Brigham Young University

BYU ScholarsArchive

International Congress on Environmental Modelling and Software

8th International Congress on Environmental Modelling and Software - Toulouse, France - July 2016

Jul 12th, 9:10 AM - 9:30 AM

Integration of Land Use and Land Cover Models: Coupling Two Existing Models to Improve the Simulation of Location Choice

Harutyun Shahumyana
National Center for Smart Growth Research and Education, University of Maryland, University College Dublin, harutyun.shahumyan@ucd.ie

Rolf Moeckel
Technische Universität München, rolf.moeckel@tum.de

Peter Claggett
U.S. Geological Survey, pclaggett@usgs.gov

Frederick Ducca
National Center for Smart Growth Research and Education, University of Maryland

Follow this and additional works at: https://scholarsarchive.byu.edu/iemssconference

Part of the Civil Engineering Commons, Data Storage Systems Commons, Environmental Engineering Commons, Hydraulic Engineering Commons, and the Other Civil and Environmental Engineering Commons

Shahumyana, Harutyun; Moeckel, Rolf; Claggett, Peter; and Ducca, Frederick, "Integration of Land Use and Land Cover Models: Coupling Two Existing Models to Improve the Simulation of Location Choice" (2016). *International Congress on Environmental Modelling and Software*. 17.
https://scholarsarchive.byu.edu/iemssconference/2016/Stream-C/17

This Event is brought to you for free and open access by the Civil and Environmental Engineering at BYU ScholarsArchive. It has been accepted for inclusion in International Congress on Environmental Modelling and Software by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

International Environmental Modelling and Software Society (iEMSs)
8th International Congress on Environmental Modelling and Software
Toulouse, France, Sabine Sauvage, José-Miguel Sánchez-Pérez, Andrea Rizzoli (Eds.)
http://www.iemss.org/society/index.php/iemss-2016-proceedings

Integration of Land Use and Land Cover Models: Coupling Two Existing Models to Improve the Simulation of Location Choice

Harutyun Shahumyan[ab], Rolf Moeckel[c], Peter Claggett[d], Frederick Ducca[a]
[a]*National Center for Smart Growth Research and Education, University of Maryland*
[b] *University College Dublin, harutyun.shahumyan@ucd.ie*
[c] *Technische Universität München, rolf.moeckel@tum.de*
[d] *U.S. Geological Survey, pclaggett@usgs.gov*

Abstract: Single models do not usually provide all answers required for complex policy decisions, so an integrated modelling is often applied to inform policy makers and urban planners. Developing a fully integrated model is an expensive and time consuming task, thus, coupling existing models is often used for model integration. The paper provides an overview of potential model integration approaches, briefly describes the Simple Integrated Land Use Orchestrator (SILO) model and the Chesapeake Bay Land Change Model (CBLCM), and focuses on the integration method applied to link those models. Particularly, Python wrappers were developed to loosely couple SILO and CBLCM; while ArcGIS Model Builder was used to provide a graphical user interface. The suggested approach is especially efficient when the models are developed in different programming languages, their source code is not available or the licensing restrictions make other coupling approaches infeasible.

***Keywords**:* SILO; CBLCM; land use; land cover; model integration; loose coupling; Python wrapper.

1 INTRODUCTION

Land use and land cover models form critical elements in understanding location choice and its impact on the environment. Traditionally land use models have been used in conjunction with transportation models, enabling modellers to estimate the origins and destinations of trips, the location of demand and how residential and employment location interact with the transportation system. Land cover models, on the other hand, focus on the usage of the land; whether it is agricultural, forest, housing, employment or any one of a wide range of covers. While these types of models have evolved from different needs and disciplines and may be applied for different purposes, the outputs of these two types of models are closely related and can contribute each other. In analysing policies it is important that the results of these models be reasonable consistent with each other.

In this paper we explore a method to link a residential location choice model, the Simple Integrated Land Use Orchestrator (SILO) with the Chesapeake Bay Land Change Model (CBLCM), a model which predicts land cover at the parcel level. We address how these linkages can be made, the analytic and software challenges of making these linkages, and the results from linking these models together.

