Data Models and Query Languages for Linked Geospatial Data

Manolis Koubarakis, Kostis Kyzirakos and Charalampos Nikolaou

Dept. of Informatics and Telecommunications
National and Kapodistrian University of Athens

TELEIOS

Tutorial Organization

14:30 – 14:45 Introduction

14:45 – 15:15 Background in geospatial data modeling

15:15 – 16:00 Geospatial data in RDF - stSPARQL

16:00 – 16:30 Coffee break

16:30 – 16:45 Geospatial data in RDF - GeoSPARQL

16:45 – 17:00 Implemented RDF Stores with geospatial support

17:00 – 17:50 Geospatial information with description logics, OWL and rules

17:50 – 18:00 Conclusions, questions, discussion

Introduction

Presenter: Manolis Koubarakis

Dept. of Informatics and Telecommunications
National and Kapodistrian University of Athens

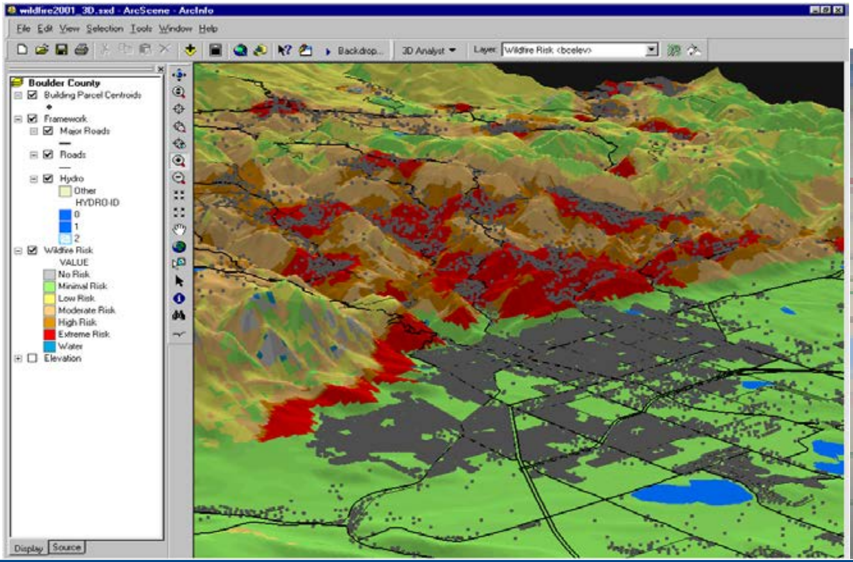
Outline

- Why should you be interested in geospatial information?
- Why should you attend this tutorial?

Why Geospatial Information?

- **Geospatial,** and in general **geographical,** information is very important in reality: everything that happens, happens somewhere (**location**).

- **Decision making can be substantially improved** if we know where things take place.



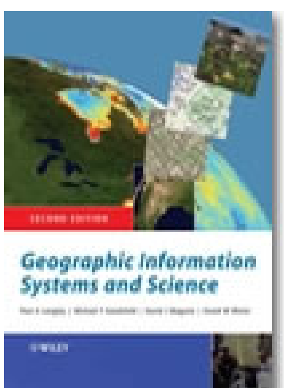
Geography

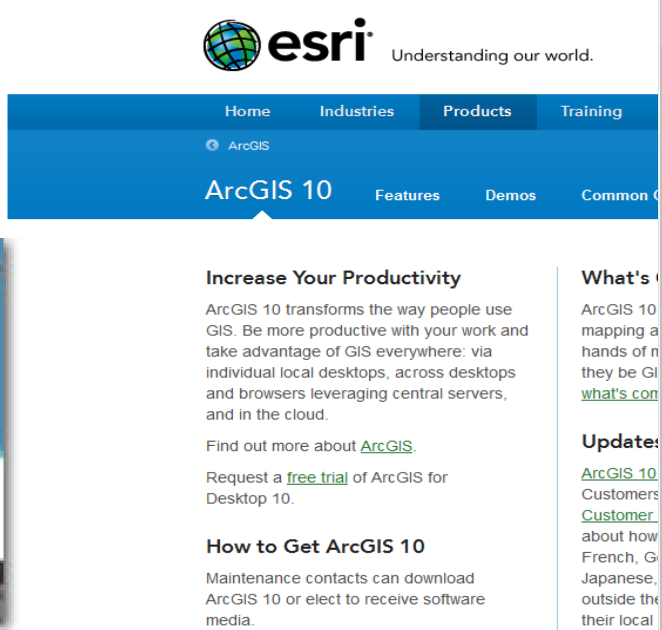
- From http://en.wikipedia.org/wiki/Geography
  - **Geography** is the science that studies the lands, the features, the inhabitants and the phenomena of the Earth.
  - From the Greek word **γεωγραφία (geographia)** which means “describing the Earth”.

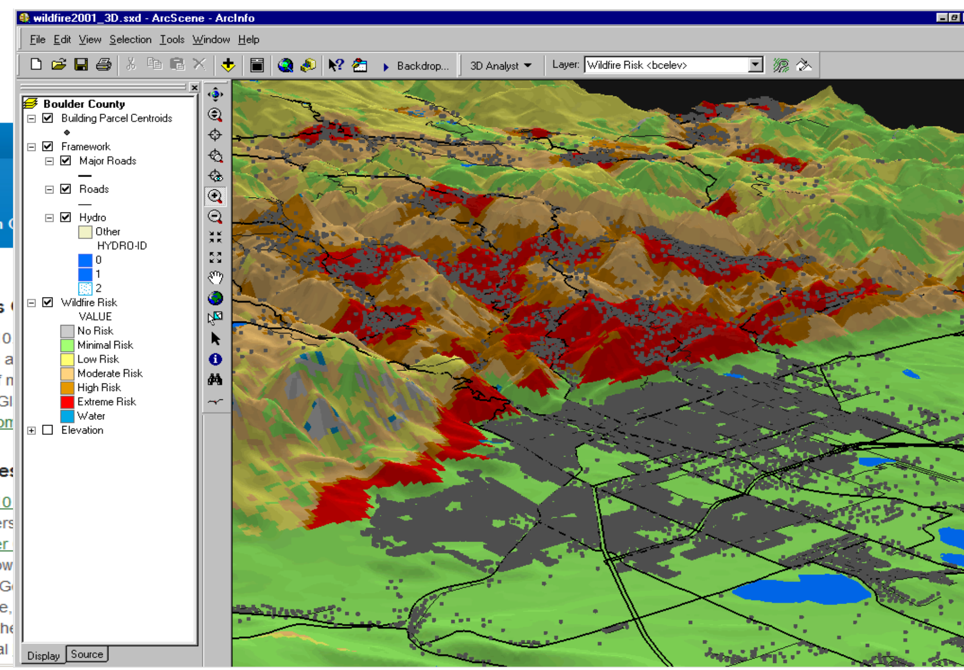
Geographical Information Systems and Science

- A **geographical information system (GIS)** is a system designed to capture, store, manipulate, analyze, manage, and present all types of geographical data.

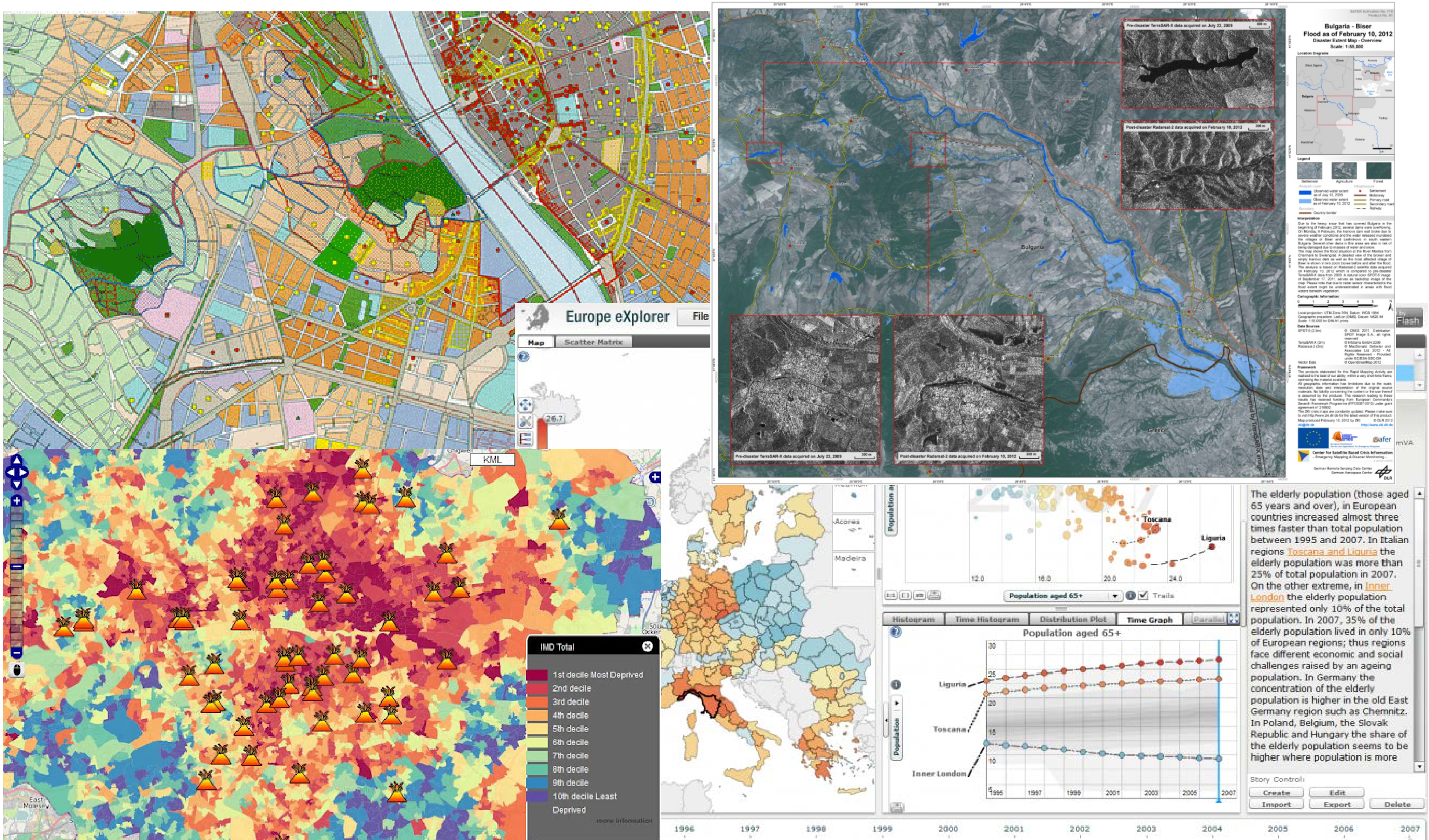
- **GIS science** is the field of study for developing and using GIS.







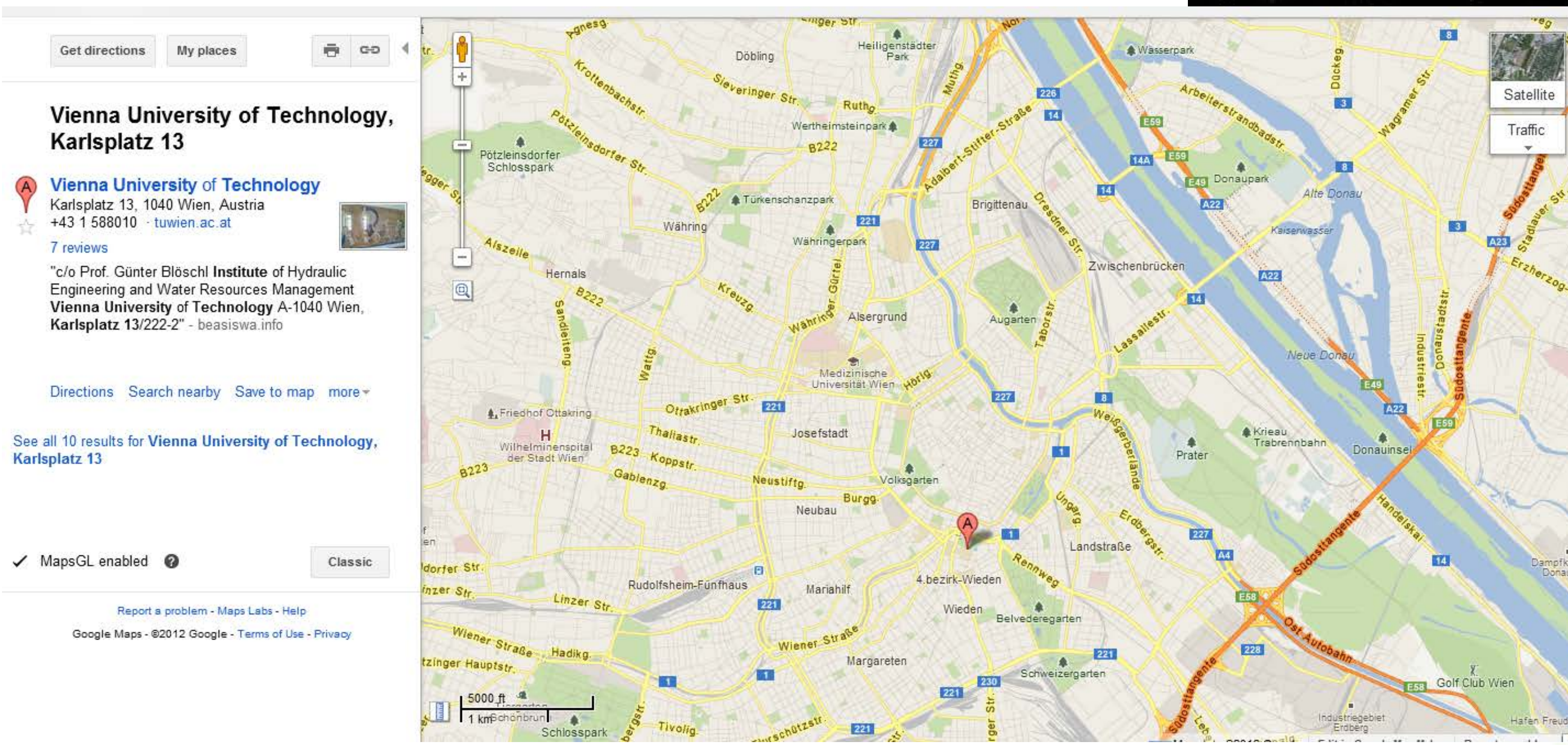
Combining GIS Data for Decision Making



Why this tutorial?

- Lots of **geospatial data is available on the Web** today.
- Lots of **public data** coming out of open government initiatives is **geospatial**.
- Lots of the above data is quickly being **transformed into linked data!**
- People have started building **applications** utilizing linked data.

