POINT BASED INTEGRATION FOR GEO-SPATIAL DATA INFRASTRUCTURES

Ismail Wadembere, Patrick Ogao
Faculty of Computing & Information Technology, Makerere University
P.O. Box 7062, Kampala, Uganda
iwadembere@cit.mak.ac.ug, ogao@cit.mak.ac.ug

Abstract

There exists a lot of spatial data due advances in geo-spatial data capturing and collection technologies and tools like GPS, Satellite, Hand held Cameras, and total station based ground survey methods which are now available in most parts of developing world. But these data are not fully utilized when making geo-spatial decision not because of lack of analysis and modeling packages and personal skills, but mainly to due lack of an easy way to bring those data together so the different data sets can support and supplement the understanding and utilization of each other.

In this paper, we present an approach which can be used to integrate geo-spatial data basing on points to represent the different spatial elements (point, polylines, and polygons) which make up spatial objects/feature and for correlating the features in different data sets and for building spatial data infrastructures (SDI).

With is approach, the nodes and vertices are the characteristics of points that are used to integrate and related different data in SDI. It has the advantage that GIS practitioners deal with one type of spatial element and since it is the bases of all geo-database manipulation, it makes it easy to exchange data in different form. This provides a start towards solving the issue of spatial integration of data comprising of many complex features with varying geometries which makes it difficult for users to develop one approach to adjust the discrepancies among thematically similar data sets. It also provides avenue for geo-spatial data storage and manipulation in opens standards like Geography Markup Language (GML)

Keywords: Spatial Data Integration, GIS, Geometry Adjustment, Spatial Data Infrastructures

1 INTRODUCTION

In order to make most of spatial decisions, users need to use and integrate geo-spatial data from different sources. This is almost done by every geospatial practitioner as it is difficult for an individual to have all the data required as geo-spatial data sets are expensive to capture and huge to manage. But, as users integrate different data sets, there are some shortcomings, one of them being the inability to systematically adjust the geometries used in representing the different spatial objects/features in data sets so that they can overlay perfectly fulfilling the topological requirements needed for spatial analysis and modeling.

Geospatial data sets can be combined and integrated by overlying and merging layers, but this is not enough and can not solve the above problem as geometrical differences within individual features among thematically similar data sets can still remain present even after overlaying and merging. This calls for adjustment of primary elements – the primitives (points, polylines, and polygons) of individual geometrical features (Masuyama, 2006).

The point based geospatial feature geometry adjustment approach is being adopted as spatial data has been conceptually defined as geometric data consisting of points, polylines, and polygons (Lu et al., 2007) and these conceptual objects (geometric data) that are used to represent GIS objects/features which should be robust and efficient in terms of storage, retrieval, processing, and good graphical reconstructing.

However, to handle the different geometry primitives (points, arcs, and polygons) at the same time brings in challenges in terms of requiring different approaches and algorithms to adjust the different primitives (Masuyama, 2006). This can be observed by taking a close look at the different primitives, as it is evident they have different geometries and situation could be made simpler if only one type of primitive could be handled during geometry adjustment. In this research we look at representing arcs (polylines) and polygons (areas) using points as a move towards providing only one primitive type - points on which geometry adjustment can take place.

It is also important to note that as we get motivated in developing this approach, we need to think about robust GIS data storage, sharing, and retrieval. Thus, the need for Geography Markup Language (GML) - an environment to be used in storage, management, and sharing of the primitives. According to Chang and Park (2006) spatial data is better handled in GML due to its strength in storage, modeling, transportation, and sharing of geo-spatial data.

GML is currently being used by many to encode spatial data using the eXtensible Markup Language (XML). Since XML uses portable data, it enables GML to provide an open bridge between closed vendor-specific systems and file formats (OGC, 2007). This has made GML to be adopted by many for geospatial data sharing. GML was developed by OpenGIS Consortium and it is a standard for encoding geographical information (Lu et al., 2007). It is now the de facto standard for electronic spatial data exchange among the applications of Web and distributed GIS.

The importance of GML has been demonstrated and emphasized by Zhang et al. (2003) and they concluded that “As an open, non-proprietary industry standard, GML overcomes the problems of current GIS software proprietary data models and database structures. Compared with other standards, such as Geographic Data File (GDF) and Spatial Data Transfer Standard (SDTS), GML approach has the advantage of enabling on-line data exchange. GML holds promise in providing a standard way to share and use the existing spatial data over the Web”.

In this paper, we develop an approach for representing spatial data primitives (point, arc, and polygon) into point primitives in GML. We start by looking at the spatial data integration problem, geometrical integration requirements so that we focus our efforts towards geo-spatial data geometrical correction and integration. Then we look at the nature of spatial data, GML and geometry representation. This is followed by the decomposition of arc and polygon features into the point primitives. We end by outlining further research and a conclusion of our findings.

2 THE SPATIAL DATA INTEGRATION PROBLEM

According to Global Spatial Data Infrastructure (GSDI) Secretariat (GSDI, 2005) and the Federal Geographic Data Committee (FGDC) (FGDC, 2007), the goal of Spatial Data Infrastructure (SDI) is to reduce duplication of effort among organizations, improve quality and reduce costs related to geographic information management, to make geographic data more accessible to the public, to increase the benefits of using available data, and to establish key partnerships with various users to increase data availability. As SDI in different countries is being developed, the issue of data management (Malhotra, 2000) and data integration for use in GIS analysis and modeling is needed to cement achievements of SDI.

