Integration of multiple representations of elevation and process models

Laércio Massaru Namikawa

Instituto Nacional de Pesquisas Espaciais - INPE
Caixa Postal 515 - 12245-970 - São José dos Campos - SP, Brasil
laercio@dpi.inpe.br

Abstract. This paper presents a framework to integrate process models to multiple representations. The targeted type of process models is the dynamic spatially explicit process model, an environmental process models where values are spatially distributed and the term dynamic is used to emphasize the explicit time dependency of the process model. Of the available methods to link representations stored in a Geographic Information System (GIS) to process models, the tight coupling was selected since values are directly in GIS, being adequate for the computationally intensive requirements of dynamic spatially explicit process models. The choice of which GIS to use was driven by the need to access implementations of spatial algorithms. Since only open source software provides this capability, TerraLib, an open software GIS, was selected. TerraLib provides functions to access representations stored in relational databases. Metadata about the representation is stored using a database table and the spatial capabilities of TerraLib. This representation of information contains the area where valid values exist in the representation of elevation and values describing modeling procedure including date and time of the measurement, techniques used during measurement and production, and accuracy of the representation. Optimizations were implemented to support the computationally intensive characteristic of TITAN2D. They include a cache of past queries and prediction of future queries. The solution to select the best representation is based on a ranking system that considers resolution and date and time of the representation acquisition.

Keywords: dynamic process model, representation, simulation, TerraLib, TITAN2D.

1. Introduction

This paper presents a framework to link process models to multiple representations. The objective of the framework is to use multiple representations of the same geographic object to simulate a geographic phenomenon. In addition, the framework minimizes the generation of additional representations, provides fast retrieval of information from representations, and provides information from the representation best suited for process model requirements.

The focus is on representations of elevation since elevation is the base representation required for the type of process models targeted by the framework. Process models also require representations derived from elevation and those are handled by the framework as well. The term multiple representations term is used to refer to any set of representations of one geographic object and the term multiple representations of elevation is used to refer to representations of the geographic object elevation.

A geographic object is defined in this study to be a substantial entity of SNAP (derived from snapshot) subontology. SNAP and SPAN (derived from entities that span time) are subontologies of the Basic Formal Ontology (BFO), a information systems ontology being developed at the Institute for Formal Ontology and Medical Information Science (IFOMIS) (see Grenon and Smith (2004) and Feng et al. (2004) for more information on BFO). Conversely, process is one of the processual entities of the SPAN subontology, and defined as an entity extended in space-time that does not have discontinuities (Grenon and Smith 2004).

The focus of the framework is integration with dynamic spatially explicit environmental models, one of the most demanding process models in terms of computational and representation requirements. Dynamic process models are defined as models of processes in their transient states. The equations that describe the model include at least one derivative in time. Computational solutions for the differential equations must choose a time resolution that better approximates the computational and mathematical solution. While the simplest computational solution uses a fixed time difference (Burrough 1998), more sophisticated ones predict time differences at every step of the computation through the speed of propagating wave (Pitman et al. 2003; Patra et al. 2005).

In a spatially explicit process model, values of the model variables vary with their location; therefore, the model equations include derivatives of space (Kemp 1993). The simulation results of these models are also distributed over the region. The most common computational solutions for derivatives in space use finite element and finite difference methods (Mitasova and Mitas 2002).

Simulation of geographic process models requires values of spatial entities to solve process equations. A Geographic Information System (GIS) manipulates representations of spatial entities; therefore, a linkage between process models and GIS is essential. Processes and GIS are linked through loose, tight, and embedded coupling (Brimicombe 2003).

Loose coupling has the advantage of keeping the process model and GIS independent and is effective when the amount of values exchanged is small, such as in lumped process models. An embedded coupled integration is obtained when either a GIS or a process model has the capabilities of the other. Since a GIS developer can not predict all possible implementations to solve process equations and a process model can not support all types of representations that may become available, this integration limits some functionalities. When the process model is spatially explicit, the tight coupling is preferred since values can be directly accessed from GIS representations. Therefore, the framework uses a tight coupling.