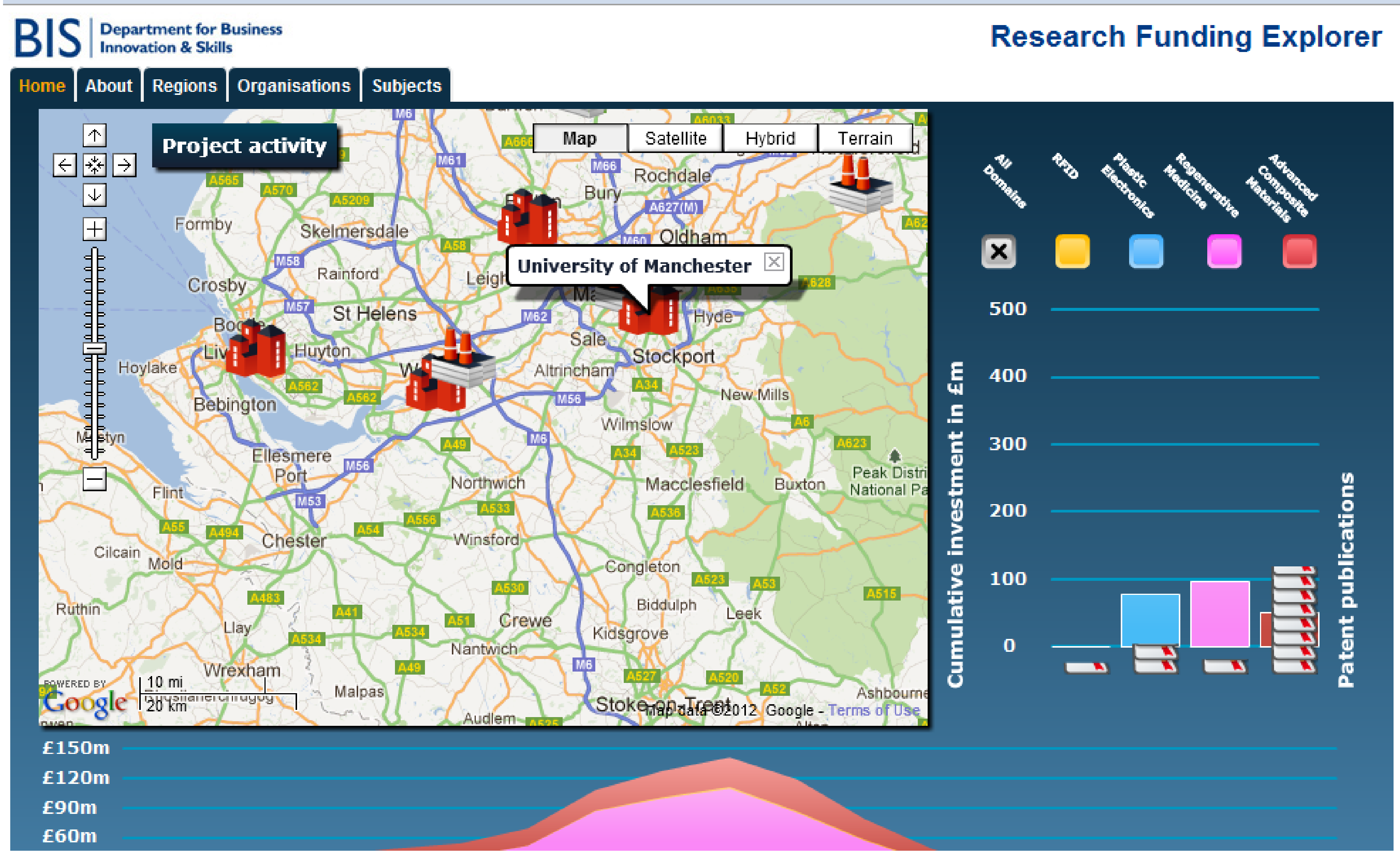
Geospatial data on the Web



Open Government Data

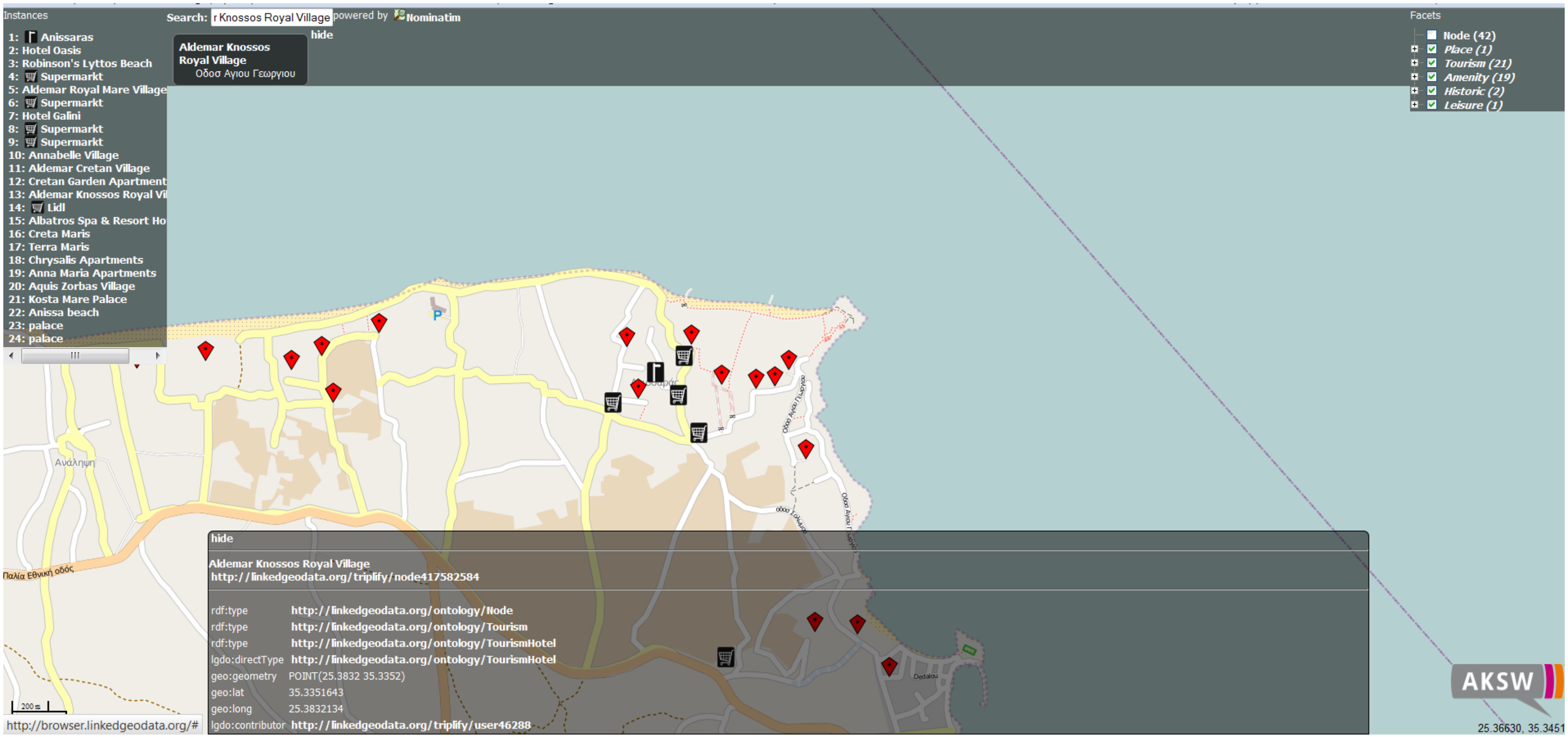
Linked geospatial data – Ordnance Survey

Linked geospatial data – Research Funding Explorer



Linked geospatial data – Spain

Linked geospatial data – Open Street Map



Conclusions

- **Introduction**
  - Why should you be interested in geospatial information?
  - Why should you attend this tutorial?

- **Next topic:** Background in geospatial data modeling

Background in geospatial data modeling

Presenter: Manolis Koubarakis

Dept. of Informatics and Telecommunications
National and Kapodistrian University of Athens

Outline

- Basic GIS concepts and terminology
- Geographic space modeling paradigms
- Geospatial data standards

Basic GIS Concepts and Terminology

- **Theme:** the information corresponding to a particular domain that we want to model. A theme is a set of **geographic features.**

- **Example:** the countries of Europe

Basic GIS Concepts (cont'd)

- **Geographic feature or geographic object:** a domain entity that can have various **attributes** that describe **spatial and non-spatial** characteristics.

- **Example:** the country Greece with attributes
  - Population
  - Flag
  - Capital
  - Geographical area
  - Coastline
  - Bordering countries

Basic GIS Concepts (cont'd)

- Geographic features can be **atomic** or **complex.**

- **Example:** According to the Kallikratis administrative reform of 2010, Greece consists of:
  - 13 **regions** (e.g., Crete)
  - Each region consists of **perfectures** (e.g., Heraklion)
  - Each perfecture consists of **municipalities** (e.g., Dimos Chersonisou)

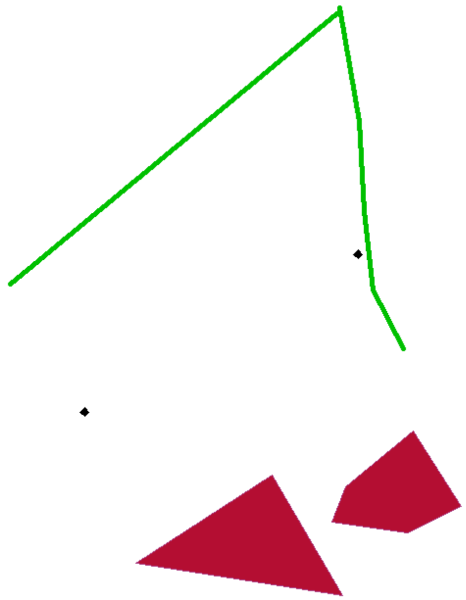
Basic GIS Concepts (cont'd)

- The spatial characteristics of a feature can involve:
  - **Geometric information** (location in the underlying geographic space, shape etc.)
  - **Topological information** (containment, adjacency etc.).

Municipalities of the perfecture of Heraklion:
1. Dimos Irakliou
2. Dimos Archanon-Asterousion
3. Dimos Viannou
4. Dimos Gortynas
5. Dimos Maleviziou
6. Dimos Minoa Pediadas
7. Dimos Festou
8. Dimos Chersonisou

Geometric Information

- Geometric information can be captured by using geometric primitives (**points**, **lines**, **polygons,** etc.) to approximate the spatial attributes of the real world feature that we want to model.



- Geometries are associated with a **coordinate reference system** which describes the coordinate space in which the geometry is defined.

Topological Information

- Topological information is **inherently qualitative** and it is expressed in terms of **topological relations** (e.g., containment, adjacency, overlap etc.).

- Topological information can be **derived from geometric information** or it might be captured by **asserting explicitly the topological relations** between features.

Topological Relations

- The study of topological relations has produced a lot of interesting results by researchers in:
  - GIS
  - Spatial databases
  - Artificial Intelligence (qualitative reasoning and knowledge representation)

The 4-intersection model

- The **4-intersection model** has been defined by Egenhofer and Franzosa in 1991 based on previous work by Egenhofer and colleagues.
- It is based on **point-set topology.**
- **Spatial regions** are defined to be **non-empty, proper subsets of a topological space.** In addition, they must be closed and have connected interiors.
- **Topological relations** are the ones that are invariant under topological homeomorphisms.

4IM and 9IM

- The 4-intersection model can captures **topological relations** between two spatial regions *a* and *b* by considering **whether the intersection of their boundaries and interiors is empty or non-empty**.

- The **9-intersection model** is an extension of the 4-intersection model (Egenhofer and Herring, 1991).

- 9IM captures topological relations between two spatial regions *a* and *b* by considering whether the intersection of their boundaries, interiors and **exteriors** is empty or non-empty.

DE-9IM

- The **dimensionally extended 9-intersection model** has been defined by Clementini and Felice in 1994.

- It is also based on the **point-set topology** of R^2 and deals with **"simple", closed geometries (areas, lines, points).**

- Like its predecessors (4IM, 9IM), it can also be extended to **more complex geometries** (areas with holes, geometries with multiple components).

DE-9IM

- It captures **topological relationships** between two geometries *a* and *b* in R^2 by considering the **dimensions of the intersections of the boundaries, interiors and exteriors** of the two geometries:

$$\text{DE-9IM}(a,b) = \begin{bmatrix} \dim(I(a) \cap I(b)) & \dim(I(a) \cap B(b)) & \dim(I(a) \cap E(b)) \\ \dim(B(a) \cap I(b)) & \dim(B(a) \cap B(b)) & \dim(B(a) \cap E(b)) \\ \dim(E(a) \cap I(b)) & \dim(E(a) \cap B(b)) & \dim(E(a) \cap E(b)) \end{bmatrix}.$$

- The dimension can be **2, 1, 0** and **-1** (dimension of the empty set).

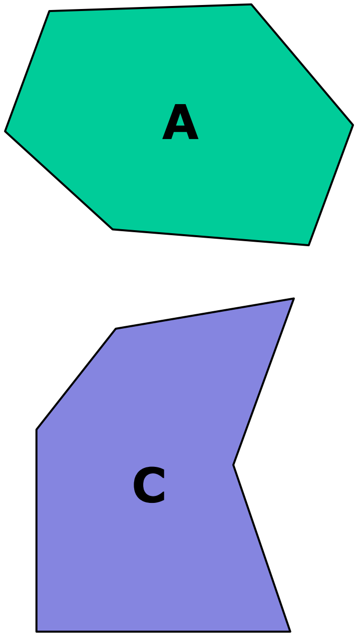
DE-9IM

- **Five jointly exclusive and pairwise disjoint (JEPD)** relationships between two different geometries can be distinguished (**disjoint, touches, crosses, within, overlaps**).

- The model can also be defined using an appropriate **calculus of geometries** that uses these 5 binary relations and boundary operators.

- See the paper: E. Clementini and P. Felice. A Comparison of Methods for Representing Topological Relationships. Information Sciences 80 (1994), pp. 1-34.

Example: A disjoint C

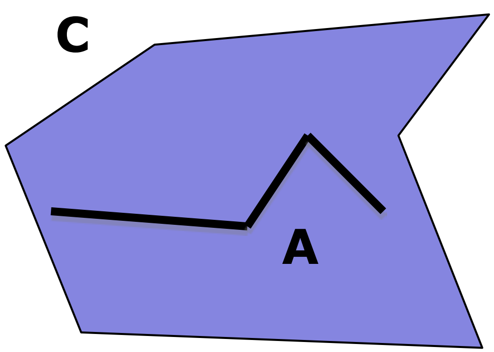


	I(C)	**B**(C)	**E**(C)
I(A)	F	F	*
B(A)	F	F	*
E(A)	*	*	*

Notation:

- T = { 0, 1, 2 }
- F = -1
- * = don't care = { -1, 0, 1, 2 }

Example: A within C

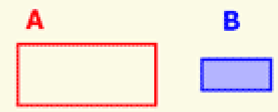
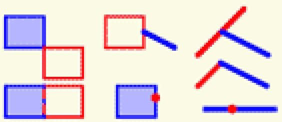
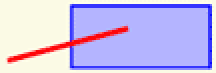
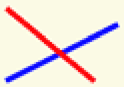
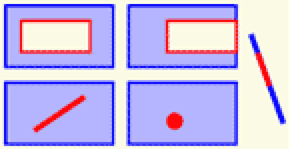
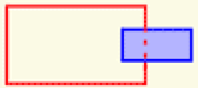
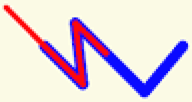


	I(C)	**B**(C)	**E**(C)
I(A)	T	*	F
B(A)	*	*	F
E(A)	*	*	*

Notation equivalent to 3x3 matrix:

- String of 9 characters representing the above matrix in row major order.
- In this case: T*F**F***

DE-9IM Relation Definitions

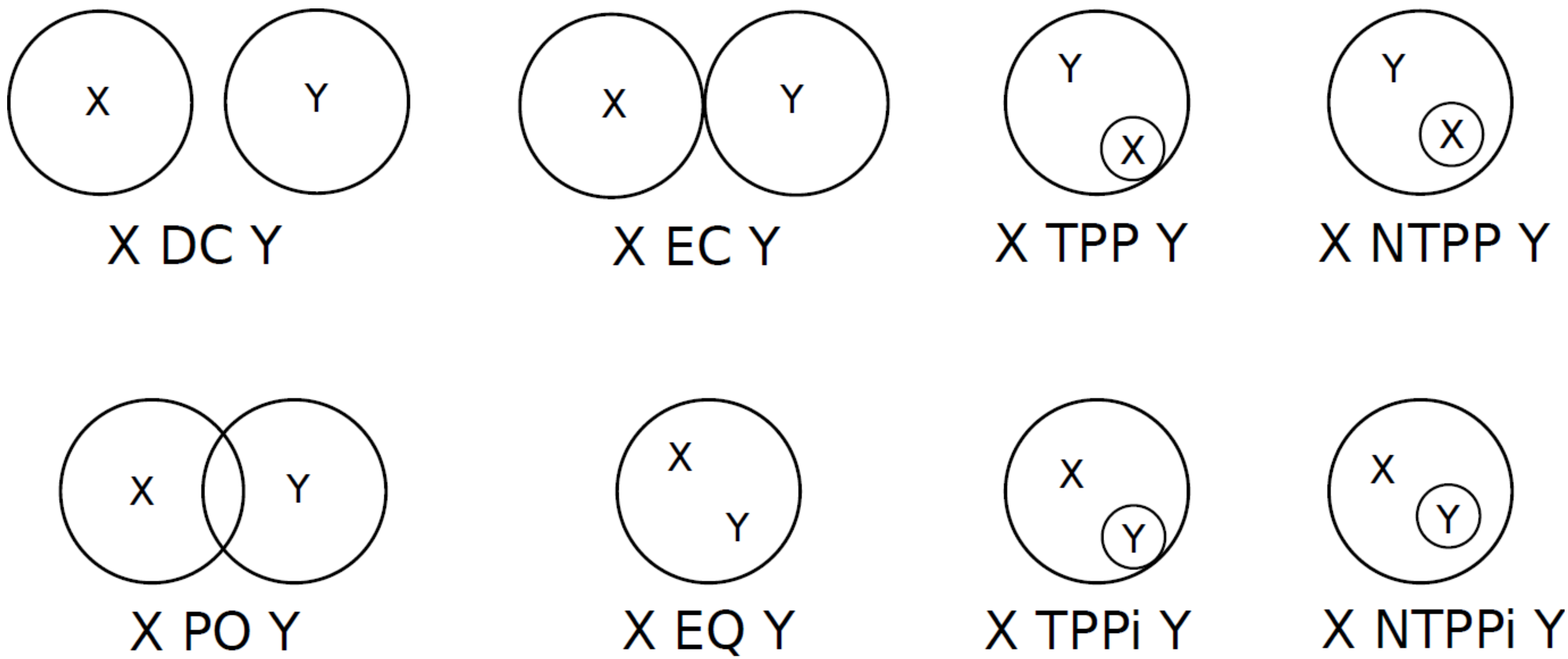
Beziehung	Definition	Beispiele
A disjoint B	$\begin{bmatrix} F & F & * \\ F & F & * \\ * & * & * \end{bmatrix}$	
A touches B (d(A) > 0 ∨ d(B) > 0)	$\begin{bmatrix} F & T & * \\ * & * & * \\ * & * & * \end{bmatrix} \vee \begin{bmatrix} F & * & * \\ T & * & * \\ * & * & * \end{bmatrix} \vee \begin{bmatrix} F & * & * \\ * & T & * \\ * & * & * \end{bmatrix}$	
A crosses B (d(A) < d(B))	$\begin{bmatrix} T & * & T \\ * & * & * \\ * & * & * \end{bmatrix}$	
A crosses B (d(A) = d(B) = 1)	$\begin{bmatrix} 0 & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$	
A within B	$\begin{bmatrix} T & * & F \\ * & * & F \\ * & * & * \end{bmatrix}$	
A overlaps B (d(A) = d(B), d(A) ≠ 1, d(B) ≠ 1)	$\begin{bmatrix} T & * & T \\ * & * & * \\ T & * & * \end{bmatrix}$	
A overlaps B (d(A) = d(B) = 1)	$\begin{bmatrix} 1 & * & T \\ * & * & * \\ T & * & * \end{bmatrix}$	

The Region Connection Calculus (RCC)

- The primitives of the calculus are **spatial regions**. These are non-empty, regular subsets of a topological space.
- The calculus is based on a single binary predicate *C* that formalizes the "**connectedness**" relation.
    - ***C(a,b)* is true** when the closure of *a* is connected to the closure of *b* i.e., they have at least one point in common.
- It is axiomatized using first order logic.
- See the original paper by Randell, Cui and Cohn (KR 1991).

RCC-8

- This is a set of **eight JEPD binary relations** that can be defined in terms of predicate *C*.



