ELSEVIER
CrossMark
Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 32 (2014) 405 – 412
Procedia Computer Science

The 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014)

Integrating GIS and FEATHERS: A Conceptual Design

Sungjin Cho[a*], Tom Bellemans[a], Davy Janssens[a], Geert Wets[a]

[a] *Transportation Research Institute (IMOB), Hasselt University, Wetenschapspark 5 bus 6, 3590 Diepenbeek, Belgium.*

Abstract

This study proposes integrating GIS and FEATHERS in order to improve the capability of data management, spatial analysis and visualization in the model framework. GIS provides geodatabase for effectively storing and edit (non-)spatial data, useful functions of spatial analysis for defining spatial interactions between phenomena simulated by the modeling system, and interactive visualization tools. Thus, this study mainly focuses on three topics: i) why FEATHERS needs a GIS module, ii) how the GIS module is designed, and iii) what functions can be supported by the GIS module in the FEATHERS system. Moreover, we overview some transportation software adopting GIS to catch up a general trend in GIS and transportation and also suggest data schema for creating geodatabase and a source code for making map layers and map tools in python. At the end, we conclude this paper with summary and plan for the future work.

Selection and Peer-review under responsibility of the Program Chairs.

Keywords: FEATHERS; GIS; geodatabase; spatial analysis; visualization; Python; PostGIS; QGIS.

1. Introduction

FEATHERS is an activity-based modelling system framework for predicting traffic demands. It functions as a policy measure by predicting how people change their activity-travel behaviors in response to a new transportation policy. Although the current version of the FEATHERS system shows enough performance in forecasting transportation demands to evaluate a policy, there are some challenges for improvement of the system, especially in spatial analysis and visualization (or mapping) of the simulation output. Thus, we propose a GIS module as an alternative to those challenges in FEATHERS. In that sense, this study mainly focuses on three topics: i) why

* Corresponding author. Tel.: +32-11-269-137; fax: +32-11-269-199.
E-mail address: Sungjin.Cho@UHasselt.be.

Selection and Peer-review under responsibility of the Program Chairs.
doi:10.1016/j.procs.2014.05.441

FEATHERS needs a GIS module, ii) how the GIS module is designed, and iii) what functions can be supplied by the GIS module in FEATHERS. This paper is organized as the following chapters. First, we introduce some cases of GIS modules in transportation modelling systems, such as TRANSIMS and UrbanSIM. Then, we briefly explain FEATHERS with a focus on system architecture, data requirement and outputs. After that, as a main part of this paper, we will deal with some issues on requirement to integrate between GIS and the FEATHERS system in terms of module development, functionality and initial condition. At the end, we conclude this paper with a research plan for developing a GIS module in the future.

2. Related Applications

2.1. TRANSIMS

TRANSIMS (Transportation Analysis and Simulation) is an integrated micro-simulation software based on a cellular automata and functions as a regional transportation system analysis. The software, originally developed at Los Alamos National Laboratory, is currently made available and furthermore developed as an open source project (http://www.transims-opensource.net). TRANSIMS provides GIS tools and formats. The GIS tools provide us with the function of editing and visualization by converting TRANSIMS network data into GIS shape files. The examples of the GIS tools supported are as following:

- Standardization: A wide variety of GIS tools exists to visualize results in maps.
- Layer concept: Data from different sources and applications can be visualized in a single map.
- Projections: Data formats are standardized to recognize projection systems of sources as well as destinations.

2.1. UrbanSim

UrbanSim is an open source urban simulation system, designed by Paul Waddell (University of California, Berkeley), for supporting metropolitan land use, transportation and environmental planning. UrbanSim requires a large amount of data representing a study area and they should be geo-coded in raster grid cells and stored as a tabular form in SQL database. All data are collected at a fine, micro-level, including household and employment data, GIS-based land- use and policy data and environmental constraints, travel access indicators and population data. The simulation output can be exported as ASCII text files to be transferred to GIS or other statistical software for further analysis and visualization. Moreover, GIS can directly link to UrbanSim in both input and output sides. UrbanSim is implemented in OPUS (Open Platform for Urban Simulation), which is a software platform written using Python and Numpy (python library).