2. A spatially explicit process model: TITAN2D

The proposed framework benefits spatially explicit process models that require multiple representations of elevation. TITAN2D is the process model that requires elevation and its derivatives at different resolutions (Patra et al. 2005). In addition, TITAN2D simulates processes that drastically change surface and measurement date and time influences its results.

TITAN2D is a tool for simulation of geophysical mass flows developed at The State University of New York at Buffalo by the departments of Geology, Mathematics, Mechanical and Aerospace Engineering, and Geography (Patra et al. 2005). The process model of TITAN2D is an averaged granular flow model based on laws of conservation of mass and momentum (Sheridan et al. 2005).

The governing equations of the flow model form the system of partial differential equations given by the depth-averaged mass balance, X-direction momentum balance, and Y-direction momentum balance equations presented by (1) (Patra et al. 2005):

$$\begin{cases} \partial_t h + \partial_x(hu) + \partial_y(hv) = 0 \\ \partial_t(hu) + \partial_x\left(hu^2 + \frac{\beta}{2} g_z h^2\right) + \partial_y(huv) = hg_x - h\beta\,\mathrm{sgn}(\partial_y u)\partial_y(hg_z)\sin(\varphi) - \frac{u}{\sqrt{u^2+v^2}}\left[hg_z\left(1 + \frac{u}{r_x g_z}\right)\right]\tan(\delta) \\ \partial_t(hv) + \partial_x(huv) + \partial_y\left(hv^2 + \frac{\beta}{2} g_z h^2\right) = hg_y - h\beta\,\mathrm{sgn}(\partial_x v)\partial_x(hg_z)\sin(\varphi) - \frac{v}{\sqrt{u^2+v^2}}\left[hg_z\left(1 + \frac{v}{r_y g_z}\right)\right]\tan(\delta) \end{cases} \quad \textbf{(1)}$$

where: ***XYZ*** is the Cartesian coordinate system, with plane ***XY*** parallel to the basal surface; $\boldsymbol{h}$ is the flow layer thickness; $\boldsymbol{r_x}$ and $\boldsymbol{r_y}$ define the radius of curvature of the local basal surface; $\boldsymbol{g_x}$, $\boldsymbol{g_y}$, and $\boldsymbol{g_z}$ define the local gravity vector; $\boldsymbol{\beta}$ is a function of the so-called pressure coefficient; $\boldsymbol{\varphi}$ is the flowing pile internal friction angle; and $\boldsymbol{\delta}$ is the basal friction angle.

The solution of the governing equations uses the parallel adaptative mesh Godunov scheme (Patra et al. 2005). The Godunov scheme represents and evaluates partial differential equations using finite volumes (Patra et al. 2005). The adaptative mesh characteristic of the equation solver implies that the size of the volume base is subjected to local refinement of the mesh to minimize approximation errors (Patra et al. 2005). Parallelism is supported in the solution by distributing volumes among available processors (Pitman et al. 2003).

The local direction of gravity ($\boldsymbol{g_x}$, $\boldsymbol{g_y}$, and $\boldsymbol{g_z}$) and the radius of curvature ($\boldsymbol{r_x}$ and $\boldsymbol{r_y}$) of the local basal surface are estimated from the first and second order derivatives of the elevation. The adaptative scheme used for the numerical solution requires elevation and its derivatives to be given at different resolutions for the different times of the simulation.

2. A GIS functions library for tight coupling: TerraLib

Tight coupling of environmental process models and GIS require process models to be open, that is, process models can be modified to provide linkage to representations in a GIS. In addition, GIS should also be open to provide access to its representations.

Most commercially available GIS have an open interface to access representations. However, the access is restricted to the portions of the software selected by the vendor. Furthermore, access is provided through interfaces that hide implementation algorithms and can not be audited. In addition proprietary GIS cost is high since the "market is an oligopoly in which two companies … have a market share of 50%" (Camara and Fonseca 2006).

Although the cost of a proprietary GIS is not a problem for most researchers, simulation of environment processes must benefit everybody, especially populations of hazardous areas. Since most hazardous areas are located in less economically favored regions, local population are less likely to be able to afford the high costs of software. Therefore, the selection of which GIS to use in this study was restricted to open software GIS.

