations for modelling languages which are used to simulate geographical processes.

INTRODUCTION

Some of the difficulties presently reported by workers linking simulation models with GIS provide an interesting base for investigation. Problems with linking simulation models to GIS are often epitomised by weaknesses in the ability to program exchanges of both data and of control between components which should co-operate. This is sometimes attributed to incompatible 'views' of the objects of interest between the databases of the two systems or by a lack of support by the programming language of one application to perform an operation requested by the other.

Further difficulties concern the need for the user to interact through multiple and sometimes unfamiliar interfaces to the software. The result is that many research prototypes are far from seamlessly integrated. The linkage is often loose and much of the semantic description of the geographic relationship between objects can be lost during transfer of data between applications.

This approach to co-operation through a linkage of two systems has been characterised by Fedra (1993) as a low-level approach to systems integration. In a thorough analysis of the problems created, Fedra argues that progress to more co-operative applications may be best achieved through higher level integration. This higher level of integration is characterised by database objects, by programming constructs and by a graphical interface which are more shared between the co-operating applications.

This paper contributes mainly to the discussion of the semantic aspects of the inter-operation of GIS, since it assumes a considerable technical base, including access to a distributed spatial database, a flexible programming language and graphical user interface around which any integrated system would be implemented. We shall show that by analysing present approaches to designing and implementing co-operation between GIS and software for simulation modelling, one may identify several semantic mismatches which limit the scope for inter-operation. Typically, the two kinds of systems exhibit differing levels of sophistication for representing relevant aspects of geographic entities and for performing operations upon them. Further difficulties pertain to the technical approaches by which data is exchanged and control is passed between the two technologies and the means by which the user may interact with the various components and see the results of these operations.

The paper begins by reviewing recent approaches by which environmental simulation models typically co-operate with GIS. Two end points of the present range of approaches are identified. The first case, where the modelling calculations are mainly performed using data and programming constructs contained within the GIS, we call GIS - based modelling. The second case, where the GIS largely acts as a pre- and post processor for data whilst simulation calculations are performed in an external model, we call simulation modelling with a GIS link. We argue that the study of many real world phenomena benefits most from an approach which allows greater interaction between the descriptions of landscape and process. This requires greater exploration of the middle ground where neither technology dominates, but rather where both co-operate more extensively.

Drawing more specifically on examples in hydrological and ecological modelling, the paper investigates the case for supporting extensions to geographical representations, perhaps through a higher level interchange format - and identifies some examples of what these extensions might usefully include, for workers in environmental modelling.

AN ILLUSTRATIVE EXAMPLE

Let us consider briefly the easily understood problem of modelling the sequence in which a series of low-lying depressions will be inundated by floodwater. This is one example of the simulation of a physical process which presently requires extension of geographical representations, including support for higher order entities. Whilst the exact hydrological pathways and processes can never be defined exactly, an approximate model for this problem may be obtained and if we have a realistic representation of the terrain and an estimate of the amount of water entering the area in unit time (e.g. from a river-bank failure).

Several GIS now incorporate functions for processing a terrain model to compute a local drainage direction, from which higher level entities such as watersheds can be iteratively defined. Once these structures (which are essentially static) have been created, simulation might proceed by progressively adding volumes of water onto the model and at each time step, calculating when there is sufficient water to fill one depression to the point of overflow. At this moment, a method is needed to keep track of which depression(s) further water flows into next, and so on as the water level rises. Over time, it is likely that small depressions will coalesce into larger ponds and the sequence in which these events occur may become quite involved. Present approaches to simulating this type of phenomenon require the creation of transient, dynamic data structures. These maintain information about the present status of inundation, lists of those areas that will be flooded next and at what threshold water level this will occur. After some time, the input of water is likely to reduce or stop, so a robust implementation would also need to be able to track the drainage of water and the separation of flooded areas. This example illustrates that to simulate even a relatively simple physical process requires a variety of additional structures, which effectively extend the representational capability of the GIS data models. These transient structures which often exist only during the run time of a simulation are usually most easily created in a standard programming language. Programming languages typically provided with GIS offer limited flexibility to create new data structures.

Given that hydrological analyses is arguably one of the better supported sets of methods within present GIS, this example serves to illustrate that present geographical representations often need to be considerably extended to allow co-operation with simulation modelling. In this example, several extensions were identified, such as:

- Flexible definition of groups of pixels to zones (and at different resolutions); e.g. 'depression'
- Hierarchical nesting of entities; e.g. 'contained within'
- Concept of interacting entities (which may or may not be coterminous); e.g. 'flows to/from'
- Duration for which a given state applies; e.g. 'period of inundation' for temporal animation, etc.
- Concept of forward and backward chaining e.g. 'currently flooded' objects/ extents,

The example also indicates the additional work that programmers presently have to undertake to ensure that the spatial description of the state of the simulation can be maintained within the GIS.

The present limitation of programming languages provided with GIS may be as much of a hindrance to inter-operation as the crudeness of the existing data structures provided within GIS. Indeed, many applications developers would argue that, given the unique complexities of spatial data, the programming languages provided for writing operations within GIS should be more powerful, rather than presently being less powerful, than standard programming languages. Solutions to this may lie in present efforts to create libraries of spatial data processing routines, which like conventional graphics libraries, may be accessed from standard programming languages. A further virtue of creating libraries is that standard methods are used, which can enable more direct comparison of results. For instance, at present many GIS offer an option to calculate flow direction from a grid DEM. But it is sometimes less than clear whether this uses the standard D8 method, or one of at least five other variants which are known to be in circulation, some of which may be preferable (Moore, 1995). Other work to identify standard operations and hence to create a generic language for GIS data processing and perhaps even for spatial modelling and simple simulation within GIS - such as the DYNAMO programs (van Deursen & Burrough, 1996) may also assist here.

CONCLUDING REMARKS

The emergence of large databases of environmental and other types of geographic information suggest that GIS should have a role to play in future environmental analysis. Yet for more realistic modelling and planning to proceed, a more flexible computing environment is needed. This environment which will be presented to the user through a single user interface, seems likely to incorporate GIS, simulation modelling and other complimentary technologies in an inter-operating manner, where no one technology is likely to dominate.

Steps towards closer co-operation involve developing a common understanding of the representation of entities between the different applications and defining a common elementary language for accessing and manipulating entities. These high-level formalisms may then aid in designing database structures which enable a greater sharing of data and programming structures which provide a tighter control over the behaviour of objects during simulation.

This paper analysed some of the difficulties that applications developers presently encounter as they attempt to move from present co-operation towards a greater inter-operation between GIS and these other complimentary technologies. Through such an analysis of the concepts and structures which presently need to be created in programming languages by simulation modellers, we can identify classes of extensions to the present forms of geographical data representation within GIS. When applied to a variety of applications, this begins to reveal the common constructs on which a generic language for manipulation and simulation of data in different GIS should be based.

Using Design Pattern to Define Interoperable GIS Models

F. Balaguer, S. Gordillo*.
LIFIA-Departamento de Informatica, Facultad de Ciencias Exactas, UNLP
CC 11 (1900) La Plata, Buenos Aires, Argentina

*also CIC-Pcia de Buenos Aires

1.-Introduction

The interaction among GIS applications and different kinds of systems (like databases, internet browsers, etc.) generates the need to solve difficult aspects at the implementation level. But, being application design critical for maintenance, reuse and extensibility, these additional problems should be also reflected in the design process.

Designing for reuse and extensibility usually requires interoperability: for example, if we want to reuse a legacy system adding spatial characteristics to it or if we need to use others systems not necessarily built with a GIS software. Our claim is that all these aspect must be (and can be) reflected during the whole design process to deal with them not only at the implementation level, but also in previous stages .

Design Patterns define design solutions to known problems, they constitute a way to record a recurrent problem, the core of the applied solution, its relationships and its responsibilities. Since Design Patterns are like templates, they can be used in many situations that are representing similar problems. Design Patterns are defined in [Gama95] where a vast catalogue explaining different design problems is presented. Also in [Gordillo97] it is shown how to use some existing Design Patterns to design GIS applications; some new ones have been defined specifically for this domain [Balaguer97].

In this paper we show how to use specific Design Patterns in GIS, for taking into account interoperability concerns by showing some examples. The first one, Decorator, allows us to extend a legacy system, built using an Object Oriented Methodology, adding spatial features to it. . Since the intent of the Decorator Pattern is to extend the functionality of objects in a dynamic way, it makes possible to reuse the legacy systems without re-defining it. Using this pattern, the new model (the spatial model) interacts with the legacy system to manage all responsibilities defined in the object model and it manipulates its own spatial responsibilities. The Mediator Pattern allows us to define an object that encapsulates the behavior specifying how a set of objects interact, decoupling in this way, the responsibility of objects from the interaction concerns. Reactor [Schmidt95] provides a simple way to support the demultiplexing and dispatching of multiples services, which are triggered concurrently by multiple events. We next describe the intent of each one of these patterns and give an example of situations in which they can be used.

2.-Design Patterns

In this section we explain different problems that usually appear in the GIS field and their solutions in terms of the application of previously mentioned patterns.

2.1.-Decorator

Problem:
Designers usually face some kind of "hybrid" applications that deal with conventional transaction-based systems and that must be upgraded to include spatial features. . However, ddding behavior and knowledge to existing classes, produce a "dirty" solution, since it mixes responsibilities of the application domain and the geographic one. To extend a traditional (for example dbms based) application with geographical features, we propose the use of the Decorator design pattern.

Solution:
Identify two underlying models: the conceptual model and the geographic one. It makes possible to decouple the problem into two different stages, thus limiting the concerns that designers simultaneously have to deal with.

To extend the conceptual model to a geographic one, the solution is based on the construction of a decorator object which has the same protocol of the decorated one. The finall implementation of these messages is delegated to the decorated object (this object is known by the decorator). Also the decorator object implements its own set of messages which are implemented by itself. In this way, it is possible to add spatial behavior to conceptual objects. Since Decorators provide a flexible alternative to subclassing for extending functionality [Gamma95], the obtained solution is a design where both, conceptual and geographic objects can be manipulated, without re-define the whole application but generating interaction between both models.

Also, there may be geographic objets that do not have an associated conceptual one (because they only have spatial features) and therefore, they only belong to the geographical model. We define an abstract class, which groups the common behavior of those objects belonging to the GIS application model, plus those that have been wrapped from the conceptual model.

2.2.-Mediator

Problem:
It is necessary to use services and geographical data from other applications. The usual strategy is based on exporting and importing raw data from and to a server application. Suppose a simple model to represent natural disasters such as storms, twisters, etc.; a logical extension to this model could use a cadastral application in order to calculate the economical impact in the affected area. The model has to use the cadastral application as a server; this application imposes its interfaces and

protocols, finally the server application could be a legacy system, afterward it could be unfeasible to modify or extend.

Solution:
Build an object which hides the complex process of sharing data with the server application. This object behaves following all steps (protocol) which are used to connect the client with the server application. A hierarchy with different mediators could be defined allowing to interact with different applications in order to share their data and services.

2.3.- Reactor

Problem:
A geographical model has to be modified or extended in order to allow connections to other applications. The model could act like a server of data and services. Suppose you have to extend a cadastral model with the objective to support remote calls or information requests from other applications. The model has to be preserved from changes related to connections and sockets services. The model has to be extended to support different kinds of connections, allowing non OO applications to interact with the geographical model, in order to request information or services.

Solution:
Create an object which has the responsibility for managing request events. It creates an specialized object in order to satisfy each request. The Reactor pattern helps to decouple application-independent mechanisms from application-specific functionality [Schmidt95]. The application-independent mechanisms are reusable components that demultiplex events and dispatch pre-registered event handlers.

3.-Conclusions

Patterns are the best way to record designers experience solving recurrent problems, also they are the media to reflect the state of the art of design in some fields. Interoperability into GIS field has become an issue to solve because of complexity and diversity of technologies applied on it.
We presented three design patterns stressing problems related with interoperability in GIS domain such as: extension of legacy models, communications between client and servers applications (built from geographical model).

4.-References

[Balaguer97] F. Balaguer, S. Gordillo, F. Das Neves: "Patterns for GIS Applications Design". In the proceedings of Patterns Language of Programming 1997

[Gamma95] E. Gamma, R. Helm, R. Johnson, J. Vlissides: "Design Patterns. Elements of reusable Object-Oriented Software". Addison Wesley, 1995.

[Gordillo97] S. Gordillo, F. Balaguer, F. Das Neves. "Generating the Architecture of

GIS Applications with Design Patterns" 5th ACM Workshop on Geographic Information Systems. Las Vegas, USA.November 1997.

[OGIS] Open GIS Consortium (OGC) (1996b), The Open GIS Guide -A Guide toInteroperable Geoprocessing, Available at http://ogis.org/guide/guide1.html

[Schmidt95] D.C. Schmidt, "Reactor: An Object Behavioral Pattern for Concurrent Event Demultiplexing and Event Handler Dispatching". (J.O. Coplien and D.C. Schmidt, eds), Reading, MA: Addison-Wesley, 1995

Implementation of the OGIS Simple Feature Interface

Scott Morehouse
Environmental Systems Research Institute
Redlands, Californla 92373 USA

This presentation will cover ESRI's experience with implementing RFP 1 and will address some of the issues in taking interoperability from "talk" to "technology". I will compare and contrast the different distributed computing platforms (and different SQL approaches defined in RFP 1 for SQL) in this context.

Query Processing in Distributed Spatial Databases

Walid Aref
Panasonic Information and Networking Technology Laboratory
Panasonic Technologies, Inc.
Two Research Way
Princeton, NJ 08540

1. Introduction

Spatial databases are naturally distributed due to the data collection process, the distances between the geographical entities of interest, and the physical geographical boundaries (e.g., between countries, etc.). This calls for ways of interoperation among the distributed spatial databases.

In order for the interoperability among distributed spatial databases to be possible, several issues have to be addressed. Some of these are: resolving the heterogeneity among the spatial databases, e.g., in their data models and metadata, providing open standards and architectures to facilitate the interoperation and the scalability of the interoperability solutions, efficient distributed query processing and optimization for user queries that involve more than one spatial database.

In this paper, I focus on the latter issue; namely providing efficient algorithms for spatial query processing in distributed spatial databases.

In a distributed environment, the execution cost of a query involves the message (or communication) cost, in addition to the CPU and I/O costs.

Many research papers and prototype distributed database systems address the issue of distributed query processing and optimization. Although most of the techniques prescribed in the literature may apply to distributed spatial databases, there are optimizations that are of spatial nature that can help further reduce the overall cost of a distributed spatial query. I illustrate some of these optimizations in the paper for several spatial queries that are of common use in spatial databases.

The queries of interest here are the distributed spatial proximity and spatial join queries. In the following sections, I address some of the issues that are unique to distributed spatial query processing, explain briefly the distributed version of the spatial queries of interest and address issues related to answering each of them efficiently.

I assume that there are m distributed spatial databases $g1, g2, ... gm$). Each database has multiple collections of objects (e.g., tables), where each collection represents objects of a certain spatial data type. For example, in $g1$, one might have a table containing point objects representing the fire stations in the geographical area covered by the spatial database $g1$, a second table containing line segments representing the road network of the same area, and a third table containing polygons that represent the land use map of the area covered by $g1$, etc.

2. Distributed Spatial Query Processing

A *distributed query* is any data manipulation statement that references databases at sites other than the *query site*; the site that initiated the query request. The *processing sites* are the sites where the actual processing of the query takes place, e.g., a join site is where one of the joins in the query takes place.

Several strategies for evaluating distributed queries exist in the literature, e.g., the ship whole and fetch matches strategies, joins using dynamically creating indexes, semijoins, and joins using hashing filters.

