Internet GIS and Water Resource Information

P. Deeprasertkul, R. Chitradon

Hydro and Agro Informatics Institute,
National Science and Technology Development Agency,
Bangkok, Thailand

Abstract. GIS is Geographic Information System, a computer system capable of integrating, storing, editing, analyzing, sharing and displaying geographically referenced information. At present, GIS is not only limited to cartography but also involves in various activities i.e. scientific investigation, natural resource management, environmental impact assessment, etc. Internet GIS allows more information sharing as many users can access GIS at the same time. Another progress in GIS is GIS/MIS where non geographical information (customized to users' purposes) regarding the particular area was overlaid with GIS. Internet GIS/MIS is useful for water resource management as it gives users better understanding of the overall picture i.e. GIS: locations of rivers/basins, topography of the flooded/drought areas, linkages of geographical factors and natural disasters occurred and MIS: water demand and supply thus gives users the ability to find best solution for each area and manage water resource in a sustainable manner.

1. Introduction

GIS can be linked to location data, such as people to addresses, reservoirs to plant, or rivers within a network [1]. This information can be laid to give a better understanding of how it all works together. Internet GIS/MIS is a research and application that utilizes the Internet systems to facilitate the access, processing, and dissemination of geographic or non-geographic information and spatial analysis knowledge [4]. The Internet GIS/MIS system has become a technology to support spatial analysis and expand information representation in GIS and MIS. These display usage can also meet water demand for management, analysis, estimation, and decision.

2. Internet GIS/MIS System Architecture

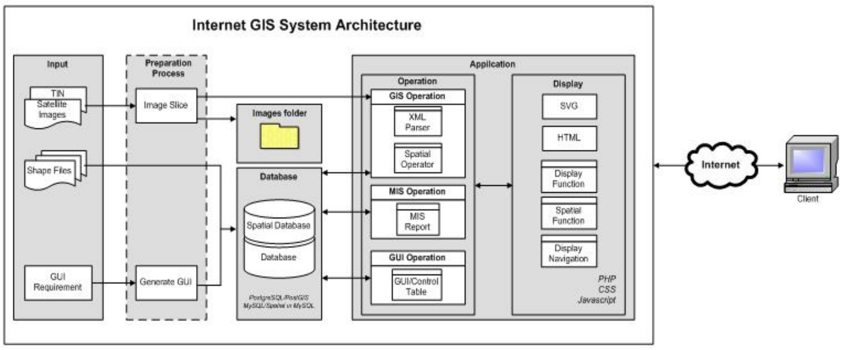


Figure 2.1 Internet GIS/MIS System Architecture

Architecture of Internet GIS/MIS System is shown in Figure 2.1. **GIS:** Geographical features can be described with raster data, or vector data (points, lines and polygons). Raster data, for example, aerial or satellite images are sliced into the strip map by a tool called *Image Slicer*. For vector data, they are imported to database and spatial database, respectively. **MIS:** The output reports are created and displayed on the Internet GIS/MIS. The data are queried from databases. **GUI Generator:** Internet GIS/MIS gives users the ability to easily generate GUI. The GUI and control table are created when the system receives the requirement.

3. XMLCompiler

The architecture of XMLCompiler is shown in Figure 3.1. XMLCompiler accesses and queries the non-spatial and spatial data to produce an input file. A software tool for parsing and converting the input file to *svg* files in XMLCompiler is XMLParser. The svg files are compressed to the *svgz* files.

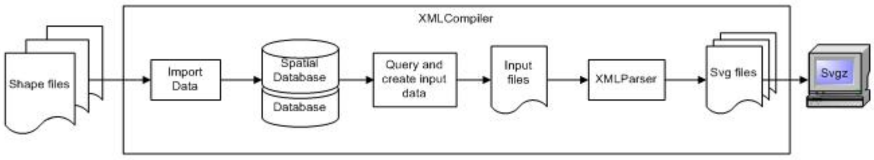


Figure 3.1 A XMLCompiler architecture

Definition 1: Let I be an input file of XMLParser and contains a lot of stream lines.