2 MODELS

2.1 Simple Integrated Land Use Orchestrator (SILO)

SILO is a microscopic discrete choice land use model[1]. The model consists of three main modules, namely household relocation, demography and real estate development. The core module is household

[1] The model is described in more detail at www.silo.zone.

relocation, in which individual households consider moving to alternative dwellings subject to their available housing and transportation budgets as well as to commute travel times for each worker in the household. Because SILO functions as a microsimulation in which households with their respective income and household members with their individual workspace are represented explicitly, the model is able to respect monetary and time constraints explicitly (Moeckel, forthcoming).

The demography model covers all relevant demographic events, including aging, marriage and divorce, birth of children, children leaving the parental household, death, car ownership and change of employment status or workplace. The real estate development module simulates developers who invest into new dwellings if the demand is high. Developers imitate household location preferences to build the most marketable housing stock. Some dwellings deteriorate over time and may be demolished eventually. Other dwellings are renovated. This module also includes a price model that adjust housing prices upwards (high sensitivity) under high demand and downwards (low sensitivity) if demand for this dwelling type in this neighbourhood is slow.

SILO starts in 2000 and runs through 2040 in one-year increments. Population data are validated at the county level for the year 2012 (so-called backcasting). As the model is capable to reasonably represent changes from 2000 to 2012, it is assumed to capture behavioural principles of household relocation, demography, and real-estate development.

2.2 Chesapeake Bay Land Change Model (CBLCM)

The CBLCM is pseudo-cellular automata simulation model that translates exogenous projections of population and employment into spatially mapped patches of residential and commercial development on the landscape after accounting for infill and redevelopment. The CBLCM produces 101 stochastic Monte Carlo simulations of future growth for any given set of assumptions (e.g., a future scenario) over user specified time intervals. Each CBLCM simulation represents an equally plausible independent representation of future urban growth patterns. The Monte Carlo simulations enable the quantification of uncertainty associated with each future scenario. The amount of growth simulated is controlled by exogenous household and employment projections, estimated infill/redevelopment rates, and housing and employment densities. The CBLCM is dynamic in that growth attracts growth and housing and job densities can gradually increase through time relative to their proximity to urban centres. Moreover, by explicitly representing housing and job densities for every residential and commercial patch, future development can be reclassified consistent with local land use definitions. The CBLCM is designed for use at the regional-scale with multi-temporal land use/cover data such as the 30-meter resolution National Land Cover Database developed for the conterminous United States (Homer et al., 2015).

3 METHODOLOGY

This section identifies requirements for model integration and briefly reviews coupling approaches described in the literature. The challenges of coupling of particular models are discussed and an alternative method of loose coupling is suggested to link SILO and CBLCM models.

3.1 Model Coupling Approaches

For the coupling of environmental models, Brandmeyer and Karimi (2000) identify a five-level coupling hierarchy. The *manual data transfer* method is the most basic level of model coupling. Though this approach requires minimal initial cost and time to apply, it is not convenient when multiple runs and frequent data exchange are required (Brandmeyer and Karimi, 2000). Models integrated by *loose coupling*, for which the data exchange between models is automated, still work independently and the user interacts with each model separately (Wong et al., 2009). Loose coupling has low initial cost, requires minimal changes to existing codes, and the models can be developed independently. In *shared coupling*, the models either share a user interface, which hides the internal coupling method (Berry et al., 1997), or they share database, for which the models are kept separate but share the data storage (van Walsum and Veldhuizen, 2011). *Joined coupling* employs both the common user interface and data storage and may use two structurally different approaches: embedded coupling, where one model contains another (Liu et al., 2014); or integrated coupling, where each model is a peer of every other

model (Sudicky et al., 2003). For *tool coupling*, the models are coupled using a modeling framework (Babendreier and Castleton, 2005; Moore and Tindall, 2005). Though it has higher initial costs due to framework design and development, several such tools have been developed, such as the Open Modelling Interface (OpenMI) (Gregersen et al., 2007), the Model Coupling Toolkit (MCT) (Warner et al., 2008), the Community Surface Dynamics Modelling System (CSDMS) (Overeem et al., 2013), the Earth System Modelling Framework (ESMF) (Hill et al., 2004), the Framework for Risk Analysis Multimedia Environmental Systems (FRAMES) (Shah et al., 2004; Whelan et al., 2014), PCRaster (Schmitz et al., 2009), O-PALM (Piacentini et al., 2011), OASIS (Valcke, 2013) and ICMS (Rahman et al., 2004).

