IOSR Journal of Mobile Computing & Application (IOSR-JMCA)
e-ISSN: 2394-0050, P-ISSN: 2394-0042.Volume 3, Issue 4 (Jul. - Aug. 2016), PP 40-46
www.iosrjournals.org

The Role of Object-Oriented Programming (OOP) in Modeling of Geographic Information Systems (GIS)

Onu F. U[1], Uche-Nwachi E. O[2], Chigbundu K. E[3]
[1]*Computer Science Department, Ebonyi State University, Abakaliki – Nigeria*
[2]*Computer Science Department, Ebonyi State University, Abakaliki – Nigeria*
[3]*Computer Science Department, Michael Okpara University of Agriculture, Umudike – Nigeria*

Abstract: *The popularity of Geographic Information System (GIS) has grown tremendously such that software developers can no longer neglect that sector when fashioning out new technologies for software development. The concept of Object-Oriented Programming (OOP) has revolutionalised software development industry including the GIS and software modeling in general. This paper reveals how the concepts inherent in Object-Oriented Programming Languages (OOPLs) can be deployed to model a GIS software. Using materials from secondary sources and adopting a case study method, the paper unraveled the desirable features of GIS software with a view to determining the relevance of OOP in GIS modeling. Furthermore, current GIS Modeling software features were critically studied and were found to imbibe several OOP techniques and concepts. A GIS software, ArcGIS was used as a case study to reveal that GIS software models have close affinity with OOP implementation languages like C++, Java, C#. The paper concluded that OOP concepts indeed were very relevant in GIS modeling, even though some GIS related problems still needed to be fully tackled by existing software.*

Keywords: *OOP, GIS, Modeling, Inheritance, ArcGIS*

I. Introduction

Geographic Information Systems (GIS) are special class of information systems that keep track not only of events, activities and things but also where those events, activities, or things happened. Almost anything can happen anywhere and sometimes it becomes critically important to know where those things occurred. Things like position of a country's boundaries, location of hospitals or fire stations, etc are usually very important to human welfare.. GIS is a collection of computer hardware, software, geo-data, etc, for capturing, managing, analyzing, and displaying all forms of geographically referenced information. Of course, everything involves the people who will manage the system, not just the computers and procedures [1]. Figure 1 shows the interconnection of the components of a GIS in general terms.

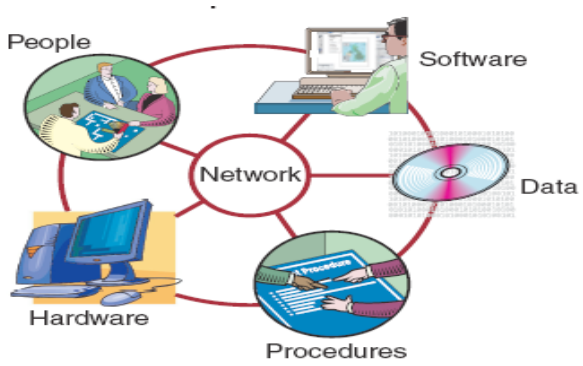

Figure. 1: Six parts of a GIS [1]

GIS can be described as a process for input, storage and retrieval, analysis, and output of geographical information [2]. GIS software has become the most widely used software by social scientists in the past two decades. It has become a worldwide application that can integrate and relate any data with a spatial component in order to support the decision making criteria in many organizations. GIS has made decision making easier, faster, and usually more accurate. Areas where GIS find application include zoning and land use planning, crime analysis and prevention, traffic analysis, natural resource management and environmental assessment, demographic analysis, emergency management planning and disaster recovery, public health and healthcare access, parks and open space planning, and transport and public transit planning GIS as a software is developed by software developers using different concepts and techniques. OOP is one of the most popular techniques of