SDI has two data types: the foundation data (spatial data) and the framework data (thematic data) (FGDC, 2007) (GSDI, 2005). The major classes of spatial data sets include cadastral, topography, boundary, geodetic control, surface water, ground water, soils, geology, land cover, agriculture, and forestry. And thematic spatial data include social, economic, health, demographics, famine, and poverty. The main focus in our approach is the spatial data which is the foundation and it is the type that requires much disintegration handling as framework data is based on the foundation data.

We should note that even foundation data sets from different sources must be made available continuous at an appropriate level so that it is possible to combine them in a consistent way at any time. This will facilitate to share them between several users under conditions that do not restrict their extensive use. Making it easy to discover available spatial data, evaluate their fitness for different purposes and to know the conditions under which they are applicable (GSDI, 2005).

Data integration is one of the oldest research fields in the database area, but still remains a difficult issue to ignore as spatial data need to be integrated during the process of bring it to a common sharing platform. Integration of spatial data from different sources (multiple geodatabase or information systems) generally aims at combining/merging selected sources, databases, systems so that they form a unified new whole and saves users the bidden of interacting with many systems.

Some data sources, databases, and information systems are not designed for integration and whenever integrated access to different sources is desired, the sources and their data in many cases do not fit together. They have to be coalesced by additional adaptation and reconciliation functionality. We should note that there is no one single integration approach although the goal is always to provide a homogeneous, unified view on data from different sources, the particular integration task may depend on (1) the architectural view of system, (2) the content and functionality of the component systems, (3) the kind of information that is managed by component systems (spatial data, alphanumeric data, multimedia data; structured, semi-structured, unstructured data), (4) requirements concerning autonomy of component systems, (5) intended use of the integrated system (read-only or write access), (6) performance requirements, (7) the available resources (time, money, human resources, know-how, etc.) and (8) the source and type of method used to collect the data (Ziegler and Dittrich, 2004).

Additionally, several kinds of heterogeneity typically have to be considered. These include differences in (1) hardware and operating systems, (2) data management software, (3) data models, schemas, and data semantics, (4) middleware, (5) user interfaces, and (6) business rules and integrity constraints (Ziegler and Dittrich, 2004).

In terms of geospatial, (Budak et al., 2006) defines spatial heterogeneity as “non-stationary nature of most geographic processes, where global parameters do not reflect well the process occurring at a particular locality”. We expand on this definition to explain the problem being addressed in paper - spatial heterogeneity is the situation where the data sets which are supposed to be the same for a particular locality do vary as a result of differences in the geometrical make up of the primitives (point. arcs, and polygons) which make up features of the objects.

It should be noted that massive and heterogeneous databases and data collection approaches are not static and do not need to be combined at once (Makinnon, et al, 1999). This calls for dynamic techniques that can be used at any time in a more end user familiar format, which necessities an element of automation to make the whole process practical.

It should be dynamic so that technology-based approach will not constrain the ability to effectively meet the information requirements of users as their requirements evolve over time (Ma et al., 2000). This is the basis for our thinking of a point primitive based approach of integratino as it is the smallest geometrical object which makes it easier to alter – dynamic.

3 SPATIAL DATA GEOMETRICAL CORRECTION

Feature geometrical correction in GIS is needed to make any layers the same or to make features of thematically similar layers fit onto one another or to certain format or pattern. In this paper the focus is on geometrical objects similarity not the semantics (Budak et al., 2006) which handles the meaning of the data set by looking at the ontologies and semantics annotation of the data.

The conventional ways of carrying out geometrical correction is by taking layers as the spatial objects on which the geometry correction is done. This faces shorting as whole layers could be correct but the individual component (features) may need geometry adjustment to fit with other features. This calls for the need to refocus the elements of geometrical correction from layers to features. But before that can be done, there is need to identify which elements of features need to be adjusted and this can be achieved by breaking down the layer into features, then features into individual primitives (points, arcs, and polygons) and these into the smallest primitive – the points.