An open software GIS that facilitates applications to access representations stored in relational databases through library functions, TerraLib (available at **www.terralib.org**), was selected for the implementation of the linkage. TerraLib is being jointly developed by various groups headed by the Image Processing Division at INPE (Brazilian National Institute for Space Research) (Câmara et al. 2000). The advantages of using TerraLib are: availability of the source code; ability to create query functions using algorithms that are independent of data structures; and use of database and GIS functions that have already been implemented.

TerraLib is built using C++ programming language since the language support to object-orientation and generic programming facilitates a collaborative development environment (Câmara et al. 2000). Object-oriented computer languages define classes that are abstractions of data types with functions that manipulate the contents of the class (Stroustrup 2000). Generic programming relies on containers classes and generic algorithms (algorithms that do not depend on the implementation details of the classes in the container) that manipulate containers (Stroustrup 2000). In TerraLib representations of spatial entities are stored in database tables using either the capability of handling spatial representations of spatial database or a Binary Long Object (BLOB) type (Vinhas and Ferreira 2005).

3. Adding multiple representations capability in TerraLib

A representation of elevation can use either a vector or raster format accordingly to the choice made during the modeling procedure. Modeling geographic objects consists of making measurements of their geometry and of their values and structuring the measurements for querying and manipulation (Laurini and Thompson 1992). In TerraLib, a representation of elevation in raster format is stored in a table identified by the prefix ("**Raster**"). A vector format representation of elevation using contour lines is stored in tables identified by the prefix "**Lines**" (for the contour lines) and the prefix("**Points**" (for sample points). The capabilities for using multiple representations of elevation were implemented only for raster format, although support for other formats is similar.

During a simulation, representations are queried to obtain values of elevation and its derivatives at locations where the processes occur. Since each representation of elevation has its own cartographic projection, a transformation of projection would be required at each request. In addition, information about the choices made during elevation modeling is not available for the queries. Therefore, an additional geometric representation that stores the geographic extents of the representation of elevation and the modeling choices is specified and included in the TerraLib database.

The proposed solution takes advantage of TerraLib query functions with spatial restrictions. A new representation using polygonal geometries that correspond to the spatial extents of each representation of elevation is created (identified by "**RepresInfo**", derived from Representation Information). "**RepresInfo**" has a projection that is suitable for the whole Earth to allow queries on representations of elevation for any location. The projection using latitude and longitude values is the most convenient and it is identified here by "**LatLong**" name. The datum of this projection must be the one that best fits the whole Earth; therefore, the WGS-84 geoid was selected.

External tables may be linked to TerraLib database tables. A table (named "**represattrib**") that contains detailed information about the choices made during the modeling of elevation is created and linked to the polygonal geometry with the extents of each representation of elevation. This table contains attributes of each representation of elevation that are obtained from the metadata available for the representation.

The metadata should describe all the choices made during modeling of the elevation. Existence of metadata associated with a representation of elevation is crucial for proper use of the representation. For example, if the metadata does not provide the information about when elevation was measured, and the application assumes that the elevation corresponds to values at another time and date, the difference in value between the expected by application and the provided by the representation may be in the range of tens of meters.

A standard metadata, such as the Content Standard for Digital Geospatial Metadata, published by the United States Federal Geographic Data Committee, usually includes information about the representation producer, source, geographic reference, and availability (Goodchild and Longley 1999; Guptill 1999). However, adoption of standard metadata has been slow due to the high workload of documenting existing representations and there are few representations of elevation with a complete metadata. Due to this constraint, only information that is useful for selecting a representation of elevation is included in the metadata of the framework. The attributes used to describe the representation metadata and stored in "**represattrib**" table are described in Table 1.

Using the "**represattrib**" table and the standard tables of TerraLib, the representations stored in a TerraLib database are queried through a standard SQL query or a spatial SQL query. For example, a standard query to retrieve the name of the layers with vertical accuracy better than 10 meters would be: *SELECT repres_id FROM represattrib WHERE vertical_resolution < 10 AND measure_unit = Meters*

An example of spatial query is obtained adding to the query presented in the previous example a constraint that it must contain a location given by coordinates 150, 260. The spatial query would be: *SELECT repres_id FROM represattrib WHERE vertical_resolution < 10 AND measure_unit = Meters AND contains(Point(150,260))*

A TITAN2D simulation query specifies the resolution and the date and time when elevation was measured. The availability of multiple representations of elevation with differences in resolution avoids problems arising from the use of a representation that is too coarse or too fine. For example, if the representation resolution is too coarse, a simulated flow may bypass a ridge that is too thin to be represented at that resolution. On the other hand, if the resolution is too fine, the computational resources will not be optimally used and the time to compute may be too long. In addition, the date and time of the measurement of the selected representation of elevation may be from after the simulated event, and the surface may have changed drastically due to the event or posterior processes.