software development. The general approach of Object-Oriented Programming is to view a software system as a collection of interacting entities called *"object"* each of which is defined by an *identity*, a *state* described in terms of *member variables*, and a *behaviour* described in terms of *methods* that can be invoked[16]. Object-Oriented Programming (OOP) refers to a programming methodology based on objects, instead of just functions and procedures. These objects are organized into classes, which allow individual objects to be grouped together. Most modern programming languages like Java, C/C++, and PHP, are object-oriented languages, and many older programming languages now have object-oriented versions [12]. Object-oriented concept has traversed every area of computer technology. The gains of Object-Oriented technology are not only being appreciated by software developers. Experts in the Information Technology (IT) fields are now deeply applying the technology. That is the reason we popularly hear about Object-Oriented Database Management System (OODMS), Object-Oriented Programming Language (OOPL), Object-Oriented System Design (OOSD), Object-Oriented System Analysis (OOSA), Object-Oriented Geographic Information System (OOGIS) and so on.

Object-oriented technology has been incorporated into Geographic Information System to make real complex objects easier to model. The GIS data models could be one of three forms: vector data models represent geography as collections of points, lines, polygons; raster models represent same as cell matrices that store numeric values; while the Triangulated Irregular Network (TIN) model represents geography as sets of contiguous three dimensional coordinates (x, y, z) arranged in a network of non overlapping triangles [25]. OOP allows for immediate implementations of object concepts rather than simulating them with traditional languages [3][4]. Many Systems like Ietergraph's TIGRIS, Smallworld GIS, GeoStar, and Laser Scan's GAE have been designed and implemented using the Object-Oriented approached [5].

Research Objectives

The overall objective of this research work is to reveal the relevance of Object-Oriented Programming concepts and techniques in modeling of Geographic Information System (GIS). The specific objectives are as follows:

- To review features of object oriented programming languages
- To show the place of these OOP in GIS modeling.
- To show that current GIS modeling software are also OOP based

II. Literature Review

In 1960, the world's first true use of operational GIS was developed in Ottawa, Ontario, Canada by the Federal Department of Forestry and Rural development by Dr. Roger Tomlinson. It was called Canada Geographic Information System (CGIS) and was used to store, analyze, and manipulate data collected for the Canada Land Inventory. Tomlison is also recognized as the "Father of GIS" because he was able to use overlays to promote the spatial analysis of convergent geographic data [6]. Majority of the early researches were carried out on data structures, algorithms, and indexing schemes, and, consequently, had strong links to emerging agendas in computer science. GIS was then faced with several problems such as converting the content of a paper map to digital form, storing results on magnetic tapes, and how to compute the area of patches. But these problems were solved when the research agenda of GIS expanded to issues of data quality and certainty, the cognitive principles of user interface design, the cost and benefits of GIS, and social impacts of the technology.

Before computers, GIS involved tedious manual map analysis procedures. A transparency, for instance, was taped over a contour map of elevation and areas where the contour lines were closely spaced (steep) were outlined and filled with a dark color, all in an effort to identify areas of high or low susceptibility to certain elements.

GIS has grown to become a very significant application of computing. In 1995, the initial notion of GIS as an assistant performing tasks that the user found difficult, complex, tedious, or expensive to do by hand, was replaced by one in which GIS became the means by which humans communicate what they know about the Earth's surface, with which they make collective decision about land management, and by which they explore the effects of alternative plans [7]. That means GIS is both a database system with specific capabilities for spatially-referenced data, as well as a set of operations for working with the data [8]. There are generally two types of GIS Models: cartographic models and spatial models. Cartographic Models are applied in the automation of manual techniques which use drafting aids and transparent overlays, such as a map identifying locations of productive soils and gentle slopes. Spatial Models use expressions of mathematical relationships among mapped variables, expressed as parameters and relationships.

There are a number of GIS software available today ranging from high powered analytical software to visual web applications [1]. Three groups of such software can be identified:

- **Desktop GIS**: Allows work with spatial data on a personal computer. e.g. ArcGIS
- **Geobrowser**: An explorer for geographic information that allows combination of different types of geographic data from different sources. e.g. Google Earth.