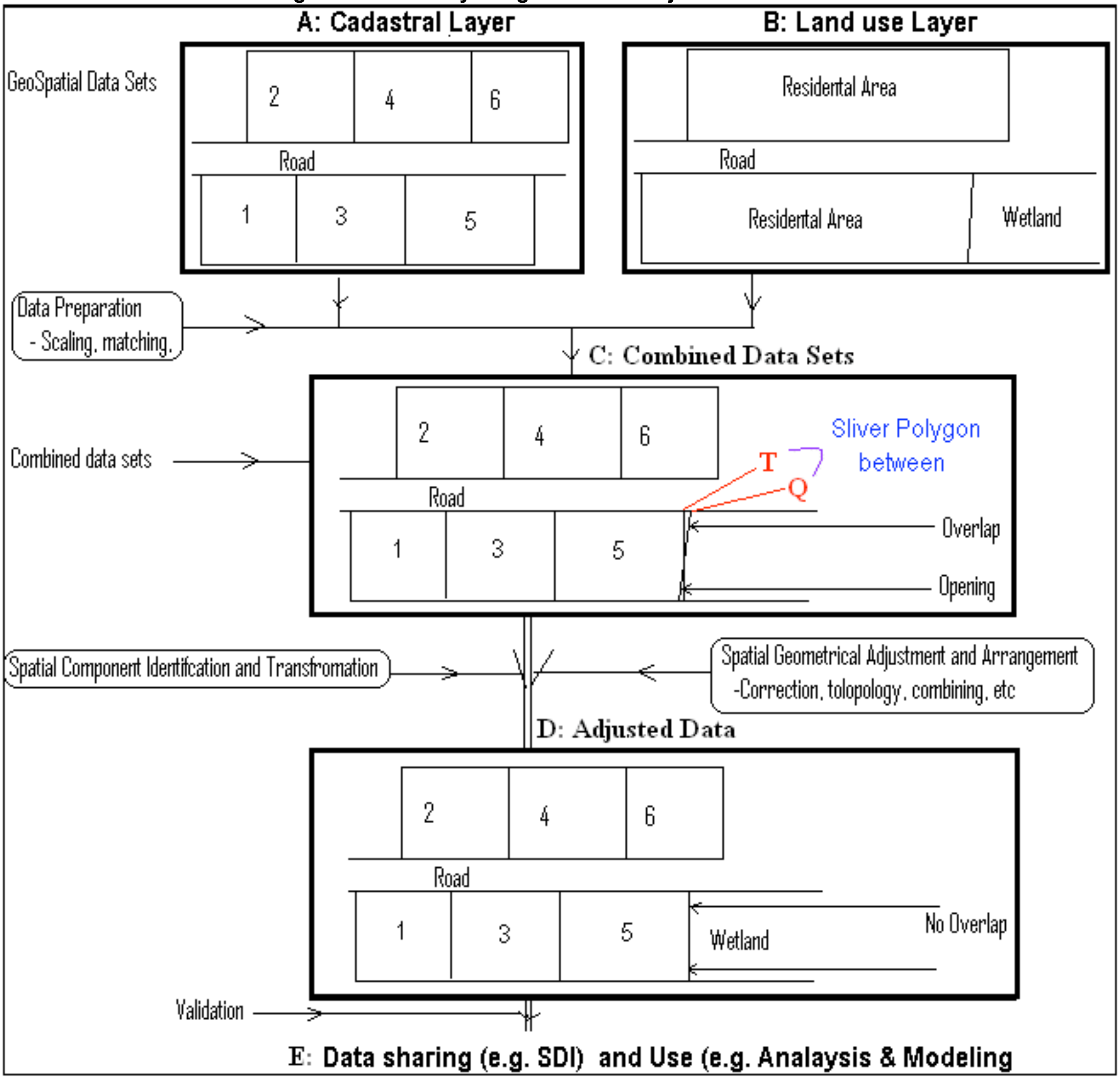
Geometrical integration incorporates schema integration and solving semantic conflicts of the datasets (Kampshoff, 2005). The interest here is the schema integration at the model level as the spatial feature geometry will be broken into primitives (points, arcs, and polygons) which can be done under different environment like GML. The promise we base our design on is that the primitives are expected to have same spatial properties for similar thematic datasets (datasets representing the same feature).

3.1 Nature and Disharmony in Spatial Data

In this paper, we consider only vector GIS data, where features can be disaggregated and transformed into primitives under GML. The assumption being that the integration of spatial features should be of thematically similar layers (layers containing data/information of same feature). For example roads should be integrated with roads not with rivers or any other feature unless they share a common boundary.

Differences in geospatial data are caused by many factors, for example it can be as a result of capturing thematically similar data independently by different individuals with no common framework being followed (Kampshoff, 2005). This leads to same points on different layers having different coordinates, arcs which are supposed to be the same ending up intersecting, and neighboring areas overlapping. These lead to geometrical and topological inconsistencies like creating sliver polygons, dangling arcs and nodes (see figure below).

Figure 1: Geometry Integration and Adjustment Problem



On the figure, we have two layers: - layer A: cadastral layer having plots 1, 2, 3, 4, 5, and 6 demarcated for construction of residential houses and layer B: land use layer which indicates the areas set aside for different land use activities (residential use and wetlands). When we carry out data preparations (including scaling and matching) on the two layers and we combine them by merging their geometries and observe the result on layer C (combined data sets). We notice that plot 5 has an overlap with the wetland and also there an opening which is neither plot 5 nor part of wetland. This is when we need to adjust the geometries of individual features/objects so that we eliminate the overlaps and openings which create sliver polygons (like polygon between P & Q), dangling arcs and nodes.

4 GEOGRAPHY MARKUP LANGUAGE (GML) AND GEOMETRY REPRESENTATION

Geography Markup Language (GML) has been defined as an XML encoding for geographic information (Sripada et al., 2004). GML has become popular as geospatial data are stored in multiple data models and schemas and data sources have differing query processing capabilities (Levy, 2005), (Kampshoff, 2005), (Friis-Christensen et al., 2005), (UCGIS, 2004), and (Kilpelainen, 1997) and GML's strength is in storing data in such a way that it is independent of the data model. Also in the model-mapping approach, a fixed database schema is used to store any GML documents without the assistance of GML schema or Document Type (DTD) (Lu et al., 2007).

In order to achieve that, GML provides a variety of kinds of objects for describing geography including features, coordinate reference systems, geometry, topology, time, units of measure, and generalized values. Thus, GML is based upon an abstract model of geography and describes the world in terms of geographic entities called features, which are represented as a list of properties and geometries. In GML, a feature is an abstraction of a real world phenomenon and since it is a geographic feature, then it is associated with a location on earth (OGC, 2007) and it has properties, some of which have values as geometry.

A feature can be composed of other features and the geometry of a geographic feature can also be composed of many geometry elements. Geometries are composed of basic geometry building blocks such as points, arcs, and polygons (Lake, 2000) and the properties of features are the name, type, and value description.