The order of running the models and the data feedback frequencies also effect model coupling choices. The 'sequential' coupling scheme provides the weakest form of the integration, where the first model runs the required time step/period and provides the output to the second model, which only runs after getting the results of the first model (van Walsum and Veldhuizen, 2011). This scheme is often used for manual data transfer or loose model coupling. A drawback of such an approach is that the stability between the two linked models is determined by the model that gets updated first, which can lead to inconsistencies for the second model. In contrast, the 'fully coupled' scheme supports the full feedback between models within each time step. However, full coupling usually requires code modification. Moreover, it may result in iterations within iterations and increase its computational load essentially, reducing the overall efficiency (van Walsum and Veldhuizen, 2011).

Each of the described approaches has its advantages and disadvantages (Brandmeyer and Karimi, 2000; Droppo et al., 2010) and the selection of the method mainly depends on the model requirements, research goals and available resources.

3.2 Challenges in Coupling SILO and CBLCM

Model integration can be challenging when models are developed independently. Often, as in the case of this study, models are developed in different programming languages and software environments (Table 1). SILO was written in Java and CBLCM was written in C++, and both were developed independently in different organizations. Therefore, keeping them separate and not changing the source codes was an important aspect for keeping the integrated suite compatible with the continuously updated versions of the models. Moreover, the models run sequentially, and there is a one-way data flow from SILO to CBLCM. More specifically, SILO provides households and employment data to CBLCM. Though technically this data transfer could be done for every simulation year, it is implemented only for selected simulation years. SILO runs internally in one-year increments and while the CBLCM is flexible to use any user-specified time increment, it frequently takes years to fully realize development on the ground such as the build out of a residential subdivision. Therefore, it's unrealistic to run the CBLCM in increments less than 5-years and makes more practical sense to run it in 10-year increments given the volume of output data produced. Thus it is optimal for SILO to provide decadal inputs to the CBLCM (Table 1).

Table 1. Main characteristics of the used models.

Model	Programming Language	Operation System	Simulation Years	Number of runs	Time per model year	Total Runtime*
SILO	Java	Multi-platform	2001 - 2040	39	9 min	6 hour
CBLCM	C / C++	CentOS Windows	2030, 2040	2	3 hour	6 hour

* *20 x AMD Opteron Processor 6328 @ 3.20GHz, 42GB RAM, Windows 7.*

Taking into consideration those factors, loose coupling method was identified as the most appropriate approach for linking SILO and CBLCM models. However, the existing coupling tools could not have been applied for various reasons, including the absence of support for the Windows or for programing language used, or the requirements for changing the original source codes. Even widely used integration frameworks such as OpenMI or CSDMS aiming to couple models with minimal changes

require the modification of the source codes (Belete et al., 2014; Whelan et al., 2014); and using generic model wrappers provided by FRAMES can also be quite labour-intensive (Droppo et al., 2010). The aim of this research was to couple particular models with minimal efforts, without changing their source codes and in the meantime providing an option for easy plug-in for other models. Specific wrappers have been developed in Python to do that; while ArcGIS Model Builder was used as a graphical user interface for the integrated suite.

3.3 Using GIS and Python for Model Integration

GIS has been used widely as an integration tool for different environmental models including soil erosion (Brazier et al., 2005), land use (Clarke and Gaydos, 1998), hydrologic (Devantier and Feldman, 1993), water quality (León et al., 2002), pollution and watershed (Basnyat et al., 2000; Kittle et al., 2006) and other models (Argent, 2004). In most of these cases, GIS is used for loose coupling of the models, implementing data exchange and visualization. It helps to overcome some of the problems related to data interoperability (Goodchild et al., 1997), particularly organizing data manipulation and exchange (Argent, 2004). Moreover, availability of flexible scripting languages like Python in modern GIS packages (e.g. ArcGIS) allows to develop interactive user interfaces within GIS under which the models can be linked (Tao et al., 1996).