Table 1 Description of the attributes used to provide description of a representation.

Attribute	Description
repres_id	Identifies to which representation the metadata is about.
description	Used to provide a brief report about the representation.
measure_name	Used to identify the type of information.
measure_unit	Identifies the measure unit.
resx and **resy**	Describe resolution of the representation.
measure_datetime	Used to store the date and time of measurement.
production_datetime	Identifies when the representation was created.
production_method	Describes techniques or algorithms used to create the representation.
production_source	Describes the measurement instrument.
source_scale	Indicates the scale of the original source of information.
horizontal_accuracy and **vertical_accuracy**	Describe the declared accuracy for the representation. Units are the same as the "**measure_unit**" attribute.

4. Implementing the link to the process model

The framework handles adaptative mesh by selecting the representation with the resolution that is closest to that required by TITAN2D and applying interpolation or aggregation algorithms to provide the required resolution. The interpolation method used is the bilinear interpolation, which estimates the value at a point as the distance-weighted average of values at the closest four grid cell centers. Bilinear interpolation performance is just slightly inferior when compared to more computationally expensive methods such as bicubic interpolation and kriging (Rees 2000). For aggregation, a method that estimates value by averaging all grid cells covered by a cell with the queried resolution is used. This aggregation method is used since it gives a surface that is smoother than the nearest neighbor method.

TITAN2D is computationally intensive and the framework needs optimizations to allow simulations to be executed effectively. A cache of queries and prediction of queries are implemented for the required optimization. The queries cache avoids making a query to the TerraLib database for every location where a value is required. In the cache, the parameters of a query and the values that satisfied the query are stored. When a new query is made, its parameters are checked for a match with previous queries in the cache. If there is a match, the value is obtained from the cache and not from TerraLib database, improving the performance.

The adaptative mesh of TITAN2D requires values at various resolutions. The prediciton is based on the assumption that if a query is made at a given resolution at a location, another query at the same resolution will be made for a different location. The implementation avoids making a new query to the TerraLib database by storing all the values of a representation and not only the value for a location.

TITAN2D requires values of slope and curvature at various resolutions. Representations of slope and curvature are created when the query can not be satisfied by the cache or by the TerraLib database. New representations are created from an elevation representation with the same resolution as the resolution of the derivative required by TITAN2D. Slope and curvature are estimated using a third-order finite difference method (see Horn (1981) for a description of the method). The new representations of slope and curvature are stored in TerraLib database with the metadata describing the estimation method and the source representation.

When the query specifies only resolution, the best representation for the query is the one with the closest resolution to query resolution. When the query specifies only the time of the case to be simulated, the answer to the query is not straightforward. In addition, if the query involves resolution and time, the answer is even more complicated. Therefore, a system that ranks representations according to how well they satisfy the query is necessary and was implemented. There are no rules to define the best representation for a general case and each process model must specify a ranking based on its own requirements. The framework allows the process model to specify the ranking system; however, a default system is provided.

The default system ranks higher representations acquired before the time constraint of the query, with ranking scores getting lower as the time difference becomes larger. When the acquisition time is just after the query time, the rank is lower than the rank for just before. In addition, the ranking scores of representations acquired after the query time decrease faster as the difference gets larger when compared to representations acquired before. When resolution and time of the case are constraints of the query, the solution proposed in the framework is to add the ranking scores given by resolution and time.

The C++ computer language classes that implement the linkage to a process model are the "**TeModelInterface**", "**TeGlobalModelInterface**", "**TeModelQuery**", "**TeModelQueries**", and "**TeRepresentationInfo**" classes. "**TeRepresentationInfo**" is used in the program that inserts a representation in TerraLib database. The optimizations required by the process models are provided by the other four classes. The classes are presented in Table 2.