2.3. TransCAD

TransCAD is the GIS software that is designed especially for transportation professionals to manage, visualize and analyze transportation data. By integrating GIS techniques and transportation modeling models in a single platform, TransCAD supports a powerful GIS engine with special extensions for transportation, mapping, visualization, and analysis tools designed for transportation applications, and application modules for routing, travel demand forecasting, public transit, logistics, site location, and territory management. TransCAD runs on desktop computing standards under Microsoft Windows. Therefore, we can acquire and install TransCAD at a much lower cost than any other integrated GIS and transportation modeling solution. Besides, we don’t have to build custom applications or complicated data interchange modules to perform transportation analysis with GIS data.

2.4. Cube

Cube is the transportation planning software with a broad range of capabilities for the comprehensive modeling of transportation systems. Cube allows us to analyze and evaluate the impacts of a wide range of transportation infrastructures and policies. Cube produces decision making information by using transportation modeling and GIS techniques, statistics and descriptive graphics and animations. Using the GIS tools, we can analyze, edit and visualize geodatabase data (which contain all kinds of data, except for trip table and impedance matrices) in travel demand models, such as transportation networks and transportation analysis zones. Since the GIS in Cube uses ESRI MXD[i] files to store all settings, we can directly use a MXD file in Cube (or ArcGIS).

3. FEATHERS

FEATHERS (Forecasting Evolutionary Activity-Travel of Households and their Environmental RepercussionS) is an activity-based transportation forecasting model framework that simulates the full pattern of activity and trip episodes of individual in each day [1]. FEATHERS was developed by IMOB (transportation research institute in Hasselt University, Belgium) in 2005, to facilitate the development of dynamic activity-based transportation demand model for Flanders, Belgium. Like other activity-based model frameworks, FEATHERS also requires a large amount of microscopic data representing activities, trips and environment constraints. Currently the FEATHERS framework adopts the core of the Albatross rule-based scheduler assuming that people attempt to make a decision to meet their needs and preferences within a given spatio-temporal constraints. In this context, the actual scheduling process is defined as an ordered sequence of activities and derived travel in which individuals try to set their goals, given some constraints that limit the number of feasible activities [2].

3.1. Framework

To implement several theoretical advances, such as dynamic and full microscopic ABMs (activity based models), FEATHERS provides the modular functions needed so as to develop and maintain ABMs in a particular study area. These modular functions allow for easily controlling functions and adopting a new theory into the model framework [3]. First, every FEATHERS module obtains its required settings from a configuration module (*ConfMod*) by communicating with each other. This configuration module functions as a central controller which can operate each of the modules and (in-)activate modules. This way enables to easily modify a parameter setting for a module by updating the XML configuration file. Second, data module (*DatMod*) is one of the main modules in the framework, which supports access to the data throughout all other modules. Third, population module (*PopMod*) deals with the different agents that are used in the synthetic population consisting of a group of agents who are characterized by a number of socio-economic attributes, such as age, gender and driver license ownership. Next, schedule module (*SchedMod*) links to the population module due to the fact that the schedule module operates input data from the population module and also stores outputs of the scheduler in the population module. Schedule execution module (*ExecMod*) implements a schedule execution mechanism and simulates the simultaneous and synchronous execution of all activities and journeys for population. Learning module (*LearnMod*) manages the learning process of the agents by means of the supervisor in combination with the scheduling and execution for that agents. Lastly, statistics module (*StatMod*) provides reports regarding the synthetic population and the activity-trip schedule to the end user. On the other hand, the visualization module (*VisMod*) creates graphic reports about the results from the FEATHERS simulation. However, the report only describe a basic statistic information and graphs about the simulation result, neither geostatistical information nor map visualization. Therefore, this study suggests that this module needs to be improvement by integrating a GIS function.