To illustrate the differences in distributed spatial query processing, consider the ship whole and the fetch matches strategy. In the ship whole strategy, the distributed query processor ships the entire tables to the processing site and saves it in a temporary table, performs the required operation (e.g., a join), and ships the result of the operation back to the query site. On the other hand, in the fetch matches strategy, when performing a join, the outer table of the join is scanned sequentially, and the distinct key values of the join column are shipped to the site where the inner table exists to perform the join.

We observe the following differences in distributed spatial databases that affect significantly the strategies for distributed spatial query processing.

1. spatial objects (e.g., polygonal objects) are usually approximated by simpler shapes (e.g., their bounding boxes), in order to speed up the processing. Therefore, it may be a cost effective policy to ship the approximate objects in order to reduce the messaging (or the communication) cost when processing a distributed query.

2. Because of this approximation of spatial objects, spatial query processing is performed in two steps: the *filter* step and the *refinement* step. In the filter step, only approximations of the spatial objects participate in the query operations in order to filter out the objects that are not likely to be part of the answer to the query. The output of the filter step is fed into the refinement step, where the exact descriptions of the qualifying spatial objects are then used to further refine the answer to the query.

   In a distributed environment, choosing the sites where each of the filter and the refinement steps are to be performed gives another dimension for the optimization of the distributed spatial queries.

3. In order to perform the spatial operations, it is sometimes possible to ship some aggregate geometric representation of the entire spatial table or parts of it. For example, in order to perform a spatial join of two spatial databases, at sites $s1$ and $s2$, one can ship a region, say $r1$, representing the union of the area coverage of the objects in one of the spatial tables, say the ones at $s1$, to the site $s2$ containing the second spatial table. Only the objects in the second spatial table (at $s2$) that overlap the region $r1$ are the ones that need to participate in the distributed spatial join. Moreover, one can build a region representing the union of the intersection region, say $r2$, and ship $r2$ back to $s1$. By intersecting $r2$ with the first spatial table at site $s1$, one can find the objects in $s1$ that actually participate in the distributed spatial join. This technique may reduce the message cost of the distributed queries significantly.

3. Spatial Proximity

One example of a spatial proximity query is the following:

> Given the m databases of objects ($g1, g2, ... gm$) , find the m-tuples of objects ($o1, o2, ..., om$) such that each object oi E gi is within r units from each of the other objects of the same m-tuple.

This variant of the spatial proximity query has many practical applications. One example of a query that has this form is the following query: find the shopping centers, libraries, and schools that are within "r" miles from each other. This reflects a proximity query that is of a *two*-dimensional nature, assuming that the geographical locations of the shopping centers, libraries, and schools are described using two-dimensional coordinates. Other examples exist in the one-, three-, and multi-dimensional space (e.g., in spatiotemporal queries).

A straightforward query plan for evaluating this query is by a query processing pipeline. There are drawbacks, that will be explained in the full paper, for using this technique to answer proximity queries both in the centralized and distributed cases. In the centralized case, one can find more efficient ways of answering this query.

In the full paper, I present a new technique for answering proximity queries in a distributed spatial database environment that uses the spatial characteristics of the objects to reduce the overall message cost of the query. This is based on techniques borrowed from computational geometry.

4. Distributed Spatial Join

In addition to the distribution of the filter and refinement steps over different sites, the full paper presents special issues related to the distributed spatial join operation. To illustrate a more complete example, the full paper focuses on spatial join over a distributed collection of *polygonal* databases.

5. Concluding Remarks

New issues and cost parameters arise when processing distributed queries over distributed spatial databases. The paper illustrates some of these issues and gives two concrete examples of distributed algorithms that handle the distributed spatial proximity and join queries.

Supporting Interoperation of GIS Objects

Silvia Nittel and Richard R. Muntz
Computer Science Department
University of California, Los Angeles
Los Angeles, CA 90024

Today, a promising approach to large-scale interoperability in geographic information systems (GIS) is a GIS implemented via a distributed object management system (DOMS) (e.g. CORBA, Java, OLE/COM) based on a standardized spatial object model (e.g. OGIS [1]). Geographic entities are available as location- and platform-independent objects, and are accessible via their interface, i.e. their attributes and a set of operations. Using a DOMS, much of the hardware, network and communication heterogeneity between interoperating GIS objects on different platforms and machines is handled automatically by the DOMS software, and is of no concern to the user. Also, via the notion of strongly typed object interfaces, type checking is also automatically performed at compile time to ensure that correct argument types are passed between interoperating GIS objects.

However, to resolve the correctness of an argument passed from a GIS client to a GIS object, it is not sufficient to only verify the correctness of the type of the argument, but it is necessary to ensure that the content of an argument is matched. By content, we mean constraints related to the actual properties of an argument that are not easily captured in the type hierarchy. For example, let us assume that the operation "intersect()" of the GIS object type "LineString" has the following signature:

```
Geometry intersect(in Geometry other);
```

"Intersect" expects an input argument of type "Geometry". To perform the intersection operation correctly, both geometries have to be in the same spatial reference system (SRS), i.e. in the same coordinate system with the same metrics for the axes. However, the requirement of this argument conformance rule is not explicit in the operation's signature, and can not be automatically checked at compile time. Also, the client cannot be sure in which spatial reference system the returned "Geometry" object will be defined, and cannot specify explicit rules for the kind of spatial reference system it would require to be returned (e.g. for displaying objects). Other examples include a GIS service that handles the type "Image", and has to apply operations to images of different formats, resolution, size, compression, etc.

Today's DOMS system offer little help to resolve issues of interoperability of GIS objects regarding the content since its interpretation is usually based on property values of the user-defined GIS object rather than supported through a type system. This makes it difficult to define general conformance rules between interfaces in distributed GIS systems. Today, it is the server's task to check which input parameters can be handled, and which cannot, and either provide implementation for different kinds of parameter content or return an error message. On the other hand, a client might have to explicitly conform to "content" rules of a service (e.g. providing a geometry object within a specific SRS), and have to re-match the return value from a GIS server to its requirements. This is error-prone, cumbersome and makes server and client code bulky.

In the proposed paper, we examine the issue of GIS object property typing with a view to formalize how to cope in a systematic way with the heterogeneity of GIS object properties in a general distributed environment. Key issues for which we discuss solutions concern the problem of how to decide whether two given GIS objects are compatible based on the content of their arguments, or can be made interoperable through the introduction of ancillary mechanisms such as object property reconciliators.

In our approach, we introduce a type model in which we interpret a type specification as a GIS object property descriptor. A GIS object property descriptor consists of a generic type descriptor such as SRS, IMAGE, or GRID, and a set of attributes specifying the properties of a specific GIS object; for example, the notion IMAGE[640,480,JPEG] can be used to denote the image properties of a map object with width, height and encoding equal to 640, 480 and JPEG encoding, respectively. This type model supports the definition of a variety of GIS object property type relationships.

In particular, we define subtyping rules supporting the conformance of GIS object interfaces regarding the content of arguments; e.g. SRS[Cart,2D,1,km] describes a 2-dimensional spatial reference system with a Cartesian coordinate system, tickmarks as multiples of 1, and kilometer as unit. A SRS property of this value can substitute a SRS with the value SRS[Cart,2D,1,m]. In some cases, a subtyping relationship might be too restrictive, i.e. for interoperability it is sufficient if two GIS types support at least one common kind of property. If they support more than one common kind of property, a GIS object property negotiation may take place to choose the property that is actually used.

If two GIS object properties do not support a common property requirement, they might still be interoperable, if it is possible to use a property reconciliator such as a converter of a spatial reference system or a converter for image encoding. We also use the GIS object property type model to define the characteristics of a facility supporting a dynamic reconciliator based on GIS object property type relationship.

Using this GIS object property type model provides for automatic reconciliation of the content of arguments passed between two GIS objects. The mechanism is reusable for all kinds of different GIS objects. Relinquishing both services and clients from the task of checking the arguments makes their implementation simpler and more lightweight. Since the GIS object property reconciliation mechanism is extensible, i.e. new description types and conformance rules can be added, GIS services can profit from the extension without any code having to be added to their implementation.

References:

[1] Buehler, K., McKee, L.: Introduction to Interoperable Geoprocessing, OGIS Project Technical Committee of the Open GIS Consortium, Inc., OGIS TC Document 96-001, 1996.

Hot Links as a new Way of Data Integration in a Distributed Computing Environment

Andre Hagehuelsmann
Freie Universitaet Berlin/
Intergraph Germany
Adalperostr. 26
85737 Ismaning, Germany

INTRODUCTION

The issue of data sharing and integration has to be reconsidered in the context of open, distributed computing environments. Conventional approaches to data sharing utilize conversions from one data structure to another and if necessary from one data model to another. This procedure results in both the loss of information to a certain extent and redundant data without linkage to their source. The intention of the Open GIS Specification is to resolve this problem by dynamically interpreting and representing geodata from various sources in a unified, comprehensive, and generic form, that is known as the Open Geodata Model. Inherent incompabilities are planned to be resolved by semantic translation services. The overall objective is the interoperability of multiple geodata sources.

MOTIVATION

Assuming the geodata exist as self-contained entities, this concept should serve well. But the possibility that even single datasets are conceptually distributed deserves further consideration. This scenario occurs when the following situation is given:

- There is a geometrically accuracte geographic dataset (i.e. cadastral data), that is proclaimed to be the geometric basis for geodata that occupy the same area.
- There is a thematic geodataset (i.e. a vegetation map), which should refer to that base geometry, since their spatial extents overlap and they are set up on the same scale range.
- The geometries that describe the same or similar objects in both datasets do not actually match. The geometry of the thematic dataset is entirely or in part a less accurate representation of the base dataset's geometry.

Both datasets are distributed in so far as:

- The geometry of the thematic data is entirely or partially described by the base geometry dataset.
- The thematic dataset holds information that also describes objects of the base dataset.

As long as consumers want to use such a combination of base geometry and thematic information, for example for land use planning, the problem of how to achieve and maintain this integration remains.

SOLUTION

In summary, the presented approach shows a functional integration that is established through links pointing to services that support the information to be integrated. These links can be handled as properties in the scope of the Open Geodata Model without the need of modifying this model, in contrast with conventional approaches to data integration which require fundamental extensions and modifications to the data models and data types in order to merge them. As a result of the integration, the thematic dataset no longer contains any coordinate information of geometry that is described elsewhere. Because this approach no longer requires data copies, inconsistencies resulting from redundant data are eliminated. Opposed to the exchange of actual data, information to the links is provided, moving the task away from simple data fusion to intelligent combination of information.

The presented approach has been implemented in an OLE/COM-environment in consideration of the OGIS concepts, namely the Open Geodata Model, the OGIS Services Model, and the Information Communities Model. The integration is achieved through a kind of communication between the applications and services that support both datasets involved. The service of the thematic data accesses the base dataset via a trader using the spatial extent and the conceptual scale of the thematic information as query parameters.

The emerging semantic gap between the involved data models is resolved by comparing their data dictionaries to provide a comprehensive schema that represents interrelationships of terms and definitions. The schema is mapped to a simple relational database that supports tables for one-to-one-, one-to-many-(aggregation) and many-to-one-relationships(generalization). Each relationship is given a priority as well as the information whether one or both parts have to be derived. This means that in cases where no relationship between feature definitions can be retrieved directly, sophisticated queries have to supply appropriate objects of one dataset that are compatible with the specific feature definition of the other dataset. These queries are implemented as interface members of the services that support the particular dataset.

According to the comprehensive schema, each object in the thematic dataset is given control over the further communication. The particular object then acts as an independent entity which queries the base dataset for a counterpart which geometry shall be used. This counterpart can be a single object, a collection of objects or a part of an object. Based on specific parameters, the thematic object decides whether and to what extent it can take over the given geometry. This step notably involves the most extensive algorithms, since data specific integrity rules (i.e. space-filling) have to be applied during geometry modification. In this case, the thematic object will lose its previous geometric part and will instead take up link information that points to its present geometry. Conversely, the base dataset's object(s) whose geometry was used to check the geometric similarity is (are) assigned a link to the thematic object(s) in order to be described more precisely. The links simply provide the name of the specific dataset along with the object's identity number. The type and function of the dataset as well as its supporting service are derived from its meta-information. To maintain the integration established by means of the process described above, the thematic geodata - each time they are called up - queries via its links for the accurate geometry in order to put the geometric information in a cache if necessary. As a consequence, this interoperable environment facilitates a single workflow for land use planning which considers existing accurate geometry along with appropriate thematic information.

IRIS: a Tool to Support Data Analysis with Maps

Gennady and Nathalia Andrienko
GMD - German National Research Center for Information Technology
Schloss Birlinghoven, Sankt-Augustin, D-53754 Germany

IRIS is a software system designed to assist users in exploration of spatially referenced statistical data such as economical, demographic or ecological data related to geographical locations. Analysis of such data is impossible without representing them on maps. The software systems known as GIS (Geographic Information Systems) can be used for data mapping, but, despite their power in operation with geometric information, the visualization facilities of GIS need to be improved. The most serious shortcoming is that no guidance is provided to the user in selection of presentation techniques for data to be analyzed, whereas finding an adequate presentation, on the one hand, is crucial for successful analysis and right conclusions, on the other hand, requires special knowledge from the field of thematic cartography. IRIS incorporates this knowledge in the form of generic, domain-independent rules. This allows to generate automatically thematic maps correctly presenting users data.

By automatic map generation IRIS releases the user from the necessity to think how to present her/his data and from the routine work on map building and allows her/him to concentrate on data analysis. To get a cartographic presentation with IRIS, the user needs only to select the fields to be presented.

To choose the adequate presentation techniques for given data, IRIS takes into account data characteristics (types of fields: numeric, categorical, logical; number of different values or value range for a field) and relations among data components (whether the fields to be analyzed are comparable, whether they can be summed to produce some meaningful total, whether some of the fields is included into another). Some of the techniques emphasize these relationships (see Fig.1).

Different presentation techniques provide different opportunities for analysis. For example, bars allow easy estimation of absolute values and differences, pies are good for seeing proportions, painting area objects in colors or shades according to values of an attribute gives an integral view on variations of this attribute through the territory. All these opportunities can be useful during exploratory data analysis when the user does not know beforehand the inherent features of the data. Therefore, whenever different presentations of the same data are possible, IRIS offers all of them so that the user could switch between them in the process of data analysis.

Maps on computer screen should not be mere reproductions of paper maps. The new (comparing to the history of cartography) output medium offers new opportunities for analysis: a map can be dynamic and reactive to users actions. In IRIS we develop tools for interactive manipulations with maps that are intended to strengthen the potential of different visualization techniques in data exploration. By "interactive manipulations" we do not mean such basic operations as zooming or direct access to data values through the map. Our idea is that each presentation method requires a specific interactive manipulation tool that exploits the principal features of the method and helps in the kind of data analysis this method is suitable for. We expect that in this way the user can utilize the potential of each presentation more completely and effectively.

An example of interactive manipulation is visual comparison designed for the so called choropleth maps representing values of a numeric attribute into color shades in which objects are painted: the greater is a value, the darker is the color. Choropleth map is good for the study of spatial distribution of attribute values: colors are promptly perceived by a human; similarly colored neighboring spots tend to be perceived together as larger patterns (figures, images), and this favors finding interesting spatial patterns. By comparison of two or more choropleth maps one can reveal relationships among several attributes: relatedness will manifest itself in similar patterns.

In "visual comparison" some number N between the bounds of the value range of the shown attribute is selected, and the map is redrawn so that values grater than N are depicted by shades of green and those less than N are shown by shades of cyan. The greater is the difference between some value and N, the darker is the shade used to represent it. The values exactly equal to N are shown in light yellow (see Fig.2).

So, visual comparison adds color hue to the expressive means used in the map. This encourages visual grouping of objects: neighboring objects painted in the same color tone tend, despite differences in shades, to be perceived together as a single figure. This evidently favors revealing spatial patterns. Thus, in our example (Fig.2) it is clearly seen that the least percent of children (less than the lower quartile) in Bonn is in the center of the city.