- **Web-based GIS (WebGIS):** These are online GIS applications which may not be as fully functional as software stored on a PC.

The relational model upon which early GIS software systems were built has been acknowledged as an insufficient model for applications that deal with spatial data .The application of object-oriented techniques for the design of future Geographic Information Systems has been proposed on several stages such as message-passing programming language, object-oriented database management systems, and object-oriented software engineering techniques[9]. A non-exhaustive description of object oriented concepts exposed by OOP languages includes the following.

- **Class:** This is a blueprint or prototype that defines the variables and methods common to all objects of a certain kind. Once a class is defined, any number of objects can be created that belong to the class.
- **Object**: The basic entity of an OOP system. It is an instance of a class. During program execution, objects interact with each other without knowing the details of their data or code.
- **Instance:** This is the actual object created at runtime. One can have an instance of a class or that of an object.
- **Classification:** This is the mapping of several objects (instances) onto a common class. All objects that belong to the same class are described by the same properties and have the same operations. For every object in the object-oriented approach there exists at least one corresponding class; classification is often referred to as the *instance_ of* relationship [13].
- **Generalization:** The process of grouping several classes of objects which have some properties and operations to a common general *superclass* is called generalization. The term *subclass* and *superclass* characterize generalization and refer to object types which are related by an *is_a* relation, since the object of the subclass is also an instance of a *superclass* [13].
- **Association:** Association is a relationship between similar objects that can be grouped together to form a higher level set object [14]. The term *set* is used to describe the association, and each associated object in the set is called a *member* [20].
- **Aggregation:** This is the process where several objects can be grouped together to form a composite object, creating a higher-level object which may itself be referred to [15]. The terms *subpart* or *component* refers to parts of the composite object. Operations of aggregates are not compatible with operations on parts, and vice-versa. Details of the constituent objects are suppressed when considering the aggregate object. Every instance of an aggregate object can be decomposed into the instances of the corresponding component objects [13].
- **Object Identity:** Object Identity differentiates one object from the other. Object name is used to identify the object. Hence the object name itself is an identity.
- **Encapsulation:** This has often been described as information hiding. It is the process of making the detailed implementation of an object hidden from its user. Objects generally do not expose their internal data members to the outside world. Object-oriented programming encapsulates data (attributes) and methods (behaviours) into objects [16].
- **Inheritance:** Inheritance is a major-type/sub-type relationship. Inheritance is the mechanism by which a type inherits properties (data structure and methods) of its super type. A class inherits from another class when it receives some or all the qualities of that class. The starting class is known as the base, super, parent, or generalized class, while the inheriting class is called the derived, sub, child, or specialized class [18].
- **Polymorphism:** It is derived from poly (meaning many) and morph (meaning form). Therefore it means *"many forms"*. Polymorphism represents the state of an object in which it can assume different forms. It is implemented by inheriting some of the functions from the parent classes and overriding or modifying part of them. One of the benefits of polymorphism is that it offers great flexibility in class derivation [18].

In the early 1990s, based on the object-oriented paradigm, a number of modeling languages proposed that geographic phenomena were modeled as specialized subclasses of a set of predefined classes representing geometric objects. Consequently, the class Street would be modeled as a subclass of the class Line, a class Pole would be modeled as a subclass of the class Point, etc. Laser-Scan developed the Gothic geospatial object-oriented processing environment in late 1980s. Gothic represented real world objects by object models in classes, according to the schema. Instances of objects had behaviours, attributes, identity, and geometry [7].

With respect to OOP, Simula 67, the first programming language to use objects, was developed in mid-1960's by Ole-Johan Dahl and Kristem Nygard at the Norwegian Computing Center Oslo for creating simulations. Ole-Johas and Kirstem were working on simulation that deals with exploding ships, and realized they could group the ship into different categories. Each ship type would have its own class, and the class would generate its unique data and behavior. Simula 67 was not only responsible for introducing the concept of class, but it also introduced the instance of a class [10].