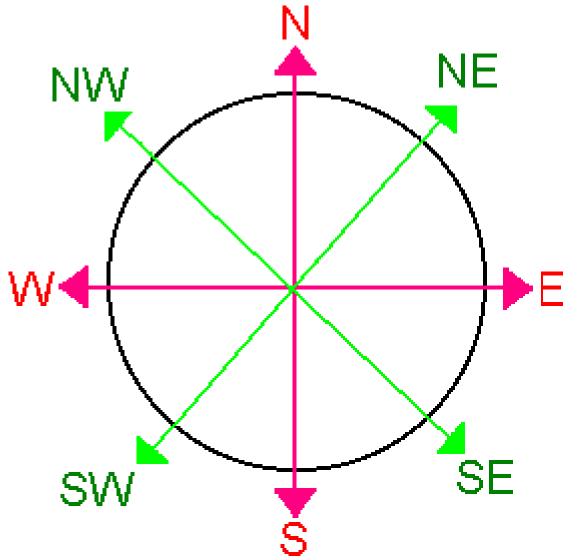
RCC-5

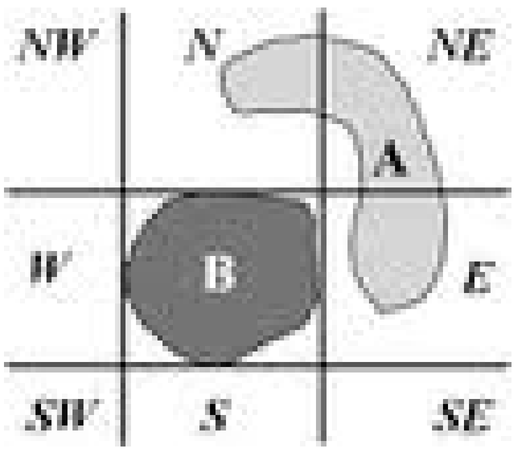
- The **RCC-5 subset** has also been studied. The granularity here is coarser. The boundary of a region is not taken into consideration:
  - No distinction among DC and EC, called just DR.
  - No distinction among TPP and NTPP, called just PP.

- RCC-8 and RCC-5 relations **can also be defined using point-set topology**, and there are very close connections to the models of Egenhofer and others.

More Qualitative Spatial Relations

- **Orientation/Cardinal directions** (left of, right of, north of, south of, northeast of etc.)



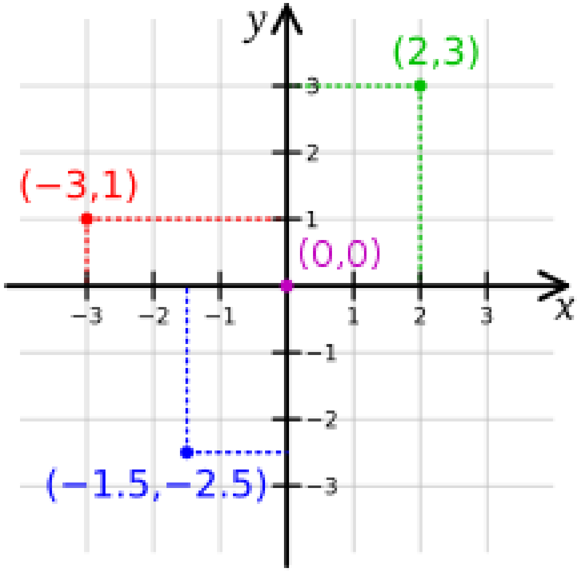


- **Distance** (close to, far from etc.). This information can also be **quantitative**.

Coordinate Systems

- **Coordinate:** one of *n* scalar values that determines the position of a point in an *n*-dimensional space.
- **Coordinate system:** a set of mathematical rules for specifying how coordinates are to be assigned to points.

- **Example**: the Cartesian coordinate system



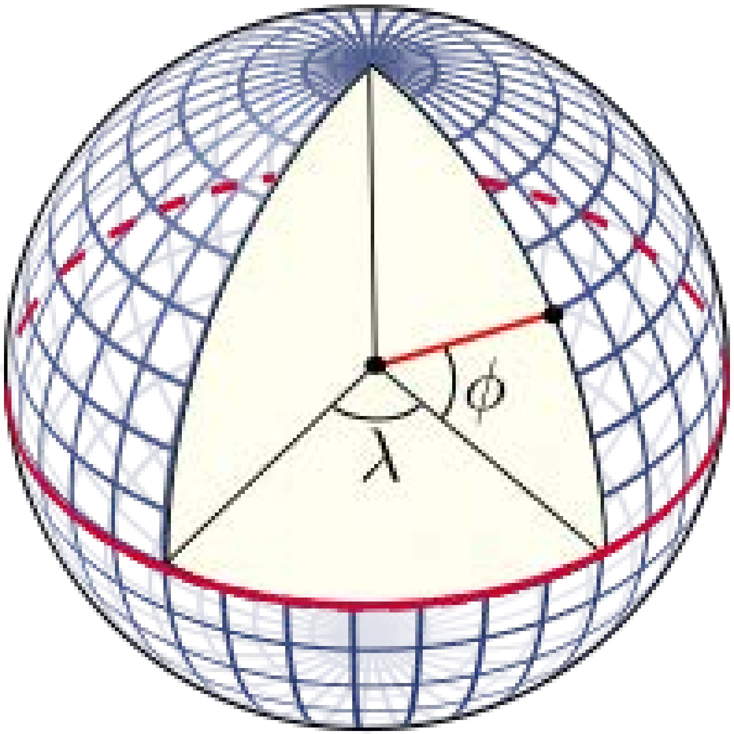
Coordinate Reference Systems

- **Coordinate reference system**: a coordinate system that is related to an **object** (e.g., the Earth, a planar projection of the Earth, a three dimensional mathematical space such as R^3) through a **datum** which specifies its origin, scale, and orientation.

- The term **spatial reference system** is also used.

Geographic Coordinate Reference Systems

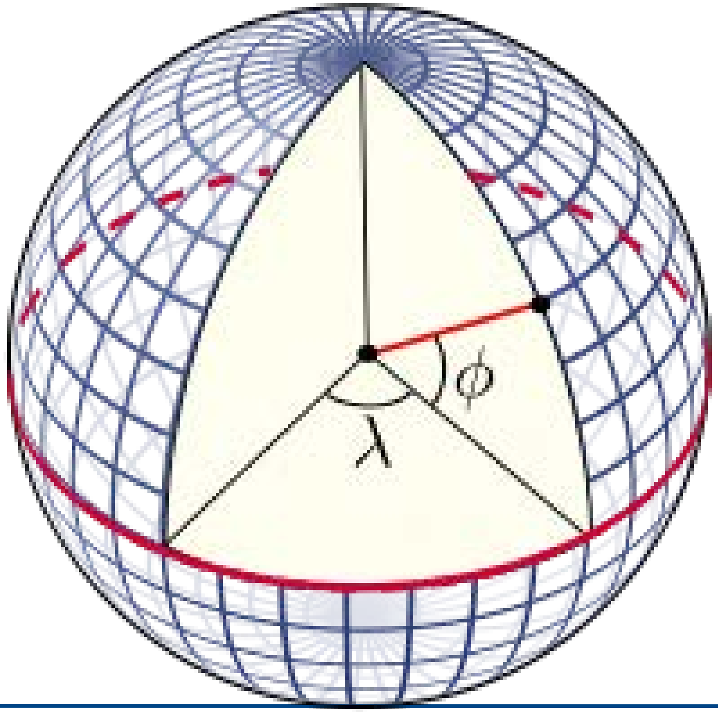
- These are 3-dimensional coordinate systems that utilize **latitude (φ)**, **longitude (λ)** , and optionally **geodetic height (i.e., elevation)**, to capture geographic locations on Earth.



The World Geodetic System

- The **World Geodetic System (WGS)** is the most well-known geographic coordinate reference system and its latest revision is **WGS84.**

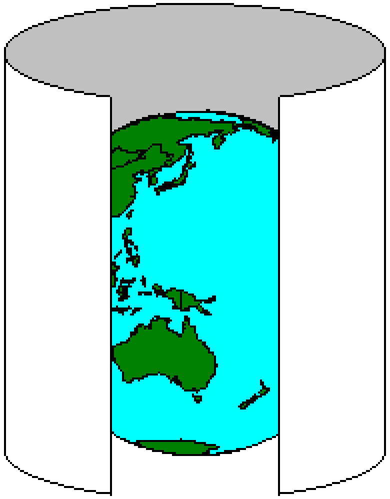
- **Applications:** cartography, geodesy, navigation (GPS), etc.



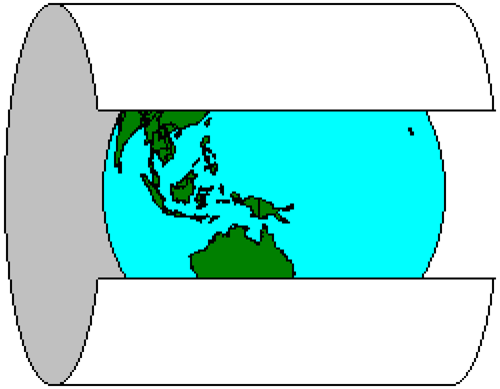
Projected Coordinate Reference Systems

- **Projected coordinate reference system:** they transform the 3-dimensional approximation of the Earth into a 2-dimensional surface (distortions!)

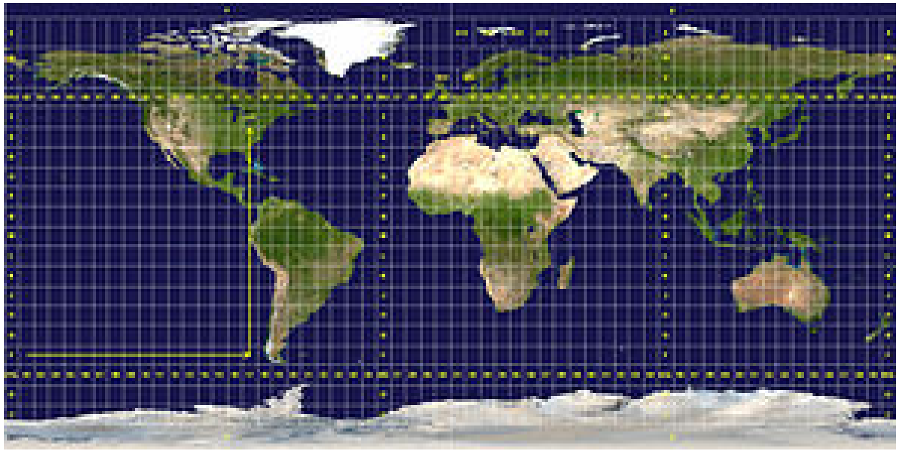
- **Example:** the **Universal Transverse Mercator (UTM)** system



Mercator projection



Transverse Mercator projection



Coordinate Reference Systems (cont'd)

- There are well-known ways to **translate** between co-ordinate reference systems.

- Various authorities maintain lists of coordinate reference systems. See for example:
  - **OGC** http://www.opengis.net/def/crs/
  - **European Petroleum Survey Group** http://www.epsg-registry.org/

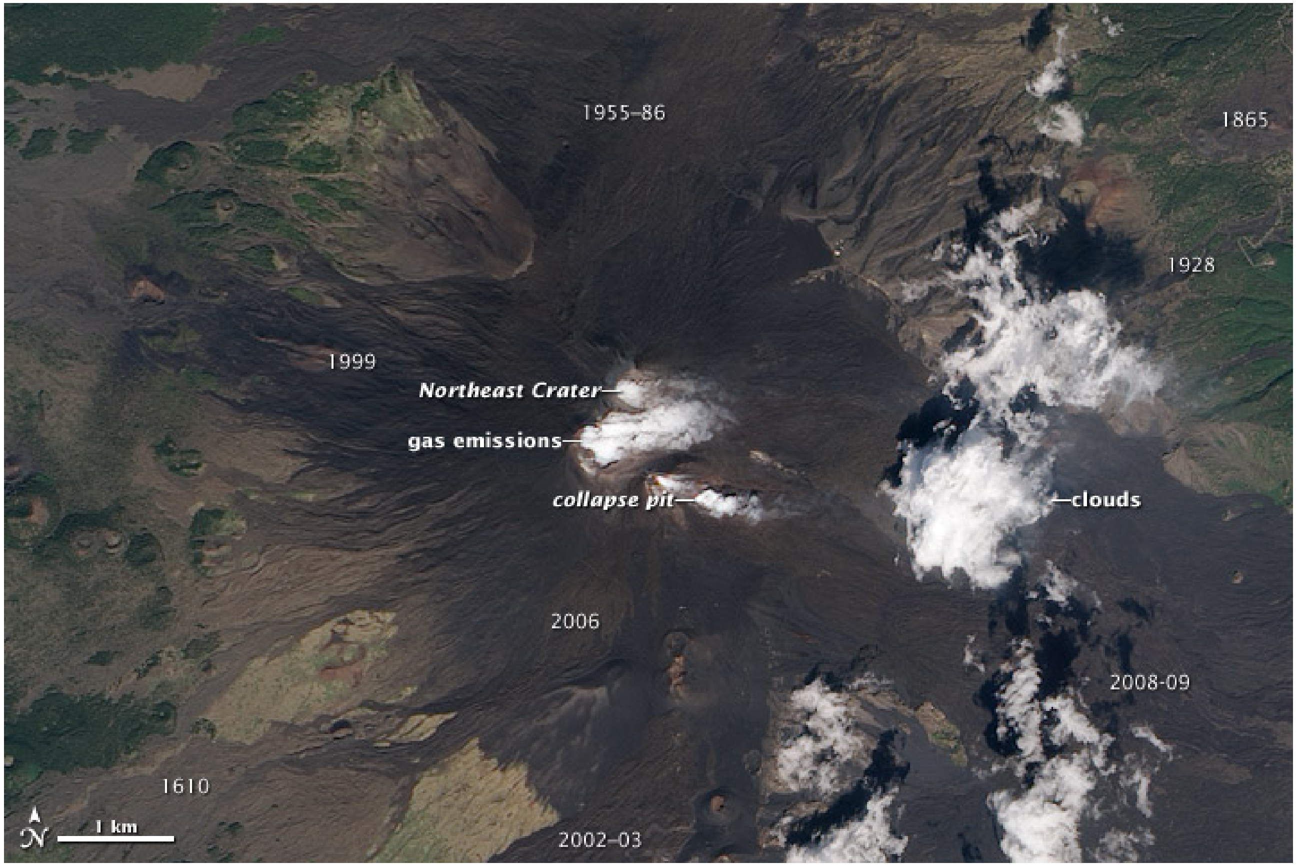
Geographic Space Modeling Paradigms

- **Abstract** geographic space modeling paradigms: discrete objects vs. continuous fields

- **Concrete representations**: tessellation vs. vectors vs. constraints

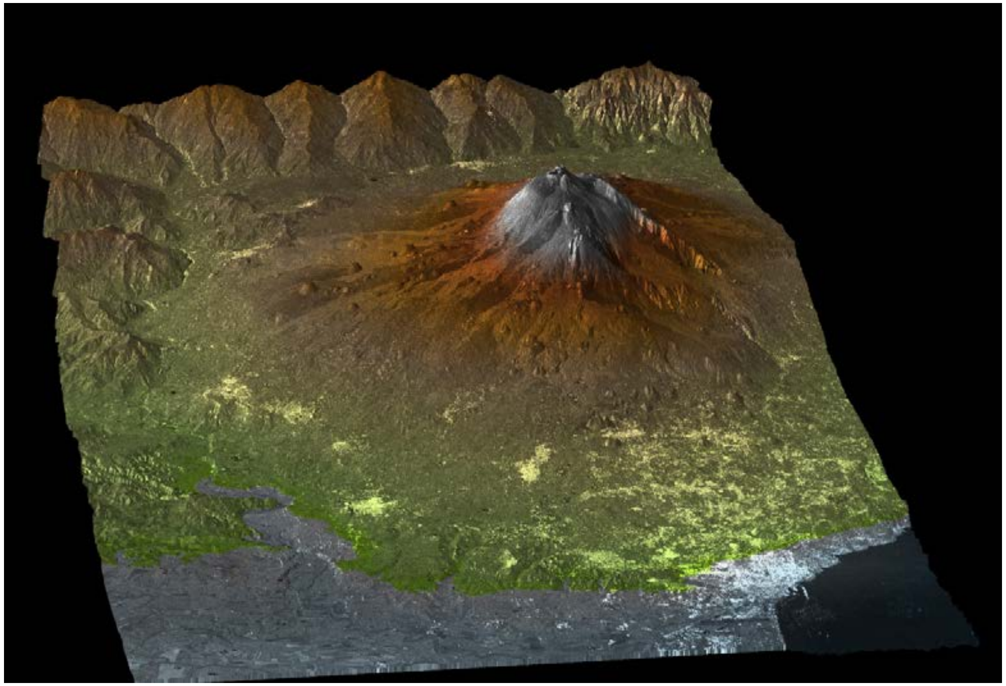
Abstract Modeling Paradigms: Feature-based

- **Feature-based** (or **entity-based** or **object-based**). This kind of modeling is based on the concepts we presented already.



Abstract Modeling Paradigms: Field-based

- Each point *(x,y)* in geographic space is associated with one or several attribute values defined as **continuous functions** in *x* and *y*.
- **Examples:** elevation, precipitation, humidity, temperature for each point *(x,y)* in the Euclidean plane.



From Abstract Modeling to Concrete Representations

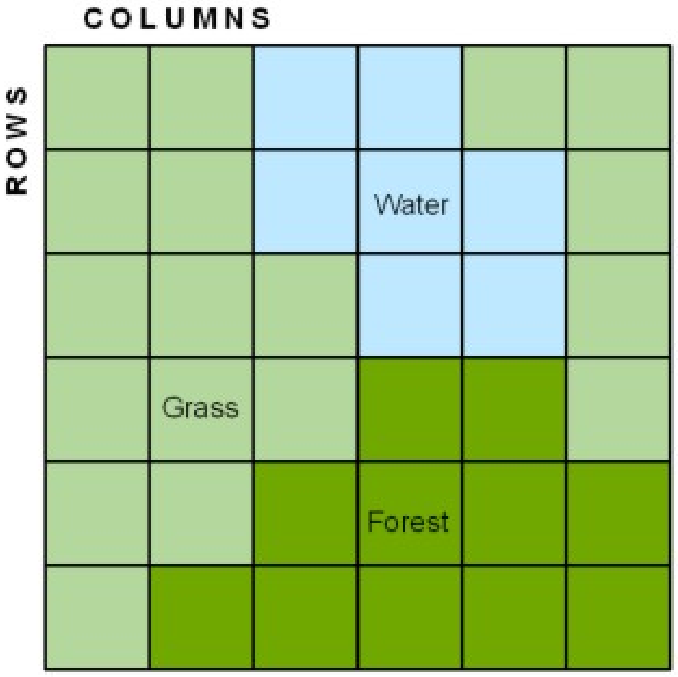
- **Question**: How do we represent the **infinite objects** of the abstract representations (points, lines, fields etc.) **by finite means** (in a computer)?