5. A simulation session using TITAN2D and multiple representations of elevation

A simulation using TITAN2D requires representations of elevation, in a rectangular grid format to be imported into a TerraLib database. Next, the user defines the temporal and spatial limits of the simulated geophysical mass flow, that is, when and where the simulated event happened. The temporal limits are given by the event's initial and the final date and time. The spatial limits require definition of the projection of the limits coordinates. In addition, the user defines the computational parameters of the simulation and the flow pile parameters.

For every step of the simulation, at each cell of the adaptative grid, required values of elevation and their derivatives are queried at a resolution defined by the cell size and constrained by the date and time of the simulated event. Derivatives and representations of elevation are created when the query can not be satisfied. The sequence for a simulation session is presented in Figure 1a, and a summary of results from simulation of a block-and-ash flow event (at Colima Volcano, Mexico) is presented in Figure 1b.

6. Summary

A framework to link dynamic spatially explicit environmental models was presented, with details of the implementation of the framework to link TITAN2D, the process model used as an example of process model, and a GIS that was built using TerraLib. The framework supports storage of representations in a relational database. Representations are queried by the process model, using spatial and metadata constraints, to provide values that best match the requirement to solve its differential equations.

Table 2 Description of the classes that implement the linkage to a process model.

Class Name	Description
TeRepresentationInfo	Facilitates reading and writing of representations and their information into TerraLib database tables.
TeModelInterface	Implements the interface to process models.
TeGlobalModelInterface	Provides the unique global access to the interface to assure that the interface to process models is not duplicated.
TeModelQuery	Implements a cache entry.
TeModelQueries	Implements the cache of queries.

The framework supports the concept of generating new representations only when an application requires it. Thus in the case implemented in this framework, representations with different resolutions are created when a query needs it. In addition, dynamic spatially explicit environmental models use differential equations in space and the solution for these equations require slope and curvature values. New representations of slope and curvature, derived from elevation, are created when queried using a finite element estimation method.

Due to the requirements that TITAN2D be computationally efficient, optimizations were implemented. The optimizations are achieved using a cache of previous queries and the values satisfying those queries. In addition, values of all locations for a selected representation are stored in the cache.

The problem that arises when querying multiple representations is how to rank when more than one representation satisfies the query. The solution proposed in the framework is to let the process model specify the ranking system. In the implementation of the framework, the default ranking gives a score that is higher for resolution matching the query and decreases the score as the difference increases and ranks higher for representations derived from measurements taken just before the query time. The time rank decreases as the time difference increases, with a higher decrease if measures were taken after the time required in the query.

References

Brimicombe, A. **GIS, Environmental Modelling and Engineering**. London; New York: Taylor and Francis, 2003. 312 p.

Burrough, P. A. Dynamic Modelling and Geocomputation. In: **Geocomputation: a primer**. P. A. Longley, S. M. Brooks, R. McDonnel and B. Macmillan. Chichester, England: John Wiley and Sons, 1998. p. 165-191.

Camara, G. and F. Fonseca. Information Policies and Open Source Software in Developing Countries. **Journal of the American Society for Information Science and Technology**, accepted for publication, 2006.

Câmara, G., R. C. Souza, B. M. Pedrosa, L. Vinhas, A. M. V. Monteiro, J. A. Paiva, M. T. Carvalho and M. Gattass. TerraLib: Technology in Support of GIS Innovation. In: 2nd Brazilian Symposium on GeoInformatics, 2000, São Paulo, Brazil. **Proceedings**.

Feng, C.-C., T. Bittner and D. M. Flewelling. Modeling Surface Hydrology Concepts with Endurance and Perdurance. In: **Giscience 2004**. Berlin, Heidelberg: Springer-Verlag, 2004.

Goodchild, M. F. and P. Longley. The Future of GIS and Spatial Analysis. In: **Geographical Information Systems**. P. A. Longley, M. F. Goodchild, D. J. Maguire and D. W. Rhind. New York: John Wiley & Sons, 1999. p. 567-580.

