E-Tech: Tecnologias para Competitividade Industrial, Florianópolis, v. 4, n. 1, p. 82-89, 2º. Sem., 2011.

INDEPENDENT INTERFACE FOR SCHEMATIC DIAGRAMS IN GEOGRAPHIC INFORMATION SYSTEM

Hideson A. da Silva
Companhia Paranaense de Energia (COPEL) / Pontifícia Universidade Católica do Paraná (PUC-PR), hideson@gmail.com

***ABSTRACT*.** This paper presents Independent Interface for Schematics Diagrams (IISD) to facilitate and make the development of schematic diagrams applications more dynamic. Nowadays there are many options of tools for databases, graph visualizations and Geographic Information Systems, with many updates and sofwtare corrections, and usually these tools are necessary to develop schemactic diagram application. IISD helps this development and the maintenance during its lifetime. The proposal is to divide the application in functions, according to the tools used. In IISD the parts of the application are implemented independently, where each part can be updated, and other parts can be added, without remaking what has already been built in the application.

***Keywords**:* Schematic diagram. Graph representation. GIS.

E-Tech: Tecnologias para Competitividade Industrial, Florianópolis, v. 4, n. 1, p. 82-89, 2º. Sem., 2011.

1 INTRODUCTION

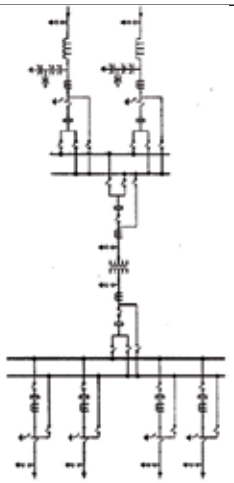
A schematic diagram is a graphic that represents situations disregarding information about the shape, size or location. In general, it is an illustration with standard symbols from a data structure graph representation.

Schematic diagrams have been studied for a long time [Laric 1946, Hurst 1969 and Smith 1971] and can be used in many areas, like networks (telecom, power, water, gas, etc), maps, electrical diagrams, etc. It is an important application, because it makes it easier to analyze a large volume of data and information than in tables or in large geographic maps.

Johnson (1994) writes about the fundamentals of schematic diagrams. Pretorius and Wijk (2008) used diagram schematics in their approach to the visual analysis of multivariate system traces. Shiran (1989) proposed a new framework for layout design verification and built an algorithm to generate the switch-level schematic diagrams. Schematic diagrams are largely used for circuit layouts and some researches can be found in Wu (2011) and Green (1990). In Figure 1 are shown two examples of schematic diagrams.

Figura 1 – Example of schematic diagram.

a) Sao Paulo Brazilian metro. [São Paulo 2011]



b) Eletrical diagram of substation power plant. [Ferrari and Da Cruz 2003]

Currently, the data about maps and networks are stored with the help of geographic information systems (GIS). GIS is an used software to store and manage data and information which contains geographic coordinates [HARMON and ANDERSON 2003]. Today the most popular softwares for GIS are: ArcGIS, MapInfo, IDRISI, InterGraph, SmallWorld, GRASS, which are listed in internet page of Stanford University (2011).

E-Tech: Tecnologias para Competitividade Industrial, Florianópolis, v. 4, n. 1, p. 82-89, 2º. Sem., 2011.

There are applications of GIS in many areas, for example: Utilities (electrical, gas water), agriculture, forestry, geology, hydrology, local and national government, military, risk management, transportation, disaster prevention, telecommunication, etc.

Therefore, considering the importance of GIS applications and the schematic diagrams for better analysis of GIS applications data, it is proposed an Independent Interface for Schematic Diagrams (IISD). This paper covers aspects of applications that work with GIS and schematic diagrams.

Talking about the maps data visualization in diagrams, Klippel et al. (2005) discusses different views of schematization from cartography. Parikh and Nielsen (2009) presented one research about IEC Common Information Model (CIM), which is a data model used to share information about power system feeder connectivity. Han and Kim (2009) presented the New Distribution Information System (NDIS) application using GIS schematic diagram to visualize power distribution in Korea.

2 OBJECTIVE

The IISD is an interface specification used to integrate and build schematic diagrams. It is possible to use it with any database, geographic interface, graph visualization and GIS.

Today the software industry is more complex, there are many options of software and tools (databases, graph visualization, GIS) and consequently more updates (“service packs”), new versions with new facilities. IISD was built to integrate all these items and give support during their lifetime. IISD will assist the department of Information Technology (IT) of companies, which must look at these items to build and buy solutions.

Therefore IISD proposes an interface that separates the parts, in which each part is responsible for a specific function of “business”. It makes it easier to add new tools and update current tools that are used in schematic diagram.

The main objectives are:

- **Simplicity.** It is easy to add more modules and facilities to the system.
- **Flexibility.** IISD permits many interfaces for each part of the specification.
- **Independence.** Each part of the interface can be developed independently.
- **Reliability.** The specification brings only one interface to read data and build the graph structure in all the system.
- **Portability.** To facilitate the replace of a part, like a database, graph visualization or GIS, in diagram schematic system.