Python is an open source object-oriented programming language balancing high-level programming with low-level optimization (Aruoba and Fernandez-Villaverde, 2015). Though Python programs usually run slower than FORTRAN, Java or C/C++ programs, Python has become popular due to its simpler syntax and less requirements on specialized knowledge about operation system and memory management (Schmitz et al., 2009). From the model integration perspective, Python has specific libraries supporting scientific programming (SciPy), modelling and data analysis (Pandas), as well as visualizations and parallel computing (IPython). Another advantage is its language interoperability often used to glue other programming languages. Thus, Python has libraries supporting function calls from MatLab (MLabWrap), R (RPy), Excel (OpenPyxl), FORTRAN (F2PY, PyFort), Delphi (Python4Delphi), Java (Jyton, JPype, Jepp), Perl (PyPerl), PHP (PiP), C/C++ (Ctypes, Cython, SWIG), etc. Moreover, Python runs natively on Windows, Mac and Linux operation systems. Thus, Python can facilitate interoperating modules implemented in other programming languages (Roberts et al., 2010) and has been successfully used to link such models (Schmitz et al., 2009).

3.4 Python Wrapper

A Python script was developed to wrap existing independent models and provide a user-friendly interface through ArcGIS Model Builder for data exchange with other tools or models. As each model has specific run parameters, input and output files, in spite of having similar general structure, the wrapper was fine-tuned explicitly for SILO and CBLCM.

Table 2 presents the main components of the wrapper, while the complete code is provided at GitHub[2] and publicly available for other users. The ESRI Arcpy library was used for getting and setting the model parameters within a wrapper as well as for displaying status messages in the ArcGIS geo-processing window. The os.system() standard function was used to call the actual models from the wrapper. In some cases, access to a model source code may be restricted or changing the code is not desirable. To avoid any problems with the file paths used in the model, the default path was set to the folder containing the model using the os.chdir() command. The exchangeable output files were set as wrapper output parameters and can be used by other models or tools. A wrapper with such structure can be easily imported in the ArcGIS Model Builder environment.

[2] https://github.com/Shahumyan/ModelPythonWrappers

Table 2. The main components used in the wrappers.

Component	Code example

Initialization Stage
Organizes the initialization of the model based on the provided parameters such as the path to the model program (executable or script), parameters file, the scenario name, simulation period, etc.
	# Get input arguments
	in_Program = arcpy.GetParameterAsText(0)
	in_Scenario = arcpy.GetParameterAsText(1)
	in_Year = arcpy.GetParameterAsText(2)

Running Stage
Makes the folder containing the model executable as a default folder and runs the model from there.
	# Run the model / program
	desc = arcpy.Describe(in_Program)
	sourceFilePath = desc.path
	os.chdir(sourceFilePath)
	os.system(in_Program)

Results Output Stage
Export output files (filename1, filename2, …) from the model output folder (outfolder) into a specified folder from where other models can use them.
	# Export output files
	Output1=sourceFilePath+"\\outfolder\\filename1"
	Output2=sourceFilePath+"\\outfolder\\filename2"
	Output3=sourceFilePath+"\\outfolder\\filename3"
	arcpy.SetParameter(5, Output1)
	arcpy.SetParameter(6, Output2)
	arcpy.SetParameter(7, Output3)

3.5 Data Flow

Even if the models are comparable in terms of data being exchanged, it is rare that one model output identically matches the input format of the other model (Droppo et al., 2010). The data processing toolset of ArcGIS can be effectively used if data manipulation is required before passing an output from one model to the second.