- **Answers**:
  - **Approximate** the continuous space (e.g., $\mathbb{R}^2$) by a discrete one ($\mathbb{Z}^2$).
  - Use **special encodings**

Approximations: Tessellation

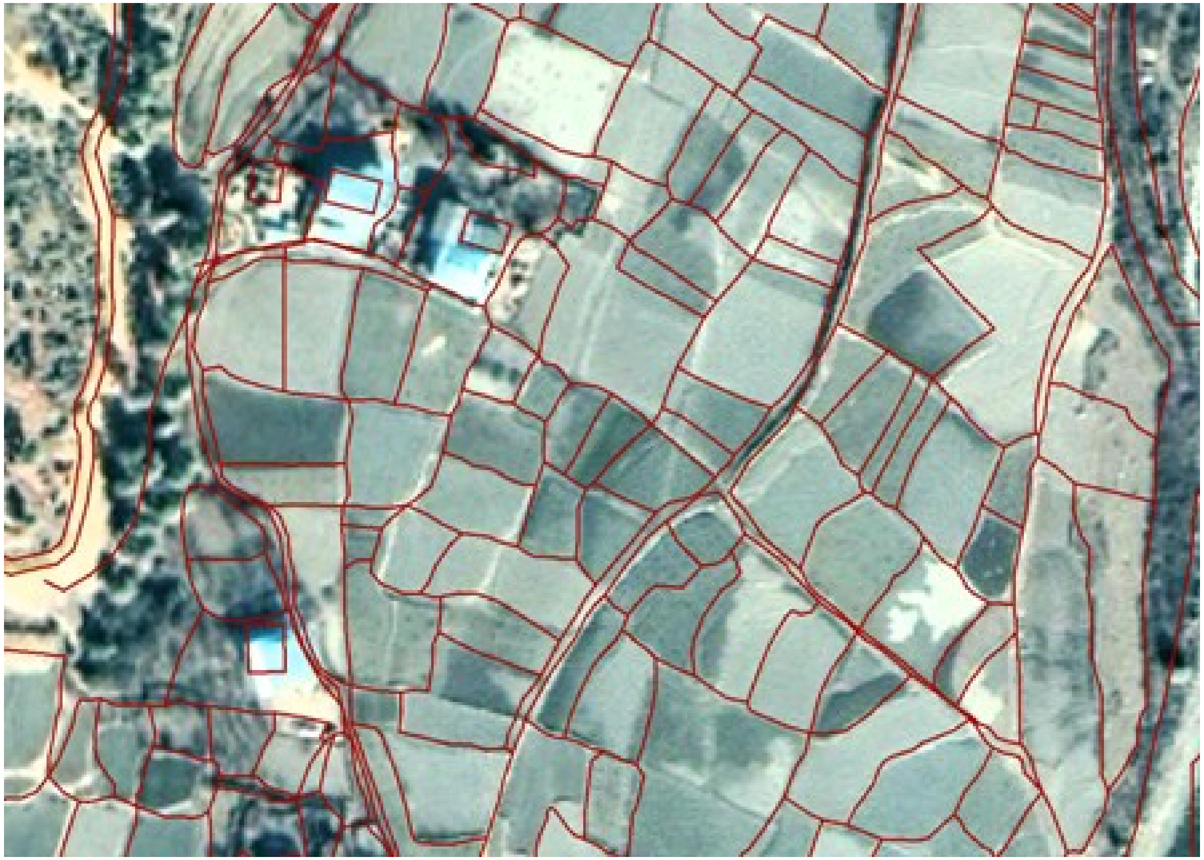
- In this case a **cellular decomposition of the plane** (usually, a grid) serves as a basis for representing the geometry.

- **Example:** raster representation (fixed or regular tesselation)



Example

- **Cadastral map (irregular tessellation)** overlayed on a satellite image.

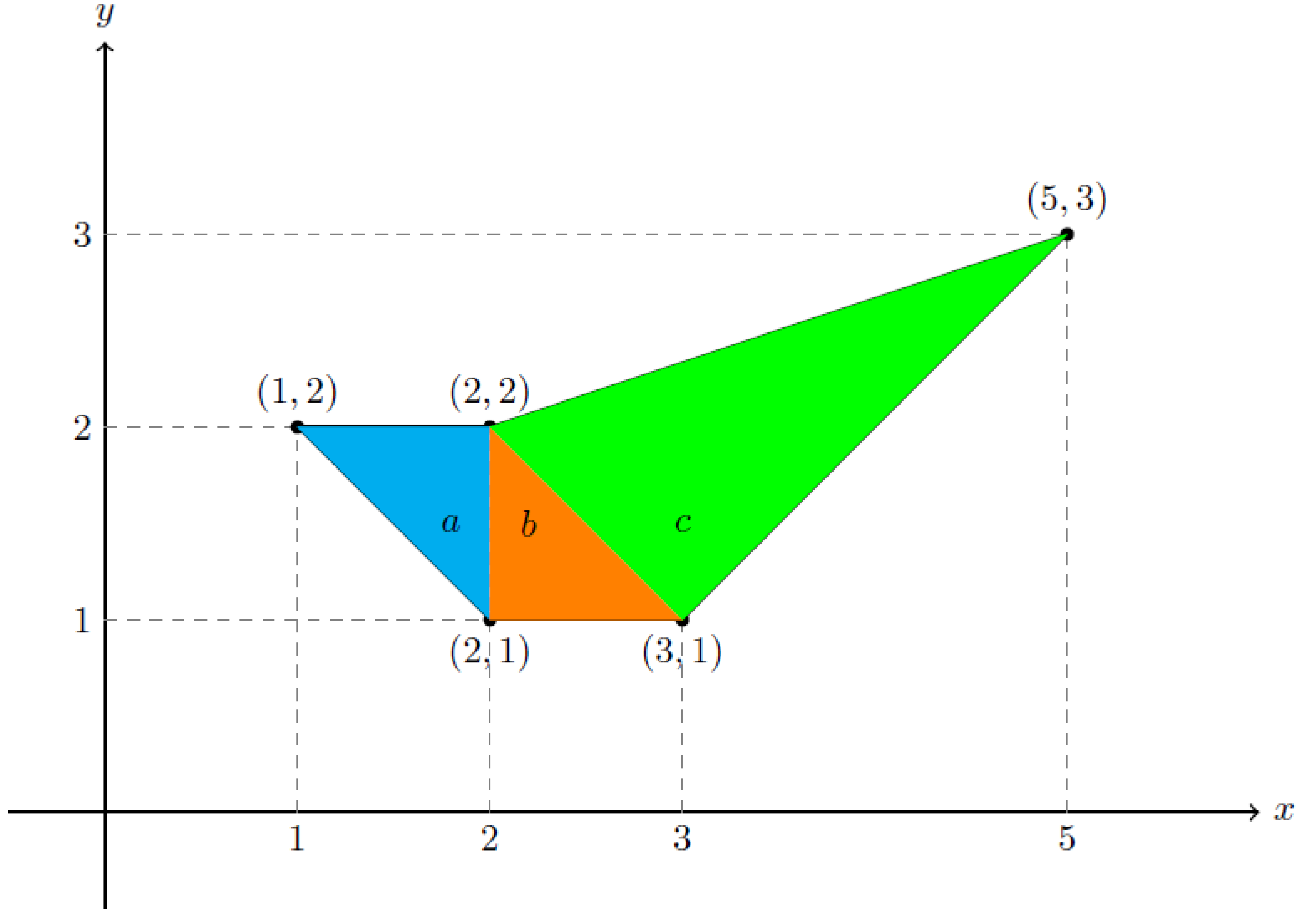


Special Encodings: Vector Representation

- In this case objects in space are represented using **points** as primitives as follows:
  - A **point** is represented by a tuple of coordinates.
  - A **line segment** is represented by a pair with its beginning and ending point.
  - **More complex objects** such as arbitrary lines, curves, surfaces etc. are built recursively by the basic primitives using constructs such as lists, sets etc.

- This is the approach **used in all GIS and other popular systems today. It has also been standardized** by various international bodies.

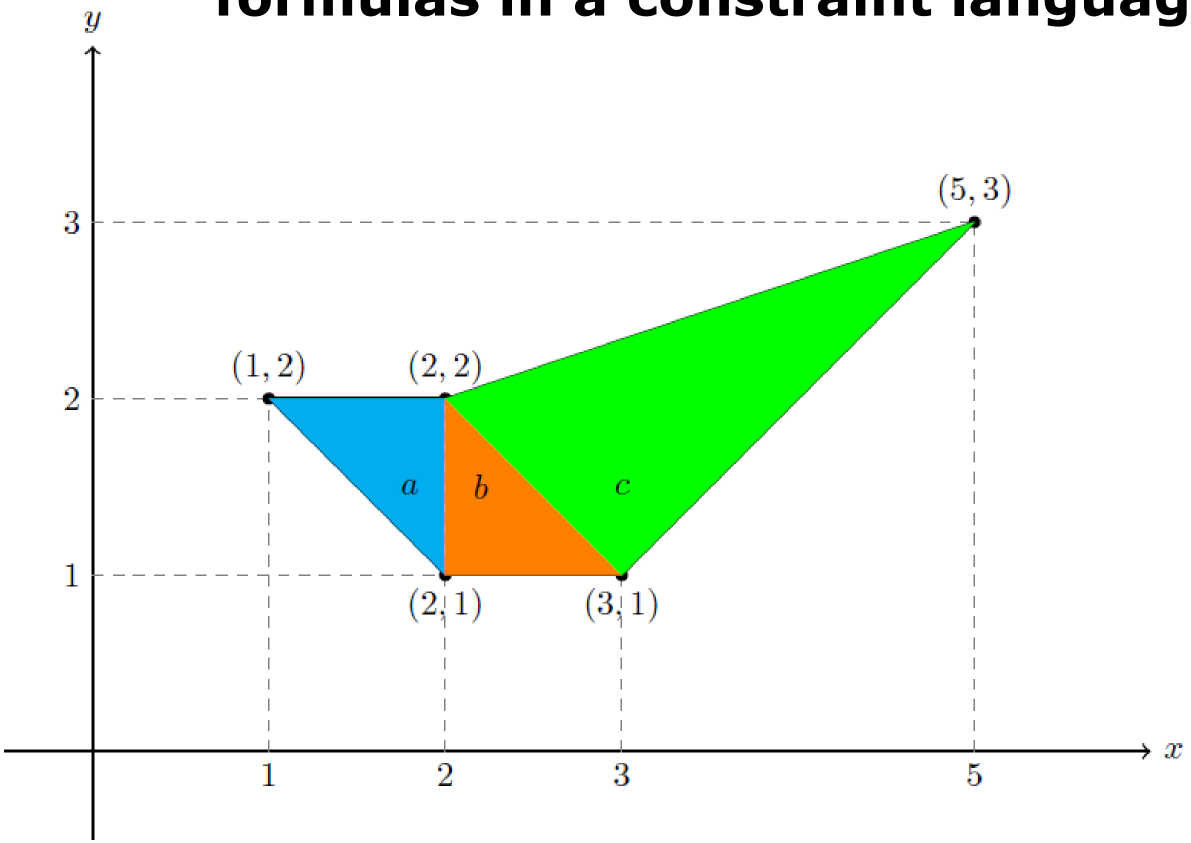
Example



[(1,2) (2,2) (5,3) (3,1) (2,1) (1 2)]

Special Encodings: Constraint Representation

- In this case objects in space are represented by **quantifier free formulas in a constraint language** (e.g., linear constraints).



$$(y + x \geq 3 \wedge x \leq 2 \wedge y \leq 2) \vee (y + x \leq 4 \wedge x \geq 2 \wedge y \geq 1) \vee (y \geq 3 \wedge x \leq 5 \wedge y - \frac{x}{3} \leq \frac{4}{3})$$

Constraint Databases

- The constraint representation of spatial data was the focus of much work in **databases, logic programming and AI** after the paper by Kanellakis, Kuper and Revesz (PODS, 1991).

- The approach was very fruitful theoretically but was not adopted in practice.

- See the book by Revesz for a tutorial introduction.

Geospatial Data Standards

- The **Open Geospatial Consortium (OGC)** and the **International Organization for Standardization (ISO)** have developed many geospatial data standards that are in wide use today. In this tutorial we will cover:
  - **Well-Known Text**
  - **Geography Markup Language**
  - **OpenGIS Simple Feature Access**

Making location count.
www.opengeospatial.org

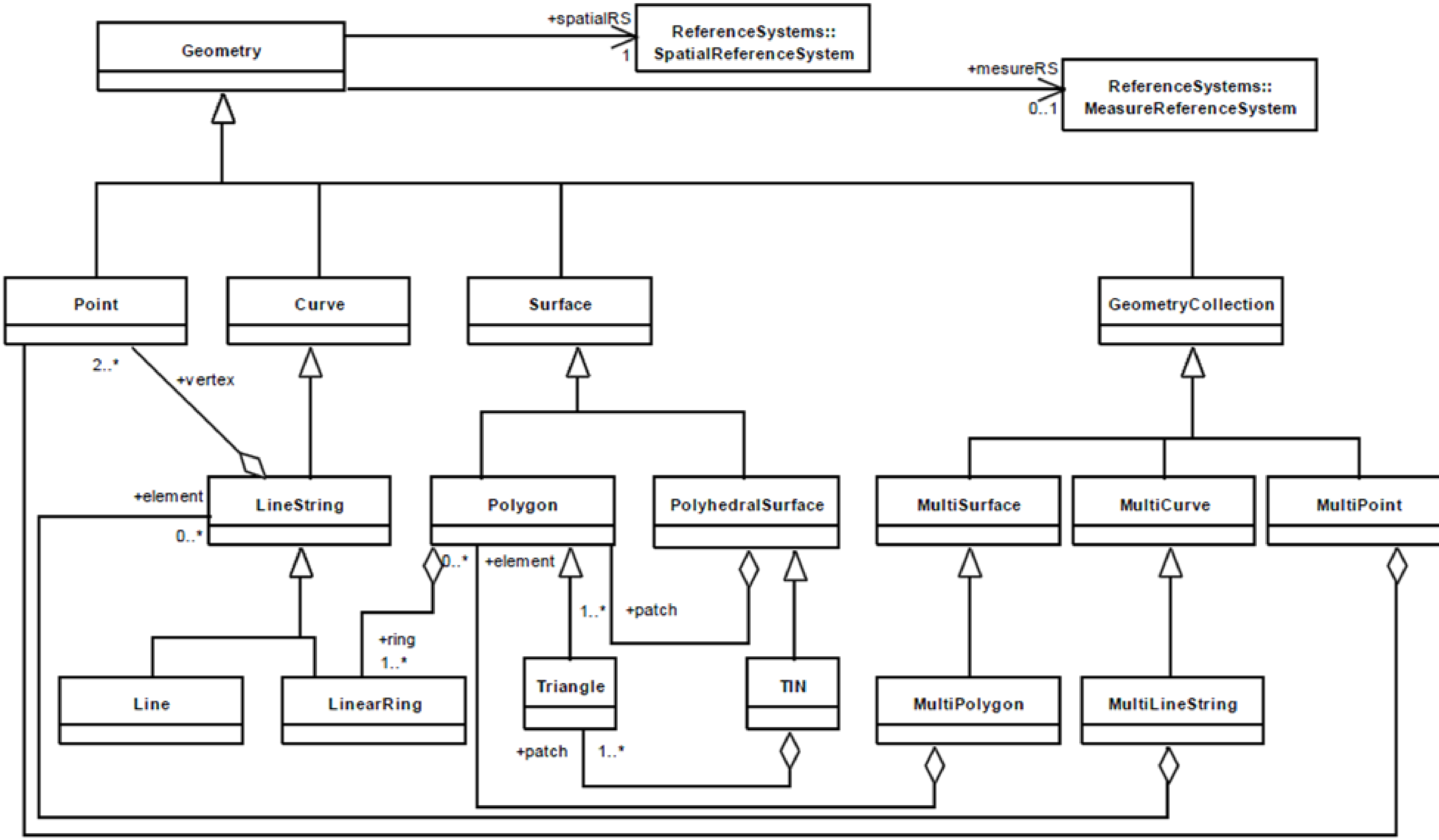
Well-Known Text (WKT)

- WKT is an OGC and ISO standard for representing **geometries**, **coordinate reference systems**, and **transformations** between coordinate reference systems.

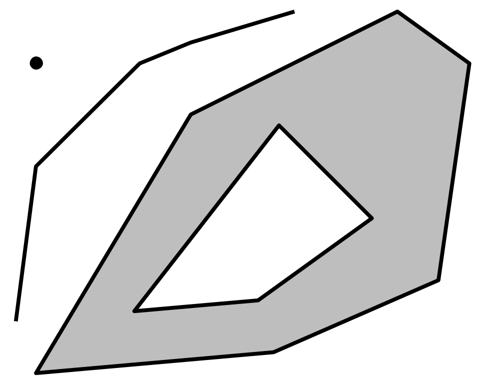
- WKT is specified in **OpenGIS Simple Feature Access - Part 1: Common Architecture** standard which is the same as the **ISO 19125-1** standard. Download from http://portal.opengeospatial.org/files/?artifact_id=25355 .

- This standard concentrates on **simple features:** features with all spatial attributes described piecewise by a straight line or a planar interpolation between sets of points.