Grenon, P. and B. Smith. SNAP and SPAN: Towards Dynamic Spatial Ontology. **Spatial Cognition and Computation** , v. 4, n. 1, p. 69-10, 2004.

Guptill, S. C. Metadata and Data Catalogues. In: **Geographical Information Systems**. P. A. Longley, M. F. Goodchild, D. J. Maguire and D. W. Rhind. New York: John Wiley & Sons, 1999. p. 677-692.

Horn, B. K. P. Hill Shading and the Reflectance Map. **Proceedings of the IEEE**, v. 69, n. 1, p. 14-47, 1981.

Kemp, K. K. **Environmental Modeling with GIS: A Strategy for Dealing with Spatial Continuity**, NCGIA - National Center for Geographic Information and Analysis, 1993. p. 142.

Laurini, R. and D. Thompson. **Fundamentals of spatial information systems**. London: Academic Press, 1992.

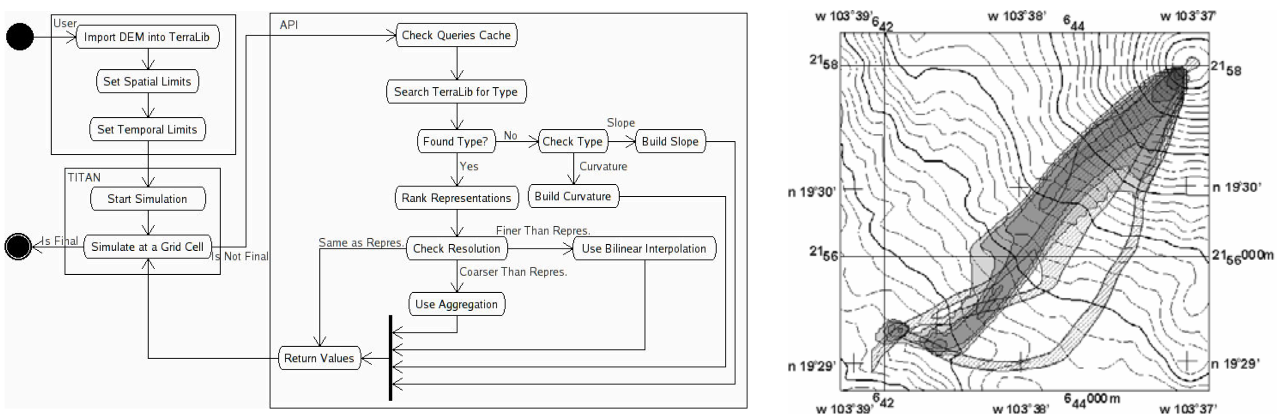
Mitasova, H. and L. Mitas. Modeling Physical Systems. In: **Geographic Information Systems and Environmental Modeling**. K. C. Clarke, B. O. Parks and M. P. Crane. Upper Saddle River, NJ: Prentice Hall**,** 2002. p.189-210.

Patra, A. K., A. C. Bauer, C. C. Nichita, E. B. Pitman, M. F. Sheridan, M. Bursik, B. Rupp, A. Webb, A. Stinton, L. M. Namikawa and C. Renschler. Parallel Adaptative Numerical Simulation of Dry Avalanches over Natural Terrain. **Journal of Volcanology and Geothermal Research**, v. 139, n. 1-2, p. 1-21, 2005.

Pitman, E. B., C. C. Nichita, A. Patra, A. Bauer, M. Sheridan and M. Bursik. Computing granular avalanches and landslides. **Physics of Fluids**, v. 15, n. 12, p. 3638-3646, 2003.

Rees, W. G. The accuracy of Digital Elevation Models interpolated to higher resolutions. International Journal of Remote Sensing, v. 21, n. 1, p. 7-20, 2000.

Vinhas, L. and K. R. Ferreira. Descrição da TerraLib. In: **Bancos de Dados Geográficos**. M. Casanova, G. Câmara, C. Davis, L. Vinhas and G. R. Queiroz. Curitiba: Editora MundoGEO, 2005. p. 383-426.



(a) (b)

Figure 1 (a) Main sequence for simulation of a geophysical mass flow using the link between TITAN2D and the multiple representations of elevation. (b) Summary of results from simulation of block-and-ash flow event at Colima Volcano.