In addition to visualization and related interactive operations IRIS contains such facilities for data analysis as querying, calculations in spreadsheet manner, generation of derived attributes. IRIS is implemented in 2 variants: as a program running on PCs under Windows and as a WWW application with interface in Java language running under any WWW browser. The system can be accessed remotely at the address: http://allanon.gmd.de/and/java/iris. The WWW variant of the system was called more than 4500 times by people all over the world. IRIS was included in Top 1% web applets and Top 10 web applets lists (September 1996) by independent Java Applet Rating Service.

Information Brokers for a Web-Based Geographic Information System

Ian Finch and Eleanor Small
Connect Centre
Department of Computer Science
University of Liverpool
Liverpool L69 3BX
United Kingdom

INTRODUCTION

Although there is a large amount of geographic information available on the World Wide Web, it is extremely difficult for a user to locate the data they need. Typically they must take one of the following approaches:

- Ask colleagues for the address of such information
- Use a search engine to locate the information
- `Surf' the net, following links from appropriate Web sites and stumbling (hopefully) across the information

The fundamental problem is the lack of computer-readable information about the geographic data on the Web; that is, an almost total absence of metadata.

This paper describes a method to combat this problem, using `information brokers' which store the metadata about geographic information on the Web. Using a graphical user interface (GUI), coded as a Java applet and embedded in a Web page, the user enters into a conversation with one or more brokers to locate information of interest.

The power of this approach arises from the fact that the information brokers (which have been coded in Perl in the prototype system) and the GUI speak a common language (the Knowledge Query and Manipulation Language; KQML) and can therefore easily exchange information. Furthermore they share a common vocabulary (or ontology) which ensures that when they refer to some entity (for example a map) during a conversation, they both have the same understanding of what that entity is and how it can be manipulated.

A method has also been developed for allowing the information brokers to gather their metadata from the Web and this too will be described.

FROM METADATA TO A COMMON LANGUAGE

As mentioned in the introduction, apart from asking a person where to obtain information, there are two main methods for locating geographic information on the Web; using a search engine or surfing in appropriate areas of the Web. The major problem with both of these approaches is the same; they rely somewhat on luck. With a search engine, the user hopes that they have chosen the correct keywords to search with. If a suitable Web page exists, it is also a requirement that the search engine knows of that page's existence and has `up to date' knowledge of its contents. With the surfing approach, luck plays an even greater role, with the user only finding the desired information if they happen to follow the correct set of links.

Thus, in the current situation, a person seeking geographic information may spend a considerable amount of time searching the Web and may not find the best source of data for their requirements (or indeed any useful information at all).

The underlying problem is a lack of metadata associated with the geographic information on the Web. It must also be noted that this metadata offers no advantages unless a way of manipulating it is also devised. This then allows any agents (such as the GUI and the information brokers) to communicate with each other in the most efficient way. The system described in this paper defines five levels of information, each layered on top of the other. These are:

- Information
- Metadata
- Vocabulary (ontology)
- Sentences
- Conversations

Information is, quite simply, the geographic data which may be of interest to a user. This could be, for example, population statistics or an image of a political map of some region.

Metadata is the information about this geographic information. Continuing with the map example, this would contain the location and extent of the area to which the map refers, together with information about the projection used for the map.

The vocabulary (or ontology) is the glue which enables the various agents (brokers and GUI) to communicate. Just as for human communication, it is essential that all participants mean the same thing for each word they use. Each word in the vocabulary will therefore be known by each agent and there will be a single definition of its meaning. The word `map' will thus refer to an image, together with metadata giving its location, extent and projection. Note that both the image and its metadata must be present for some entity to be a `map'. Thus, if an image is clearly a map to a human, but has no metadata associated with it, then it cannot be a map as far as the agents are concerned. Conversely, an image may not appear to be a map to a human, but as long as it has the correct metadata associated with it, then the agents will consider it to be a map.

Moving on from the vocabulary, it is necessary to create rules which specify how the words from the vocabulary can be combined to create valid requests and answers. These requests and answers (or sentences) make it much easier for the agents to understand each other. The rules restrict the infinite number of combinations of words to a very small set.

Finally, to allow exchanges between agents to take place easily, it is necessary to have rules governing how sentences can be combined. These rules allow standardisation of the conversations between agents. For example one valid conversation may be a request for information from one agent followed by a rejection of the request by another agent.

To clarify these levels of information and their relationship with each other, the following section examines how the types of information were represented in the prototype system, using the Knowledge Query and Manipulation Language.

THE KNOWLEDGE QUERY AND MANIPULATION LANGUAGE

The Knowledge Query and Manipulation Language (KQML) was developed by DARPA to allow agents to exchange information in a standard manner. It has been used by a number of researchers in the fields of Artificial Intelligence and Distributed Systems, and has proved to be both powerful and versatile. It must, however, also be noted that more recent information interchange protocols have been developed, notably CORBA.

The basic idea behind KQML is that agents send each other `performatives'; instructions to perform some operation (such as evaluating an expression and sending back the result). These performatives are encoded as lisp expressions for easy parsing, and sent as simple ASCII strings. A performative will consist of a keyword indicating the type of performative it is (an `evaluate' performative, for example) and then a series of keys and values indicating the language (or protocol) which is to be used, the ontology to use and the actual content of the performative (which in the case of the `evaluate' performative is the expression to be evaluated). As an example, one agent might send the following performative to another agent asking it to perform an addition:

```
(evaluate :language arithmetic
          :ontology general
          :content "1 + 7")
```

This can now be tied into the levels of information described above, beginning with the data level. Since all data must be available on the Web, each piece of data can be uniquely described by its Web address (that is, its URL).

Metadata must then be associated with this data. The actual method for implementing this would be up to each agent in the system. There is no need to standardise this across the system, provided that each agent has the appropriate metadata available for it to use or pass on. This leads on to the vocabulary (or ontology) level.

In order that the agents can communicate effectively, they need to share a vocabulary, knowing, for example, the metadata that is associated with a satellite image. Thus, one agent (the user's Web browser, for example) can request a map from another agent (an information broker, maybe), confident of obtaining the correct response. As an example, consider the map. In the system discussed here, a map is an image which can be manipulated by the user, allowing points and areas to be specified. Thus, the user could call up a map of Europe, draw a box around the British Isles and then zoom in on this area, perhaps automatically loading a new map with more detail (rather than simply scaling the area in question). It is obvious, therefore, that a map is more than a simple image; there must be associated metadata linking the pixels in the image to a geographic location reference system. So, within our ontology, a map is an image, together with metadata which allows it to be linked to the real-world area it represents. A simple scan of a page from an atlas, with no associated metadata, is therefore not a 'map', but just an 'image'. Conversely, a radar image which has the appropriate metadata associated with it, is a 'map'. The vocabulary as a whole is given a name (for example `gis') and this can then be used as the value for the 'ontology' key in the performative, to ensure that a common vocabulary is being used.

The next level above the vocabulary is the sentence level, combining the words from the vocabulary into sentences. A sentence in english might be `find all maps of Liverpool', this would become a KQML performative in this system, for example:

```
(evaluate :language gisLang
          :ontology gis
          :content (find :type map
                         :area liverpool))
```

Note the use of the language key to specify the language being used. This is similar to the ontology, but instead of defining the meaning of words (such as `map' and `liverpool' in the above example), it defines the meaning of actions (such as `find' above).

The final level of information is the `conversation', which consists of a sequence of sentences between two or more agents, according to a set of rules. In the system described here, one agent sends an `evaluate' sentence. This can only be responded to using one of four performatives; sorry, error, later or reply. The `sorry' response means that this is a valid sentence, but the second agent is unable to fulfil the request. The `error' response means that there is something wrong with the performative. The `reply' response means that the evaluation has been successful and here is the result. The `later' response means that the evaluation can be performed, but the result will not be available immediately. The `later' response will then cause the first agent to respond indicating whether it wishes to `callback' to retrieve the result at some later time, `subscribe' (let the second agent send the result when it evaluates it) or `discard' the result (in which case the second agent can abort its evaluation). In this way, a set of rules can be constructed, specifying whether a given conversation is legal or not.

Thus, using KQML, the five levels of information described in the previous section can be represented in a computer-understandable form.

THE PROTOTYPE SYSTEM

As discussed above, the prototype system uses a structured communication protocol, consisting of five levels of information; from the actual geographic information to the rules for conversation. The system consists of three types of agent; the user's GUI (implemented as a Java applet running in a Web browser), the information brokers (implemented as servers written in Perl) and the Web servers (not implemented as part of the prototype system).

An example interaction will illustrate the operation of the system. Suppose that the user wishes to obtain some radar imagery of Liverpool. The first step is for the user to open the Web page containing the Java GUI. This will automatically connect them to their default information broker. The GUI begins by requesting from that broker the most general map they possess (usually a map of the world). This is sent to the GUI, which then displays it for the user. The user can now use a `rubber rectangle' to select portions of the map and zoom in for more detail. Each time the zoom button is pressed, the GUI asks the broker for more detail. If the broker knows of a more detailed map, it retrieves it for the user and that is displayed. Otherwise, the GUI simply scales up the appropriate portion of the map which it already possesses. Notice that the GUI must have the more detailed map to display immediately (depending, of course, on the speed of the network/modem). Thus, if the broker send back the `later' response (discussed in the previous section), which means that the information will be available, but not immediately, the GUI will discard that request and scale the current map (not informing the user of this).

When the user is happy with the area they have selected, they can fill in information about the imagery they want (for example the resolution or the band, in the case of Synthetic Aperture Radar). The GUI will ask the broker for this information. In this case, if the data is not immediately available, the user will be consulted (using a dialogue box) to check whether it is acceptable to wait for the data.

CONCLUSIONS

This approach offers many advantages over other ways of locating geographic information on the Web:

- The system understands about geographic information, due to the various layers of information; notably the metadata and vocabulary levels. It is therefore more suited to the needs of the GIS community than a general search engine.

The system reduces traffic to the Web servers containing the actual information because the information broker only passes on the location of relevant data (as opposed to the greater amount of data sent when browsing the Web site).

- Information brokers can be tailored for specific groups of users. For example, brokers can be set up to know about the UK or maritime-related GIS or radar-related GIS.

Currently, the system does not automatically obtain information from Web servers, but the information brokers use a hand-built database. This is obviously the next stage in the development of this system. The authors will also consider interaction between the classes of agents which do not currently communicate, for example, allowing two brokers to exchange information.

A Request Specification Language for Spatial Internet Marketplaces

Volker Gaede, Kerry L. Taylor, and Xiaofang Zhou
CSIRO Mathematical and Information Sciences
GPO Box 664
Canberra, ACT 2601
Australia

Notes: slides from this conference presentation can be found at http://www.wiwi.hu-berlin.de/~gaede/rsl.slides.ps.gz

Internet Marketplaces have recently been proposed as a new and interesting model to make data and computational services available to a broad public. Unlike electronic commerce or on-line shopping services, services are both requested and delivered through the Internet. The underlying idea of Internet Marketplaces is that providers make a range of services available on the Internet and customers rent them whenever necessary. Such a market model is attractive both to customers and providers.

The idea of an Internet Marketplace has been adapted to a range of different application areas including decision support systems, mathematical software and spatial information systems. Among the different proposals for marketplaces, Spatial Internet Marketplaces seem to have a somewhat distinguished role in that they address a wide range of different communities of interest (e.g., Earth Observation, global positioning, environmental management), that geographic space acts as a common underlying domain and there is a pressing need for data and computational services. Consequently, Spatial Internet Marketplaces might have a much greater commercial potential than other marketplaces. For example, advanced spatial applications require access to up-to-date data, specialized hardware and tailor-made software modules. Answering complex questions posed by spatial applications can require the combination of various disparate and distributed services.

However, before service combination is possible, there are many issues to resolve, especially that of interoperability standards. The recent Open Geodata Interoperability Specification, offered by the Open GIS Consortium is a first framework on which to build a Spatial Internet Marketplace. With respect to concrete implementations, recently a first Spatial Marketplace design has been proposed in the context of the SMART project currently implemented at CSIRO Mathematical and Information Sciences. The proposed spatial market consist essentially of two different kinds of services: function and query services. Services participating in the Spatial Marketplace have to register with a distinguished service, called the registry. In addition to the basic provider services, there is also a service class which provides support for planning. Planning services turn a declarative request specification into an executable plan that invokes a combination of distributed data and computational services, to materialize the request.

In order to invoke the wide range of services conveniently, we need a standard request language. The language is to be used for expression of requests---both requests to be resolved by a planner and requests directed to individual function and query services. For this role we propose the Request Specification Language (RSL).

The RSL is designed specifically for requests for planning, function and query services in the SMART model; dealing uniformly with each. It also supports expression of desired result formats through type descriptors and type constructors. Although it is primarily intended for machine-based generation and interpretation, it is also conveniently human-readable and writable.

RSL requests are framed with respect to a novel abstract data model called SDM (Smart Data Model). It is designed primarily to allow uniform representation of computational procedures and structured databases in a way that is convenient and natural to each. Expressions of schemas for popular data models (relational, object-oriented, and so on) should be easily translated to this data model, enabling easy wrapper development for such existing systems.

The relational model is not suitable for SMART services because it lacks a sufficiently rich semantic structure. Although the relationship of fields within a row of a single table is clear enough, the relationships between fields across tables is not adequately represented. The object-oriented model captures the relationship of attributes of an object, but cannot represent relationships between objects. Unlike the relational model, the object-oriented model deals uniformly with computational procedures that are methods, but it loses its uniformity as soon as a method requires access to some data that is external to the object.

Unlike traditional models the SDM is concerned with data description, not with data representation or manipulation, and therefore has no difficulty with issues of duplication, unique naming and efficient retrieval.
The RSL distinguishes object selection---identifying the objects and data of interest, and object refinement---identifying what part of the described objects are to be returned and how. In its current form it is equivalent to a non-recursive first-order constraint language with arithmetic constraints. It is not a programming language. There is, for example no provision for iteration or conditional execution in the language. Rather it is a declarative query language.

A service request consists of two parts: a constraint specification which specifies the set of objects of interest and a target list describing the required output resulting from the operation and the representation of the result. In other words, the target list may not only consist of the required attributes but also of a range of type constructors or transformation routines. This allows the local service to generate an output which can be readily used by subsequent services without having to ship it to another site.

Clearly, leads to a significant reduction of the computational overhead associated with evaluation a given request.

In the full paper, we identify the need for the SMART RSL and present a detailed description of it together with the underlying SDM. We show by means of various examples that the language is suitable for the role we have outlined in a spatial internet marketplace. We also discuss the push and pull models for controlling the distribution of data in distributed systems and discuss how to support them in the RSL.

Interoperability by exchanging executable content, or what have PostScript and Java in common?

Andrej Vckovski
Spatial Data Handling Division
Department of Geography
University of Zurich

Abstract

In the area of Geographical Information Systems (GIS) , the requirements posed upon an set of "heterogeneous but interoperable systems" are not easily met. The particular systems most often possess a certain autonomy and are designed to meet specific requirements for the respective application domain their are used in. In order to be more than mere "data exchange", interoperability needs to be based on a framework which is easy to understand for all participating communities and yet expressive enough to meet those application specific requirements.

This contribution shows how "executable content" known from systems such as PostScript or Java can provide a very flexible method to exchange rich information in a way where the data consumer does not need to have the full knowledge of the data producer's application domain.

PostScript has been designed to overcome hardware dependencies on raster output devices when exchanging data describing text and graphical shapes. The heterogeneity of the output devices was addressed by incorporating much of the device specific issues within the device itself and define a standardized data exchange format (contrary to approaches that define a set of device-specific drivers on the data producer side). The data exchanged consist actually of an executable program written in the PostScript language. The device peculiarities are handled by an interpreter for that programming language. The PostScript approach is based on a semantically rich description of the data on a high abstraction level, semantically rich in the sense that the approach is very expressive.