SILO provides the household and employment numbers by zones, but CBLCM needs those numbers at the county level and distributed to urban/rural areas. To implement this efficiently, a special geoprocessing tool (SILO2CBLCM) has been developed (

Figure 1). It applies existing functionality provided through ArcGIS (Reclassification, Tabulate Area, Join, Calculate Field, Query, etc.) to count residential and commercial cell number ratios distributed on rural and urban areas; then, those are used with the zonal household and employment outputs from SILO to estimate number of households and number of jobs on urban and rural growth areas by counties. Specific Python scripts where used to format and export those estimations as demand tables required by CBLCM.

3.5 Integration in ArcGIS

The geoprocessing model developed in ArcGIS Model Builder consists of a set of modules, including SILO and CBLCM models linked through the SILO2CBLCM tool. Solid lines in Figure 2 represent data flow directions between modules, while the dotted lines represent the preconditions to run the modules. All module parameters have their default values, which can be changed using standard ArcGIS windows. The wrapper codes can be viewed or edited by the ‘Edit’ function available in the context menu. The actual model code is not accessible from here, though their file paths are defined in the wrapper and can be used to open the models in their specific environment (e.g. Microsoft Visual Studio, IntelliJ IDEA). Models are executed for a specific simulation year or period defined by the model’s parameters. Thus, the simulation presented in the Figure 2 starts with SILO modelling the land use changes for the years 2012-2030, followed by CBLCM modelling land cover for the year 2030.