3.2. Data requirement

FEATHERS requires three types of data layers: activity-based schedule diary, synthetic population data and environment information in a study area. First, the activity-based schedule diary contains information about socio-

[i] MXD is a file extension for a map document used by ArcMap. ArcMap is a component of ArcGIS, a GIS (geographic information system) software suite.

demographic characteristics and activities and trips executed in a day. Typically, such diary data are collected by a survey asking respondents to record their personal information and daily schedule in a given day.

Second, FEATHERS also needs detailed information about population in the study area. In general, a related government institute, for instance the Census Bureau in the United States, regularly operates a census of the population and then reports the census data every some years. Nevertheless, some personnel information, such as a residence, income level and so on, is typically missing due to a privacy protection. Thus, synthetic population data are normally generated by using synthetic population applications, for example IPF (iterative Proportional Fitting) and CO (combinatorial optimization). To run the synthetic population applications, two types of data sets are required: seed and marginal data. Seed data contain detailed and disaggregate population information and typically has a small number of sample. On the other hand, target marginal data describe the sum of one dimension in each attribute for 100% population (or a desired fraction of population) in the study area.

Lastly, the environment information consists of land use data and transportation system. The land use data provides indicates of availability and attractiveness of facilities for conducting a certain type of activity in a certain area. Therefore, it describes the number of school children (for a bring/get activity) and employees in a (non-)daily goods (for shopping), catering (for a leisure activity), and bank/post offices (for a service) and so on. Transportation system consists of road network for privacy and public transport, as well as the network for slow mode. Each of these networks is defined by particular attributes that might influence transport mode choice decisions. In detail, those networks contains information about travel time and distance from origin to destination in a study area.

3.3. Outputs

As for a simulation output, FEATHERS produces an individual diary accounting which activities are conducted where, when, for how long, with whom, the transport modes involved and ideally also the implied route decisions. In addition, we can derive traffic demands from the predicted diary by extracting traffic volumes on the network, as OD matrices. using the OS matrices, we can plot time-dependent traffic flows on network. Moreover, FEATHERS provides a report with some basic statistics about the simulation output in a text or tabular form. To spatially analyze or visualize the simulation outputs, hence, end-users have to manually transfer the outputs to GIS or statistical software because of neither geo-coded outputs nor supporting spatial analysis and visualization functions.

3.4. Why FEATHERS needs GIS

To sum up, FEATHERS predicts individual activity-trip diary as a simulation result for population in a study area by means of complex mechanisms manipulating a decision-making process with activity-trip schedule, population and environment data. Based on the simulation result, it then provides a report with basic statistics of activity-trip patterns in text or tabular form and impedance matrices of representing traffic demands derived from the activity-trip schedules predicted. In other words, while FEATHERS treats a large amount of spatial data sets indicating population, land use and transportation in a study area to simulate individual activity-trip behaviors, it does not support any functionality of data processing for editing spatial parameters in using data, for example different zoning system or geographic coordination system. Even, there is no functionality for spatially linking between different input data sets, for instance between land use and transportation data. Those data sets are only connected to each other by using an identical zone ID. However, it is difficult for users to manage data connection because one should update the zone ID in all other data sets when the zone ID is modified in any data sets (it might happen when administrative districts are reorganized in the study area). Moreover, since there is no visualization in FEATHERS, one should manually transfer the output to GIS or statistical software and then operate the software separately. It requires additional efforts to transfer data and run other software.

Thus, this study proposes GIS as a solution to the above issues in FEATHERS that has problems with difficulties in managing heterogeneous data sets, limited analysis and visualization techniques. In general, GIS provide three main functionalities: integrated data management, spatial analysis and visualization. First, GIS allows us to effectively manage data by means of capability of integrating spatial and non-spatial data. Using GIS, a large amount of heterogeneous data can be stored in one database and organized along spatial, thematic or temporal considerations. The integrated environment guarantees that the data can be easily accessed to support various application needs.