In the early 1970s, Alan Kay and his team at Xerox Parc Alto Research Center developed the first computer called the Dynabook. Smalltalk was the object-oriented language developed for programming the Dynabook. Smalltalk is still in existence today, though it is not commercially viable. The Smalltalk teams was inspired by the Simula 67 project; Smalltalk was designed to be dynamic rather than static, that is the Smalltalk objects could be changed or deleted. Smalltalk was also the first programming language to introduce inheritance concept [11].

Simula 67 was a groundbreaking system that has inspired a large number of other programming languages. By 1980s object-oriented programming had become prominent, and the primary factor in this was C++ programming language. Byarne Stroustrup from Bell Laboratory in the mid 1980's created C++. Until most recently C++ was the language of choice of an average programmer [10]. Current GIS software systems built around these object-oriented programs make use of Object Oriented Technologies which employ the following OOP techniques.

- **Object Oriented User Interfaces (OOUI):** The use of icons in place of command lines/menu selections to launch task. We drag, drop, slide, toggle, push, click, and point and click rather than type and retype. Also use of glyphs as graphical objects controlled by programs to perform specific functions
- **Object Oriented Programming Systems (OOPS)**: Uses widgets to develop computer code. e.g. drop a command button on a design surface, series of dialog boxes specify action and code (or a macro) is written behind command button for desired function.
- **Object Oriented DataBase Management (OODBM):** Provides database procedures for establishing and relating spatial objects or data objects, couples data and the processing procedures that act on the single data object through encapsulation, polymorphism, structure and inheritance. A single query defines the where (locational attribute), what (thematic attribute), spatial context and linkages [23].

III. Discussion

The earlier discussed concepts that form the core of Object-Oriented Programming Languages (OOPLs) can be deployed to model GIS software. Relatively common concepts such as classes, objects, instances of objects and inheritance, plus others that are less known such as classification, generalization, association, and aggregation find direct relevance and equivalence in many desirable features of GIS software models. The role of OOP in these models is emphasized in the next section.

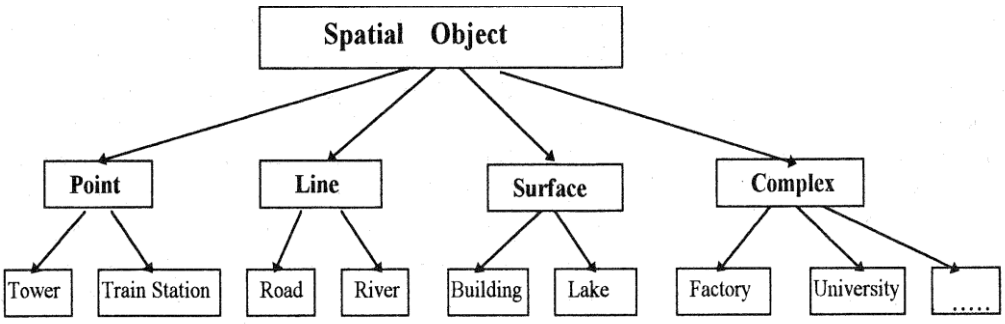

Figure 2: Feature classification [19]

Place of OOP in GIS modeling

In geometry, there are four highest super classes (POINT, LINE, SURFACE, and COMPLEX) abstracted from all features of GIS. We can identify *buildings, road, river, land, parks, lakes, electric polls, trees, railroads* from maps or images. Each of the identified features can be classified into feature classes. Furthermore, buildings can be specialized into *post office, school, town hall and hotel*. etc. Some of the feature classes with the same geometric properties can be combined. For example *buildings, parks, lake, land* can be combined to superclass called *surface.* Road, river, railroads can be combined to a superclass called *line.* Train station and *electric polls* and *trees* can be combined to a superclass called *points.*

Several features might be aggregated to a complex objects, for instance, some buildings and other features can be grouped to a factory. Each spatial features class is declared as belonging to a superclass of four geometry classes. For instance, the class BUILDING has attributes like building number, owner, date built, street etc. Specialized subclasses of a feature class like HOTEL can have additional attributes and still inherit the common attributes from the superclass (parent) class (BUILDING). For example, HOTEL class might have manger, staff, tax, bed number, and room number as its attribute and still inherit attributes like owner, built date, and building number from the superclass BUILDING.