There are many GML concepts that can be adopted in representing geospatial objects (Doddapaneni and Trivedi, 2005) including hierarchical model and complex elements composed from simple elements and rich set of geometries; which can be user defined as well as standard GML defined. The integral part of any GML data model is the GML application schema (Doddapaneni and Trivedi, 2005), which describe the aspects of geo-spatial data requirements and the issues to put into consideration when accessing it.

GML is rich in geometry representation - its geometries are extensible, and provide additional flexibility in defining geometries that are domain relevant or user-relevant. GML documents nest spatial data types, permitting the effective representation of the various components of spatial data (Lu et al., 2007). GML application schema(s) define data types for properties of the feature and provides the information on features with their appropriate geometries and these data types include primitive predefined data type as well as user-defined types. There by paving a way for the GML software applications to represent the feature with its designated spatial construct (Doddapaneni and Trivedi, 2005).

GML is being employed as an environment to be used to break down geospatial data geometry into point primitives as geographic features can be accessed and presented in the form of GML features/objects (Chang and Park, 2006) which contain geometries as well as other properties in themselves. As a result we obtain GML data which is self-descriptive and serves as a mechanism for information discovery, retrieval, and exchange (Lu et al., 2007). We use GML to define geometric features, select geometric primitives of the features and identify spatial relationship between the features.

All these can be stored and represented/model geographic objects as GML feature instances and can be shared and transmitted as GML documents or messages and transported across the Internet (Lake, 2005). This approach is adopted since GML is a text-based language, provides an efficient means for archiving geospatial data, making it unlikely that future software will be incompatible with it (Lu et al., 2007). We are also encouraged by other researchers like (Chang and Park, 2006) when they say that “GML is very important for the components development and integration, as well as for interoperable geospatial data”. Also the representation and the modeling aspect are critical as we engage them in the GSIM.

As stated by (Lu et al., 2007), that GML is based on a GIS framework, it has the relationship maintenance as one of its aspect and it helps to build multidirectional associations among different spatial features and their properties using XLink and XML pointer language (Xpointer). With that, the data can be seamlessly integrated with other GML documents, providing different information about the same spatial data.

4.1 Why Point Geometry in GML

Many commercial packages and modeling approaches are good in handling objects, which have regular and pre-determined structure, but the discrepancies in these methods arise when it comes to objects that have multiples of these regular structures (Sagayaraj et al., 2006). As a matter of fact, most of geospatial objects are irregular due to being composed of different primitives. That is the main reasons we are looking at the simplest primitive – the point, which is not affected by these characteristics and limitations.

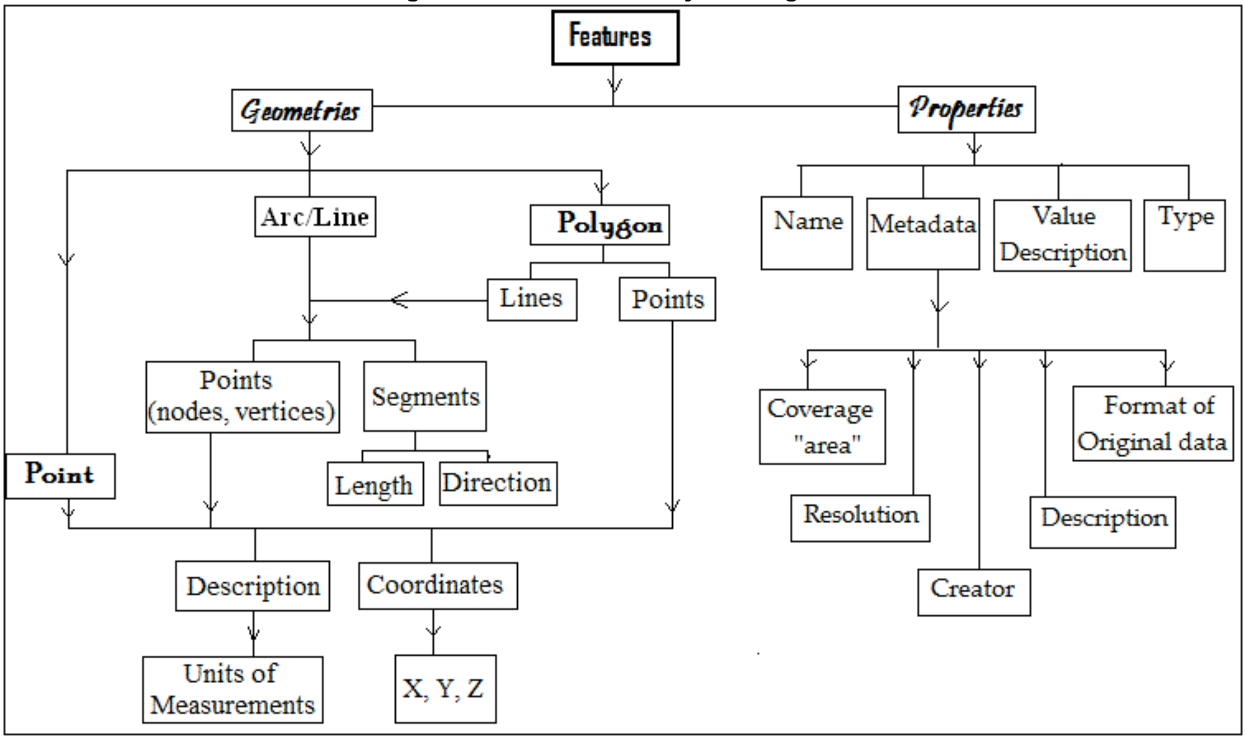
The point primitive is further supported by the fact that the geometric property of geographic features which define their position is a point. So it can be used to store other geometric properties in addition to their position. GML provides geometry features corresponding to geometry classes of Point, LineString, LinearRing, Polygon, MultiPoint, MultiLineString, MultiPolygon, and MultiGeometry (OGC, 2007); but the focus in this paper is the point.