However, the PostScript change history has shown also some drawbacks of this approach. For example, the understanding of data encoded in PostScript" (i.e., a PostScript program or "file") needs a full-blown PostScript interpreter, which is not a trivial thing considering the expressiveness of the language. This contribution discusses some of the lessons learned in PostScript's history.

Java is an object-oriented programming language and distributed computing environment which has gained a lot of in the last few years (however, partly due to excellent marketing activities). One of the core concepts of Java is the same as used in PostScript and other interpreted languages: Hardware and operating system heterogeneities are resolved by interpreting a well-defined format on the target

platform. However, unlike PostScript Java uses an intermediary format ("byte-code") which makes this stage of execution somewhat independent on the language itself (i.e., Java byte code needs not necessarily to be produced by a Java-compiler).

Java offers also other features which make it even more suitable for the exchange of information in an heterogeneous world:

- Java-objects can be -- platform independent -- dynamically loaded into a running program. A GIS could load Java-objects describing some data into its running image and execute its content.
- Java has been partly designed to be used in distributed environments and supports therefore many corresponding primitives.
- Java supports interfaces and interface inheritance which is a flexible mean to provide a common ground within interoperable systems.

The contribution will discuss some of these features of Java and give some examples on how Java can be used as a framework for interoperability, a framework which is based on the exchange of executable content.

The use of Functional Programming in the Specification and Testing Process

(Extended Abstract)

Werner Kuhn
Institute for Geoinformatics
University of Muenster

Andrew U. Frank
Dept. of Geoinformation
Technical University Vienna

1. Introduction

The problem with formal specifications is that formal tools can only check that the specifications are internally consistent. No method is conceivable to automatically check that the specification expresses the intentions of the (collective) mind of the specification body, i.e., that it is correct. However, humans can easily detect if the result of an operation corresponds to their expectations. The idea of this paper is to show how this human capability can be exploited in the writing and testing of specifications for interoperable GIS.

A problem related to that of deciding the correctness of specifications is that of checking the compliance of an implementation with a specification. It is practically (though not fundamentally) impossible to prove the compliance of a program with a specification, but one can test (with a limited set of examples) if an implementation produces no results prohibited by the specification.

The same solution applies to both cases: if specifications are written in an executable language, they can be run and the results of specific test cases can be directly inspected. The results can be used to demonstrate and better understand what the specification means, and to check compliance of implementations. The paper applies this idea to the specification of semantics for interoperable components. It shows how the use of a functional language for specification writing (as explained in the companion paper, Frank and Kuhn) achieves the goal of unifying the specification and testing processes.

2. Current Situation: a three-step specification process

Today's specifications formally describe the signatures of operations, assuming an implicit type system. The semantics of operations are typically expressed using first-order logic or natural language. Such expressions often assume the semantics of analytical geometry which are not supported by straight-forward implementations.

They are difficult to combine and to understand in their collective meaning. It is not unusual that there is no implementation (i.e. no model) satisfying all of them. Operations from analytical geometry (e.g., line intersection or point equality) are notorious examples.

Faced with these difficulties, current specification practice follows a three-step process:

- Specifications of operation semantics are written as natural language expressions. They contain to a large degree pre-existing understanding of terms and expected behavior. A substantial amount of wishful thinking is involved in both, reading and writing the specifications.
- Implementation according to specification is attempted.
- The implementation is checked against a set of examples for which the answers have been computed (e.g., by hand) to see if the results of operations correspond (compliance testing, see OGC 1997).

This separation of specification from testing is dangerous for the success of the whole process and potentially very expensive, because only phase three can reveal if an implementation is feasible or not. It has to be assumed that an implementation (a model, in terms of proof theory) exists. Even if such a model exists, it remains to be tested whether it does what the specifiers had in mind. And there is no guarantee at all that the semantics of the model have been specified such that ambiguities cannot endanger interoperability.

3. Application of Proof Theory to Specification Writing

Proof theory is the appropriate theoretical framework to discuss specifications and to improve the process:

- specifications have to be independent of implementations; thus, they are a theory;
- implementations are concrete realizations of a theory and multiple implementations can co-exist; they are models which must comply with the theory.

The relationship between theory and model is blurred in most specification and programming environments. Pure functional languages (Bird and Wadler 1988) with classes, like Haskell or Gofer (Peterson et al. 1996; Jones 1991), provide a clean and useful separation of the two levels.

4. Executable Specifications

If specifications can be executed, they can serve as prototypes for an implementation. This has tremendous advantages for the specification process, by moving the testing phase up to the specification writing phase and allowing for early error recognition and rectification.

Writing specifications in an executable language, however, reduces them automatically to constructive expressions (of the kind of rewrite rules). This is a serious restriction, but it can be circumvented by expressing what cannot be stated in the theory as (constructive) test functions and supplementing the theory with a simplistic execution model. This model is not optimized for effective execution, but can demonstrate that an implementation is possible and support human understanding of what is going on in the specified operations.

5. Demonstrating the Semantics of a Specification

Executable specifications allow to compute results for any input data. They allow the specifier to observe if her intuition about what the specification should say corresponds to what the specification actually says. The specification can easily be distributed, so that any interested parties can submit test cases and convince themselves that the specification matches the collective understanding of the specification group.

6. Compliance Testing

Outcomes for test cases can be computed with an executable specification. These results can then be compared with the output for the same test cases from the implementation to be tested. An automated compliance test system can thereby be designed.

If one has a hypothesis about a "wrong" implementation, it is possible to produce specific test cases where the outcome from a "wrong" implementation would be detectably different.

7. Conclusions

The processes of specification and compliance testing must be closely linked to avoid that the specification of semantics is effectively in the compliance testing program. Functional programming allows for such a close link. Pure functional languages with a class system separate the levels of theory and model. Thereby, they achieve both conceptual simplicity and practical utility for specification and testing.

References

Jones, M. P. (1991). An Introduction to Gofer, Yale University.

OGC (1997). A compliance testing strategy. The Open GIS Consortium, Document 97-7. *http://www.opengis.org.*

Peterson, J., K. Hammond, et al. (1996). Report on the functional programming language Haskell, Version 1.3. *http://haskell.cs.yale.edu/haskell-report/haskell-report.html*

A Web-based Scientific Data Server for Accessing and Distributing Earth Science Data

Liping Di, R. Suresh, K. Doan and Doug Ilg
Hughes STX Corporation
7701 Greenbelt Road, Suite 400
Greenbelt , MD 20770

Ken McDonald
NASA/GSFC, code 505
Greenbelt, MD 20771

In recent years, one of most exciting developments in information technologies is the proliferation of Internet connectivity and the popularity of WWW services. Web technologies have provided a convenient way to access multimedia information through the Internet. Almost all users with Internet access use Web browsers to access information. Although currently most information accessed through the Web are descriptive, the possibility of distributing scientific data through the Internet using the Web will immediately offer scientific datasets to millions of potential users with little effort. Therefore, the data will be more widely used, and cost savings in data distribution will be achieved. Currently, many prototyping efforts have been done to show the applicability of the Web technologies for the scientific data distribution. The particular one which promotes NASA data standards and provides accesses to NASA datasets is the prototype Data and Information Access Link (DIAL) system developed by Hughes STX Corporation and National Center for Supercomputing Applications (NCSA) with NASA DIAL is the software designed specifically for type of what could be generally termed a "scientific data repository", which is designed to provide scientific data over networks. There will soon be many thousands of such repositories, varying in scale from large global archives to science teams or even individual scientists. These repositories must all interoperate as seamlessly as possible, while operating in heterogeneous computing environments. Groups of interoperating servers will form one or more federations of repositories. The key software technologies to create the DIAL system are: 1) catalog interoperability, and 2) efficient access to complex scientific data objects over networks.

The development of DIAL system is a on-going process. The fully-developed DIAL system will consist of: 1) a catalog interoperability layer to support major protocols, such as Z39.50, CEOS CIP, EOSDIS V0, for interoperation among DIAL servers and between DIAL and other data systems in support of NASA concept of the federation of small data producers; 2) a scaleable scientific data server to serve complex scientific data and data objects in multiple formats and to allow data users to interactively manipulate the selected data so that they can obtain the data in their favorite form in terms of spatial and temporal coverage and resolution, parameters, and format. The data server will address the dynamic and object serving; 3) an open database interface layer to interface with Open Database Connectivity (ODBC) capable metadata catalog databases for powerful data search and finding; 4) a user interface layer to support user-friendly interaction between client and server; and 5) a search engine with a file-based catalog database to provide basic metadata database capability in case the data producer does not have an expensive commercial database system. In addition, a suite of software tools will be available to help data producers to ingest the data into the system and to help data users to download and analyze the data.

Currently a prototype of the DIAL system has been developed. The prototype system has all above mentioned components in some stages of the development. The prototype DIAL is a compact yet powerful Web client-server based data browsing and distribution system, providing low-cost on-line data access to users through the Internet. DIAL enhances data access and interoperability through a simple web interface. It permits data producers to set up a low end workstation as a data server for distributing their data and metadata quickly and easily. Users can access the data server through any web browser, search and query the data archival based on geographic location, time, and other relevant parameters, locate data of interest, view the metadata, browse the data, view, subsample, subset, and download the data to local disk. Data may be download in one of three formats (HDF, ASCII, and binary), or displayed on the screen as a GIF image, HTML text or tables. Binary executables of the prototype software are free to any interested people, and are available on all major UNIX platforms, and will soon be available on the Window/NT platform. The address for the demo site of this software is http://hops.stx.com:8080/dialhome.html.

Geospatial Modelling:
A Case Study for a Statewide Land Information Strategy

Dr. David Pullar
The University of Queensland
Brisbane, QLD 4072. AUSTRALIA

Ms. Kristin Stock
Queensland University of Technology
Brisbane, QLD 4001. AUSTRALIA

Abstract

Most information applications are built on an underlying business model that includes a semantic description of the information content for a system. Using expressive and powerful modelling techniques is imperative to accurately capture this business model. This paper examines modelling techniques with regard to developing spatial information products for a statewide land information system. These information products convert spatial data held by state government agencies into useful and timely information that may be accessed by users in a networked computer environment. Handling of spatial information requires special application software, such as a geographical information system (GIS). Likewise application modelling techniques for GIS applications impose special modeling requirements. The paper reviews geomodeling approaches in terms of (1) special spatial modelling languages, and (2) extending general modelling languages to incorporate spatial requirements. The latter approach is adopted to describe a geomodel for spatial domains in terms of the abstract syntax (metaclasses, relationships, and constraints), well-formedness rules (rules and constraints on valid models) and semantics (model usage). This is applied to one spatial information product, namely for dealing with land pest infestations, to describe an information model from primary data collection in the field to networked access to information.

1. Introduction

This paper goes beyond the exchange of spatial information to look at how tools for manipulating that spatial information are commonly understood and transferred across application boundaries. In this respect interfaces for spatial information systems are examined at a conceptual level. We are more concerned with the expression of user requirements and system structure than the component level interfaces between system modules.

The paper reviews data modeling methodologies for GIS-specific applications. In everyday terms we are interested in specifications, design artifacts, and database schemas for land information systems. Two forces motivate our work. Firstly, we believe appropriate attention has not been given to design and analysis methodologies within the spatial information industry. A great deal of attention within the GIS scientific community has focused on spatial query operators and spatial access methods. By account, the amount of research into application level design has been lacking. This is despite the fact that a larger number of people would be drawn into defining system requirements and design objectives for a normal application than those people tasked with building the system. While many papers espouse the virtues of capturing user intent, the fact is there are very few tools or guides to assist in GIS conceptual modeling. A second motivating force is that most user requirements are either expressed or understood to relate to a small set of concepts, e.g. maps, coverage's, networks, etc. While new paradigms will eventuate, we believe most applications can be expressed in terms of a key set of design concepts. These design elements define high-level type systems with application specific semantics.

The contribution of this paper is to:

- review the applicability of object-oriented design and analysis methods for GIS
- describe object-oriented design elements for application objects used in GIS
- test the interpretation of a spatial metamodel in the context of land information products (PROPOSAL ONLY).

This paper looks at interoperability from an application perspective. The next section describes the setting for the development of information products in a statewide land information system. This is called the Queensland Spatial Information Infrastructure Strategy (QSIIS). Section 3 discusses object-oriented design and analysis (OODA) methodologies in relation to modeling techniques. Section 4 reviews specific OODA modeling languages for spatial classes. Section 5 describes how a general OODA modeling language may be extended to include spatial requirements. It describes this in the context of information products being developed for QSIIS. The conclusion describes a proposal to test the usability of our geomodel and its applicability to providing interoperable tools in a statewide networked information system.

2. QSIIS Interoperability Environment

The Queensland Spatial Information Infrastructure Strategy (QSIIS) is a plan for providing spatial information services to Queensland. It is largely driven by State Government departments, and its aim is to provide 40 information products that are defined by the collection, maintenance, and use of spatial data sets by each department. This is based on the assumption that the departments involved (13 out of a total of 18) are representative of the main purchasers and providers of spatial information (Alexander-Tomlinson, 1997).

Each of the 40 information products is developed to fulfil a particular set of requirements. For example, the property interests analysis product (currently under development) is concerned with land administration. Currently, to obtain full information about a property, users must search across multiple sites holding land information records, with related issues in timeliness; incomplete or non existent electronic records and difficulties in knowing where to search. The property interests analysis information product aims to allow single point access to all property based information. This is expected to provide benefits including:

- increase efficiency and accuracy in assessing land values and setting rents for government land
- simplify bureaucratic processes for registration of interests in land (with implications for decision making on land use)
- make information about native title claims available to avoid potential loss in international competitiveness (Alexander-Tomlinson, 1997).

The immediate concern of system designers is to make information available across a number of different sources by providing interoperability. However, as indicated by the above list of requirements, for users the concern is more focused on the outcomes of the availability of information, in terms of what economic and administrative impact it might have.

3. Information Modeling

We need to build a model for a geographic information system when it is too difficult to comprehend the system in its entirety. As the complexity of the system increases, so does the importance of good

modeling techniques. Some essential factors for geographical application development are the use of a standard application design methodology (Hadzilacos and Tryfona, 1996) and a model specification using a rigorous modeling language. Geographical applications make use of mathematical models of space and their representation, including geometry, map projections, cost surfaces, statistical rendering, etc. If modeling space from a mathematical perspective is considered to be an open problem then it is not surprising that geographic application methodologies are also open to various interpretations. This is evident in the constant tension between users who want semantically expressive models for their particular application domain and vendors who need to provide general purpose tools. So given that it is a fallacy to believe in the existence of a *universal data model* for geographic reality (Morehouse, 1990), it seems appropriate that we concentrate our energies on describing metamodels from which we can build different models.

So what are metamodels? If models are representations of some reality then metamodels are descriptions of models. Metamodels use a rule language to define all well-formed models that may be represented. Likewise to flexibly describe a metamodel, in theory, sometimes requires a meta-metamodel. Architecturally this is shown in Figure 1. A meta-metamodel defines an abstract language for specifying metamodels. A metamodel is an instance of a meta-metamodel, which is a language for specifying models. As with any modeling process, the model cannot be expected to provide complete knowledge of its subject but a good model should provide a reasonable interpretation of the real situation.