Real world GIS objects can easily be modeled using the concepts of Object-oriented programming such as object classification, generalization, association, aggregation, object identity, encapsulation, inheritance, and polymorphism. Samples of these concepts are demonstrated below

- **Object classification**: Classification can be described as mapping several objects to a common class. The GIS model of a School may include the classes BUILDING, TEACHER, and STUDENT. A single instance, such as the building with building Id number "*Junior Secondary 1A,"* is an object of the corresponding object type, that is, the particular object is an instance of the class BUILDING. Operations and properties are assigned to object types, for example, the class BUILDING may have the property *YearBuilt*, which is specific for all buildings. Likewise, the class TEACHER may have an operation to determine all *TotalNumberOfCourseTaught*, and STUDENT may have an operation to determine all Students all *Student'sAverageScore*.

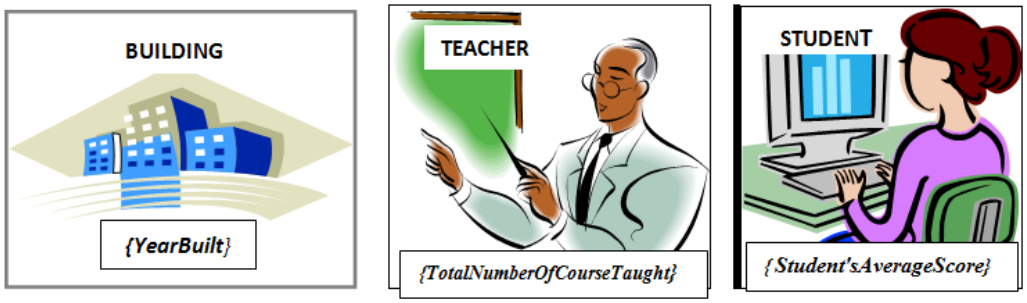

Figure 3: Graphical representation for *classes* BUILDING, TEACHER, and STUDENT.

The property values differentiate between object of the same class. Property values describe the individual characteristics of each object. For instance, two STUDENTSCORES may be distinguished by their studentId, and studentName. Geographic relationships between object classes are based on geographic operators (such as overlap, adjacency, inside, and touching) that determine the interaction between objects.

- **Generalization:** Generalization groups classes of objects with some operations in common into a *superclass. Superclass* BUILDING can have subclass CLASSROOM, LIBRARY, and LABORATORY. *Superclass* and *subclass* are abstractions for the same object, and do not describe two different objects. The classroom *"junior secondary 1A,"* for example, is simultaneously an instance of the classes CLASSROOM and BUILDING.

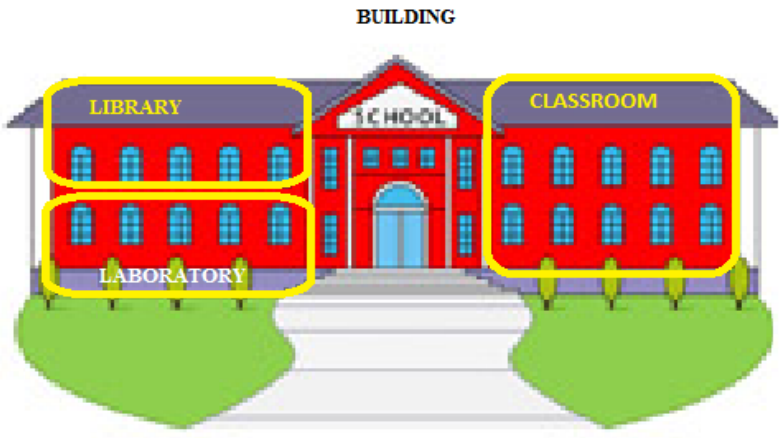

Figure 4: A generalization with multiple subclasses of a superclass.