In adopting our approach, we were encouraged by Sripada et al.(2004) when they stated that "GML is a comparatively new language in the field of geographic information systems and still in its developmental stage. Most of the data processing techniques need to be further developed in order for GML to be an efficient medium for geographic data storage and processing". We think our contribution is in that line and it will provide another alternative to the different initiatives and innovations taking place in GML and GIS data management.

5 GEOSPATIAL DATA GEOMETRY BREAKDOWN ONTOLOGY

The primary element and which is used for spatial adjustment in paper is a point. A spatial point is a concept used to define an exact location in space or place on a plane. In GIS, topology is a must in order for data to be used for analysis and modeling and to have topology, we need points since any form of space is considered to be made up of points as the basic elements. Thus, polylines (arcs) and polygons are converted into points before adjustment as they can be handled in GML using the point profile (Lake and Reed, 2005). Polygons, for example, are made of an array of arcs, line segment, and points. Each line segment contains an array of points (ESRI, 2004) since segment is the shortest distance between any two vertices (points). Therefore polygons are broken down into arcs and points; arcs/lines are disintegrated into points (see figure below).

Figure 2: Feature Geometry Disintegration

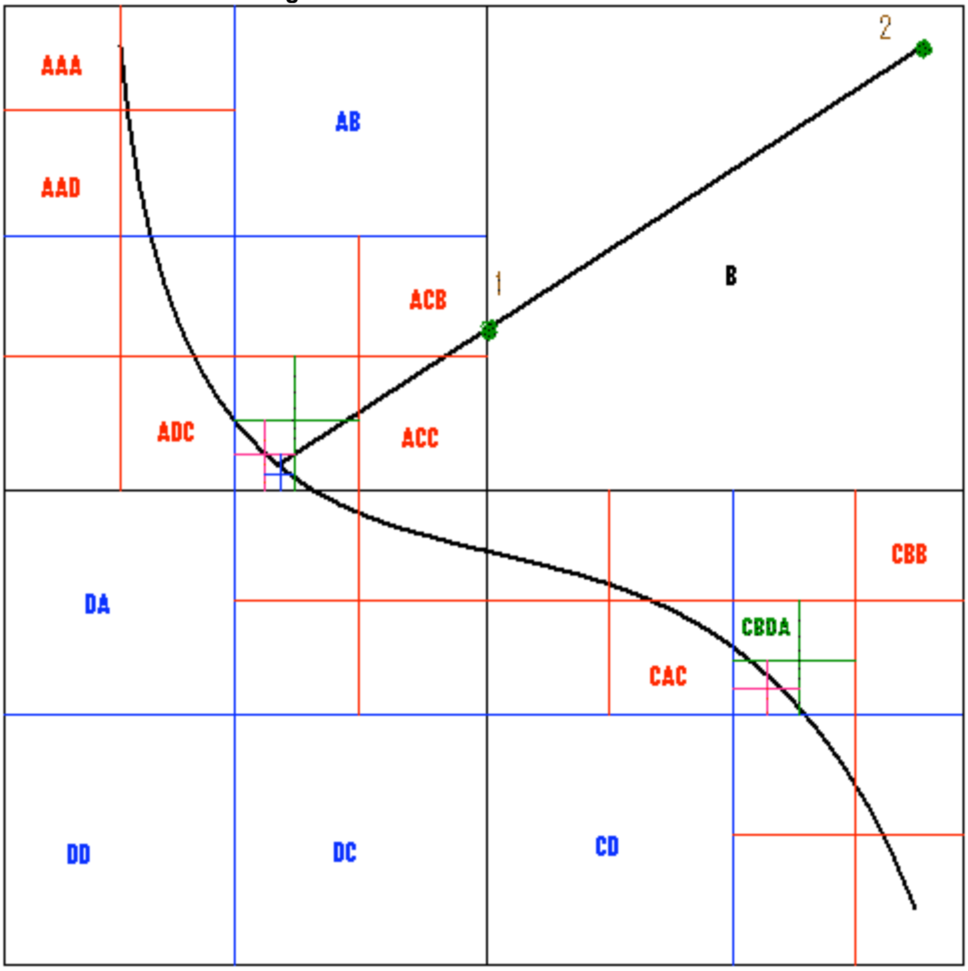


On the figure, we start from the feature which is composed of geometries and properties. Before the geometry adjustment, the properties (which contain name, metadata, value description, and type) are extracted and kept so that they can be re-attached to feature after adjustment. For the geometries, we identify the primitives (points, arcs, and polygons). The polygons are first broken down into points and lines, and then the lines/arcs are broken down into points and line segments. The line segments are broken into points and at this moment it is only the points existing as primitives. It is these points that are used for spatial geometrical adjustment.

For polygons to be adjusted, they are first broken down into points and arcs. This starts off by identifying the components of the polygon by selecting the straight arcs and converting/naming them as lines. For any arc which is not straight is broken down into smaller parts and checked to confirm if they are line segments. If not, the arcs are broken down further and the process continues until the polygons are line segments and points. Where line segments join and where they start and end, we get the points.