Second, GIS offers not only standard GIS functions (e.g. spatial query, geocoding, buffer, etc.) but also spatial analysis tools (e.g. spatial autocorrelation, interpolation, regression and interaction). Using those GIS tools, we can define spatial properties (e.g. spatial dependency or auto-correlation) of the activity-trip behavior resulted from the simulation, as well as basic activity-trip patterns. At last, GIS supports interactive tools to visualize complex information, such as patterns and relationships, in a visual format on soft or hard copy.

4. GIS in FEATHERS

This section provides detailed ideas of requirements for integrating between GIS and FEATHERS with a focus on three aspects: data management, spatial analysis and visualization. To design the GIS module, we choose Python as a programming language because Python is often referred to as a “scripting language” that means you type in commands and immediately see the results. This is suitable for doing quick calculations or figuring out how a particular mechanism works that often happens in scientific project. This allows us to focus on the task at hand. Furthermore, many of the GIS software adopt the Python language for scripting GIS functions and libraries in the software. Therefore, we create geodatabase and design the GIS module in Python. The following subsections take into account the structure of spatial data in FEATHERS and the design of the geodatabase and the GIS module in Python.

4.1. Data management

To adopt GIS functions in FEATHRES, spatial data must be correctly encoded with a topology and a coordination system and non-spatial data, including both qualitative and quantitative data, must also be encoded and associated with their respective spatial components at first. All together with both spatial and non-spatial data are stored in geodatabase (also known as spatial database) which is a central data repository for spatial data and management. Once built the geodatabase, both managers and users can easily access to the geodatabase and manage or use the data they need. The geodatabase requires GIS data associated with each of the existing FEATHERS input, in particular land use and network data. Once acquiring the GIS data, it should be connected to the corresponding (non-GIS or attribute) data by joining those with a (primary) unique key (“ZONE ID).

In FEATHERS, land use and network data have to be stored in the geodatabase because those data sets contain spatial attributes, such as “Zone ID” in land use and “Origin/Destination ID” in network data. Thus, those spatial attributes are applied as a unique key for joining GIS data. Note that FEATHERS applies a hierarchical zoning system consisting of three levels: SUPERZONE, ZONE and SUBZONE for those land use and network data. Each level of zones represents a different level of spatial districts in the study area. Therefore, the original inputs are divided into three different data sets according to the three zone levels. However, in the geodatabase it is not necessary to divide the input data because a different level of zones are managed as individual objects with a different number of digits of zone ID and the attribute of the “ZONE LEVEL” field in the same data sets, not the separate ones.

Once spatial and attribute data are ready, and then we should create geodatabase with those data. By doing that, we can import spatial and attribute data from the geodatabase and make use of the GIS functions in FEATHERS. There are a number of options to create geodatabase: open source databases (e.g., MySQL, PostGIS and SpatiaLite) and commercial databases (e.g., Oracle, MS SQL Server). Each one of these databases has its own pros and cons so that the choice all depends upon your situation, such as the size and type of data you store (the open source databases normally have a smaller data storage than the commercial ones), your available budget for a license if you choose one of the commercial databases (whereas the open source databases grant a free license for users) and some functional requirements in your system. In this study, PostGIS is chosen because it provides enough storage for the spatial and attribute data in FEATHERS and is an open source database so we do not need any budget for a license. Moreover, it is compatible with QGIS (also known as "Quantum GIS") that provides GIS libraries we will use. QGIS will be explained in more detail later.

PostGIS is an open source software, released under the GNU General Public License, and an extension to the PostgreSQL[ii] object-relational database with supporting for geographic objects that allow location queries to be run in SQL. PostGIS supports spatial data types, spatial index and spatial query. Figure 1 shows the database schema of the location data in FEATHERS. Each of entities in the location data contains a unique ID ("id" in figure 1) and these IDs are linked to the IDs in a lower level of the entity, for example "czone (most central zone in superzone)" in superzone and "id" in zone. This is because FEAHTES uses a hierarchical zoning system consisting of 3 layers wherein each entity entirely belongs to only one entity at the uppers level.