WKT Class Hierarchy



Example



WKT representation:

```
GeometryCollection(
  Point(5 35),
  LineString(3 10,5 25,15 35,20 37,30 40),
  Polygon((5 5,28 7,44 14,47 35,40 40,20 30,5 5),
          (28 29,14.5 11,26.5 12,37.5 20,28 29))
  )
```

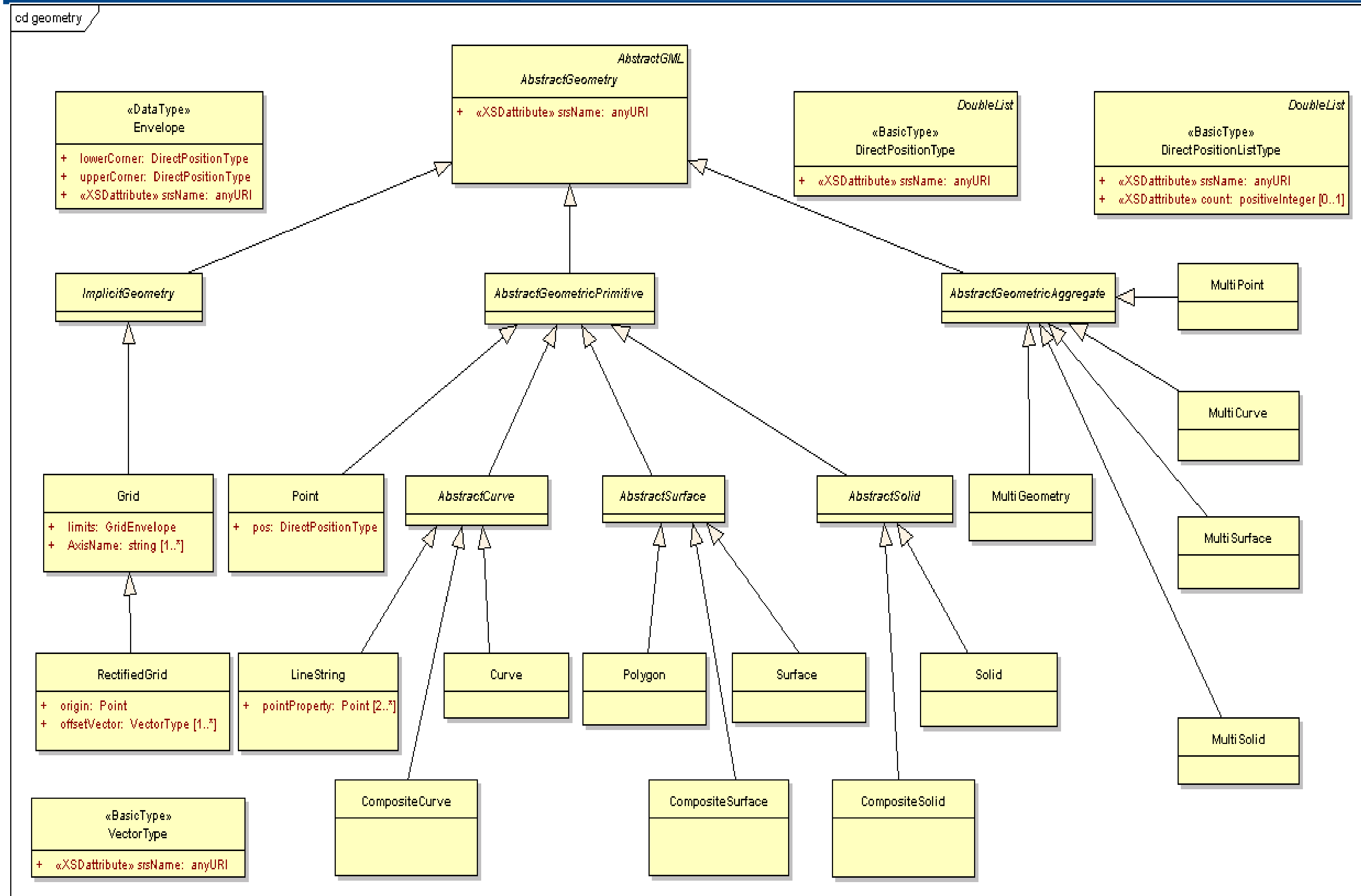
Geography Markup Language (GML)

- **GML** is an **XML-based encoding standard** for the representation of geospatial data.

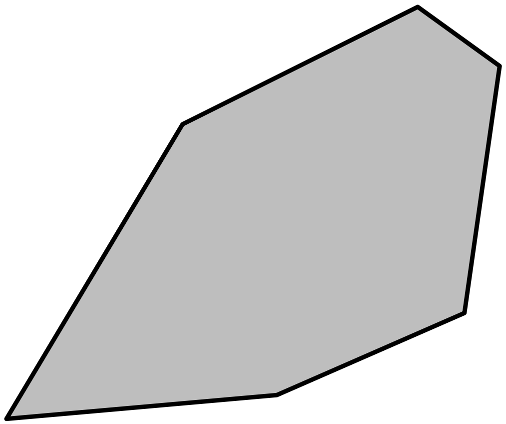
- GML provides XML schemas for defining a variety of concepts: **geographic features, geometry, coordinate reference systems, topology, time** and **units of measurement**.

- **GML profiles** are subsets of GML that target particular applications.
  - **Examples**: Point Profile, GML Simple Features Profile etc.

GML Simple Features: Class Hierarchy



Example



GML representation:

```
<gml:Polygon gml:id="p3" srsName="urn:ogc:def:crs:EPSG:6.6:4326”>
    <gml:exterior>
        <gml:LinearRing>
            <gml:coordinates>
                5,5 28,7 44,14 47,35 40,40 20,30 5,5
            </gml:coordinates>
        </gml:LinearRing>
    </gml:exterior>
</gml:Polygon>
```

OpenGIS Simple Features Access (cont'd)

- OGC has also specified a standard for the **storage, retrieval, query and update of sets of simple features using relational DBMS and SQL.**

- This standard is **"OpenGIS Simple Feature Access - Part 2: SQL Option"** and it is the same as the **ISO 19125-2** standard. Download from http://portal.opengeospatial.org/files/?artifact_id=25354.

- **Related standard**: ISO 13249 SQL/MM - Part 3.

OpenGIS Simple Features Access (cont'd)

- The standard covers two implementations options: (i) using **only the SQL predefined data types** and (ii) using **SQL with geometry types**.

- **SQL with geometry types:**
  - We use the WKT geometry class hierarchy presented earlier to define **new geometric data types for SQL**
  - We define new **SQL functions on those types**.

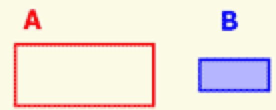
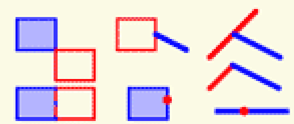
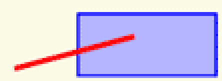
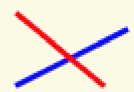
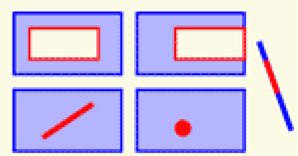
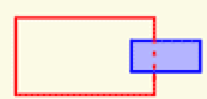
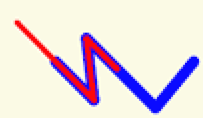
SQL with Geometry Types - Functions

- Functions that **request or check properties** of a geometry:
  - `ST_Dimension(A:Geometry):Integer`
  - `ST_GeometryType(A:Geometry):Character Varying`
  - `ST_AsText(A:Geometry): Character Large Object`
  - `ST_AsBinary(A:Geometry): Binary Large Object`
  - `ST_SRID(A:Geometry): Integer`
  - `ST_IsEmpty(A:Geometry): Boolean`
  - `ST_IsSimple(A:Geometry): Boolean`

SQL with Geometry Types – Functions (cont'd)

- Functions that test **topological relations** between two geometries using the **DE-9IM**:
  - `ST_Equals(A:Geometry, B:Geometry):Boolean`
  - `ST_Disjoint(A:Geometry, B:Geometry):Boolean`
  - `ST_Intersects(A:Geometry, B:Geometry):Boolean`
  - `ST_Touches(A:Geometry, B:Geometry):Boolean`
  - `ST_Crosses(A:Geometry, B:Geometry):Boolean`
  - `ST_Within(A:Geometry, B:Geometry):Boolean`
  - `ST_Contains(A:Geometry, B:Geometry):Boolean`
  - `ST_Overlaps(A:Geometry, B:Geometry):Boolean`
  - `ST_Relate(A:Geometry, B:Geometry, Matrix: Char(9)):Boolean`

DE-9IM Relation Definitions

Beziehung	Definition	Beispiele
A disjoint B	$\begin{bmatrix} F & F & * \\ F & F & * \\ * & * & * \end{bmatrix}$	
A touches B (d(A) > 0 ∨ d(B) > 0)	$\begin{bmatrix} F & T & * \\ * & * & * \\ * & * & * \end{bmatrix} \vee \begin{bmatrix} F & * & * \\ T & * & * \\ * & * & * \end{bmatrix} \vee \begin{bmatrix} F & * & * \\ * & T & * \\ * & * & * \end{bmatrix}$	
A crosses B (d(A) < d(B))	$\begin{bmatrix} T & * & T \\ * & * & * \\ * & * & * \end{bmatrix}$	
A crosses B (d(A) = d(B) = 1)	$\begin{bmatrix} 0 & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$	
A within B	$\begin{bmatrix} T & * & F \\ * & * & F \\ * & * & * \end{bmatrix}$	
A overlaps B (d(A) = d(B), d(A) ≠ 1, d(B) ≠ 1)	$\begin{bmatrix} T & * & T \\ * & * & * \\ T & * & * \end{bmatrix}$	
A overlaps B (d(A) = d(B) = 1)	$\begin{bmatrix} 1 & * & T \\ * & * & * \\ T & * & * \end{bmatrix}$	

- **A equals B** can also be defined by the pattern TFFFTFFFT.
- **A intersects B** is the negation of **A disjoint B**
- **A contains B** is equivalent to **B within A**

SQL with Geometry Types – Functions (cont'd)

- Functions for **constructing new geometries** out of existing ones:
  - `ST_Boundary(A:Geometry):Geometry`
  - `ST_Envelope(A:Geometry):Geometry`
  - `ST_Intersection(A:Geometry, B:Geometry):Geometry`
  - `ST_Union(A:Geometry, B:Geometry):Geometry`
  - `ST_Difference(A:Geometry, B:Geometry):Geometry`
  - `ST_SymDifference(A:Geometry, B:Geometry):Geometry`
  - `ST_Buffer(A:Geometry, distance:Double):Geometry`

Geospatial Relational DBMS

- The OpenGIS Simple Features Access Standard is today been used in all **relational DBMS with a geospatial extension.**

  - The **abstract data type mechanism** of the DBMS allows the representation of all kinds of geospatial data types supported by the standard.
  - The query language (SQL) offers the **functions** of the standard for querying data of these types.

Conclusions

- **Background in geospatial data modeling**:
  - Why geographical information?
  - Geographical Information Science and Systems
  - Geospatial data on the Web and linked geospatial data
  - Abstract geographic space modeling paradigms: discrete objects vs. continuous fields
  - Concrete representations: tessellation vs. vectors vs. constraints
  - Geospatial data standards

- **Next topic:** Geospatial data in the Semantic Web

Background in geospatial data modeling

Presenter: Manolis Koubarakis

Dept. of Informatics and Telecommunications
National and Kapodistrian University of Athens

Outline

- Basic GIS concepts and terminology
- Geographic space modeling paradigms
- Geospatial data standards

Basic GIS Concepts and Terminology

- **Theme:** the information corresponding to a particular domain that we want to model. A theme is a set of **geographic features.**

- **Example:** the countries of Europe

Basic GIS Concepts (cont'd)

- **Geographic feature or geographic object:** a domain entity that can have various **attributes** that describe **spatial and non-spatial** characteristics.

- **Example:** the country Greece with attributes
    - Population
    - Flag
    - Capital
    - Geographical area
    - Coastline
    - Bordering countries

Basic GIS Concepts (cont'd)

- Geographic features can be **atomic** or **complex.**

- **Example:** According to the Kallikratis administrative reform of 2010, Greece consists of:
  - 13 **regions** (e.g., Crete)
  - Each region consists of **perfectures** (e.g., Heraklion)
  - Each perfecture consists of **municipalities** (e.g., Dimos Chersonisou)

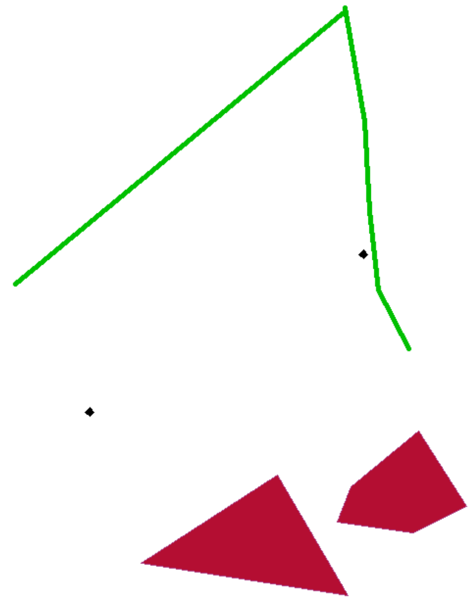
Basic GIS Concepts (cont'd)

- The spatial characteristics of a feature can involve:
  - **Geometric information** (location in the underlying geographic space, shape etc.)
  - **Topological information** (containment, adjacency etc.).

Municipalities of the perfecture of Heraklion:
1. Dimos Irakliou
2. Dimos Archanon-Asterousion
3. Dimos Viannou
4. Dimos Gortynas
5. Dimos Maleviziou
6. Dimos Minoa Pediadas
7. Dimos Festou
8. Dimos Chersonisou

Geometric Information

- Geometric information can be captured by using geometric primitives (**points**, **lines**, **polygons,** etc.) to approximate the spatial attributes of the real world feature that we want to model.



- Geometries are associated with a **coordinate reference system** which describes the coordinate space in which the geometry is defined.

Topological Information

- Topological information is **inherently qualitative** and it is expressed in terms of **topological relations** (e.g., containment, adjacency, overlap etc.).

- Topological information can be **derived from geometric information** or it might be captured by **asserting explicitly the topological relations** between features.

Topological Relations

- The study of topological relations has produced a lot of interesting results by researchers in:
  - GIS
  - Spatial databases
  - Artificial Intelligence (qualitative reasoning and knowledge representation)

The 4-intersection model

- The **4-intersection model** has been defined by Egenhofer and Franzosa in 1991 based on previous work by Egenhofer and colleagues.
- It is based on **point-set topology.**
- **Spatial regions** are defined to be **non-empty, proper subsets of a topological space.** In addition, they must be closed and have connected interiors.
- **Topological relations** are the ones that are invariant under topological homeomorphisms.

4IM and 9IM

- The 4-intersection model can captures **topological relations** between two spatial regions *a* and *b* by considering **whether the intersection of their boundaries and interiors is empty or non-empty**.

- The **9-intersection model** is an extension of the 4-intersection model (Egenhofer and Herring, 1991).

- 9IM captures topological relations between two spatial regions *a* and *b* by considering whether the intersection of their boundaries, interiors and **exteriors** is empty or non-empty.

DE-9IM

- The **dimensionally extended 9-intersection model** has been defined by Clementini and Felice in 1994.

- It is also based on the **point-set topology** of R^2 and deals with **"simple", closed geometries (areas, lines, points).**

- Like its predecessors (4IM, 9IM), it can also be extended to **more complex geometries** (areas with holes, geometries with multiple components).

DE-9IM

- It captures **topological relationships** between two geometries *a* and *b* in R^2 by considering the **dimensions of the intersections of the boundaries, interiors and exteriors** of the two geometries:

$$\text{DE-9IM}(a,b) = \begin{bmatrix} \dim(I(a) \cap I(b)) & \dim(I(a) \cap B(b)) & \dim(I(a) \cap E(b)) \\ \dim(B(a) \cap I(b)) & \dim(B(a) \cap B(b)) & \dim(B(a) \cap E(b)) \\ \dim(E(a) \cap I(b)) & \dim(E(a) \cap B(b)) & \dim(E(a) \cap E(b)) \end{bmatrix}.$$

- The dimension can be **2, 1, 0** and **-1** (dimension of the empty set).

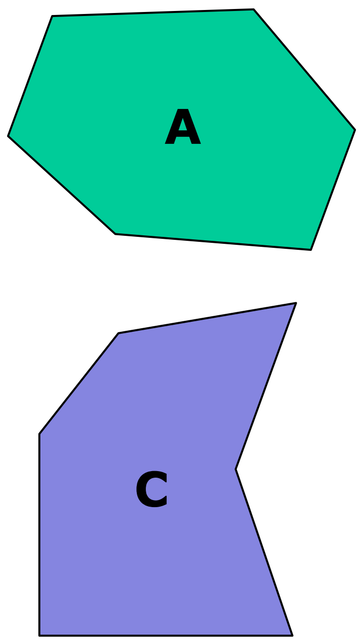
DE-9IM

- **Five jointly exclusive and pairwise disjoint (JEPD)** relationships between two different geometries can be distinguished (**disjoint, touches, crosses, within, overlaps**).

- The model can also be defined using an appropriate **calculus of geometries** that uses these 5 binary relations and boundary operators.

- See the paper: E. Clementini and P. Felice. A Comparison of Methods for Representing Topological Relationships. Information Sciences 80 (1994), pp. 1-34.