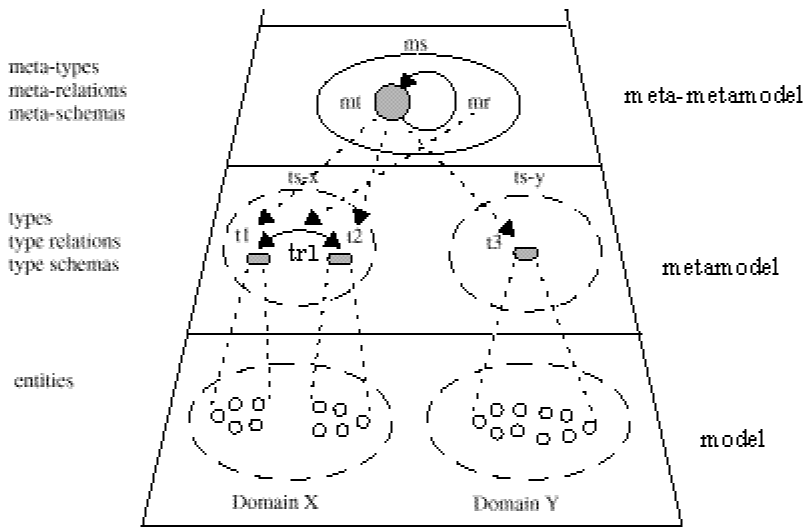


Figure 1: Meta-metamodel, Metamodel, Model Architecture (Source: Crawley et al., 1997)

The meaning of the abstract concepts in Figure 1 may be understood using some real world illustrations. These examples are adapted from Crawley *et al*. (1997) A meta-metamodel would be a type system for CORBA IDL, then the metamodel would contain schemas of CORBA IDL types and relationships between them, and the entities in the model would be CORBA objects. Alternatively, if the meta-metamodel defined a language system for design notations, then the metamodel elements would be specific notations, and the entities in the model would be design diagrams.

We want users to construct object models for geographic applications. The metamodel language we chose to describe these models is the *Unified Modeling Language* (UML). It represents the best practices in the object technology industry derived from leading object-oriented methods, including OMT (Rumbaugh *et al*., 1991). UML (1997) is "a language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems." UML's modeling language includes:

- Model elements **—** fundamental modeling concepts and semantics
- Notation **—** visual rendering of model elements
- Guidelines **—** idioms of usage within the trade

The most prominent aspect of UML is its visual notation. It includes a methodology to capture the dynamic aspects of a model, such as interactions, collaborations, and state histories of objects in a system. A static *class diagram* is used as an integrating framework for the system specification. The class diagram shows a collection of declarative (static) model elements, such as classes with their contents and relationships. UML has the advantage that it is relatively easy for professionals to understand and to facilitate their involvement in the design process. But still includes formal definitions for the structural aspects of class diagrams and a set-theoretic language for expressing well-formedness constraints.

A brief description of some modeling elements for UML class diagrams is shown in figure 2. Class symbol is a composition of a class name, its attributes and operations. A binary association is shown as a line connecting class symbols. The end of an association where it connects to a class is called an association role. Roles signify important aspects of associations including multiplicity, ordering, qualification, navigation, and aggregation relationships between classes. Association paths are adorned with an association name and constraints. A constraint is a semantic relationship among model elements that specifies conditions and propositions that must be maintained as true. It is shown as a text string in braces {}. All model elements that include text strings (such as class names, attributes, association names, constraints, etc.) represent information that has both syntax and interpretation. We have presented just a small subset of UML notations, a detailed description can be found in the UML documentation set (UML, 1997). Our experience suggests that UML's combination of a visual notation and some formal language rules provide a reasonable balance between expressiveness and readability.

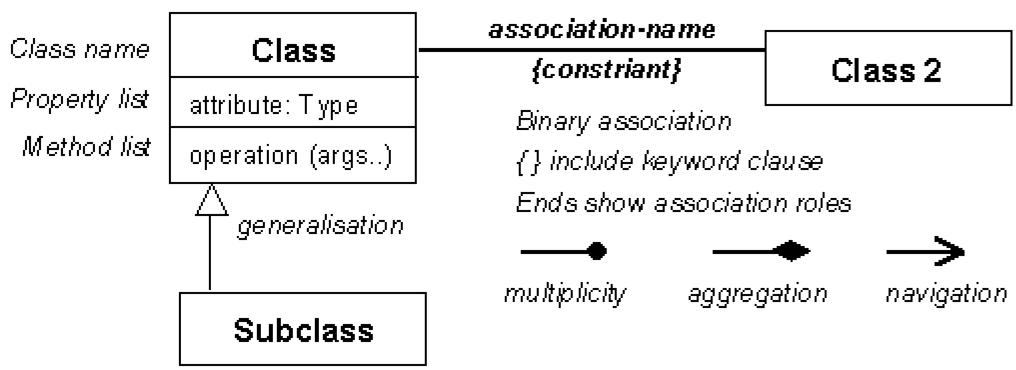


Figure 2: Notations for several model elements in UML

UML supports some core model elements that are used to define models. It also includes language extensions for specifying process-specific models, for instance to define a special type of association between classes. These metamodel constructs are sufficient for a user to create a customized specification of an application specific domain (without having to resort to a meta-metamodeling layer). The next section describes GIS specific models that have been proposed. Section 5 describes the extension mechanisms in UML and develops GIS application specific models.

4. Literature Review

Although a number of modelling methods have been used since conceptual modelling first became a distinct step in the application design process, recent years have seen a general acceptance of the object oriented method as suitable for GIS modelling (De Oliveira et al., 1997). However, a recent criticism of object-oriented modelling is that there are no standard practices or commonly accepted conventions. Worboys (1994) examines the causes for the proliferation of competing models, all claiming to represent a true object oriented modelling methodology. The main cause is the lack of general consensus on the key features and definitions that constitute an object oriented model.

In recent times, the debate over object oriented data models has been diffused somewhat by distinguishing between object oriented analysis and design (OOAD) methods, and object oriented programming (OOP) languages and databases (Booch, 1996). It is important to make the distinction between modelling concepts and implementation tools. OOAD provides the method for analysing problems and framing them in terms of objects. The benefits of

OOAD for an application can be realised without the need for an OOP implementation, but using OOP to build system applications has obvious advantages. Similarly OOAD can be implemented on a non-object oriented database management system.

Although the segregation of OOAD and OOP has gone some way towards identifying a standard set of characteristics of an object oriented model, the model has been applied to GIS in a number of different ways. Four such applications are described here:

- GeoOOA, which is an extension of the object oriented model (Kosters et al., 1997);

- GISER, which is an extension of the Enhanced Entity Relationship Model, itself an extension of the Entity Relationship Model to include some object oriented concepts (Shekhar et al., 1997);

- USM*, which is an extension of the Unifying Semantic Model, and is specifically distinguished from the object oriented model by the authors, although it exhibits many similar characteristics to the object oriented model (Park, 1997) and

- GMOD, which is an extension of Camara et al. model, itself an extension of the object oriented model (De Oliveira et al., 1997).

The remainder of this section reviews these four models in terms of five basic questions:

- What problems are the models intended to solve (that is, why were they developed)? -

- What representational semantics do the models include?

- How do the models handle space and time?

- What mechanisms do the models have for interchange and repository?

- What metamodel characteristics do the models provide?

4.1 Problems addressed by the Models

The four models differ in the problems they are intended to solve, or the justification for their existence. GeoOOA defines the following problems of the conventional object oriented model, defining an extended model to solve them:

-- there is no easy way to distinguish between spatial and non-spatial classes;

- there is no obvious way to determine the type of a spatial class (that is, point, line or region);

- there is no distinction between topological and conventional relationships;

- important topological constraints are hidden in textual specifications and

- frequently occurring topological constraints lead to redundant textual specifications (Kosters et al., 1997).

GISER's main contribution is to address the problem of the unification of field-based and object-based spatial data model approaches. It includes concepts for the discretization of field-based models using an interpolation technique (Shekhar et al., 1997), as well as procedure valued attributes and support for the entire GIS process (input, modelling, manipulation and presentation) (Shekhar et al., 1997).

USM* focuses on the provision of tools as part of a problem solving environment for ecological issues. Its main aim within this is to promote model reusability and provide a tool for high-level (semantic) specification of models. The solutions the model provides include:

- model management tools, incorporating access and assistance in modelling (to increase reusability);

- the ability to specify and store simulation models for ecological processes and

- visualisation services (Park, 1997).

Finally, GMOD's contribution to the use of object oriented modelling for geographic objects has some of its justification in common with GISER. It attempts to isolate users from implementation details, allowing the inclusion of semantics (De Oliveira et al., 1997).

4.2 Representational Semantics provided by the Models

The representational semantics provided by GeoOOA directly mirror the criticisms of the object oriented model made by the authors and outlined above. The model defines a syntax and semantics for all GeoOOA primitives and their standard attributes, services, topological relationships and constraints. It is visually explicit in that it ensures the object and their intent are visible (Kosters, Pagel and Six, 1996).

GISER provides a functional view of geographic phenomena as input, data modelling to extract spatial information content, manipulation and result presentation. Special notations are provided for non-entity information (in particular, field based model elements) (Shekhar et al., 1997).

USM*'s main contribution is its use of the semantic model to represent concepts which describe observations and the logical relationships that hold them together (Park, 1997). It does this by providing a number of multidimensional constructs (Park, 1997).

USM* is distinct from the other models in that its authors explicitly reject the object oriented model, saying that it is typically tailored to a specific implementation system. Despite this, it exhibits many of the characteristics that are typically object oriented (for example, encapsulation and aggregation) (Ram and Park, 1996).

GMOD's representational semantics define classes for both spatial and non-spatial objects. The spatial objects are not immediately obvious from the example diagrams (refer to GeoOOA's justification for existence). GMOD allows progressively more detailed models to be developed in the typical style of OOAD (Rumbaugh, 1991). Spatial representational details are invisible at the conceptual level (hence the inability to identify them from non-spatial entities in the absence of specific notation differences) (De Oliveira et al., 1997).

4.3 Model Concepts of Space and Time

GeoOOA defines geoclasses for point, line, region and raster. These classes encapsulate specific geometric behaviour and topological constraints. The model defines special composition notations for the network structure (including edge links and node junctions). In addition, it provides a particular association relationship: the whole-part composition, which can have a covering, containment or partition structure (Kosters et al., 1997). Spatio-temporal objects are also supported using a timestamping function (Kosters et al., 1997).

GISER represents the continuous field class using space/time extents, from which features (both names and unnamed) are identified in space. These may then be discretized to spatial objects in a coverage. The model also supports derived features and relationships (Shekhar et al., 1997).

USM* supports spatio-temporal and dynamic classes. The dynamic classes are specifically designed to incorporate the simulation models that are the focus of the ecological decision making system for which it was designed.

Relationship constructs that are provided by the model include spatio-temporal and causal relationships (Ram and Park, 1996).

GMOD divides its classes into conventional classes and geoclasses. Geoclasses can be either geoobjects (object-based view) or geofields (field, or continuous view) (Camara et al., 1994). Both geoobjects and geofields have an attribute which describes their georegion (geometric location). Time is included through the definition of a time class, which any other object can have as a component (De Oliveira et al., 1997). GMOD includes two temporally based relationships: versioning and causation (De Oliveira et al., 1997).

4.4 Interchange and Repository Mechanisms provided by the Models

Of the literature reviewed for the four models, only the discussions of USM* explicitly identified a repository mechanism. The USM* repository has three components:

- a metadata directory, which includes the USM* model created by users;

- a mapping directory, which maps the metadata to the underlying database and

- a model description, which describes the various simulation models (Ram and Park, 1996).

GMOD provides interchange mechanisms, and is intended to be an open model. A method for connection of the model system to a commercial GIS using an interface is provided. An External Driver layer is responsible for conversion between the underlying GIS and GMOD, so that the user has the benefit of GMOD's semantics (De Oliveira et al., 1997).

4.5 Metamodeling Characteristics Provided by the Models

All of the models reviewed (based on the literature examined) were very much focused at the model level (see Section 3), but also included discussion of the application of the model at the object level. None of the models addressed metamodeling or meta-metamodeling.

The questions examined above in relation to each model indicate that there are significant similarities between the models in terms of the problems they hope to solve, the representational semantics they use and their concepts of space and time. The differences in these three areas relate mostly to specific details of dealing with individual spatial classes and relationships.

There is more variation in the mechanisms provided for interchange, including tools like the repository. Some models do not address them at all, while others include detailed allowances for such uses. None of the models provide a metamodeling level.

The intention of this paper is not to suggest that these models (and many others that are similar) do not provide adequately for the modelling of geographic data. A large body of literature, including that reviewed here, indicates that these extensions are useful to some degree. Instead, this paper points out that none of the models reviewed here (and no other geographic models that we are aware of) provide metamodeling tools. The provision of metamodeling tools (like UML) is significant in that it removes the need for explicit extended models to be developed, as geographic models are supported by the tools provided at the metamodel level.

The next section shows how UML (a metamodeling tool) can be used to define geographic models without any need for extension to the tool itself, since the metamodel level provides for definition of particular stereotypical constructs.

5. GIS Application Specific Extensions to UML

The UML metamodel includes built-in mechanisms to facilitate domain-specific extensions to its metamodel without needing to resort to a meta-metamodeling layer. These are essentially variants of the core modeling elements (i.e. class and association) and their semantics (constraints) that can be tailored for specific application areas. UML supports extension mechanisms using stereotypes, tagged values, and constraints (UML, 1997):

> **Stereotypes** may extend the semantics, but not the structure, of pre-existing types and classes in the metamodel. Certain stereotypes are predefined in the UML, others may be user defined. The general presentation of a stereotype is to place the name of the stereotype within matched guillemets «*stereotype name*» or depict it by a graphic icon appropriate for the model element being described.
>
> **Tagged Value** is an explicit definition of a property as a name-value pair. In a tagged value, the name is referred as the tag. Certain tags are predefined in UML, others may be user defined. A tag is attached to a model element as a comma-delimited sequence of property specifications with the format $\{keyword_1 = value_1, keyword_2 = value_2, .. \}$.
>
> **Constraints** are semantic conditions on the relationship between model elements expressed as a text string. Certain constraints are predefined in the UML, others may be user defined. UML does not prescribe the language in which the constraint is written, but ideally a process-specific constraint is described in a formal language with a specified syntax and interpretation. A constraint is shown as a text string in braces { }.

It is very conceivable that the models described in Section 3 may be specified using the process-specific extensions from UML. Taking GeoOOA for example. Stereotypes may be used to describe *geoclasses* with an identical graphical syntax. Topological *whole-part* structures, including specializations for *covering, containment*, and *partition*, may be defined as association stereotypes. This would appear as a text string (as yet there is no graphical syntax in UML for user defined association stereotypes). The *abstract network* structure may be expressed as a constraint between association paths for *network-link* classes and *network-node* classes. The semantics for geometric standard services may be completely specified using behavioral model aspects of UML.

Therefore UML is capable of expressing concepts of space and time using special geomodeling constructs. It provides the same desirable features as GeoOOA; such as visible distinction of spatial classes and explicit topological constraints. Both syntax and semantics can be expressed in a mathematical language to enforce domain information. UML also considers a supplementary methodology (i.e. guidelines) as an essential part of building models for complex systems. The additional benefits of using UML are that it allows tailoring of the metamodel to allow for variant models, and provides a mechanism for model interchange and interoperability.

5.1 Example of a UML Geomodel

As an example of a UML metamodel we will relate our own experience of developing geographical applications. One of the more surprising aspects we have encountered is that users make explicit reference to very few spatial concepts in their requirements. Often the spatial concepts are implicit in the business case for the system of interest. Spatially explicit requirements (dimension, scale, and accuracy) are collected during the final stage in the requirements process. In the initial stages we found that users frequently make reference to (1) the scope or realm of spatial information and (2) to its thematic qualities. The notion of a *realm* appears to be important in several phases of the application development methodology. At a conceptual level a realm defines the sphere of influence of geographical information. This eventually is refined in the design phase to a map view control with a map projection, area-of-interest, display rendering, etc. A *theme* class represents a set of spatial entities with similar properties that are part of the realm, it corresponds closely to a map theme. Users often make reference to semantically rich descriptions of spatial *themes* without explicitly namely their spatial characteristics. This eventually is refined in the design phase to express spatial properties such as dimension, topological relations, scale, accuracy, scale-dependent display characteristics, etc. Figure 3 shows stereotypes for realm and theme classes with an aggregation relationship.