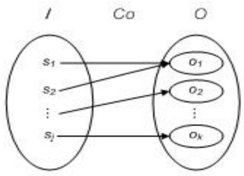
$I = \{s_j \mid s_j$ is the input stream and j is a number of input streams$\}$.

Definition 2: Let O be a set of XMLParser output files.

$O = \{o_k \mid o_k$ is the output file and k is less than or equal a number of *geocode*$\}$,

where *geocode* is a geographical code to identify a point or area at the surface of the earth [7].

Definition 3: Let *Co* be the mapping function from I to O: $Co: I \rightarrow O$. *Co* is the XMLParser processes and many-to-one function. Since many input streams containing the similar *geocode* are mapped to one output file.



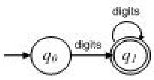
The XMLParser is correct with respect to the input assertion s_j and the output assertion o_k. Suppose that s_j is true, so that s_1 contains data in database and has satisfying pattern. Then XMLParser converts s_1 to o_1. Hence, o_k is true [5]. However, when a number of the input streams in I are added and associate with the original, a closure occurs. Hence, the new input files I^* have both original and added input streams. Thus, $I \subseteq I^*$. In case of O, the closure of output relation is also constructed in the similar condition. Thus, $O \subseteq O^*$ where O^* is a new set of output files. Then, the assertion $s_j \in I^*$ if $s_j \in I$ and $I \subseteq I^*$. Also, the assertion $o_k \in O^*$ if $o_k \in O$ and $O \subseteq O^*$. As the correctness assertion of I, O, and closures of their relations, XMLParser can be inferred that it is the robust program.

3.1. XMLParser

XMLParser is written by C programming language and applied Lex and Yacc for generating a lexical analyzer and a parser. Lex is a program generator designed for lexical processing of character input streams. The Lex source file associates the regular expressions and the input file fragments [3].

Definition 4: A *finite automaton* is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ [6], where

1. Q is a finite set called the *states*,
2. Σ is a finite set called the *alphabe*t,
3. $\delta: Q \times \Sigma \rightarrow Q$ is the *transition function*,
4. $q_0 \in Q$ is the *start state*, and
5. $F \subseteq Q$ is the *set of accept states*.



This state diagram of finite automaton (M) accepts all strings that have at least one digit. Thus $L(M) = \{w \mid w$ contains at least one digit$\}$. For Yacc, the heart of source is a collection of grammar rules [2].

Definition 5: A *context-free grammar* is a 4-tuple (V, Σ, R, S) [6], where

1. V is a finite set called the *variables*,
2. Σ is a finite set, disjoint from V, called the *terminals*,
3. R is a finite set of *rules*, with each rule being a variable and a string of variables and terminals,
4. $S \in V$ is the start variable.

The structure of XMLParser is shown in Figure 3.2 which has two following steps.
Step 1: The *xmlparser.lex* is the source specification which specifies the regular expressions. It recognizes expressions in a stream and performs the specified actions as it is detected.
Step 2: The *xmlparser.yacc* specifies rules describing the input structure and code to be performed when these rules are recognized. The Lex output file *lexxml.c* included in the Yacc source file and *xmlparser.yacc* are compiled and generated the *yxml.c*. Therefore, XMLParser is the resulting execution program from the compiling *yxml.c*.