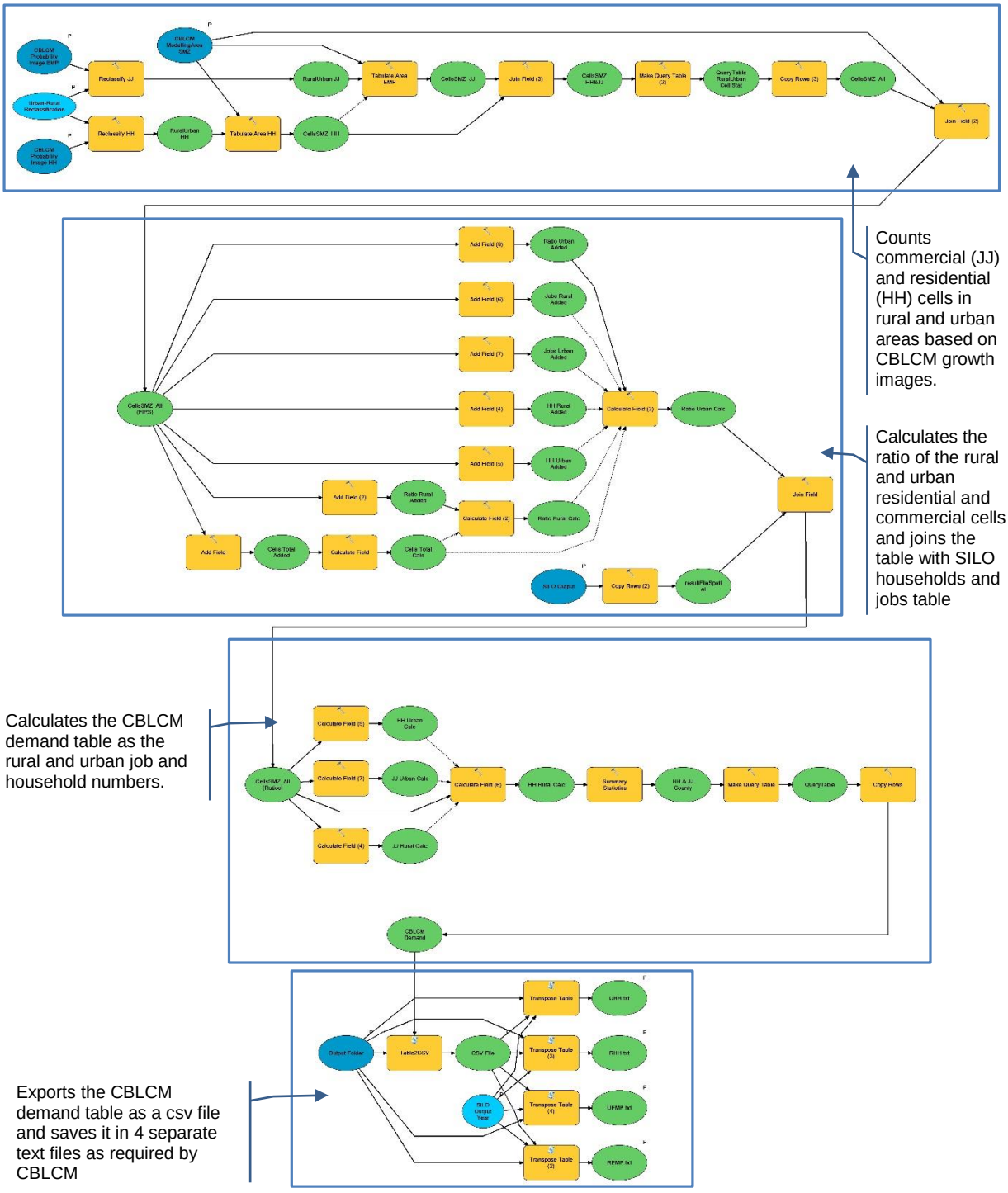

Figure 1. ArcGIS tool (SILO2CBLCM) to prepare SILO household and employment data for CBLCM.

4 RESULTS

With the use of Python wrappers, the implementation of the coupler is separated from the models' source codes. The loosely coupled geoprocessing model in ArcGIS Model Builder provides a user-friendly interface and uses a data flow diagram paradigm to organize models and information flow. This provides plenty of flexibility, which helps in terms of portability, performance and maintenance of the codes. While there are limitations, this approach supports different types of components and links them under a single user interface without changing their original source codes. The presented approach

allows adding a new model easily, requiring major revisions neither of the existing suite nor the model to be added. Key benefits and limitations of this approach are summarized below.

Benefits

- No need to change the source code of the models.
- Allows to run models developed in different programming languages.
- Can be extended with additional models over time.
- User interface to show process flows and linkages between the models.
- Easy tracking of model input and output files.
- Rich visualization, spatial analytic and mapping capabilities through integration with ArcGIS.
- Easy to implement with regard to required time, resources and programming experience.

Limitations

- Parallel model runs and dynamic data exchange during simulation time steps is not supported.
- Model processes run independently from one another.
- Data exchanged between modules need to be written to and read from a hard drive. No in-memory data exchange is available.

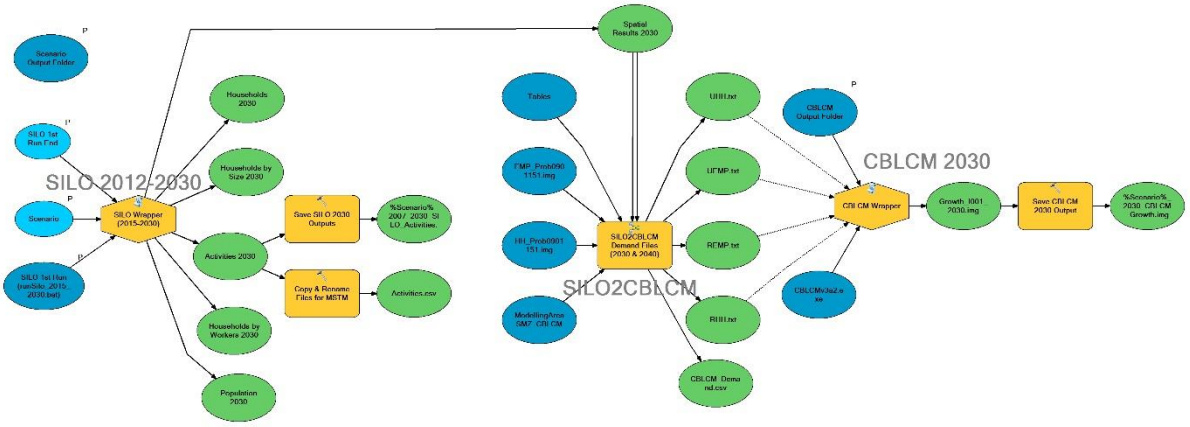

Figure 2. ArcGIS geoprocessing model to couple SILO and CBLCM.