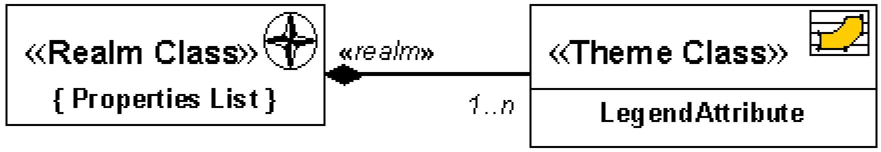


Figure 3: Class Stereotypes for Geomodel

5.2 Pest Information Application

We describe a sample application dealing with land pest infestations to demonstrate the use of these geomodeling elements. The requirements for the pest infestation information system are as follows. Plant and animal pest *infestations* are recorded by an *inspection* in relation to land ownership *properties*. Land resource officers, who each have an assigned district, make the on-site *inspection* and enters it into their *field system*. Besides entering infestation details they may optionally enter the *extent* of the infestation from a GPS traverse or by a free hand sketch. Periodically, each land resource officer connects to the *central office system* and synchronizes its records, the main office maintains a registrar for each district. The central system is used for decision support on remedial activities and for answering public queries on infestations. Figure 4 shows part of the class diagram. A full specification would include attributes and operations in the class diagrams, and behavioral modeling diagrams for interactions and collaborations between objects.

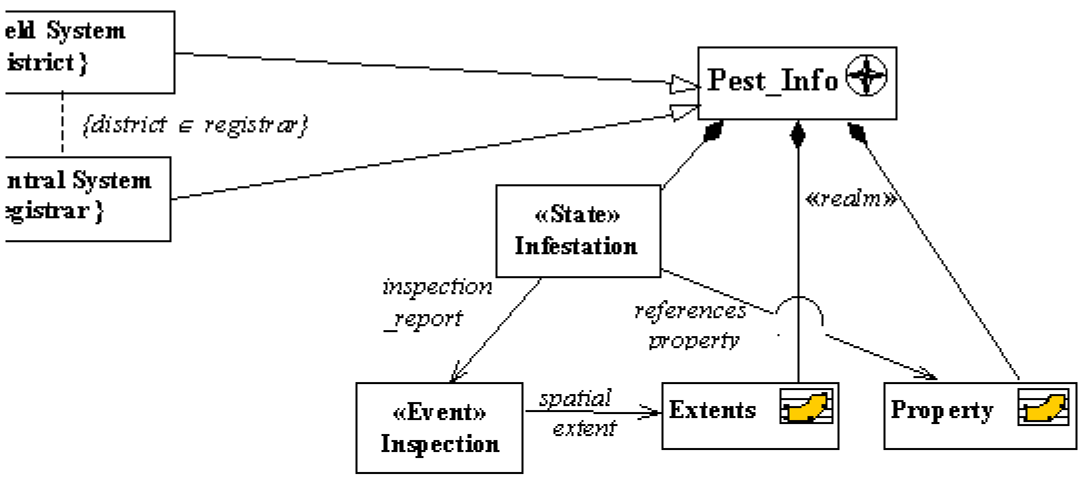


Figure 4. Geomodel for Sample Pest Information System

Even this simple example embodies several complex interactions and interoperable aspects of the application. Firstly the inspection record acts as an *event* to record the current *state* of a pest infestation. These are modeled as a stereotype based upon concepts for temporal systems (Langran, 1992). The spatial extent for an infestation may be measured in the field using a GPS device. Presently this is handled by an "import facility." Periodically the field system is synchronized with the central office system using a "connection facility" or an "interchange facility". The former uses an RPC connection, and the later uses transfer via a file format. Aspects of synchronization are explored further in the system design phase. It is modeled as a replicated database with a replication manager to control authorization and upload/download of data.

The sample application described was developed using Arcview from ESRI. Our work suggests that the extended geomodel from the system design can be translated in a very mechanical fashion into a prototype application. This includes map displays, tables, interface menus, and code scripts. It would then be the responsibility for the system implementers to add operational code consistent with the specification from class diagrams, data dictionary, and the operation dictionary.

We have defined several metamodel elements used in the system analysis and system design. We are currently undertaking a usability study and end-user surveys to ascertain their utility. It is our belief that application design operates in a collaborative environment between end-users, experts, and software engineers. The usability tests will indicate how intuitive and readable models are. We are using four criteria to judge overall readability: usefulness, effectiveness, learnability, and likeability. These criteria are used to determine the success of computer systems (Rubin, 1996).

5.3 Land Information as a "Yellow Page" Service

The development of an information application is often subdivided into several stages: requirements analysis, design and implementation. There is a progression in considerations from conceptual descriptions of the problem, to the computing environment needed, and finally an operational application. UML compels a system level view of application problems. Their definition of a model is semantically closed abstraction of a system. Therefore even during requirements analysis system specific terminology is introduced. This bias is well suited to a requirement analysis for information products described in the Queensland Spatial Information Infrastructure Strategy (QSIIS). Unlike environmental planning and control applications where there is a high level of semantic information related to the real world process (De Oliveira et al., 1997), most of the applications for QSIIS reflect fiscal and institutional systems. In fact most the requirements relate to access of information products and services within a state government organizational structure.

Future developments of applications, like the pest information system, will be part of an online directory. A trial technology architecture was explored in 1996 that linked directory access to several GIS databases. The objective of the trial linkage was to help identify standards and protocols to support interoperable access to spatial information systems (QLIS, 1996). It is simple to connect two sites, but adding several sites with various client users and service problems adds significantly to the complexity. The trial architecture explored linking several sites through a "yellow page" directory. This is implemented as a broker that maintains a directory catalogue of services and providers, which it can look-up in response to user requests. Subscribers, such as Pest_Info, would advertise the services they offer and unadvertise services being withdrawn. This is a very dynamic environment where the QSIIS broker adds and removes Yellow Pages listings as instructed, and also knows how to dispatch user requests to the appropriate service. All user requests are processed through the QSIIS directory and not directly to the service's application. The connected client broker and server broker communicate service transactions. The transaction protocol begins by a user request. The QSIIS broker sets up a communication session to the listed service. The client broker will advise if it can answer the request giving the cost, in time and monetary units, to fulfill the request. The server broker relays this information back to the user, once accepted the server broker using the same session identifier communicates again with the client broker to complete the request. The QSIIS directory records the details of the transaction for business accounting purposes. Figure 5 shows this scenario with a the sample Pest_Info application.

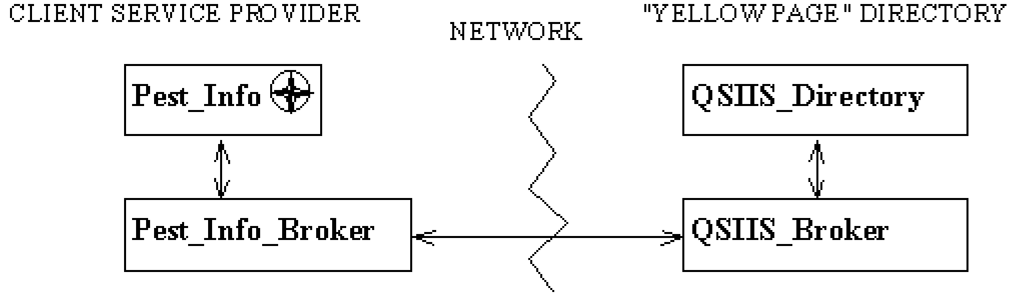


Figure 5: Client-Server Object Classes to list Pest_Info services with "Yellow Page" directory

A prototype technology architecture was built by the Queensland Government in 1996. The system was implemented in C++ and used middleware software, TUXEDO on a UNIX Server, to control multidatabase access. Arcview's SHAPEFILE format was used for interchange of spatial information. It is understood that a proposed system would provide greater dynamics for providers to list and deploy services. A review of this trial confirmed the need for high level interoperable components that adhere to standard protocols. In particular addressing the particular needs of spatial information, its access methods and interpretation, were seen as relying upon future standards efforts like OpenGIS (QLIS, 1996).

It is our belief that OpenGIS will deliver the standard interface specifications, which are in turn implemented by vendor software, for spatial data access and spatial operator interfaces. But it also our firm conviction that there does not exist one geomodel to suit all user communities. We recognize the need for a metamodeling on which service providers can build their own geomodels. Interoperability between these geomodels will rely upon well defined standards and protocols, just as interchange of data relies upon standard formats. A model interchange format should try and balance its formal specification with readability.

To test how well models may be communicated we are in process of conducting usability tests. The tests are based upon the assumption that interoperability of metamodels is qualified in terms of readability and how well it communicates the model semantics between several parties.

6. Conclusion

The future of QSIIS is to support an open system marketplace for access to spatial information and special information services. This will include an interoperable environment where providers list new and integrate existing information products. The machinery for building such a system is not available today, or at least the technology is in its infancy. The strategic development of QSIIS, and similar land information systems, will depend upon communicating meaningful data models that are still readable.

This paper discusses spatial data models with regard to application development methodologies. We have reviewed several methodologies each addressing slightly different problems and user requirements. Each has advanced a geomodel they feel incorporates the necessary concepts for space and time, and representational semantics. But each again is slightly different. Therefore it is apparent there is not one geomodel for all information communities. We have explored how metamodels can be defined using an industry accepted modeling language. We are currently testing the usability of a geomodel for system analysis and system design purposes.

We believe the standards effort for interoperability should progress at two ends of the software engineering spectrum. As is occurring now, the lower end will deliver common data types used to define spatial information classes. Specifications for such data types rely upon mathematical principles for spatial data types. They will provide the basic building blocks to describe the spatio-temporal characteristics of spatial information. The middle part of the spectrum includes defining spatial data models to suit real world situations. This part should retain a high level of user input. In other words system designers should have the opportunity to define their own geomodels (or metamodels) for an information community. This challenges the standards effort to adopt meta-metamodeling facilities to allow designers to describe their own metamodels.

Bibliography

Alexander-Tomlinson (1997). Queensland Land Information Strategy (QLIS) Benefit Study. Volume II: Situation Report (Assessment of Government and Industry Spatial Information Requirements).

Booch G. (1996). *Object Solutions: Managing the Object-Oriented Project*. Addison-Wesley, Menlo Park, California.

Camara, G., Freitas, U.M., Souza, R.C.M., Casanova, M.A., Hemerly, A.S. and Medeiros, C.B. (1994). A Model to Cultivate Objects and Manipulate Fields. *Proceedings 2nd ACM Workshop on Advances in GIS*.

Crawley S., Davis S., Indulska J., McBride S., and Raymond K. (1997). Meta Information Management. *Formal Methods for Open Object-based Distributed Systems (FMOODS) Conference*, Canterbury, UK, July 1997.

De Oliveira, J.L., Pires, F. and Medeiros, C.B. (1997). An Environment for Modeling and Design of Geographic Applications. *GeoInformatica* 1, 29-58

Hadzilacos T., and Tryfona N. (1996). Logical Data Modelling For Geographical Applications. *International Journal of Geographical Information Systems*, 10(2).

Kosters, G., Pagel, B. and Six, H. (1996). GeoOOA: Object-Oriented Analysis for GIS-Applications. *Proceedings 2nd IEEE International Conference on Requirements Engineering*, Colorado Springs.

Kosters, G., Pagel, B. and Six, H. (1997). GIS-application development with GeoOOA. *International Journal of Geographical Information Science* 11(4), 307-335

Langran G. (1992) Time in Geographic Information Systems. Taylor & Francis, London.

Morehouse, S.D. (1990). The Role Of Semantics In Geographic Data Modelling. *Proceedings 4th International Symposium on Spatial Data Handling*, Zurich, 689-698

Park, J. (1997). Geographic Information Systems and Problem Solving Environment. *Crossroads* Vol. 4(1), 3-8.

QLIS (1996). QLIS Project Report: Technology Architecture Research & Development Project. The Queensland Land Information Council, The State of Queensland Department of Natural Resources.

Ram, S. and Park, J. (1996). Modeling Spatial and Temporal Semantics in a Large Heterogeneous GIS Database Environment. *Proceedings of the 2nd Americas Conference on Information Systems* (AIS '96), Phoenix, Arizona, August, 16-18

Rubin J. (1996). *Handbook of Usability Testing: How to Plan, Design and Conduct Effective Tests*. John Willey & Sons, New York.

Rumbaugh J., Blaha M., Premerlani W., Eddy F., and Lorensen W. (1991). *Object-Oriented Modeling and Design*. Prentice-Hall.

Shekhar, S., Coyle, M., Goyal, B., Liu, D. and Sarkar, S. (1997). Data Models in Geographic Information Systems. *Communications of the ACM* 40(4), 103-111

UML (1977) Universal Modeling Language (UML) Document Set, http://*www.rational.com/uml.* Rational, Software Corporation, Santa Clara.

Worboys M.F. (1994). *GIS: A Computing Perspective*. Taylor & Francis, London.

Spatial Database Design for Geographic Information System (GIS) Interoperability

Lorri A. Peltz, Marianne H. August, and Rose L. Medina
U.S. Geological Survey
Water Resources Division
Nevada District Office
333 W. Nye Lane, Room 203
Carson City, NV 89706

The Spatial Applications Section in the Nevada District Office is charged with creating or obtaining, developing, maintaining, and archiving spatial databases for District applications. During the past 5 years, database design and development has resulted in a theme-based GIS architecture incorporating specific semantic structures for databases, which includes their attributes. Semantic structures rapidly communicate the theme, scale, and geographic region of each database. Similar semantics are used within these databases that allow attributes, such as measurement units, to be obviously defined. Databases with attributes that include numeric codes have associated, similarly named, character fields providing descriptions for each code. The spatial databases, including vector and raster formats, contain metadata in accordance with Executive Order 12906. Spatial database integrity is maintained by read-only access to users and full access to the Spatial Applications Section. The Nevada Master Database design, semantic structures, physical storage, and metadata procedures are documented in an internal report.

Implementation of the theme-based design has facilitated the development of an improved mechanism for users to browse and retrieve data. The Nevada Master Database ArcView extension loads a menu to the view document and allows the user to interactively select spatial databases through a graphical-user interface. This interface loads the spatial database, a legend file, and the option to display the metadata interactively. The graphical-user interface, integrated with the spatial database design, provides GIS interoperability to the Nevada District staff. Current development focus is on continuing installation of spatial databases, spatial indexing by theme, and improving network/system requirements to support the spatial databases.

A Framework for Geographical Modeling in a Heterogeneous Computing Environment

David A. Bennett and Raja Sengupta
Department of Geography
Southern Illinois University, Carbondale
Carbondale, IL 62901-4514

Greg A. Wade
Department of Computer Science
Southern Illinois University, Carbondale

The resolution of spatial problems often requires consensus building and compromise among decision-makers as they attempt to optimize their own set of criteria. The evaluation of such criteria often requires access to an extensive set of geographical models, analytical tools, and data. The Internet provides new opportunities for the sharing of such resources. Many potential users, however, lack the time, money, and/or technological capabilities need to integrated applicable models, tools and data into a software environment that can support spatial problem solving. To take full advantage of the opportunities afforded by increased access to data and models, new geoprocessing technologies are needed that are capable of bridging multiple vendor formats and heterogeneous computing environments.