Example: A disjoint C

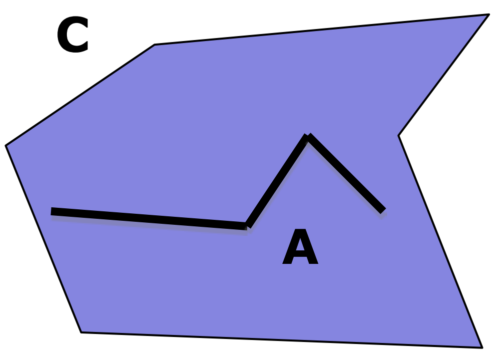


	I(C)	**B**(C)	**E**(C)
I(A)	F	F	*
B(A)	F	F	*
E(A)	*	*	*

Notation:

- T = { 0, 1, 2 }
- F = -1
- * = don't care = { -1, 0, 1, 2 }

Example: A within C

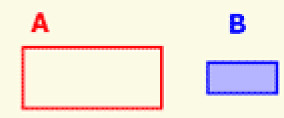
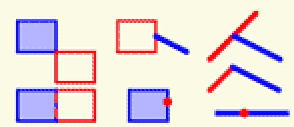
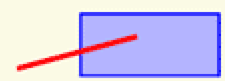
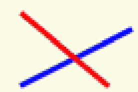
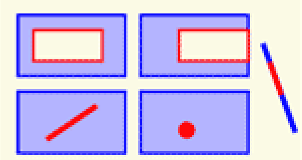
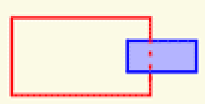
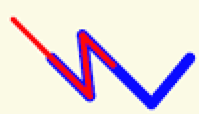


	I(C)	**B**(C)	**E**(C)
I(A)	T	*	F
B(A)	*	*	F
E(A)	*	*	*

Notation equivalent to 3x3 matrix:

- String of 9 characters representing the above matrix in row major order.
- In this case: T*F**F***

DE-9IM Relation Definitions

Beziehung	Definition	Beispiele
A disjoint B	$\begin{bmatrix} F & F & * \\ F & F & * \\ * & * & * \end{bmatrix}$	
A touches B (d(A) > 0 ∨ d(B) > 0)	$\begin{bmatrix} F & T & * \\ * & * & * \\ * & * & * \end{bmatrix} \vee \begin{bmatrix} F & * & * \\ T & * & * \\ * & * & * \end{bmatrix} \vee \begin{bmatrix} F & * & * \\ * & T & * \\ * & * & * \end{bmatrix}$	
A crosses B (d(A) < d(B))	$\begin{bmatrix} T & * & T \\ * & * & * \\ * & * & * \end{bmatrix}$	
A crosses B (d(A) = d(B) = 1)	$\begin{bmatrix} 0 & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$	
A within B	$\begin{bmatrix} T & * & F \\ * & * & F \\ * & * & * \end{bmatrix}$	
A overlaps B (d(A) = d(B), d(A) ≠ 1, d(B) ≠ 1)	$\begin{bmatrix} T & * & T \\ * & * & * \\ T & * & * \end{bmatrix}$	
A overlaps B (d(A) = d(B) = 1)	$\begin{bmatrix} 1 & * & T \\ * & * & * \\ T & * & * \end{bmatrix}$	

The Region Connection Calculus (RCC)

- The primitives of the calculus are **spatial regions**. These are non-empty, regular subsets of a topological space.

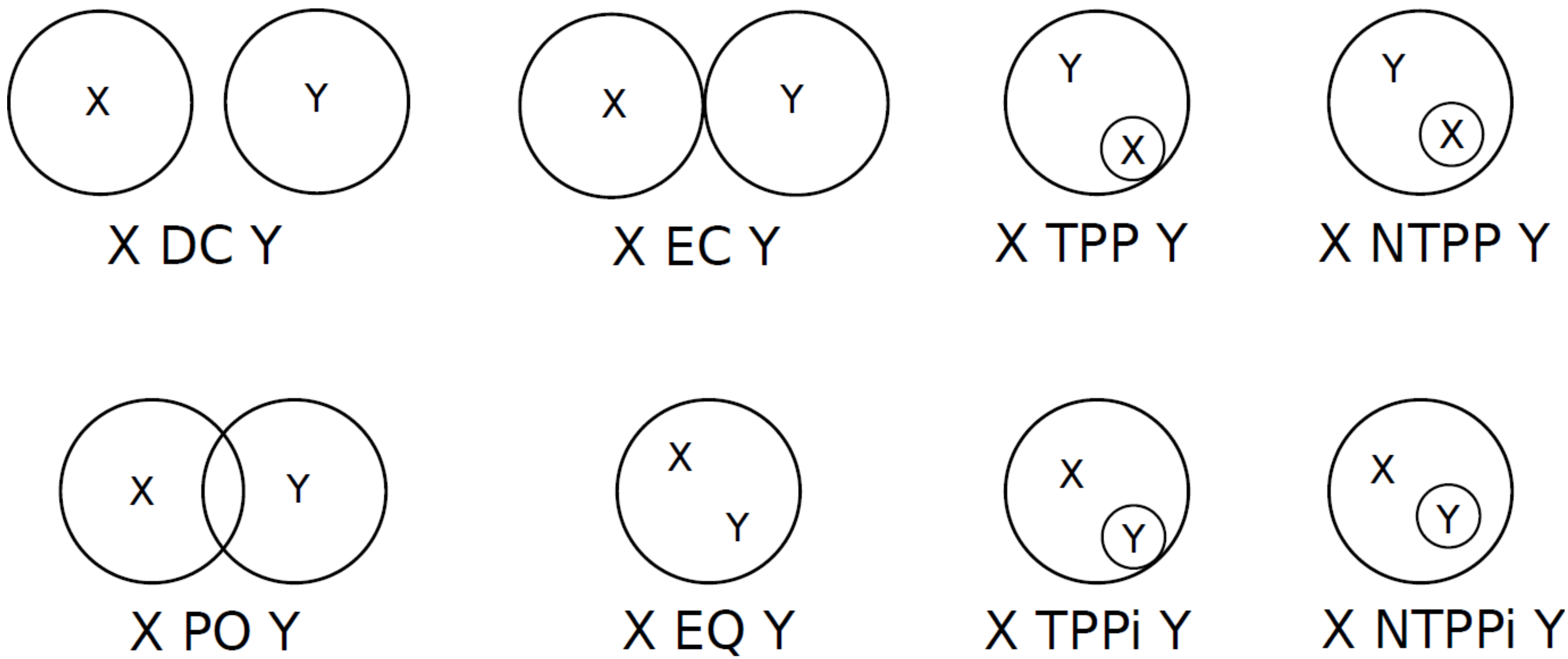
- The calculus is based on a single binary predicate *C* that formalizes the "**connectedness**" relation.
    - ***C(a,b)* is true** when the closure of *a* is connected to the closure of *b* i.e., they have at least one point in common.

- It is axiomatized using first order logic.

- See the original paper by Randell, Cui and Cohn (KR 1991).

RCC-8

- This is a set of **eight JEPD binary relations** that can be defined in terms of predicate *C*.



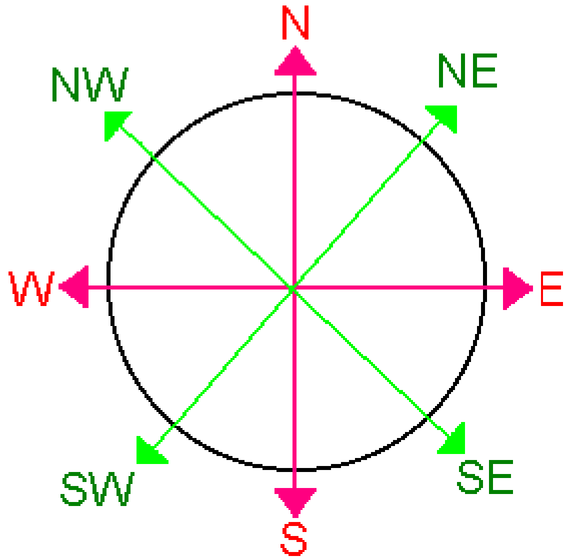
RCC-5

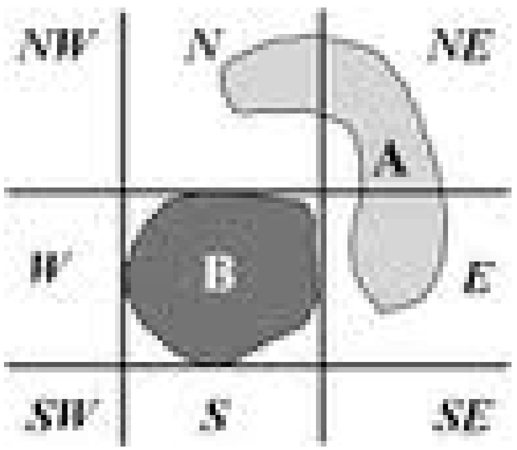
- The **RCC-5 subset** has also been studied. The granularity here is coarser. The boundary of a region is not taken into consideration:
  - No distinction among DC and EC, called just DR.
  - No distinction among TPP and NTPP, called just PP.

- RCC-8 and RCC-5 relations **can also be defined using point-set topology**, and there are very close connections to the models of Egenhofer and others.

More Qualitative Spatial Relations

- **Orientation/Cardinal directions** (left of, right of, north of, south of, northeast of etc.)



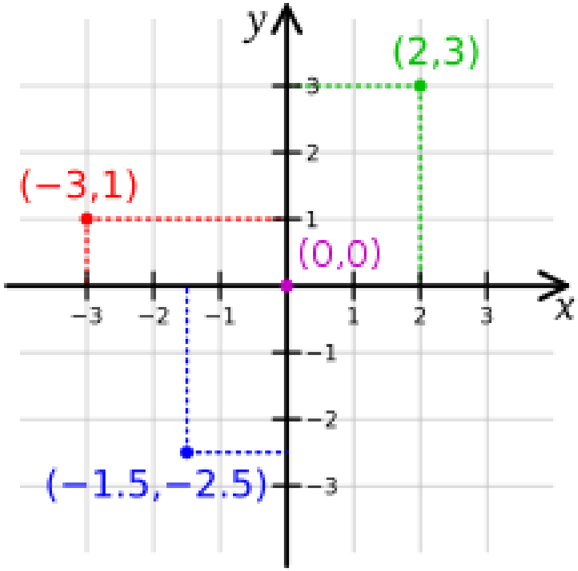


- **Distance** (close to, far from etc.). This information can also be **quantitative**.

Coordinate Systems

- **Coordinate:** one of *n* scalar values that determines the position of a point in an *n*-dimensional space.
- **Coordinate system:** a set of mathematical rules for specifying how coordinates are to be assigned to points.

- **Example**: the Cartesian coordinate system



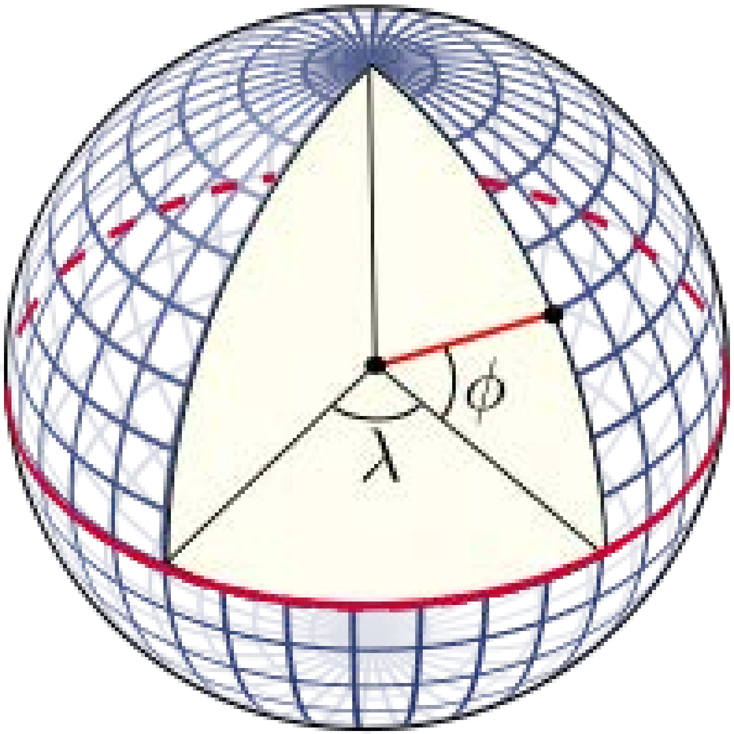
Coordinate Reference Systems

- **Coordinate reference system**: a coordinate system that is related to an **object** (e.g., the Earth, a planar projection of the Earth, a three dimensional mathematical space such as R^3) through a **datum** which specifies its origin, scale, and orientation.

- The term **spatial reference system** is also used.

Geographic Coordinate Reference Systems

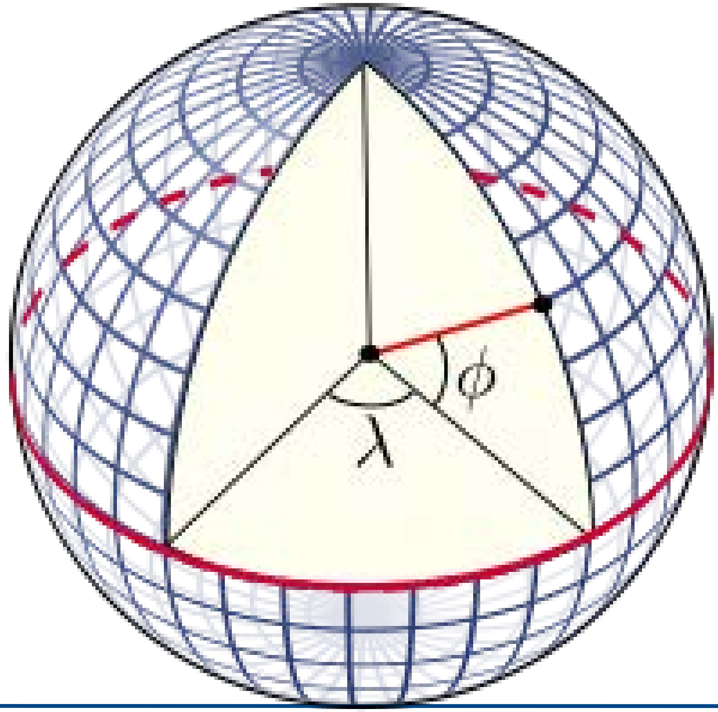
- These are 3-dimensional coordinate systems that utilize **latitude (φ)**, **longitude (λ)** , and optionally **geodetic height (i.e., elevation)**, to capture geographic locations on Earth.



The World Geodetic System

- The **World Geodetic System (WGS)** is the most well-known geographic coordinate reference system and its latest revision is **WGS84.**

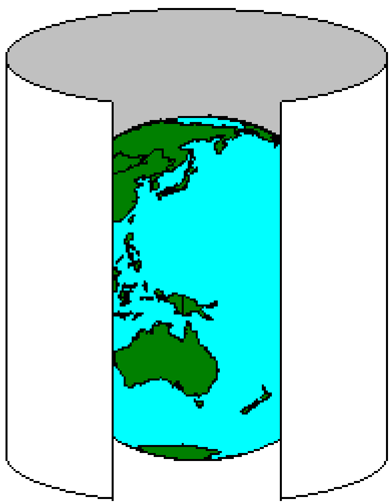
- **Applications:** cartography, geodesy, navigation (GPS), etc.



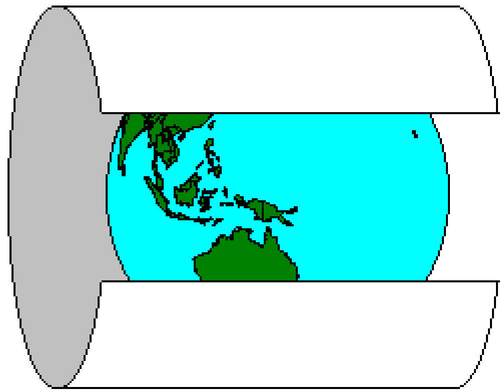
Projected Coordinate Reference Systems

- **Projected coordinate reference system:** they transform the 3-dimensional approximation of the Earth into a 2-dimensional surface (distortions!)

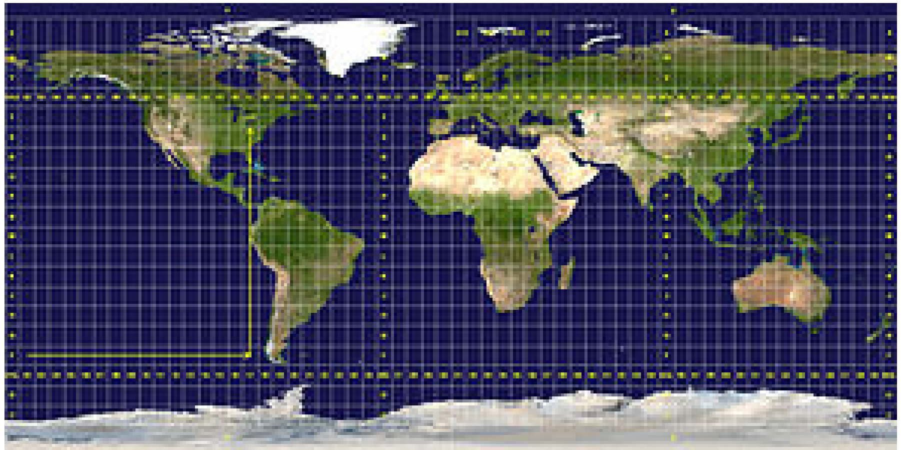
- **Example:** the **Universal Transverse Mercator (UTM)** system



Mercator projection



Transverse Mercator projection



Coordinate Reference Systems (cont'd)

- There are well-known ways to **translate** between co-ordinate reference systems.