- **Association:** Association is a form of abstraction that relates two or more similar independent objects, and the relationship between objects can be grouped to form a higher level (*group*) object. An example of association in GIS domain is neighborhood, which links parcel of land with its adjacent house lots. The detail of a *member* object is suppressed while the properties of the *group* object are emphasized. The NEIGHBORHOOD and the ROAD NETWORK are associated by the relationship *inside.* In figure 4 below, road network is *inside* neighborhood.

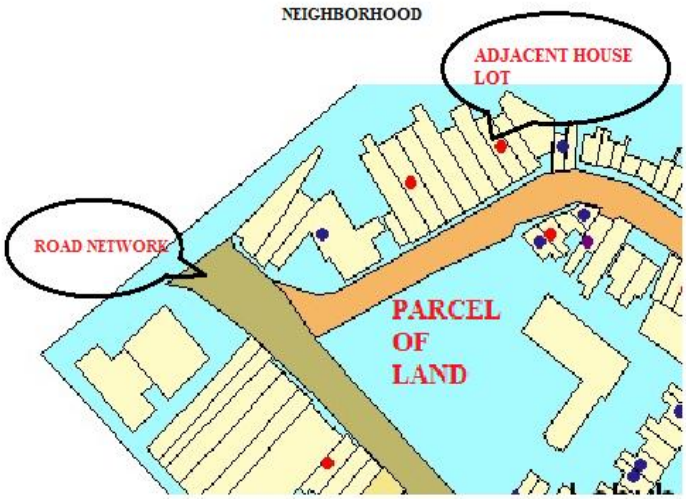

Figure 5: Neighborhood with Road Network, Adjacent House lot, and Parcel of Land [27].

Aggregation: It involves grouping together of several objects to form a high level (composite) object. The relation established by aggregation is often called the *part-of* relation, while the relationship converse to *part_of* is *consists-of*. In the modeling of a CITY, HOUSELOTS, ROADS, and PARCEL OF LANDS can be *part_of* the CITY, conversely, CITY *consists_of* HOUSE LOTS, ROADS, and PARCEL OF LANDS. Figure 3 above can also be used to illustrate the concept of aggregation; where the HOUSELOTS, ROADS, and PARCEL OF LANDS are part of the CITY. Spatial aggregation in GIS feature is an example of aggregation.

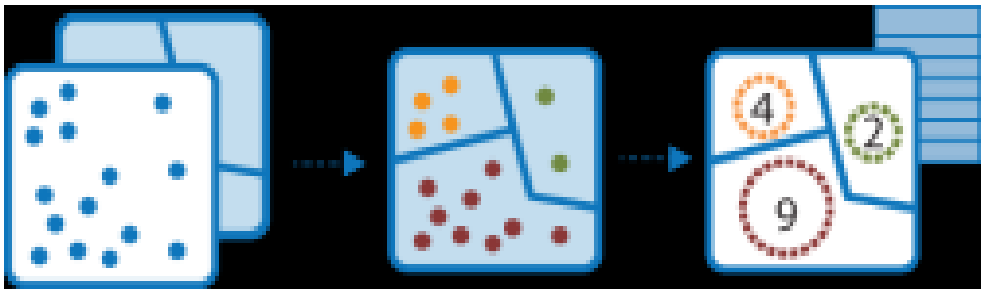

Figure 6: Graphical representation for GIS spatial aggregation [26]

Inheritance: RIVERS, SOIL, HOUSEOWNER, BUILDINGS, CITIES, STORES and, PARCEL are some of many application-specific classes of object in GIS. In object-oriented classes, some operations are associated with each class likewise some operations are associated with classes of objects in GIS. Examples of the operations that could be associated with some of the classes are: new road could be constructed, new building may be constructed, house could be sold, and hence change of house ownership will change. These operations are similar to the GIS operations like the operation to determine all CITIES that lie within a COUNTY and the operation determining all BUILDINGS within a CITY can be both interpreted as geometric operation inside [21].