To construct links between disparate software products and data formats, a common communication protocol is required. Such a protocol can be implemented if software products can import and export to a common framework or if software vendors provide "hooks" into proprietary data structures. Communication protocols are in development for geographical databases (SDTS, OGIS), however, there is little support for the distribution and sharing of the geographical models and analytical tools. The objective of this research effort is to develop a framework that supports collaborative decision making in a client/server environment that provides access to a variety of geographical models, analytical tools and data. Such an environment will provide decision makers with: 1) access to computing resources that may not otherwise be available; 2) an electronic forum for the exchange of ideas in written and/or cartographic form; and 3) a more level "technological playing field" on which to build consensus and compromise.

To reach this objective we are pursing three interrelated research objectives. These objectives include the development of a(n):

1. framework for distributed geographical modeling;
2. distributed modelbase and database management systems;
3. intelligent search engine driven by spatial metadata and a geographical modeling language

The framework presented here builds on previous work by Bennett (1997; in review), and Wade et al. (1997). Existing frameworks for the representation of geographical data are built on data models that treat the geographical database as a digital surrogate for geographical space. Yet each digital representation of geographical space is an abstraction of reality created by the user to solve a particular set of related problems. As such, the selection, organization, and implementation of those spatial elements that comprise this abstraction depends on such factors as our understanding of how spatial processes operate, the objectives of the analyst, data availability, and the spatial extent to be studied/managed. Geoprocessing technologies designed to support the study and management of complex geographical systems must, therefore, integrate methods for the representation of geographical knowledge with more traditional methods for the representation of geographical space.

In the modeling framework presented here, knowledgebase management, modelbase management and geoprocessing technologies are integrated into a single system that supports the digital representation of dynamic geographical systems. Model design is viewed as the capture of geographical knowledge and the organization of this knowledge into a model graph that emulates the spatial processes of a particular geographical system. The representation of geographical systems as graphs provides a modeling topology well suited to a distributed implementation. The management of model graphs over a computer network required the development of new interoperability tools. These platform independent tools were designed to: search network accessible repositories of geographical models, atomic model components, and geographical data; provide an interactive mechanism for integrating geographical models from atomic components; and execute these models across a distributed modelbase and database. Prototype software was developed using Java and its extensions (Wade et al. 1997). Java was created for developing network aware applications and possesses unique features that facilitate the development of software designed to be executed in a distributed environment. These features include platform independence and the ability to dynamically load and bind compiled code over the network.

The construction of geographical models from atomic components often requires the coupling of data and models derived from multiple sources. Existing coupling strategies (e.g., loose and tight coupling) that link GIS with modeling software are often either too complex or too rigid. A new method of linking GIS and analytical models is proposed here that builds on earlier work by Sengupta et al. (1996). Using the modeling framework described above as a guide, intelligent agents match datasets (irrespective of vendor formats) to models and create wrappers around modeling software and GIS datasets. The existence of these wrappers is largely hidden from the user's view.

Intelligent software agents communicate via a Model Definition Language (MDL) to integrate data with models. The MDL provides an inter-agent communication protocol for model development. Through the use of the MDL, agents retrieve, manipulate and store geographical databases and modelbases. To accomplish this task agents parse an MDL query and translate tokens into a sequence of software specific spatial operations that transform the data into a form that is usable by particular models. Within the MDL spatial queries are defined using topological relations (Egenhofer and Fransoza 1991; Clementini et al. 1993), spatial operators (Tomlin 1990; Wesseling et al. 1996), and constructs from the C programming language. Multiple agents may be invoked in the process of performing a spatial analysis. Human intervention is required only to state preferences for models and data source to be used. The agents can also act as advisors that suggest appropriate model selection and lead the user through complex spatial analyses.

The search for relevant data and the identification of data format is achieved through the use of metadata. The metadata format adopted follows the Spatial Data Transfer Standards. The Geographical Name Server (GNS) proposed by Wade and Bennett (in press) was used to further facilitate the identification and processing of this metadata by intelligent agents. The GNS provides a framework for creating a hierarchical topical nomenclature for geographic data which is required to effectively and efficiently automate the process of interpreting metadata.

References:

Bennett, DA in review. Managing geographical models as repositories of scientific knowledge. Submitted to Geographical and Environmental Modelling.

Bennett, D.A. 1997. A framework for the integration of geographic information systems and modelbase management. International Journal of Geographical Information Systems 11(4): 337-357.

Clementini, E., Felice, P. D., VanOosterom, P., 1993, A Small Set of Formal Topological Relationships suitable for End User Interaction. In Advances in Spatial Databases, Proceedings of the Third International Symposium on Spatial Data Handling (Lecture Notes in Computer Science No. 692), Ed. D. Abel and Ooi, Beng Chin . New York: Springer Verlag: 277-295.

Egenhofer, M. D., and Franzosa, R. D., 1991. Point Set topollogical spatial relations. International Journal of Geographic Information Systems, 5(2): 161.

Sengupta, R., Bennett, D.A., and Wade G.A. 1996. Agent mediated links between GIS and spatial modeling software using a model definition language. In Proceedings of GIS/LIS '96, Bethesda, MD: American Congress on Surveying and Mapping: 295-309.

Tomlin, C. D., 1990, Geographic Information Systems and Cartographic Modeling (New Jersey: Prentice Hall): 249.

Wade G.A., and Bennett, D.A. in press. GNS: A Distributed hierarchical topical nomenclature for geographic data. In Proceedings of GIS/LIS '97.

Wade G.A, Bennett, D.A., and Sengupta, R. 1997. An interactive distributed architecture for geographical modeling. In Proceedings of Auto-Carto 13. , American Congress on Surveying and Mapping, Bethesda, MD: 307-316.

Wesseling, C. G., Van Deursen, W. P. A., and Burrough, P. A., 1996. A Spatial Modeling Language that unifies dynamic environmental models and GIS. In: Proceedings, Third International Conference/Workshop on Integrating GIS and Environmental Modeling, Santa Fe, NM, January 21-26, 1996. Santa Barbara, CA: National Center for Geographic Information and Analysis. http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/main.html.

Automated Metadata Interpretation to Assist in the Use of Unfamiliar GIS Data Sources

Brandon Plewe and Steven Johnson
Department of Geography
Brigham Young University

The Open GIS Consortium (OGC) has been actively involved in the development of technical means for allowing Geographic Information Systems software to incorporate data from heterogeneous sources, both internal and external to an organization. Recently, several vendors have introduced software based on the OGC standards which begins to fulfill this vision. As with any advance in technology, interoperability has the potential to greatly enhance our work, but can also be misused, whether intentionally or not.

One of the abuses which becomes more possible with interoperability is for people to use data sets for applications for which they are not well-suited. This is largely due to the fact that GIS users will be more often using and combining data sets with which they are not intimately familiar. Examples of problems include: creating large-scale maps from low-accuracy data, combining datasets digitized from maps of very different ages, and misinterpreting attributes and classification schemes.

Metadata has long been touted as the solution to the problem. If users read the metadata, they will become familiar with the data set and will be able to make good judgments about its proper use. However, metadata records based on standards such as the FGDC Content Standard for Geospatial Metadata tend to be extremely complex, and difficult to read and understand for all but the creator. However, their complex structure is well-suited to automatic parsing by computers.

This research is building a prototype system for the automatic interpretation and use of metadata in a standard GIS environment (ESRI Arcview). As themes are added to a GIS project, their associated metadata records are also retrieved, and pertinent elements are parsed from the metadata record. This information includes the age of the dataset and its sources, the projection and coordinate system, general horizontal and vertical accuracy, quality of source information, subject matter, spatial footprint, and explanation of attributes and associated classification systems. The initial prototype focuses on issues of scale and accuracy (e.g. comparing the intended scales of two themes being displayed together). The metadata records are expected to be text files using the FGDC-standard SGML DTD, and are thus relatively simple to parse into fields. Some pieces of information can be gathered directly from the value of a single field (e.g. Horizontal Positional Accuracy Value), while others may require more extensive analysis of textual information (e.g. Horizontal Positional Accuracy Report).

This information is used throughout the GIS session when needed. For example, when a theme is added to a view window, the characteristics of the new dataset (e.g. time period, accuracy, coordinate system) are compared to those of the existing themes, and checked for compatibility. Other appropriate applications include changing the view scale (zooming in and out), performing queries, and creating maps. These hooks are added to the traditional operation interface to make the metadata system transparent; that is, the dialog boxes appear similar to the standard operations, but with added buttons or information.

When checks are performed against the metadata, a user interface assists users in their subsequent actions. This can take three forms: help boxes with descriptive text, warnings or advisory messages, and locks to prevent users from performing incorrect operations. For example, when the user adds a theme to a view, and there is a mismatch in the metadata, he or she may be presented with a table of metadata fields showing the conflict, asking them whether they still wish to add the theme. Alternatively, they may see a message box warning them of the possible repercussions of the mismatch, or be prevented entirely from adding the theme (with a proper explanation). Users can generally control (using global preferences) which level of assistance they would like. They also always have the opportunity to override the checks if they so desire. At any point, the user may directly view elements of the metadata which are pertinent to the task at hand.

In addition to preventing or warning users of potential abuses, automated metadata can also be useful for automating standard GIS processes. For example, when a theme is added to a view in Arcview, the metadata could be used to automatically give the theme a meaningful name, or set the initial scale and measurement units of the view window, or even select an initial symbology which is appropriate to the subject matter of the theme (e.g. green for a vegetation layer).

Although the system is still under development, the intended result is that the metadata are used as a partial solution to a serious problem in GIS. The advanced GIS user will be able to more intelligently use data sources with which he or she is not familiar, without having to read and understand the entire metadata records. Novice users, who may not understand many of the technical concepts discussed in the metadata, are able to use data sources more correctly, often without even realizing it.

Software Agent-Oriented Frameworks for the Interoperability of Geomatics Systems: from Fundamental Concepts to the SIGAL Project

Z. Maamar[1,2], B. Moulin[1,2], Y. Bédard[2], and G. Babin[1]
{maamar,moulin,babin}@ift.ulaval.ca, yvan.bedard@scg.ulaval.ca
Computer Science Department[1] and Research Center in Geomatics[2]
Laval University, Ste-Foy, QC G1K 7P4, Canada

1. INTRODUCTION

Within an organization that manages spatially referenced data, several types of documents are used to describe the land and its resources (ex. topographic maps, land use maps, aerial photographs, satellite images, and so on). Managing such documents is a complex task; each one is characterized by its own content, spatial reference system, quality, sources, mode of distribution, and format. Hence, Georeferenced Digital-Libraries (GDLs) can be very useful and helpful to manage this meta-information. GDLs are information bases describing the available geodocuments resources in an organization. As a result, GDLs can improve knowledge of the nature of data, identify the responsibility of "who does what, when, and how", and inform about the physical location of the documents.

There exists a number of well known GDL projects, such as Alexandria Digital Library (University of Santa-Barbara), Digital Library Project (University of Berkeley), GEOREP (Laval University), and so on. Their objectives are to help users to identify data which may be useful to them, to help producers to increase accessibility to their spatial data for potential users, and to encourage the sharing of spatial data between organizations. Users are accustomed to use these GDLs independently from each other. Yet, a more complex task is to use different GDLs, generally distributed and heterogeneous, which overlap or not on a given geographic area. This new reality claims for new alternatives for GDLs interoperability.

At Laval University, our research group under the supervision of Profs. B. Moulin and Y. Bédard is working on several GDL projects, including developing an interoperable environment for GDLs. This latter project is called SIGAL (French acronym of Geographic Information System and Software Agent). Such an interoperable environment will provide users with services which will not require them to know individual characteristics of the interconnected GDLs (a concept similar to the concept of meta-engine used to search on the world wide web).

The design of the SIGAL architecture applies principles elicited in the field of information systems interoperability, namely that an interoperable environment must:

- Maintain the autonomy and independence of the systems to be interoperated, while allowing these systems to interact despite their disparities.
- Reduce the informational disparities of the various interconnected systems by

using a knowledge which is understood by all these systems.

- Help users satisfy their needs without worrying about the distribution and disparities of the information provided by the interconnected systems.

The interoperability process has to keep the local systems, in general, and GDLs, in particular, autonomous and independent from this process. In order to reach this goal, we suggest to introduce specialized components, called software agents, based on development techniques borrowed from Distributed Artificial Intelligence. These agents are the front-ends of the systems, have the capability to act on their behalf, and can assist users to fulfill their needs. Furthermore, these software agents can be of different types (for example, facilitator agents identify the systems on communication networks, mediator agents support interactions between systems, etc.). However, given the complexity of managing operations in distributed and heterogeneous contexts, agents can be gathered into teams, which in turn evolve within what we call software agent-oriented frameworks.

2. An Application of Software Agent-Oriented Frameworks, the SIGAL Project

A software agent-oriented framework offers a set of services that can be requested either by users or by other frameworks. A framework is an environment composed of a supervisor and one or several teams of software agents. The services provided by a framework are performed by different teams set up by the framework. These teams are composed of several agents which are specific to the application to be developed and to the characteristics of the information systems to be interconnected. In the SIGAL project, our application aims at developing an interoperable environment of GDLs which involves three types of frameworks (local-source, server, client).

The local-source framework maintain the autonomy and the independence of the GDLs in the interoperable environment. Therefore, local-source frameworks interface the GDLs with communication networks and process the data requests sent by the client frameworks.

The server framework is the backbone of the SIGAL environment, since it monitors all the operations needed to support the services offered to the users and to other frameworks. In order to avoid overloading the server framework, we suggest duplicating it on mirror sites. However, it is important to maintain the coherence of the information duplicated on the server frameworks and consequently to define reliable update protocols. In the SIGAL environment, these functions are implemented as a set of services supported by server frameworks: a service to modify the informational content of a GDL and a service to connect a new GDL to the SIGAL environment.

When users need information from several GDLs, they invoke relevant services on the server framework. The invocation of such services on the server initiates the creation of a client framework generated on the user's machine. Hence, the server framework delegates operations to the client framework and limits its involvement to their monitoring. Once all operations are executed, the client framework can be either deleted or recorded for further uses.

Services provided by a framework satisfy specific users' needs such as information retrieval, etc. When a service is invoked by a user, the framework's supervisor agent activates a scripting procedure, called a realization scenario, which specifies the characteristics of the teams of agents that will perform the various operations required to carry out the service. According to that realization scenario, the framework supervisor creates teams that will play roles specified in the scenario. At their own level, team supervisor agents activate realization scenarios in order to coordinate the activities of their software agents.

For instance, a data request directed to a GDL of the SIGAL environment is defined by an access scenario which involves an agent. It possesses knowledge structures needed to transform a user's request into a data request expressed in a form compatible with the data manipulation language of the GDL. This agent activates the access plan which queries the system, in order to process the data and to transmit the answer to user.

3. Summary

To summarize, software agent-oriented frameworks can interact in order to define a global behavior which is an outcome of such interactions. These frameworks can be adapted in terms of components (types of teams of agents and agents to integrate) and functionalities (types of services to offer). These frameworks can play several roles according to the application to be developed. Finally, these frameworks can constitute an interoperable multiframework environment.

Our aim for the conference is to present the major characteristics of the SIGAL project that deals with GDLs interoperability. The results presented are part of a larger project whose objective is to propose a design method for interoperable environments based on software agent-oriented frameworks.

The SIGAL project has been the object of continuing research. We are currently testing the framework architecture described in this paper. SIGAL's prototype uses JAVA as a programming language, the JDBC driver to connect the available informational resources, i.e. GDLs, and finally the Object Request Broker VisiBroker for JAVA to specify the behavior of the SIGAL environment's components (frameworks, teams, and agents) during distributed operations.