5 CONCLUSION

The study automates the coupling of diverse models in a coherent and consistent manner without changing their source codes. Specifically, the method was applied to a microsimulation discrete-choice land use model and a pseudo-cellular automata land cover change model. Initial results of this linkage appear very promising. The models smoothly link in a form which does not require the user to track the models while running applications of these models. Instead, the user can more easily focus on the analysis and results, avoiding the need for a detailed understanding of model structure and data file systems.

Not only does the paper address the linkages, it can serve as a starting point for others interested in similar issues. The suggested methodology can be applied to other systems requiring consecutive implementation of standalone components including non-spatial models. The Python wrappers and ArcGIS models are publicly available at GitHub and can be used as templates for the integration of other models.

ACKNOWLEDGMENTS

This research was supported by a Marie Curie International Outgoing Fellowship within the 7th European Community Framework Program and by the National Socio-Environmental Synthesis Center (SESYNC) under funding received from the National Science Foundation DBI-1052875. The research benefited from many discussions with Uri Avin, Dan Engelbert, Sevgi Erdogan, Gerrit Knaap, Tim Welch and Di Yang.

REFERENCES

Argent R M, 2004, "An overview of model integration for environmental application - components, frameworks and semantics" *Environmental Modelling & Software* **19** 219-234

Aruoba S B, Fernandez-Villaverde J, 2015, "A comparison of programming languages in macroeconomics" *Journal of Economic Dynamics & Control* **58** 265-273

Babendreier J E, Castleton K J, 2005, "Investigating uncertainty and sensitivity in integrated, multimedia environmental models: tools for FRAMES-3MRA" *Environmental Modelling & Software* **20** 1043-1055

Basnyat P, Teeter L D, Lockaby B G, Flynn K M, 2000, "The use of remote sensing and GIS in watershed level analyses of non-point source pollution problems" *Forest Ecology and Management* **128** 65-73

Belete G F, Voinov A, Holst N, 2014, "An architecture for integration of multidisciplinary models", in *7th Intl. Congress on Env. Modelling and Software* (International Environmental Modelling and Software Society (iEMSs), San Diego, CA, USA)

Berry J, Buckley D, McGarigal K, 1997, "Seamlessly linking ARC/INFO to forest growth and landscape analysis models", in *ESRI User Conference*, San Diego, California

Brandmeyer J E, Karimi H A, 2000, "Coupling methodologies for environmental models" *Environmental Modelling & Software* **15** 479-488

Brazier R E, Heathwaite A L, Liu S, 2005, "Scaling issues relating to phosphorus transfer from land to water in agricultural catchments" *Journal of Hydrology* **301** 330–342

Clarke K C, Gaydos L J, 1998, "Loose-coupling a cellular automaton model and GIS: long-term urban growth prediction for San Francisco and Washington/Baltimore" *International Journal of Geographical Information Science* **12** 699-714

Devantier B A, Feldman A D, 1993, "Review of Gis Applications in Hydrologic Modeling" *Journal of Water Resources Planning and Management-Asce* **119** 246-261

Droppo J G, Whelan G, Tryby M E, Pelton M A, Taira R Y, Dorow K E, 2010, "Methods to Register Models and Input/Output Parameters for Integrated Modeling", in *International Congress on Environmental Modelling and Software*, International Environmental Modelling and Software Society (iEMSs), Ottawa, Canada

Goodchild M F, Egenhofer M J, Fegeas R, 1997, "Interoperating GISs", in *Report of a specialist meeting held under the auspices of the Varenius ProjectNCGIA* (Santa Barbara)

Gregersen J B, Gijsbers P J A, Westen S J P, 2007, "OpenMI: Open modelling interface" *Journal of Hydroinformatics* **9** 175-191

Hill C, DeLuca C, Balaji, Suarez M, da Silva A, 2004, "The architecture of the earth system modeling framework" *Computing in Science & Engineering* **6** 18-28