Inheritance is ideal for modeling such structure in a GIS, and formalizing the structure and properties of the object class. A general superclass for each concept can be identified, and the common properties defined for each of the general superclass and inherited by the subclasses of GIS applications. For instance, a superclass BUILDING could have properties like, location, ownership, and spatial relationships (intersect, neighborhood, includes). A class (HOTEL) in the user module can be defined as a subclass of the superclass BUILDING inheriting all the spatial attributes of the superclass.

IV. Arcgis 9: Desktop GIS Software

The software application ArcGIS 9, developed by Environmental Systems Research Institute (ESRI), is one of the most popular desktop GIS software. You can follow this link to download a trial copy of the software: http://www.esri.com/software/arcgis/arcview/eval/evaluate.html.

It is a family of GIS Software that include the ArcGIS Desktop, ArcGIS Engine, ArcGIS Server, ArcIMS (Internet Map Server), ArcSDE (Spatial Data Engine), etc. The ArcGIS Engine standard functionality

include map creation, map interaction, map analysis, data creation, developer controls and developer technologies. It is built around OOP components called ArcObjects that enable a user to build custom applications and implement enterprise GIS applications. ArcObjects are implemented with C++, but the ArcGIS Developer Kit provides four development Application Programming Interface (API) for the user: Component Object Model (using VB6 sp3, Visual Studio .NET 2003), .NET (with C# or VB.NET), Java (within Eclipse, JBuilder or NetBeans 3.6) and C++ Compilers (from Visual C++ sp3 for Windows)[24].

Every one of these development environments, and the programming languages used (C++, C#, VB .NET and Java) are OOP based, lending further credence to the importance of OOP in GIS modeling, although GIS areas such as exploratory spatial data analysis (EDSA), dynamic system simulation, operations research optimization, spatial statistics and visualization of multidimensional data[24] still need further research for better operations.

V. **Conclusion**

This research introduced us to the concept of Geographic Information Systems (GIS) that keep track not only of events, activities and things but also where these events or things happen. The paper showed how OOP concepts can be used to model GIS Software. It also emphasized the use of OOP in designing the features of GIS Software for modeling. The research critically examined current GIS Modeling Software features and discovered they included several OOP technologies in the user interface, design tools and coding techniques; they made use of icons, glyphs, and widgets in their operations.

The paper also discussed a popular desktop GIS software called ArcGIS 9 that is not only written with C++, one of the strong OOP languages, but provides different APIs for developers, all based on OOP principles. It was however discovered that, irrespective of the important role of OOP in GIS modeling, current GIS software needed further development in a few other areas of interest in GIS.