Using the Internet to Access Geographic Information: An OPEN GIS interface prototype

Frederico Torres Fonseca
Clodoveu Augusto Davis Jr.
PRODABEL - Empresa de Informática e Informação
do Município de Belo Horizonte S.A.
Av. Presidente Carlos Luz, 1275
31230-000 - Belo Horizonte - MG - Brazil

SUMMARY

Interfaces to access geographic data can be designed in a number of ways. When the design takes into consideration interoperability concerns, the software's architecture becomes one of the main issues. In the interfaces that intend to be interoperable, the design often includes a module that is responsible for the interaction with the user, and another which implements the connection to the GIS. The distribution of tasks between the user interface and the connection module will be discussed here. The design of an interface that allows the user to gain access to geographic data stored in one or more GIS, using the Internet as a communications medium, will be proposed as an alternative.

The architecture of this proposed solution is presented, along with its weak and strong points. In this approach, direct manipulation and menu selection tools are used to build the user interface. The connection to the GIS has been implemented to reach a single source of geographic information, but given the interoperability features included in the interface's design, the potential to access several different GIS is assured. The use of Java, a multiplatform, object-oriented language, and its potential to build geographic interfaces based on an object model, is also discussed.

OBJECTIVES

This project has started with the objective of creating "middleware" to allow an user to gain access to geographic data from several different sources, using a simple object-based model. The intention was to provide the widest possible range of users with easy and inexpensive access to public GIS data. The initial concept called for the development of an interface between the user and the GIS. Initially, this interface would only provide basic data access, and should evolve in the future to support the definition and implementation of more specialized services. However, the ability to connect simultaneously to different GIS platforms has been considered important to the success of the project.

As a development platform, the Java language has been chosen. Java includes features that provide for easy client-server computing, Internet connections and standard database access, in a multiplatform, object-oriented environment. Java applets can be made available through the Internet (or any Intranet), cutting down on software maintenance and distribution costs. The Internet also provides an excellent medium for distributed applications, as required by the project's objectives.

ARCHITECTURE

This project works with the concept of geographic objects, that can be extracted from a geographic database to be presented to the user. The modeling of the objects are based on the OPENGIS Geodata Model, although it will not be completely implemented. Geographic objects are modeled as instances of classes derived from a generic class, called GeoObject. This design suggests some basic derived classes, such as Point, Line and Polygon, but new object classes (such as Raster or DTM) can be derived from GeoObject in order to provide support for more specialized data models, as implemented by existing GISes.

The interface has two main modules: the manipulator and the extractor. The manipulator is responsible for the user interaction. Through it, the user can select information that he wishes to see on the screen. The manipulator translates user interaction into requests to be fulfilled by the extractor. The extractor works (1) receiving manipulator requests, (2) connecting to the GIS that manages the desired information, formulating queries and retrieving results, and (3) returning the retrieved results as objects to the manipulator. In turn, the manipulator will handle data visualization issues. Observe that the manipulator module, since it deals only with generic GeoObjects, can be developed to be independent from the GIS that manages the geographic database. Only the extractor module needs to be developed considering specific aspects of the host GIS architecture, since it is the responsible for actually retrieving information from the GIS. Since the manipulator can always be left unchanged, the user gets a stable interface, independent from the actual source of geographic data. This source can also be changed, without the knowledge of the user; that is, the user does not need to know from which GIS the data are coming from.

BALANCING TASKS

As proposed, the extractor module needs to include code that reflects specific knowledge about how to query an existing geographic database, and how to retrieve results. Naturally, this implementation will vary, according to which GIS is used to manage the data. In order to develop interfacing capabilities for a specific GIS product, a class needs to be derived from the Extractor class, and it must include the ability to supply the manipulator with objects in any class derived from GeoObject. Classes derived from GeoObject will inherit GeoObject's basic class properties and methods, which will be used by the manipulator. Therefore, the manipulator code can disregard such details by using encapsulation.

In order for this to work properly, there has to be a clear separation of tasks between the manipulator and the extractor. Also, interfacing software that intends to connect to several GIS can never take on functions that are GIS-specific. The degree of specialization of GIS software in their primary functions, such as spatial data retrieval and analysis, could never be achieved by a simple interfacing program. Even considering that our communications medium, the Internet, is frequently congested, it is fundamentally important to avoid making the interfacing code responsible for any geographic functions. This can mean the loss of some performance, but the loss could be compensated by the possibility of keeping the user interface stable and unchanging, GIS-independent. The use of caching in the client machine, as allowed by Java, can also alleviate performance problems. The use of this feature, however, is optional in the implemented architecture, and depends on the implementation of the extractor module.

CONCLUSION

Results achieved with the initial implementation of the above described interface indicate that the proposed architecture, even though it carries the burden of translating data between the GIS and the user interface, has benefits that overcome its limitations. Among the main benefits, we can mention the adherence to open systems concepts, the concentration on the basic functional roles of each component (manipulator, extractor and GIS), and the possibility to progressively incorporate new services and functions. Furthermore, the strong coupling between the extractor and the GIS brings the opportunity for optimization and performance gains, due to specific knowledge of the GIS internal routines, while respecting the basic interoperability rules through the weak coupling between the manipulator and the extrator.

REFERENCES

Bruns, T., Egenhofer, M.J., "WEB-Top Interfaces for GIS Map Algebra", http://www.cs.umd.edu/projects/hcil/People/tbruns/gisjournal/webalgebra.html, 1997.

Buehler, K., Mckee, L., editors, "The Open GIS Guide", http://www.opengis.org/guide, Open Gis Consortium, Inc., 1996.

Kuhn, Werner and Willauer, Langley and Mark , David M. and Frank, Andrew U.(Eds), "User Interfaces for Geographic Information Systems: Discussions at the Specialist Meeting", National Center for Geographic Information and Analysis, 1992

Oliveira, J.L., Medeiros, C.B., "User Interface Issues in Geographic Information Systems", Relatório Técnico IC-96-06, DCC, UNICAMP, 1996.

Voisard, A., Designing and Integrating User Interfaces of Geographic Database Applications, in Proceedings of 1994 ACM Workshop on Advanced Visual Interfaces, 1994.

Voisard, A., Mapgets: A Tool for Visualizing and Querying Geographic Informatio, in Journal of Visual Languages and computing, 1995.

Some examples of the usage of internet/intranet technology in GIS

Wolfgang Reinhardt
University of the federal armed forces
Department of civil engineering and surveying
Institute for Geoinformation and land development
Werner-Heisenberg-Weg 39

D-8577 Neubiberg

Internet technology is among others (e.g. CORBA, OLE/COM) the most important enabling factor for interoperating geographic information systems. This paper discusses some aspects of the impact of internet technology in the field of geographic information systems and mainly its usage in applications. The last item is illustrated by some examples.

There are mainly two issues which are considered in this paper. In the first section - as an introduction - the evolution in the development of GIS products and the new software architecture in the internet environment (GIS product architecture) is discussed. In the second section of the paper some of the new possibilities of interaction with the user and the use of this technology in specific applications (Internet applications) are presented.

The simple possibility of providing raster maps in the web is not considered here. In the section ´GIS product architecture´ an overview is given how GIS can work in the internet environment. The concept of GIS web servers and the communication between web servers and clients using CGI scripts, java aplets or something similar is roughly discribed.

In the second section the work of three relevant projects of our GIS group are introduced:

GIS INTERNET CLIENT

In one project we have developed an GIS Internet Client to view data from different servers. This client is completely written in Java and is able to zoom and scroll GIS data in rasterformat. The communication with the web server is based on CGI technique in the moment. The efficiency and the limitations of this solution as well as possible extensions are also discussed here.

ALPINE HIKING GUIDE

In another project we have developed an alpine hiking guide using internet / Java technology which helps tourists planning their outdoor activities in alpine regions.

This projects shows the advantages of multimedia applications using texts, photos, video-clips for such purposes. This guide provides the data of a scanned topographical map, terrain model data (heights) and streets and foot paths (vector data). As a sophisticated routing solution is available the user is able to plan hiking or cycling tours interactively with regard to the time he wants to spent, to maximum heights and slopes and other parameters. In addition the user can receive current data via hyperlinks from the internet e.g. about weather-forecast, snow state and others.

3D PRESENTATION OF TERRAIN SECTIONS

The goal of this project was the presentation of terrain features and the terrain in 3D also enabling walks through the region. For this reason photogrammetrically measured terrain features and a digital terrain model have been made available and have been converted in VRML format. Afterwards different standard VRML browsers have been used to visualise the data. In order to provide information on the orientation of the viewer (coordinates of the viewers standpoint and his viewing direction) and to visualise this information a Java extension for the standard browser has been written. In the last chapter the usage of the internet technology and his advantages and limitation as well as necessary extensions are discussed also.

Multi-server Internet GIS: Standardization and Practical Experiences

Carel van den Berg, Frank Tuijnman, and Tom Vijlbrief
Professional GEO Systems
Damrak 44
1012 LK Amsterdam
The Netherlands

and

Co Meijer, Harry Uitermark and Peter van Oosterom
Cadastre
P.O.Box 9046
7300 GH Apeldoorn
The Netherlands

In this paper we present an approach to an open infrastructure for geographic information on the Internet. This infrastructure enables data providers to publish their data independently, while enabling end-users to access data from several providers simultaneously, and integrate the data locally in a geographic browser. Our goal is that an end-user finds accessing geographic information in this environment as easy as if he would be working with a state-of-the-art GIS package with all data that he is interested in on his own computer. The key elements that are required are: a common format for publishing meta-data on each server, a common SQL derived query protocol, standard file formats, and standard certificate based authentication procedures, for access control and (optionally) billing. An experiment with this approach has been carried out, with three data providers in Holland: The dutch Kadaster, the municipality of Almere, and the cable-tv company Casema. In this paper we present the major design decisions, the choices that we made for the prototype environment, and the relationship to ongoing specification and standardization processes for geographic data, in particular the relationships with the proposed European CEN standards, and the recently accepted specifications from the OpenGIS consortium.

INTRODUCTION

The current wave of GIS software for Internet is based either on the file downloading paradigm, or on the picture paradigm (presenting a map as a JPEG picture), or on the client-server paradigm (creating a closed interaction between a client and a single server). Neither of these approaches can capitalize on the main potential of the Internet: integrated and easy access to a vast amount of geographic information on various servers. In addition to that the interaction protocol between client and server is typically proprietary, which means that someone who browses geographic information needs software from the same vendor as is used by the publisher.

To make GIS popular on the Internet one needs to create for geographic information the same level of uniformity as the World Wide Web has done for text. The brilliance of the World Wide Web lies in the combination of the hypertext model with the Internet, together with a formatting standard for text (HTML). The hypertext model, however, does not work for geographic data: it is not particularly useful to jump from one map to the next. So another basic metaphore has to be used.

The standard model for geographic data on the computer is the layer model. The layer model of geographic information systems relates to a paper map like the hypertext model relates to text on paper. To make geographic data on the Internet attractive one has to set a standard for the layer model, so that we can obtain a topographic layer from one source, a pollution layer from another, and a property layer from a third source, and dynamically merge them in a geo-web browser.

To achieve this the following standard protocols have to be defined (in addition to support for authentication and billing):

- a method to inquire which layers are available, and what data they contain.
- a method to ask a particular layer from a particular source.
- a standard format for returning this information.

Many aspects of these protocols are subject to ongoing standardization efforts. The format for returning geographic information is essentially a description of a file format. The method for querying the meta-data has a clear relationship with the meta- data standards, and clearinghouse related activities. The protocol for querying geographic information is new. The CEN has acknowledged the need for something like this (see CEN/TC 287, which specifies names and semantics of required spatial operators), but so far no complete proposal exists. The closest relevant specification for the query protocol is the OGC specification for SQL with simple geometric features. Despite the fact that it is relatively easy to identify protocols that can be used to address part of the problem, no comprehensive proposal exists so far that can be used to achieve open access to geographic data publishers on the Internet.

ARCHITECTURE

Our approach to the design of the meta-data structure, the query formalism and the format for the returned data is based on the object-relational formalism, where we include geographic features as attributes (that have a geometric type) within a relational table.

The motivation for this approach is based on the following considerations:

1) object relational database management systems with support for geographic data are now available from most major vendors (Informix, CA-Ingres, Oracle). Even if one wants to provide access to a file based collection of geographic data it is not difficult to implement a limited selective capability on top of it, though of course performing such selections will cost more time. So it is technically feasible for any organization to implement this functionality.

2) the object relational model is currently the only widely available formalism that can deal with geographic data, and in which all three required elements (a meta-data structure, a query formalism, and a format for returned data) are defined in an integrated manner. This is essential from a technical point of view: the meta-data does not just describes the data, it also has to provide the 'words' that can be used in queries, and it has to be clear which words can replace which syntactic element in a query. The returned data has to be understood as a response to the query, so there has to be a well defined relationship between the semantics of queries, and the actual data that is returned.

3) It can be mapped easily to the stateless http protocol, because sql is also stateless (meaning that any request can be handled independently from previous or subsequent requests).

4) It ensures that the browser requires only knowledge about which data is available, rather than detailed knowledge about file naming conventions, and tiles.

In mapping extended SQL to http, in such a way that it can be used effectively for geographic data publication over the Internet, a number of issues have to be tackled:

1. Geometric types and there semantics have to be defined.
2. A standard method has to be defined to formulate a query
3. A format has to be defined for the returned data
4. A safe and sufficient version of SQL has to be defined, in order to be sure that a server cannot be crashed by a malicious client.
5. Authentication, and billing have to be included to support commercial geographic data publication.
6. Compression has to be incorporated, in order to be able to transfer geographic data effectively over the Internet.

The full paper describes each of these aspects in detail, here we give just an example of a possible http request to a server:

```
//ooa.kadaster.nl/cgi-bin/magma?coordsys=rdm&database=kad4&relation=percelen&
attributes={magma_oid,geo_bbox,geo_pgn,owners}&
where=WRectangle.intersects(189000,485000,192000,488000)&and&
owners>='oost'&and&owners<='oostf'
```

This query requests the four named attributes (magma_oid, geo_bbox, geo_pgn, owners), for all parcels in the kad4 database, within the selected region, where the owners names prefix is between 'oost' and 'oostf'. The coordinates are given in RDM (plane state coordinates that are used as a standard in Holland). As a result a list of tuples will be returned, that match the where clause.

TRIAL EXPERIENCES

A trial with this approach has been carried out in the province of Flevoland, where three organizations, the municipality of Almere, the cable-tv company Casema, and the Cadastre, have implemented a system that allows them to access the data of the other parties directly. The data published by the Cadastre consists of the parcel boundaries. The data from the municipality consists of a large scale topographic map and large scale topographic plans for new neighbourhoods (Almere builds about 3000 new houses annually). The Casema has published the cable locations, both planned and existing. For browsing the Java based Lava GIS browser from PGS is used, and for the server the Magma GeoData publisher is used to interface between the http requests and various geographic datastores (Ingres 2.0 for the Cadastre, Illustra for the topographic data, and flat DXF files for the Casema). The three partners have installed their own servers (connected to the internet) providing spatial data on request.

In this trial the feasibility of dynamic data integration over the Internet has been demonstrated, supporting both raster and vector data at the client side, and using a Java based GIS browser to give everyone direct access. Secure communication and paid access to data are aspects that will be evaluated in the next phase of the trial.

CONCLUSION

To support open access to geographic data over the Internet three related protocols need to be defined.

- a method to inquire which geographic data is available from a particular publisher
- a method to ask a particular layer from a particular publisher.
- a standard format for returning this information.

The solution proposed in this article is based on the object relational model (with geometric types). Simple SQL request are encoded as URL's for http, and the result is returned as a list of tuples, with geometric attributes.

The trial, with the Lava/Magma software from PGS, has demonstrated that it is effectively possible to implement this approach, and to achieve in this manner open, easy and integrated access to data from different organizations. As such it can be the basis for a public infrastructure that allows each organization to publish her data independently, while at the same time it enables clients, both professionals and citizens, to have integrated access through the Internet to all available geographic information.