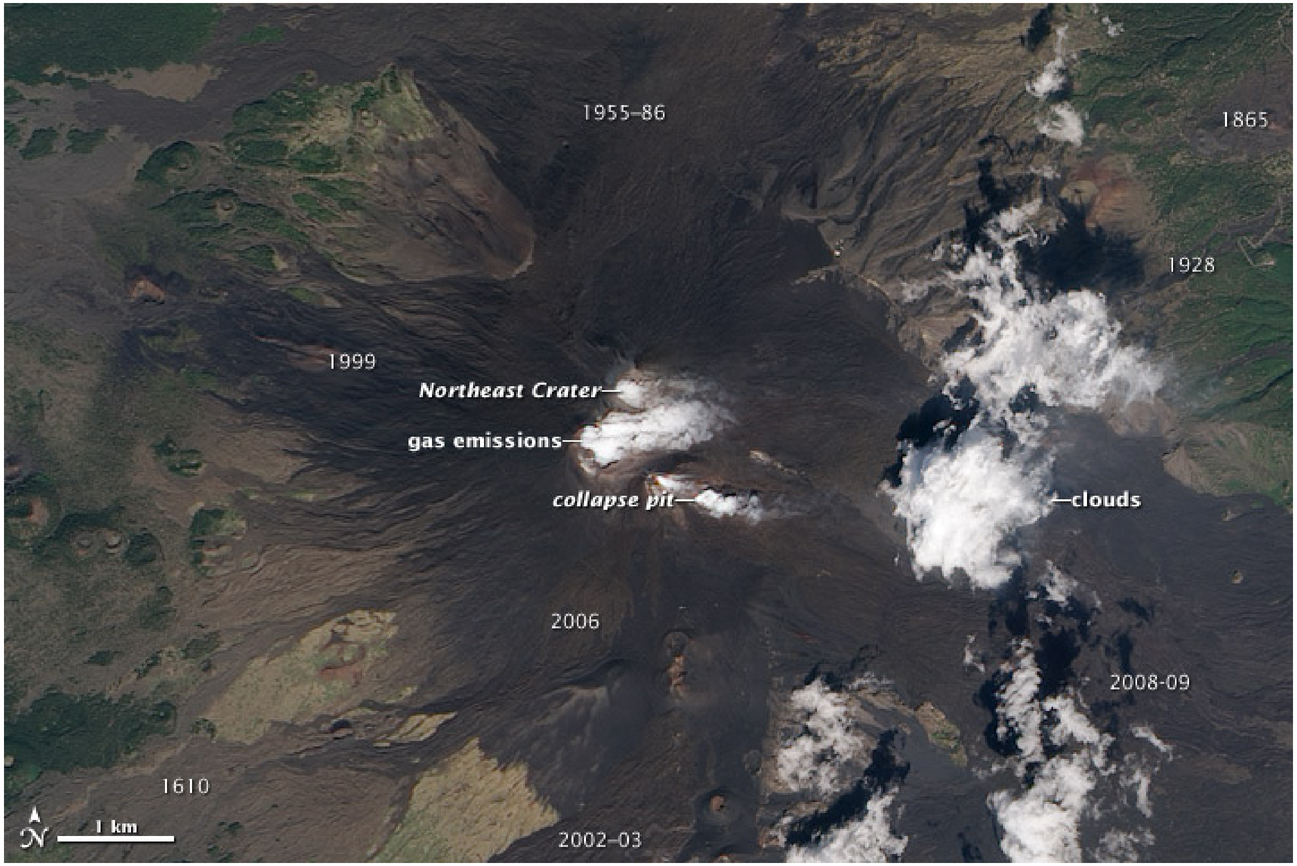
- Various authorities maintain lists of coordinate reference systems. See for example:
  - **OGC** http://www.opengis.net/def/crs/
  - **European Petroleum Survey Group** http://www.epsg-registry.org/

Geographic Space Modeling Paradigms

- **Abstract** geographic space modeling paradigms: discrete objects vs. continuous fields
- **Concrete representations**: tessellation vs. vectors vs. constraints

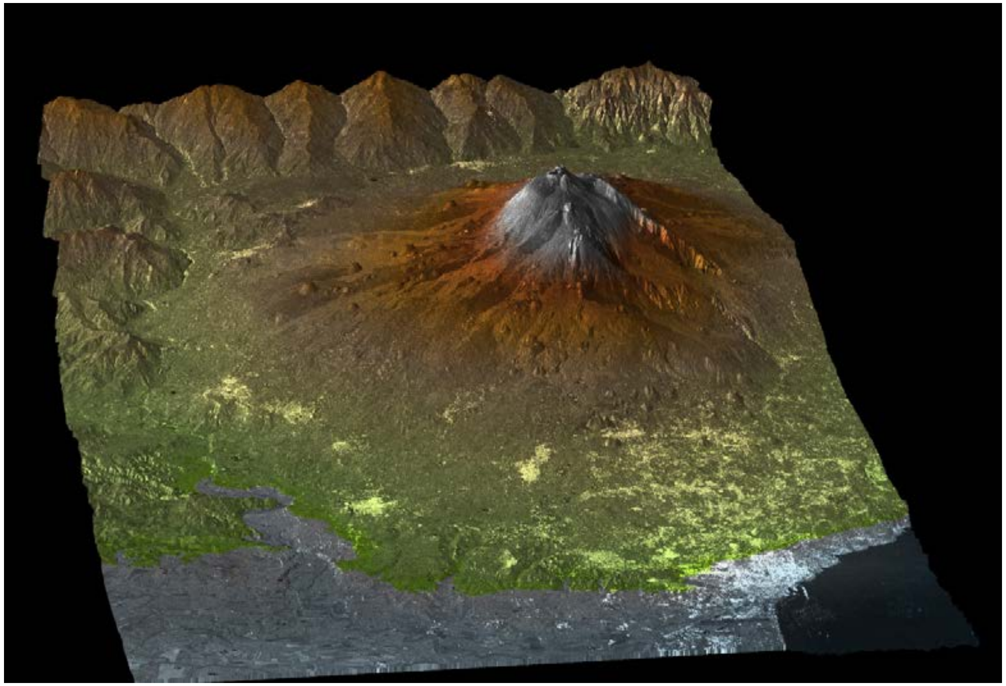
Abstract Modeling Paradigms: Feature-based

- **Feature-based** (or **entity-based** or **object-based**). This kind of modeling is based on the concepts we presented already.



Abstract Modeling Paradigms: Field-based

- Each point *(x,y)* in geographic space is associated with one or several attribute values defined as **continuous functions** in *x* and *y*.
- **Examples:** elevation, precipitation, humidity, temperature for each point *(x,y)* in the Euclidean plane.



From Abstract Modeling to Concrete Representations

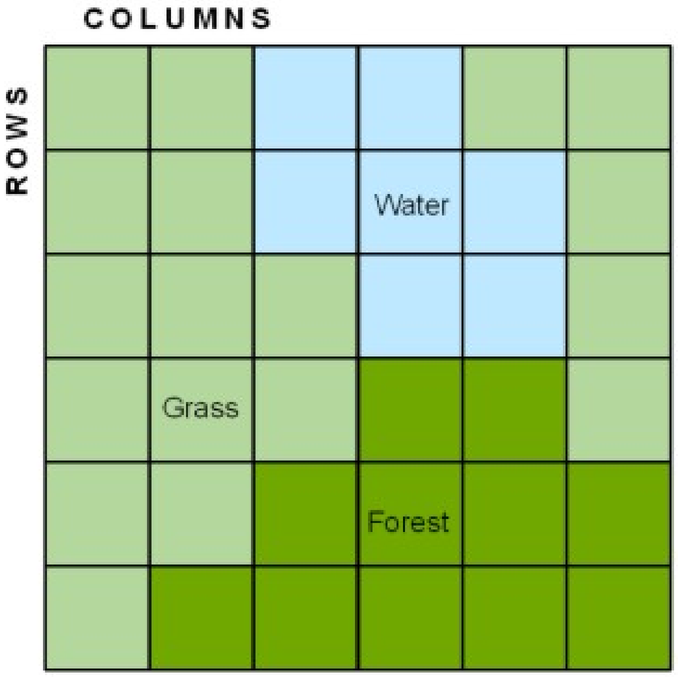
- **Question**: How do we represent the **infinite objects** of the abstract representations (points, lines, fields etc.) **by finite means** (in a computer)?

- **Answers**:
  - **Approximate** the continuous space (e.g., $\mathbb{R}^2$) by a discrete one ($\mathbb{Z}^2$).
  - Use **special encodings**

Approximations: Tessellation

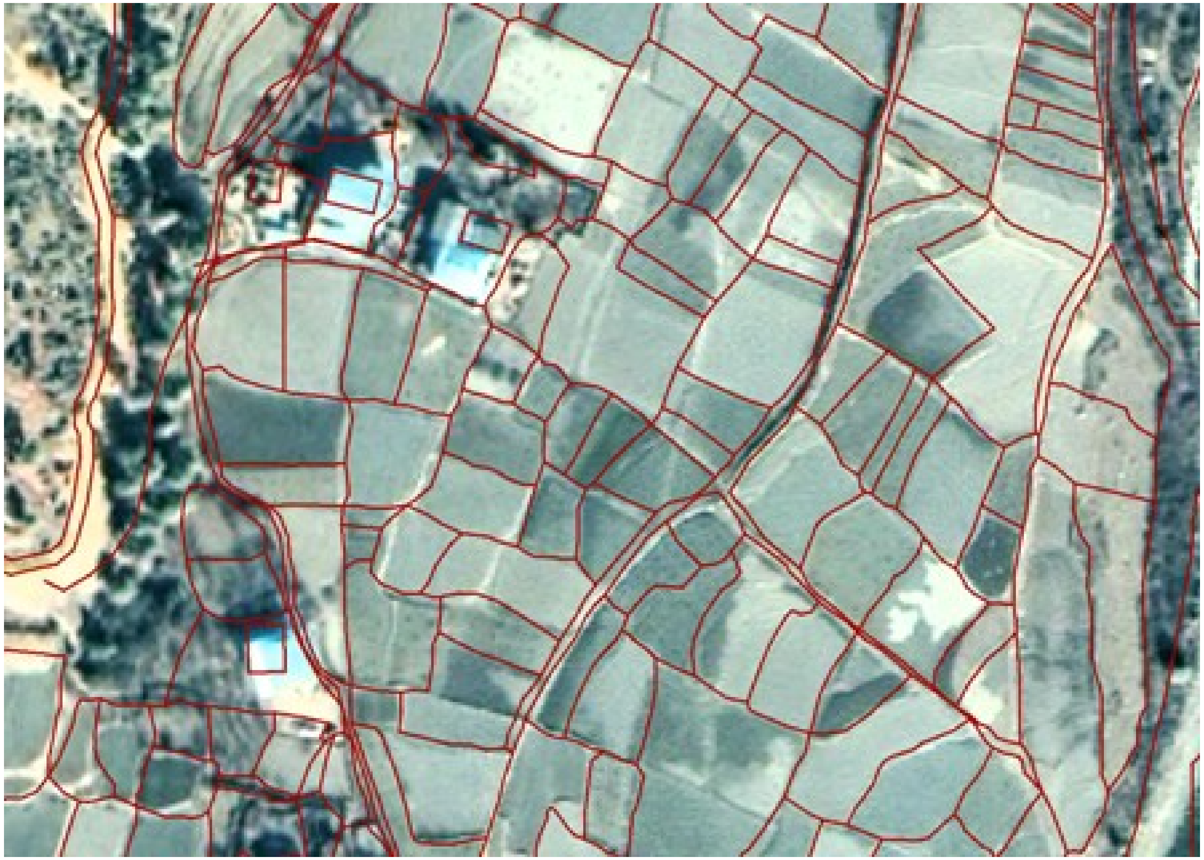
- In this case a **cellular decomposition of the plane** (usually, a grid) serves as a basis for representing the geometry.

- **Example:** raster representation (fixed or regular tesselation)



Example

- **Cadastral map (irregular tessellation)** overlayed on a satellite image.

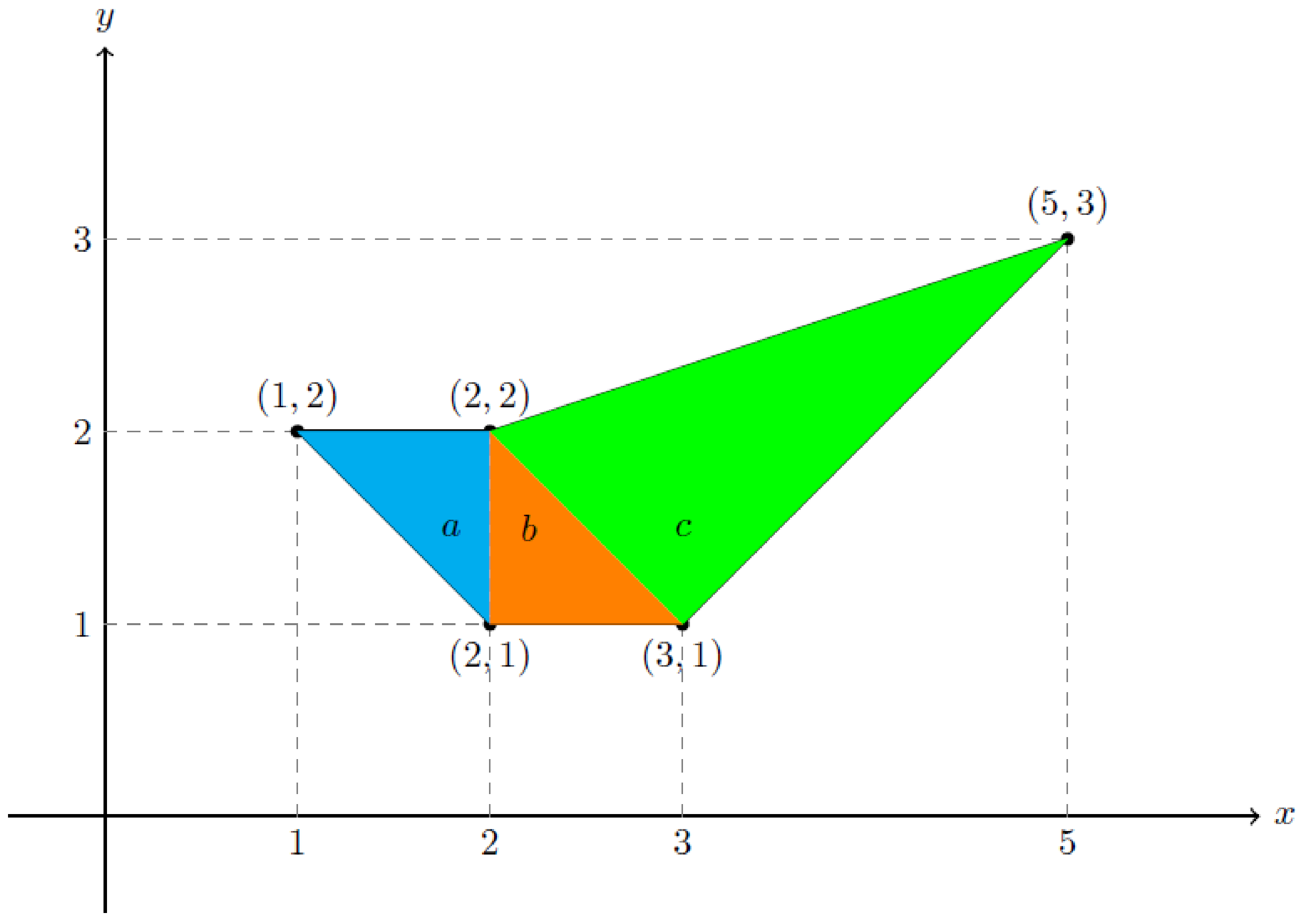


Special Encodings: Vector Representation

- In this case objects in space are represented using **points** as primitives as follows:
  - A **point** is represented by a tuple of coordinates.
  - A **line segment** is represented by a pair with its beginning and ending point.
  - **More complex objects** such as arbitrary lines, curves, surfaces etc. are built recursively by the basic primitives using constructs such as lists, sets etc.

- This is the approach **used in all GIS and other popular systems today. It has also been standardized** by various international bodies.

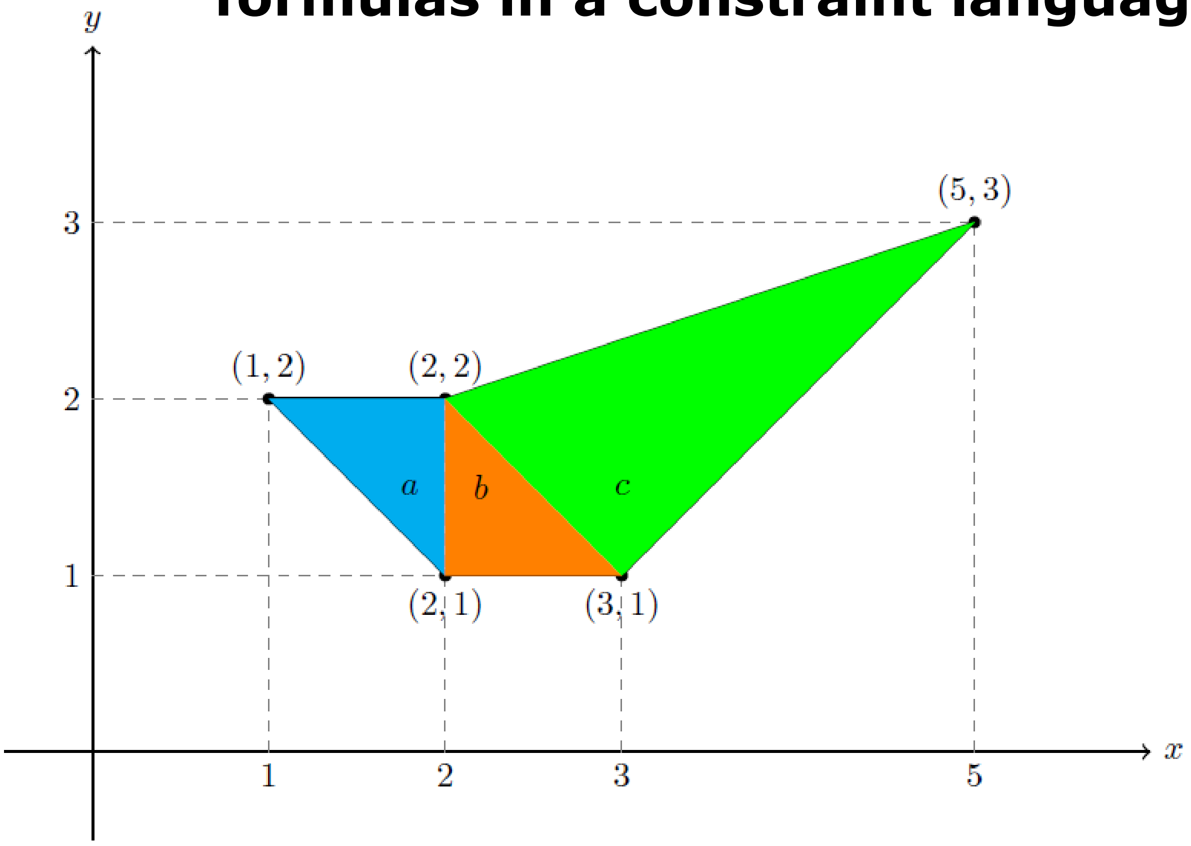
Example



[(1,2) (2,2) (5,3) (3,1) (2,1) (1 2)]

Special Encodings: Constraint Representation

- In this case objects in space are represented by **quantifier free formulas in a constraint language** (e.g., linear constraints).



$$(y + x \geq 3 \wedge x \leq 2 \wedge y \leq 2) \vee (y + x \leq 4 \wedge x \geq 2 \wedge y \geq 1) \vee (y \geq 3 \wedge x \leq 5 \wedge y - \frac{x}{3} \leq \frac{4}{3})$$

Constraint Databases

- The constraint representation of spatial data was the focus of much work in **databases, logic programming and AI** after the paper by Kanellakis, Kuper and Revesz (PODS, 1991).
- The approach was very fruitful theoretically but was not adopted in practice.
- See the book by Revesz for a tutorial introduction.