To transform an arc into points, each place on the line is considered be a point. So, we introduce points whenever there is change in direction or where we need to pick details. Putting that into consideration, let us take a look at transforming simple curve with a line attached to it into points (see figure below)

Figure 3: Arc to Points Transformation



With the arc (bold lines) in the figure above to be converted into points, we start by laying a temporary grid on the arc. The beginning grid should have the minimum number of squares (e.g. black in color on the figure) which can cover the whole layer. This is ensured by examining the squares individually in order to identify squares with uniform characteristics and these are marked according to that level of resolution (the size of the grid square). This is given labels say A, B, C, D to indicate that it is the first level. In our example we have square B which has only a line segment which can be represented by points 1 and segment 1-2.

At that level, we identify the grid squares which have features which need to be adjusted and this is done by looking at homogeneity of the feature and making decision whether to pick them or not.

The grid squares with non uniform feature at first level are further divided into smaller ones to obtain the grid squares at level 2 (AA, AB, BB, CD, etc). This process continues until all grid squares represent only one segment of arc (linear feature). Note that arcs are divided into smaller ones only if they do not form straight line segments and it is fine for points representing segments to fall on the edge of grid square.

With that, we embark to mark features in each grid square. The start and end of the feature are marked with points and the line segment joining them is noted. It is these two points which will be adjusted by giving them new coordinates or by editing their values in GML. We number points to make them easier to find. To indicate direction, we mark the starting point zero and east is positive and west negative.

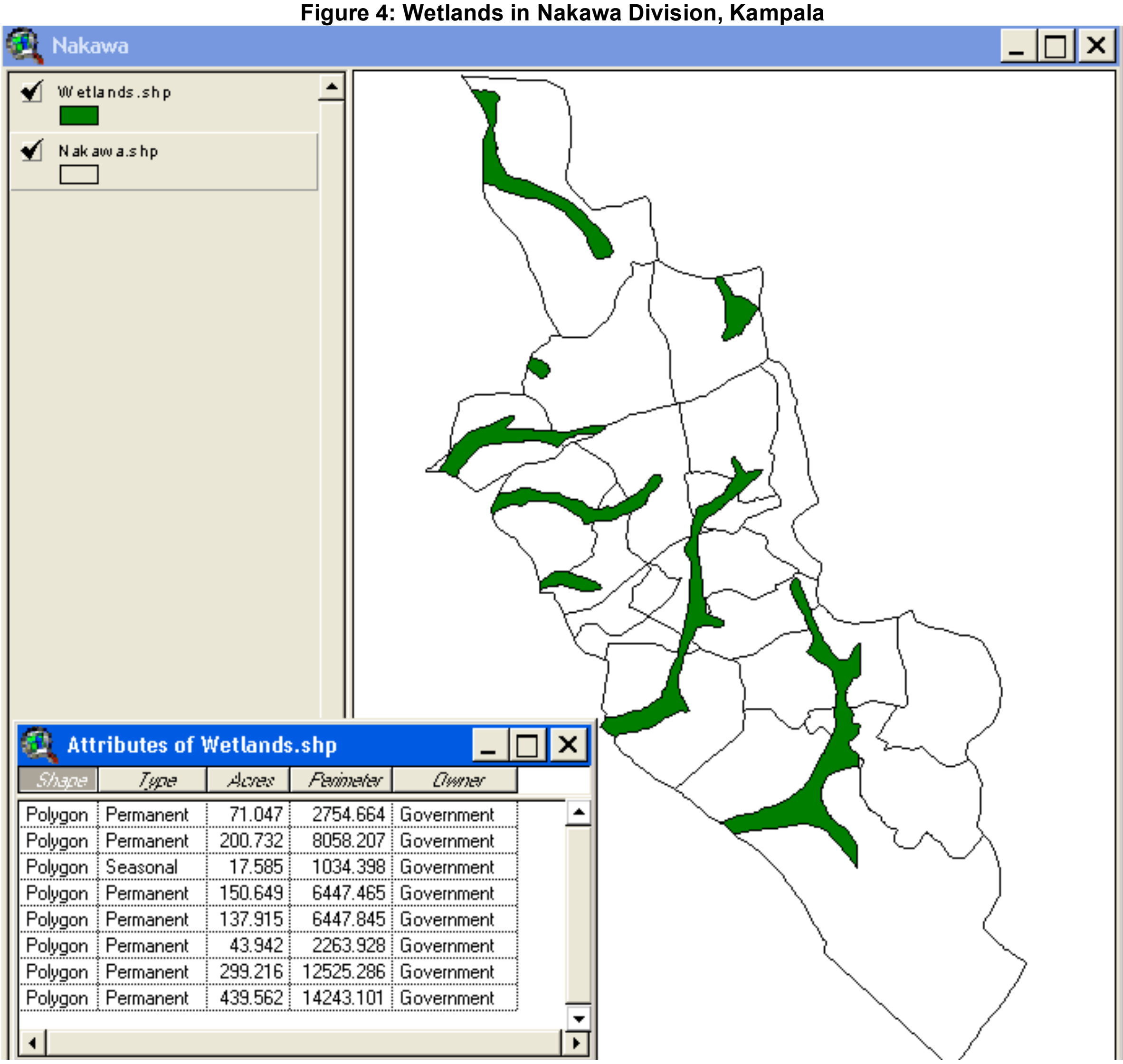
6 APPLICATION OF THE APPROACH

This section presents the application of the developed approach where a description of the data from Nakawa division in Kampala City (Uganda) which was used to test the approach is given, preprocessing which was done, the steps taken to demonstrate the approach, and subsequent results obtained after applying the technique.