References

[1]. Paul A. Longley, Michael F. Goodchild, David J. Maguire, David W. Rhind (2005). Geographic Information Systems and Science.2nd Edition. John Wiley & Sons.
[2]. Calkins H. W. & Roger F Tomlinson (1977). Geographic information systems: methods and equipment for land use planning. International Geographic Union Commission on Geographical Data Sensing and Processing. Resource and Land Investigations (RALI) Program, Virginia, 1977
[3]. Stroustrup, B. (1986). The C ++ Programming Language. Addison-Wesley Publishing Company.
[4]. Meyer, B (1988). Object-Oriented Software Construction. New York, Prentice Hall
[5]. Worboys, F. Michael (1994). Object-oriented approaches to Geo-referenced information, IJGIS, Vol.8, No.4, pp. 385-399.
[6]. Wikipedia (2003). Geographic Information System. Accessed July 10, 2016. https://en.wikipedia.org/wiki/Geographic_ information_system
[7]. Shekhar S, & Xiong, H. (2008). Encyclopedia of GIS. SpringerScience+Buisiness Media, LLC.
[8]. Pick, J. B (2005). Geographic Information Systems in Busness. Ideas Group Publishing
[9]. Sasso, D & Biles, W.E (2008). An Object-Oriented Programming Approach For A GIS Data-Driven Simulation Model Of Traffic On An Inland Waterway. Proceedings of the 2008 Winter Simulation Conference
[10]. Rentsch, T (1982). Mathematics Education: Taking a Clue From the Recent Technological Revolution -A Very Brief History of Object-Oriented Computing. Accessed July 10, 2016. http://www.cut-the-knot.org/Mset99/OOhistory.shtml
[11]. Exforsys (2006). The History of Object Oriented Programming. Accessed July 10, 2016http://www.exforsys.com/tutorials/oops-concepts/the-history-of-object-oriented-programming.html
[12]. TechTerm (2007). OOP Definition. Accessed July 11, 2016 http://techterms.com/definition/oop .
[13]. Egenhofer, Max. J & Frank, Andrew. U (1992). Object-Oriented Modeling for GIS. URISA journal 4 (2): 3-19
[14]. Brodie, M.L (1984b). On the Development of Data Models. Springer Verlag, New York, NY
[15]. Kwan, Mei-Po, Golledge, Reginald, G. & Speigle, Jon M (1996). A Review of Object-Oriented Approaches in Geographic Information Systems for Transportation Modeling The University of California Transportation Center University of Californiat Berkeley
[16]. A. Bellaachia. (2016). Object-oriented paradigm. Accessed July 11, 2016. https://www.seas.gwu.edu/~bell/csci210/lectures/oop.pdf
[17]. Shalloway Alan & Trott R James (2005). The Object-Oriented Paradigm. Net Objectives, vol 2, Issue 3
[18]. Alias Abdul-Rahman & Jane E. Drummond (2000). The Implementation Of Object-Oriented Tin-Based Subsystems For GIS. International Archives of Photogrammetry and Remote Sensing. Vol. XXXIII, Part B4. Amsterdam.
[19]. Gong Jianya & Li Deren (1996). Design and Implementation of An Object-Oriented GIS Software. International Archives of Photogrammetry and Remote Sensing, Vol.XXX1, Part B4.
[20]. Egenhofer, M.J., & Frank, A.U. (Eds.) (1989) Object-oriented Modeling in GIS: Inheritance and Propagation. Fails Church: American Congress on Surveying and Mapping, American Society of Photogrammetry and Remote Sensing.
[21]. Egenhofer, M & Herring, J. (1990). A Mathematical Framework for the Definition of Topological Relationships. In: K. Brassel and H. Kishimoto (Eds), Fourth International Symposium on Spatial Data Handling, Zurich, Switzerland, pp. 803-813.
[22]. Innovative GIS: GIS Modeling procedures. Accessed July 11, 2016 www.innovativegis.com/NR_Topic10.pdf on 10 July 2016.
[23]. Joseph K. Berry (1996). Object Oriented Technology and its GIS Expressions. Invited paper for publication in The Compiler (1996, 14:3 (16-19)), Forest Resources Institute, Florence, Alabama, USA.
[24]. Shih-Lung Shaw and Dali Wang (2006). Overview of GIS Modeling. Accessed July 11, 2016www.tiem.utk.edu/talks/shaw_gis.ppt on 2 July 2016.
[25]. Wikipedia (2015). Data Model (GIS). Accessed July 11, 2016. https://en.m.wikipedia.org/wiki/data_model_(GIS) on 2 July 2016.
[26]. ESRI (2016). ArcGIS API for JavaScripts. Accessed July 11, 2016 https://developers.arcgis.com/javascript/3/jsapi/aggregatepoints-amd.html
[27]. GeomatrixEnviroTechSolutions (2016).GIS. Accessed July 11, 2016. http://geomatrixenviro.com/gis