E-Tech: Tecnologias para Competitividade Industrial, Florianópolis, v. 4, n. 1, p. 82-89, 2º. Sem., 2011.

With IISD it is possible builds schematic diagrams system and in the future it will allow the system to upgrade with new technologies, in any part of the system, without damaging the existent parts. It is an important feature in the software industry.

3 IISD ARCHITECTURE

The Architecture of IISD is separated into five main parts. For each part it is built a process that can be developed independently, as shown in Figure 2. These parts are:

1a.) DBMS

It is responsible for data access from database.

2a.) GIS

This part is responsible for geographic data.

3a.) Graph

The Graph is a standard data structure. It applies business rules to build the graph.

4a.) Graph Visualization

There are many tools (graphic engines) for graph visualization, so a module in IISD that is responsible for the rules of diagram visualization is necessary.

5a.) Schematic Diagram

It is a main formulary of application that calls the modules to build the diagram.

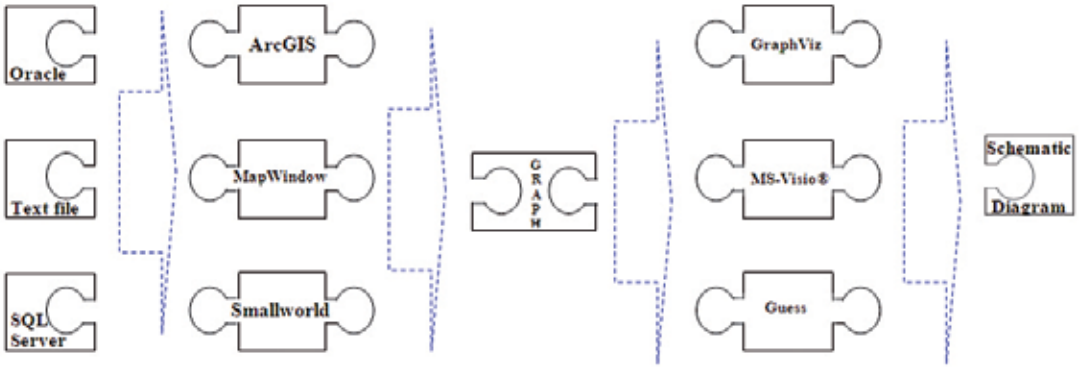
Figura 2 – Architecture proposal.

The architecture consists in dividing the diagram schematic in parts, and each part represents a specific area of expertise. Each area should be able to perform its function independently. If it is necessary to expand or modify one of the parts, this is done without interfering with other parts.

One application built with the IISD´s architecture can be easily modified and upgraded to use and receive new software. Figure 3 shows the flexibility of IISD, so that in each stage is possible to use more than one option to build the diagram.

The facility is shown when it is necessary to build one schematic diagram with more than one graph visualization. The only requisite is to develop one more “block” of graph visualization.

Figura 3 – Puzzle of IISD.

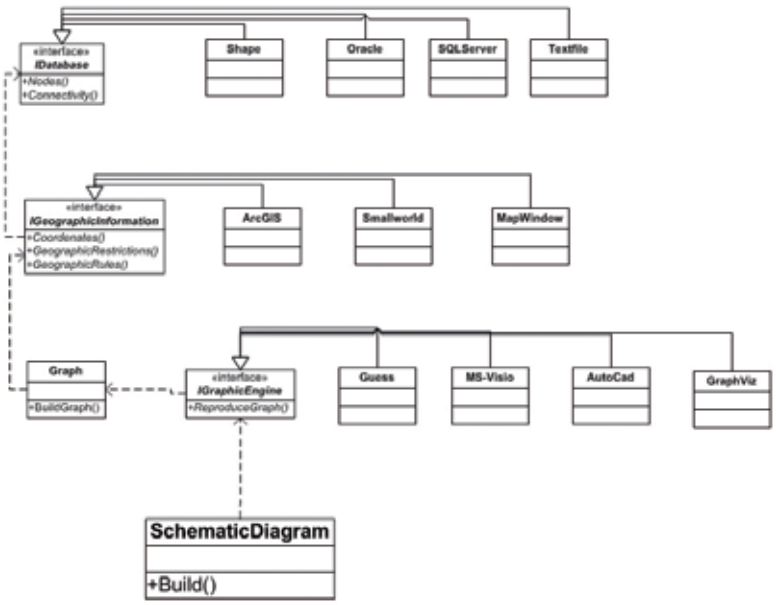


4 IMPLEMENTATION ASPECTS

The implementation of IISD proposes to use polymorphism in object-oriented (OO) languages, like C++, C# and Java. The review of OO can be found in Coad and Nicola (1993). The example of class diagram proposed for IISD, in Unified Markup Language (UML), can be observed in Figure 4. A review of UML can be found in Booch, Rumbaugh and Jacobson (2005).

In this example, there is one interface for databases, one interface for GIS and one interface for graph visualization. These separated interfaces permit that the upgrade development in each part of IISD is done more easily, without interfering in parts that are not the object of development.

Figura 4 – Class diagram.