6.1 Data Source

We used a simple spatial data set from Nakawa division in Kampala City (Uganda) to test our approach. We selected only one layer "wetland layer" see Figure 4 which shows the location of the different wetlands as polygons. The purpose was to break down the wetland polygons into polylines, then into points with can be stored in GML with the properties/attribute data stored in the associated table (see the inserted table "attributes of wetlands on Figure 4).

Figure 4: Wetlands in Nakawa Division, Kampala



Shape	Type	Acres	Perimeter	Owner
Polygon	Permanent	71.047	2754.664	Government
Polygon	Permanent	200.732	8058.207	Government
Polygon	Seasonal	17.585	1034.398	Government
Polygon	Permanent	150.649	6447.465	Government
Polygon	Permanent	137.915	6447.845	Government
Polygon	Permanent	43.942	2263.928	Government
Polygon	Permanent	299.216	12525.286	Government
Polygon	Permanent	439.562	14243.101	Government

The wetlands to be converted into points and stored in GML are the shaded (green) polygons on the figure.

6.2 Preprocessing of the Data

Before the data was disintegrated and later coded in GML, it had to be preprocessed and this involved looking the scale of the data, spatial extent. The resolution was not handled as the data used was vector data, although care was taken on the precision which was used during data capture.

We started off with spatial scaling, where we checked the wetland layer of our data to ensure that the different polygons occupy the same aerial extent i.e. they are within or equal to spatial extend of Nakawa division of Kampala City (Uganda). This had to be done as during the breakdown of the wetland polygon the assumption was that all data is within Nakawa division and same assumption was based on when separating the properties from the geometry. This was done by clipping off the wetland boundary layer using the Nakawa division layer; this left us with only wetlands in Nakawa.

The next step was spatial scaling; were we checked to make sure that the wetland polygons were at the same scale of 1:1000 which was being used. The ones found to be bigger or smaller we applied shrinkage or enlargement. This was important as the wetland layer polygons had to be at the same scale since they will be stored irrespective of the scale in GML.

Clipping and scaling were handled using ArcView GIS and ArcGIS which have all the needed tools to accomplish them. It is the geoprocessing function, it allows to cut off any part of feature.

6.3 Processing of the Data

We started off with spatial feature Component Identification. As the various components of geo-spatial data sets have to be handled differently during special adjustment, the role of this step is to identify the different primary component of the data set into individual geometry point, ploylines, and polygon.

Since different components of geo-spatial data sets have to be handled differently during spatial adjustment. The spatial objects/features have to be broken down into different primary components/individual primitive (point, arc, and polygon). We should note that although spatial data are represented as GIS objects such as rivers, buildings, roads, railway lines, etc using geometric primitives (points, arcs, and polygons); but the common primitive used is points to represent the object, store, retrieve, process, and reconstruct objects (Sagayaraj, et al, 2006).

To easy the processing of data, we used a free tool called JUMP Unified Mapping Platform, which has a GUI and API for performing spatial data processing and manipulation. It has both GML and Shapefile drivers under its API which provided a perfect environment to test our two out-comes both in GML and shape file format.

In order to use GML, an important aspect of GML was always to consider and remember that the structure of a GML document is very flexible which help in interoperability. GML consists of a series of features representing the real-world entities. These features are part of a family of a FeatureCollection and in return each feature is FeatureCollection enabling an entity to be represented by aggregations of other features.

In order not to loss the properties of wetland polygons, we took the strength of GML of being a data descriptive language, which means that the data is stored in a self-descriptive manner (Sripada et al., 2004) to distribution all the information say for points from polygons, the points have the information from the wetland. This was done by saving the properties of each wetland features into separate file. These properties were always referred to and later attached to points as explained below.

6.4 The Geospatial Disintegration Results

This sections presents the results of the applying the geometry disintegration approach on geospatial data from Nakawa division in Kampala City (Uganda).

JUMP aided to convert the shapefiles into GML; it from this point that we employ our approach to convert the polygon into the points. The was done using the strength of GML geometry properties as different shapes are defined. So to change from polygon to polylines to points, we change the geometry property from polygon to ploylines to points. That is for polygon (*<feature>* *<geometry>* *<gml:Polygon>* *<gml:outerBoundaryIs>* *<gml:LinearRing>* *<gml:coordinates>*) changed to polylines (feature> <geometry> <gml:LineString> <gml:coordinates>) or directly to points (*<feature>* *<geometry>* *<gml:Point>* *<gml:coordinates>).*

With that we can have any polygon broken down into points but it is not enough as there should be provision to keep the relationship between the points. So we made sure that the topology which is mostly affected by the geometry and sometimes limit the usability of the data set in future is handled well by presenting the data basing on GML features (Chang and Park 2006), which contain geometries as well as other properties in themselves. The wetland polygons which we picked were considered for containment, connectivity, and adjacency which are explained as follows:-

Since each polygon had properties/attributes, these were given to each point. This helps to identify points belonging to same polygon and this information was used when reconstructing the polygon.

The result is text-based file which can be archived. The GML document obtained were larger than the shape files used, this is due to their descriptive nature of GML (Sripada et al., 2004).