Homer C, Dewitz J, Yang L M, Jin S, Danielson P, Xian G, Coulston J, Herold N, Wickham J, Megown K, 2015, "Completion of the 2011 National Land Cover Database for the Conterminous United States - Representing a Decade of Land Cover Change Information" *Photogrammetric Engineering and Remote Sensing* **81** 345-354

Kittle J L, Duda P B, Ames D P, Kinerson R S, 2006, "The BASINS Watershed Analysis System - Integrating with Open Source GIS", in *Geographic Information Systems and Water Resources IV - AWRA Spring Specialty Conference* (AWRA, Houston, Texas)

León L F, Soulis E D, Kouwen N, Farquhar G J, 2002, "Modeling diffuse pollution with a distributed approach" *Water Science & Technology* **45** 149–156

Liu S, Brazier R E, Heathwaite A L, Liu W, 2014, "Fully integrated approach: an alternative solution of coupling a GIS and diffuse pollution models" *Frontiers of Environmental Science & Engineering* **8** 616-623

Moeckel R, forthcoming, "Constraints in household relocation: Modeling land-use/transport interactions that respect time and monetary budgets" *Journal of Transport and Land Use*

Moore R V, Tindall C I, 2005, "An overview of the open modelling interface and environment (the OpenMI)" *Environmental Science & Policy* **8** 279-286

Overeem I, Berlin M M, Syvitski J P M, 2013, "Strategies for integrated modeling: The community surface dynamics modeling system example" *Environmental Modelling & Software* **39** 314-321

Piacentini A, Morel T, Thevenin A, Duchaine F, 2011, "O-Palm: An Open Source Dynamic Parallel Coupler" *Computational Methods for Coupled Problems in Science and Engineering Iv* 885-895

Rahman J M, Cuddy S M, Watson F G R, 2004, "Tarsier and ICMS: two approaches to framework development" *Mathematics and Computers in Simulation* **64** 339-350

Roberts J J, Best B D, Dunn D C, Treml E A, Halpin P N, 2010, "Marine Geospatial Ecology Tools: An integrated framework for ecological geoprocessing with ArcGIS, Python, R, MATLAB, and C plus" *Environmental Modelling & Software* **25** 1197-1207

Schmitz O, Karssenberg D, van Deursen W P A, Wesseling C G, 2009, "Linking external components to a spatio-temporal modelling framework: Coupling MODFLOW and PCRaster" *Environmental Modelling & Software* **24** 1088-1099

Shah A R, Castleton K J, Hoopes B L, 2004, "Framework for risk analysis in multimedia environmental systems: Modeling individual steps of a risk analysis process" *Msv'04 & Amcs'04, Proceedings* 38-44

Sudicky E, Vanderkwaak J, Jones J, Keizer J, Mclaren R, Matanga G, 2003, "Fully-integrated modelling of surface and subsurface water flow and solute transport: Model overview and application" *Developments in Water Science* **50** 313-318

Tao C, Kainz W, van Zuidam R, 1996, "Coupling GIS and Environmental Modelling: The Implications for Spatio-Temporal Data Modelling" *International Archives of Photogrammertry and Remote Sensing* **31**

Valcke S, 2013, "The OASIS3 coupler: a European climate modelling community software" *Geoscientific Model Development* **6** 373-388

van Walsum P E V, Veldhuizen A A, 2011, "Integration of models using shared state variables: Implementation in the regional hydrologic modelling system SIMGRO" *Journal of Hydrology* **409** 363-370

Warner J C, Perlin N, Skyllingstad E D, 2008, "Using the Model Coupling Toolkit to couple earth system models" *Environmental Modelling & Software* **23** 1240-1249

Whelan G, Kim K, Pelton M A, Castleton K J, Laniak G F, Wolfe K, Parmar R, Babendreier J, Galvin M, 2014, "Design of a component-based integrated environmental modeling framework" *Environmental Modelling & Software* **55** 1-24

Wong I, Lam D, Booty W, Fong P, 2009, "A Loosely-Coupled Collaborative Integrated Environmental Modelling Framework", in *Americas Conference on Information Systems*, AIS Electronic Library