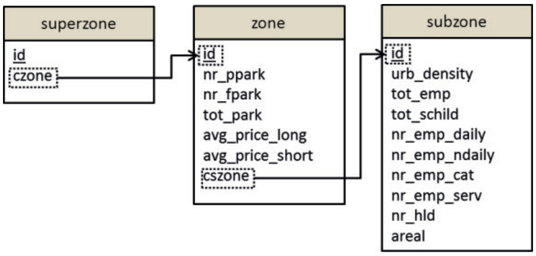

Fig. 1. Location database schema

Figure 2 illustrates a python code for designing the database (only "subzone" entity in figure 1). Procedure of building the database is as follows:

1) connect the database "FGDB (Feathers geodatabase)",
2) create table "subzone" with some columns; "id, "urb_density", "tot_emp" and so on,
3) insert a spatial column "outline" (polygon) and
4) create a spatial index "outline" for querying spatial features.

```
import psycopg2            # python library
connection = psycopg2.connect(“dbname=FGDB, user=IMOB”)       # 1) connect geodatabase
cursor = connection.cursor()
cursor.execute(“DROP TABLE IF EXISTS SUBZONE”)
cursor.execute(“CREATE TABLE subzone (id   SERIAL,                                # 2) create table “subzone”
                                      urb_density INTEGER,         # urban density
                                      tot_emp INTEGER,             # total number of employees
                                      tot_schild INTEGER,          # total number of school children
                                      nr_emp_daily INTEGER,        # number of employees in daily goods
                                      nr_emp_ndaily INTEGER,       #     in non-daily goods
                                      nr_emp_cat INTEGER,          #     in catering
                                      nr_emp_serv INTEGER,         #     in banks/post offices
                                      nr_hld INTEGER,              # number of households
                                      areal INTEGER,               # areal size
                                      PRIMARY KEY (id))”)
cursor.execute(“SELECT AddGeometryColumn(‘subzone, ‘outline’, 1234, ‘POLYGON’, 2)”)    # 3) insert a spatial column “outline”
cursor.execute(“CREATE INDEX zoneIndex ON subzone (outline)”)       # 4) index for connecting to GIS data
connection.commit()      # disconnect
```

Fig. 2. Python code for location database

[ii] PostgreSQL is a powerful, object-relational database management system (ORDBMS) which is released under a BSD-style license and is thus free and open source software.

4.2. Spatial analysis

Once the geodatabase is created, it allows us to easily edit and query (non-)spatial entities from the geodatabase. Figure 3 represent a python code for querying the zone ID with the number of households of the subzones where have more than 10,000 households.

```
import psycopg2          # import library
connection = psycopg2.connect("dbname=FGDB, user=IMOB")
cursor = connection.cursor()
cursor.execute("SELECT id, nr_hld FROM subzone WHERE nr_hld>10000")      # query attribute
for row in cursor:        # display results
   print row[0], row[1]
connection.commit()
```

Fig. 3. Python code for data query

In addition to such basic query, spatial analysis provided by GIS enable to take complex spatial properties into account, such as i) spatial autocorrelation, ii) spatial interpolation and iii) spatial regression.

First, the spatial autocorrelation calculates the degree to which a set of spatial features and their associated data values tend to be clustered together in space (positive spatial autocorrelation) or dispersed (negative spatial autocorrelation). Thus, the spatial autocorrelation defines the (hidden) spatial interaction between two attributes, such as between the number of activities executed and the number of employees in the study area. Figure 4 is the example of the autocorrelation where homogeneous areas are dispersed (negative autocorrelation) in the left side and clustered (positive autocorrelation) in the right side. This measurement provides a spatial interpretation for the simulation result in FEATHERS by accounting what extent to whether the simulated (or observed) patterns are independent or random.