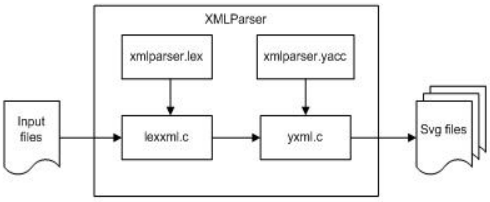
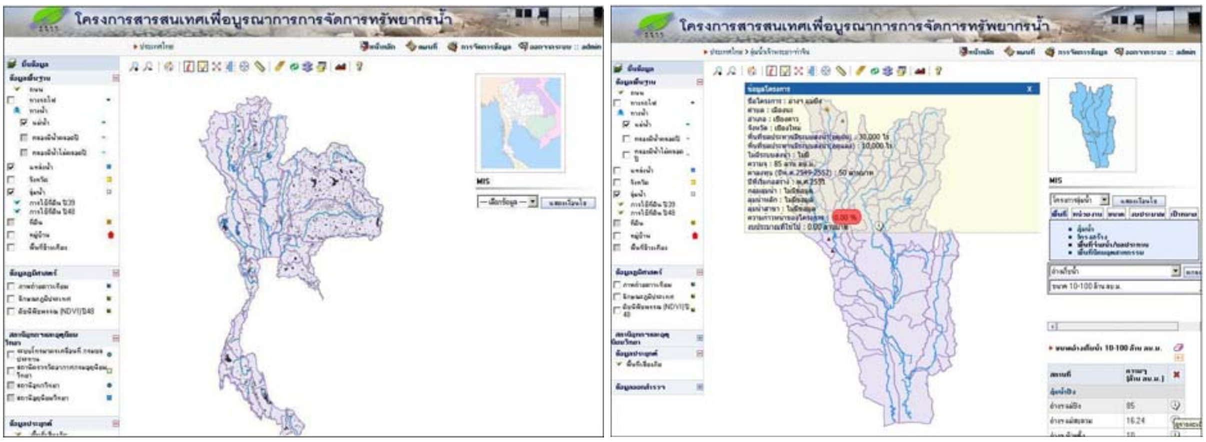


Figure 3.2 A structure of XMLParser

4. Examples

In this section, the example of Internet GIS/MIS system being GIS/MIS for Integrated Water Resources Management of Thailand project is illustrated. It is developed by Hydro and Agro Informatics Institute. Figure 4.1 (a) demonstrates a web application of GIS/MIS for Integrated Water Resources Management of Thailand. Figure 4.1 (b) shows the applied MIS in Internet GIS/MIS application. All information is displayed in table form on the MIS area and the selected information is displayed on the map.



(a) *(b)*

Figure 4.1 (a) The Internet GIS/MIS of Integrated Water Resources Management and (b) The MIS features on Internet GIS/MIS

5. Discussion and Conclusion

All software programs using in both XMLCompiler and Internet GIS/MIS system are based on the open source software. The present data of GIS developed by XMLCompiler are demonstrated high efficiency and robustly for complex spatial data. In addition to, the raster data of GIS are higher speed for loading. Anyway, MIS coupled with GIS provide a powerful tool for water resource management. It is necessary to further develop on combining MIS with GIS which can perform more perfectly and support government to make decisions better.

6. Further Development

The user-friendly XMLCompiler interface is expected that it will be released within the next few months. The tool for generating two or more spatial data by using the spatial operation function in PostGIS and Spatial Data in MySQL is developing.

REFERENCES

[1] ESRI, The Guide to Geographic Information Systems, GIS.com (2007), http://www.gis.com.
[2] JOHNSON, S.C., Yacc: Yet Another Compiler-Compiler, Yacc Online Manual, http://dinosaur.compilertools.net/yacc/index.html.
[3] LESK, M.E., SCHMIDT E., Lex-A lexical Analyzer Generator, Lex Online Manual, http://dinosaur.compilertools.net/lex/index.html.
[4] PENG, Z.R., TSOU, M.H., Internet GIS Distributed Geographic Information Services for the Internet and Wireless Networks, John Wiley & Sons, Inc., New Jersey (2003).
[5] ROSEN, K.H., Discrete Mathematics and Its Applications, Fifth Edition, the McGraw-Hill Companies, New York (2003).
[6] SIPSER, M., Introduction to the Theory of Computation, PWS Publishing Company, Boston, United State of America (1997).
[7] WIKIPEDIA, Geocode, Wikipedia Foundation, Inc. (2007), http://en.wikipedia.org/wiki/Geocode.
[8] WIKIPEDIA, Lookup Table, Wikipedia Foundation, Inc. (2007), http://en.wikipedia.org/wiki/Lookup_table.