E-Tech: Tecnologias para Competitividade Industrial, Florianópolis, v. 4, n. 1, p. 82-89, 2º. Sem., 2011.

In Figure 4 it is possible to see when it is requested one schematic diagram by the main class. Then, the main class asks for *GraphicEngine* class the schematic diagram drawing in one specific graph visualization. The *GraphicEngine* draws the diagram from one graph structure provided by the Graph class.

To build a graph structure, the *Graph* class asks for the nodes and edges from GIS. The coordinates, restrictions and rules of this data are managed by GIS application. *Database* class provides the data, regardless of where they are stored.

5 ASSESSMENT

The IISD was idealized to be implemented in any object-oriented language, but can be implemented in other language paradigms. The implementation in OO must be done using polymorphism, because when it is necessary to add another component, like database, graph visualization or GIS, it only implements the specific inherited class.

6 CONCLUSION

This paper described an interface, used to build schematic diagrams, in which each part can be developed independently, can be easily replaced and that is simple, flexible and reliable.

IISD is an option for developers to build one application that can draw and display standardized schematic diagrams. This application can be developed by system analysts with different knowledge and from different sectors, because IISD is separated in specific parts, so it is easier for developers to modify, upgrade and add a part. It is important to be able to add new tools and upgrades in an application, because it increases its lifetime.

For the company, and its users, one standard application is easier to use. IISD could be the standard model for all schematic diagrams that is necessary for all areas in the company.

REFERENCES

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. 2nd ed. **The Unified Modeling Language User Guide**. Boston, MA : Addison-Wesley, 2005.

COAD, Peter; NICOLA, Jill. **Object-Oriented Programming**.1nd ed. Prentice Hall, 1993.

FERRARI, Daniel Gomes; DA CRUZ Junior, Gelson. Classification of substation single-line diagrams based on fuzzy systems. **Proceedings of 22nd International Conference of the North American Fuzzy Information Processing Society-NAFIPS**, pp. 297-301, 2003.

GREEN, L. D. and ANDERSEN, J. Automated generation of analog schematic diagrams. **IEEE International Symposium on Circuits and Systems,** vol. 4, pp. 3197-3200, 1990.

HAN, Geoyoung; KIM, Jongmyung. **Distribution protection coordination program using GIS**. In: International Conference and Exhibition on Electricity Distribution, 20, 2009, pp. 1-4.

HARMON, John E. ; ANDERSON, Steven J. The Design and Implementation of Geographic Information Systems. John Wiley & Sons Inc, 2003.

HURST, S. L. Circuit and Schematic Diagrams. **Electronics and Power**, vol. 15, no.1, pp.24, 1969.

JOHNSON, J. Richard. **Schematic Diagrams**. , 1st Edition .Prompt Publications,1994.

KLIPPEL, A. et al. **The Cognitive Reality of Schematic Maps**. In L. Meng, A. Zipf and T. Reichenbacher, Map-Based Mobile Services – Theories, Methods and Implementations, pp.57-74. Berlin, 2005. Springer.

LARICK, S. H. Functional Schematic Diagrams. **Proceedings of the IRE**, vol. 34, no.12, pp.1005-1007, 1946.

PARIKH, P. A.; NIELSEN, T. D. **Transforming traditional geographic information system to support smart distribution systems.** Power Systems Conference and Exposition, pp. 1-9 , 2009.

PRETORIUS, A. Johannes; WIJK, Jarke J. van. Multiple Views on System Traces. **IEEE Visualization Symposium,** pp. 95-102, 2008.

SÃO PAULO, Companhia do Metropolitano de. [2011]. Disponível em: <http://www.metro.sp.gov.br/redes/mapa.pdf>. Acesso em: 07 maio 2011.

SHIRAN, Y. Reverse Engineering as a Framework for Design Verification. **IEEE International Symposium on Circuits and Systems**, pp. 969-972, 1989.

SMITH, Tom B. A Procedure for the Format of Schematic Diagrams for Design, Construction and Maintenance. **IEEE Transactions on Industry and General Applications**, vol. IGA-7 , no.2, pp. 303-313, 1971.

STANFORD UNIVERSITY. 2011. Disponível em:<http://www-sul.stanford.edu/depts/gis/soft.html>. Acesso em: 07 maio 2011.

WU, Yuping. Analog-aware Schematic Synthesis. Advances in Analog Circuits. **InTech-Open Access Publish**, pp. 247-296, 2011.

E-Tech: Tecnologias para Competitividade Industrial, Florianópolis, v. 4, n. 1, p. 82-89, 2º. Sem., 2011.

SOBRE O AUTOR

Hideson Silva

Doutorando em informática na PUC-PR, mestre em informática industrial pela UTF-PR e bacharel em informática pelas Faculdades Positivo. Possui 20 anos de experiência em sistemas de informação e atualmente é colaborador da Companhia Paranaense de Energia (COPEL), atuando na área de geoprocessamento. É docente nas Faculdades SPEI, onde leciona no curso de Bacharelado em Sistemas de Informação.

E-mail: hideson@gmail.com