Although GML files are bigger than shapefiles, but they can be compressed with a reduction up to 40-50% compared with shapefile compression which achieves 10-25% depending on the file size. With the data in text format, it makes it unlikely that future software will be incompatible with it, it can be edited by any text editor including notepad, all data is stored as one file, easy to change between the different geometry shapes, easy to add attributes to the geometry by just typing in the text, no need to store x and y coordinates in separate attribute tables as they are directly stored.

In summary, wetland polygons were a combination of lines and points and the lines were combined of points. It is that basis, that this paper used to breakdown wetland polygons from Nakawa division and lines into points so that it is only one element covered (point primitive) when it come spatial data geometry adjustment.

7 FUTURE WORK

We are looking forward to using the above results to determine quantitative geometrical differences in thematically similar spatial data sets, provide foundation for developing spatial feature primitive based integration ontologies, and implementing geometrical adjustment model which will encapsulate all the above.

8 CONCLUSION

OpenGIS developed GML to help in the interchange of geospatial data across diverse systems. In order to share such data; it requires having efficient and effective ways to exchange it between the different formats. Within this paper, we have showed how to disintegration a spatial feature into primitives (points, arcs, and polygons) and how these can finally be broken down into points and handled in GML. This paves way and provides an environment where varying geometries of thematically similar data can be adjustment. By outlining steps through which geospatial users have to go through to quickly have their data converted into points which can be handled in GML helps to take advantage of GML. In so doing making more data available to other geo-practitioners who might need it.

REFERENCE:

BUDAK, A., SHETH, A., RAMAKRISHNAN, C., LYNN USERY, E., AZAMI, M. & KWAN, M. P. (2006) Geospatial Ontology Development and Semantic Analytics. *Transactions in GIS,* 10**,** 551-575.

CHANG, Y.-S. & PARK, H.-D. (2006) XML Web Service-based development model for Internet GIS applications. *International Journal of Geographical Information Science,* 20**,** 371-399.

DODDAPANENI, K. K. & TRIVEDI, S. (2005) Insight into GML. Secunderabad - 500 003; Andhra Pradesh; India, Intergraph Consulting

Pvt. Ltd (Subsidiary Intergraph Corporation).

ESRI (2004) *XML Schema of the Geodatabase - An ESRI Technical Paper,* ESRI 380 New York St., Redlands, CA 92373-8100, USA www.esri.com, ESRI.

FGDC (2007) The Federal Geographic Data Committee.

FRIIS-CHRISTENSEN, A., NYTUN, J. P., JENSEN, C. S. & SKOGAN, D. (2005) A Conceptual Schema Language for the Management of Multiple Representations of Geographic Entities. *Transactions in GIS,* 9**,** 345-380.

GSDI (2005) Spatial Data Infrastructure. Global Spatial Data Infrastructure Association.

KAMPSHOFF, S. (2005) Mathematical Models for Geometrical Integration. *First International Workshop on Next Generation 3D City Models.* Bonn, Germany, Geodetic Institute, RWTH Aachen Univeristy, Germany.

KILPELAINEN, T. (1997) Multiple Representation and Generalization of Geo-databases for Topographic Maps. Finnish Geodetic Institute.

LAKE, R. (2000) Introduction to GML Geography Markup Language. World Wide Web Consortium (W3C).

LAKE, R. (2005) The application of geography markup language (GML) to the geological sciences. *Computers and Geosciences,* 31**,** 1081-1094.

LAKE, R. & REED, C. (2005) GML Point Profile by Open Geospatial Consortium.

LEVY, A. (2005) Logic-Based Techniques in data Integration. Department of Computer Science and Engineering, University of Washington.

LU, C.-T., SANTOS, R. D., SRIPADA, L. & KOU, Y. (2007) Advances in GML for Geospatial Applications. *Geoinformatica,* 11**,** 131–157.

MA, C., CHOU, D. C. & YEN, D. C. (2000) Data warehousing, technology assessment and management. *Industrial Management and Data Systems,* 100**,** 125-134.

MALHOTRA, Y. (2000) Knowledge Management for [E-] Business Performance. Information Strategy. *The Executives Journal,* 16**,** 5-16.

MASUYAMA, A. (2006) Methods for detecting apparent differences between spatial tessellations at different time points. *International Journal of Geographical Information Science,* 20**,** 633-648.

OGC (2007) Geography Markup Language by Open Geospatial Consortium inc.

SAGAYARAJ, F., P.THAMBIDURAI, B.S.BHARADWAJ, G.N.BALAGEI & N.HEMANT (2006) An Improved and Efficient Storage Technique for GIS Geometric Primitives Based on Minimum Bounding Rectangles. *9th Annual International Conference, Map India 2006.* India.

SRIPADA, L. N., LU, C. T. & WU, W. (2004) Evaluating GML support for spatial databases. *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International Computer Software and Applications Conference,* 2.

UCGIS (2004) University Consortium for Geographic Information Science 2002 Research Agenda. http://www.ucgis.org/priorities/research/2002researchagenda.htm.

ZHANG, C., PENG, Z.-R., LI, W. & DAY, M. J. (2003) GML-Based Interoperable Geographical Databases. *Cartography,* 32**,** 1-16.

ZIEGLER, P. & DITTRICH, K. R. (2004) Three decades of data integration - all problems solved? *Database Technology Research Group.* Winterthurerstrasse 190, CH-8057 Zurich, Switzerland, Department of Informatics, university of Zurich.