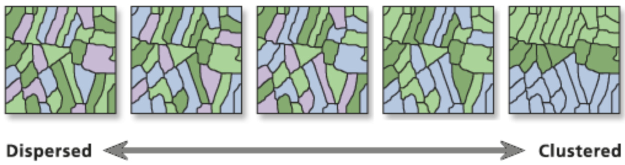

Fig. 4. Spatial autocorrelation (http://resources.arcgis.com)

Second, the spatial interpolation estimates the unobserved variables in geographic space based on the observed values. There are two methods of the spatial interpolation: inverse distance weighting (IDW) and Kriging. IDW assumes the unobserved value by attenuating the variable with decreasing proximity from the observed location. On the other hand, Kriging interpolates across space according to a spatial lag relationship with both systematic and random components. Those methods can apply for a wide range of spatial relationships for the hidden values between observed locations.

Lastly, the spatial regression captures spatial dependency that measures as the existence of statistical dependence in a collection of random variables or a collection of random variables in regression analysis. There are two methods in the spatial regression: ordinary least squares regression (OLS) and geographically weighted regression (GWR). OLS is a global statistical method for estimating unknown values in a linear regression model by minimizing the sum of squared vertical distances between the values observed in the dataset and the values predicted by the linear approximation. On the other hand, GWR is a local statistical technique that analyzes spatial variations in relationships. This allows us to evaluate how the relationship between a dependent variable and one or more explanatory variables changes as a function of the location in geographic space. Therefore, the methods enable us to explore the spatial relationships and understand the factors behind the observed spatial patterns in more detail.

4.3. Visualization

Visualization is one of the most attractive and useful capabilities of GIS. This ability enhances a viewer's perceptive abilities to understand the phenomenon due to the fact that it changes not only the look of the graphic image but also its interpretation. Visualizing data using GIS functions has many advantages over simple graphs. The following is a partial list.

- Interactive: adding new fields of data, removing them, changing the color scheme or form of the map, adding text and moving symbols
- Zoomable and pannable display: moving around in the display offers a user new perspectives, greater (or less) detail, and new insight
- Multi-dimensional (2d or 3d) display: creating the impression of three dimensions on the computer screen using complex rendering and shading algorithms

In this study, we propose PyQGIS, which is a python scripting in QGIS, for developing the visualization technique, because QGIS, an open source GIS software provides a lot of useful libraries for visualizing spatial data and also a complete QGIS API reference (http://qgis.org/api/). We can create a window that contains a map canvas and basic map tools for map panning and zooming. Panning is done with QgsMapToolPan, zooming in/out with a pair of QgsMapToolZoom instances. PyQGIS uses a map canvas which is a widget, so it allows up to simply create and show a map.

5. Conclusion

This study propose GIS as an alternative to the challenges in FEATHERS, which are spatial data management, spatial analysis and visualization. GIS allows us to effectively (non-)spatial data by storing data into geodatabase, to study spatial properties, such as spatial autocorrelation, spatial interpolation and spatial regression, and to enhance understanding of the phenomenon resulted by the simulation results. To integrate the GIS module with FEATHERS, we suggest to apply PostGIS for creating geodatabase and PyQGIS for developing GIS functions of spatial analysis and visualization. In this study, we first looked into some software adopting GIS and then proposed the design concept and architecture of the GIS module by providing some example code in python. As a part of the future work, we intend to develop the GIS module based on our conceptual design presented in this paper.

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 270833.

References

[1] Janssens D, Wets G, Timmermans H, Arentze T. Modeling short-term dynamics in activity-travel patterns: the feathers model. *Innovations in Travel Demand Modeling Conference* 2007; 71-77.

[2] Arentze T, Timmermans H. A spatial decision support system for retail plan generation and impact assessment. *Transportation Research. Part C: Emerging Technologies* 2000; 8(1-6): 361-380.

[3] Bellemans T, Janssens D, Wets G, Arentze T, Timmermans H. Implementation framework and development trajectory of FEATHERS activity-based simulation platform. Transportation Research Record: *Journal of the Transportation Research Board* 2010, 2175: 111–119.