Geospatial Data Standards

- The **Open Geospatial Consortium (OGC)** and the **International Organization for Standardization (ISO)** have developed many geospatial data standards that are in wide use today. In this tutorial we will cover:
  - **Well-Known Text**
  - **Geography Markup Language**
  - **OpenGIS Simple Feature Access**

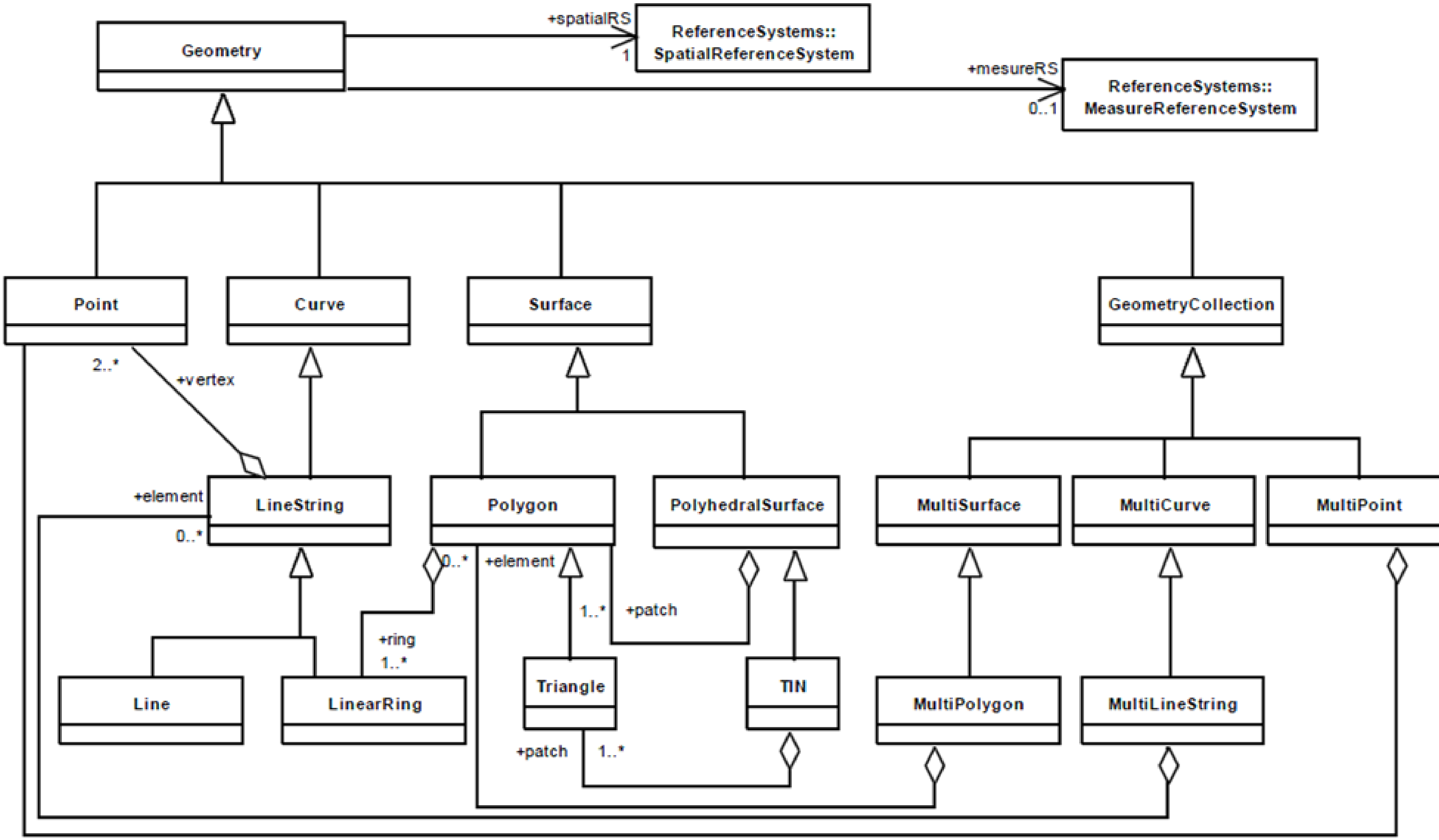
Well-Known Text (WKT)

- WKT is an OGC and ISO standard for representing **geometries**, **coordinate reference systems**, and **transformations** between coordinate reference systems.

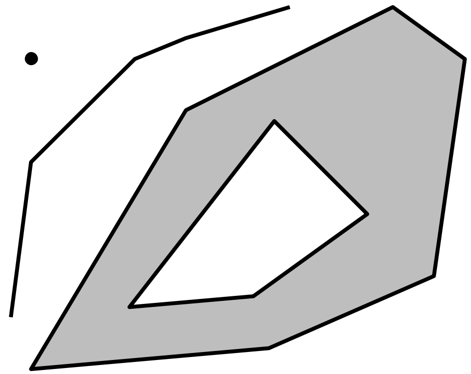
- WKT is specified in **OpenGIS Simple Feature Access - Part 1: Common Architecture** standard which is the same as the **ISO 19125-1** standard. Download from http://portal.opengeospatial.org/files/?artifact_id=25355 .

- This standard concentrates on **simple features:** features with all spatial attributes described piecewise by a straight line or a planar interpolation between sets of points.

WKT Class Hierarchy



Example



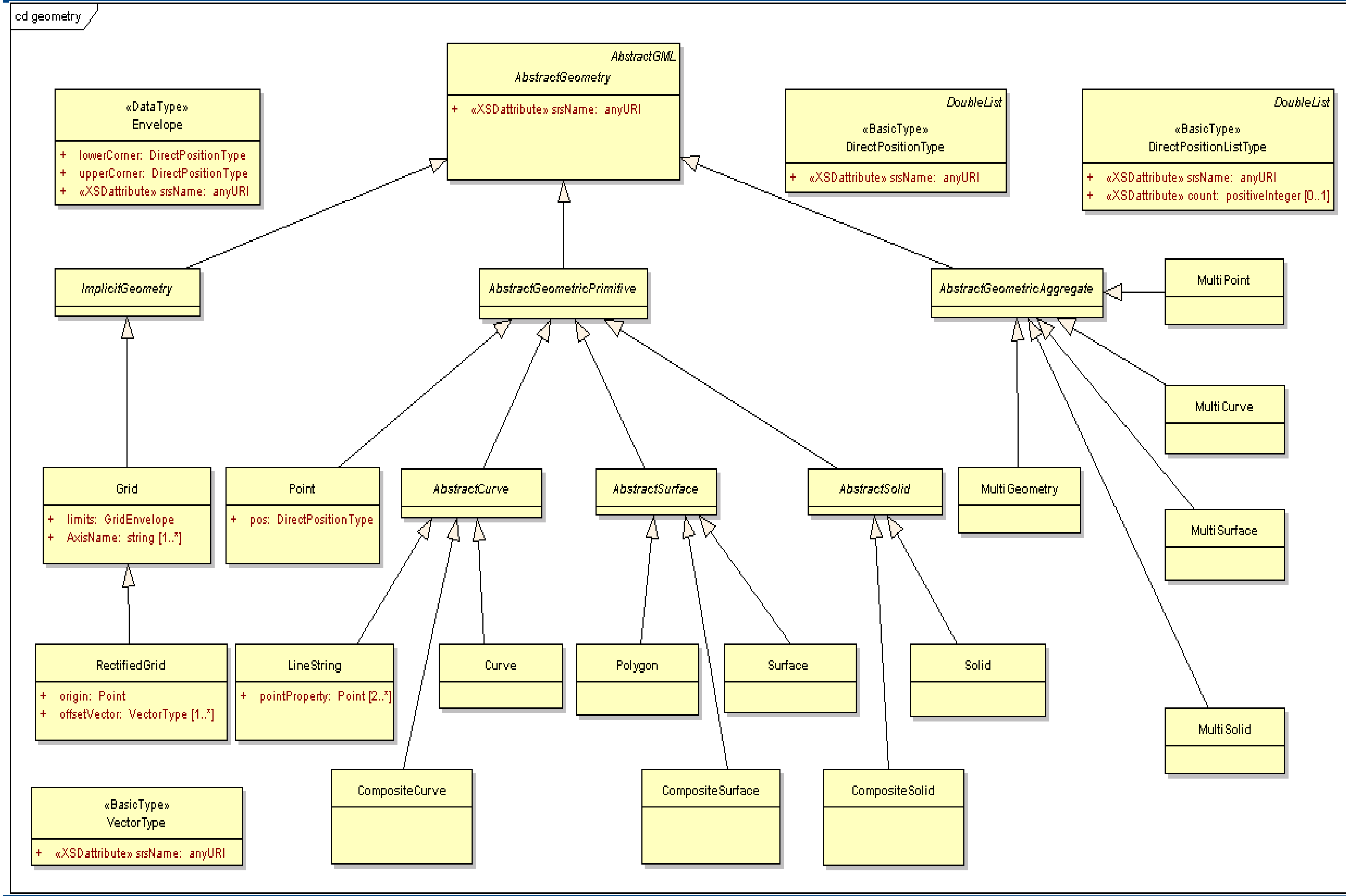
WKT representation:

```
GeometryCollection(
  Point(5 35),
  LineString(3 10,5 25,15 35,20 37,30 40),
  Polygon((5 5,28 7,44 14,47 35,40 40,20 30,5 5),
          (28 29,14.5 11,26.5 12,37.5 20,28 29))
  )
```

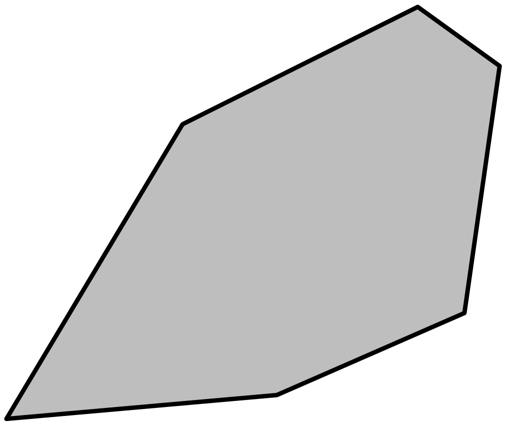
Geography Markup Language (GML)

- **GML** is an **XML-based encoding standard** for the representation of geospatial data.
- GML provides XML schemas for defining a variety of concepts: **geographic features, geometry, coordinate reference systems, topology, time** and **units of measurement**.
- **GML profiles** are subsets of GML that target particular applications.
    - **Examples**: Point Profile, GML Simple Features Profile etc.

GML Simple Features: Class Hierarchy



Example



GML representation:

```
<gml:Polygon gml:id="p3" srsName="urn:ogc:def:crs:EPSG:6.6:4326”>
    <gml:exterior>
        <gml:LinearRing>
            <gml:coordinates>
                5,5 28,7 44,14 47,35 40,40 20,30 5,5
            </gml:coordinates>
        </gml:LinearRing>
    </gml:exterior>
</gml:Polygon>
```

OpenGIS Simple Features Access (cont'd)

- OGC has also specified a standard for the **storage, retrieval, query and update of sets of simple features using relational DBMS and SQL.**

- This standard is **"OpenGIS Simple Feature Access - Part 2: SQL Option"** and it is the same as the **ISO 19125-2** standard. Download from http://portal.opengeospatial.org/files/?artifact_id=25354.

- **Related standard**: ISO 13249 SQL/MM - Part 3.

OpenGIS Simple Features Access (cont'd)

- The standard covers two implementations options: (i) using **only the SQL predefined data types** and (ii) using **SQL with geometry types**.

- **SQL with geometry types:**
  - We use the WKT geometry class hierarchy presented earlier to define **new geometric data types for SQL**
  - We define new **SQL functions on those types**.

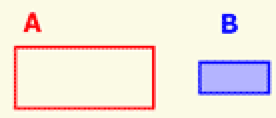
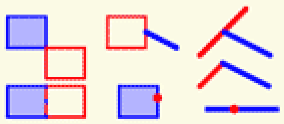
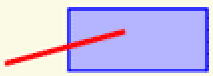
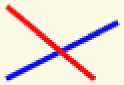
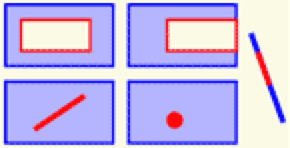
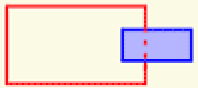
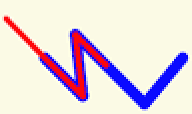
SQL with Geometry Types - Functions

- Functions that **request or check properties** of a geometry:
  - `ST_Dimension(A:Geometry):Integer`
  - `ST_GeometryType(A:Geometry):Character Varying`
  - `ST_AsText(A:Geometry): Character Large Object`
  - `ST_AsBinary(A:Geometry): Binary Large Object`
  - `ST_SRID(A:Geometry): Integer`
  - `ST_IsEmpty(A:Geometry): Boolean`
  - `ST_IsSimple(A:Geometry): Boolean`

SQL with Geometry Types – Functions (cont'd)

- Functions that test **topological relations** between two geometries using the **DE-9IM**:
  - `ST_Equals(A:Geometry, B:Geometry):Boolean`
  - `ST_Disjoint(A:Geometry, B:Geometry):Boolean`
  - `ST_Intersects(A:Geometry, B:Geometry):Boolean`
  - `ST_Touches(A:Geometry, B:Geometry):Boolean`
  - `ST_Crosses(A:Geometry, B:Geometry):Boolean`
  - `ST_Within(A:Geometry, B:Geometry):Boolean`
  - `ST_Contains(A:Geometry, B:Geometry):Boolean`
  - `ST_Overlaps(A:Geometry, B:Geometry):Boolean`
  - `ST_Relate(A:Geometry, B:Geometry, Matrix: Char(9)):Boolean`

DE-9IM Relation Definitions

Beziehung	Definition	Beispiele
A disjoint B	$\begin{bmatrix} F & F & * \\ F & F & * \\ * & * & * \end{bmatrix}$	
A touches B (d(A) > 0 ∨ d(B) > 0)	$\begin{bmatrix} F & T & * \\ * & * & * \\ * & * & * \end{bmatrix} \vee \begin{bmatrix} F & * & * \\ T & * & * \\ * & * & * \end{bmatrix} \vee \begin{bmatrix} F & * & * \\ * & T & * \\ * & * & * \end{bmatrix}$	
A crosses B (d(A) < d(B))	$\begin{bmatrix} T & * & T \\ * & * & * \\ * & * & * \end{bmatrix}$	
A crosses B (d(A) = d(B) = 1)	$\begin{bmatrix} 0 & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$	
A within B	$\begin{bmatrix} T & * & F \\ * & * & F \\ * & * & * \end{bmatrix}$	
A overlaps B (d(A) = d(B), d(A) ≠ 1, d(B) ≠ 1)	$\begin{bmatrix} T & * & T \\ * & * & * \\ T & * & * \end{bmatrix}$	
A overlaps B (d(A) = d(B) = 1)	$\begin{bmatrix} 1 & * & T \\ * & * & * \\ T & * & * \end{bmatrix}$	

- **A equals B** can also be defined by the pattern TFFFTFFFT.
- **A intersects B** is the negation of **A disjoint B**
- **A contains B** is equivalent to **B within A**

SQL with Geometry Types – Functions (cont'd)

- Functions for **constructing new geometries** out of existing ones:
  - `ST Boundary(A:Geometry):Geometry`
  - `ST_Envelope(A:Geometry):Geometry`
  - `ST_Intersection(A:Geometry, B:Geometry):Geometry`
  - `ST_Union(A:Geometry, B:Geometry):Geometry`
  - `ST_Difference(A:Geometry, B:Geometry):Geometry`
  - `ST_SymDifference(A:Geometry, B:Geometry):Geometry`
  - `ST_Buffer(A:Geometry, distance:Double):Geometry`

Geospatial Relational DBMS

- The OpenGIS Simple Features Access Standard is today been used in all **relational DBMS with a geospatial extension.**
  - The **abstract data type mechanism** of the DBMS allows the representation of all kinds of geospatial data types supported by the standard.
  - The query language (SQL) offers the **functions** of the standard for querying data of these types.

Conclusions

- **Background in geospatial data modeling**:
  - Why geographical information?
  - Geographical Information Science and Systems
  - Geospatial data on the Web and linked geospatial data
  - Abstract geographic space modeling paradigms: discrete objects vs. continuous fields
  - Concrete representations: tessellation vs. vectors vs. constraints
  - Geospatial data standards
- **Next topic:** Geospatial data in the Semantic Web