DATA STREAM MANAGEMENT IN INCOME TAX MICROSIMULATION MODELS

Molnár, István[a], Lipovszki, György[, b)]

[a] Department of Computer and Information Systems
School of Business
Bloomsburg University of Pennsylvania
Bloomsburg, Pennsylvania, 17815, U.S.A.

[b] Department of Mechatronics, Optics and Engineering Informatics
Faculty of Engineering
Budapest University of Technology and Economics
H-1111 Budapest, Muegyetem rkp. 3-9, Hungary

[a]Email: imolnar@bloomu.edu[b]Email: lipovszki@rit.bme.hu

ABSTRACT
The paper discusses the repetitive direct use of large data sets in microsimulation models. First, the income tax microsimulation models and the related data requirements are discussed. Next, the processing of the applied data set is introduced and the idea of using direct data instead of samples developed along with applied data stream management principles. The discussion of data stream management issues turns subsequently into an analysis of model control aspects of data driven simulation. Finally, some practical experiences and applied technologies are discussed.

Keywords: Data stream management, Income tax microsimulation, Microsimulation model development environment

1. INTRODUCTION

During the past 50 years, microsimulation research and practice focused on solving methodological problems closely related to mathematical modeling, mathematical statistics and economics. Less attention was devoted to model implementation, computational and application efficiency.

Recent developments show a change in focus of the efforts and recognition of the progress in the broad field of IT/IS, including use of system development environments, network oriented applications, collaborative system development and high performance computing. There is a significant stream of efforts in the field, which tries to integrate legacy systems with modern IT/IS, while using well recognized modeling and simulation (M&S) methodologies and tools.The change of obsolete world view and technology however is not fast enough and we still need to wait for a complete breakthrough.

This paper aims to contribute to these efforts and open new perspectives for microsimulation using M&S methodologies and tools widely used in engineering and sciences. The authors believe firmly that the ideas presented are powerful enough to be successfully applied in M&S of business and economic systems, including all application fields of microsimulation.

2. MICROSIMULATION MODELS' DATA REQUIREMENTS

2.1. Data sources and data analysis

One of the major components of microsimulation models (and therefore of the whole microsimulation modeling environment) are the data related to the micro units: *initial simulation model data*, *intermediate* and/or *final simulation data*. Initial data of microsimulation models are collected from *cross-sectional surveys*, and/or *longitudinal surveys*. Cross-sectional surveys collect data about a sample population for a single period of time (e.g., a survey of tourists' expenditures), while longitudinal surveys collect data about the same sample population (also called panel) for several periods of time (e.g., a household statistical survey). Microsimulation models could also use a *time-series of cross-sectional surveys*, which collect data at periodic intervals (e.g., micro-census). Special techniques (e.g., imputing, merging, synthetic data) have been developed to improve data quality and use additional data sources available. In addition, the microsimulation model consists of a series of *economic indicators* and diverse data, which are also stored for simulation as *variables and/ormodel parameters* and used for further analysis and computations.

The model behavior in microsimulation models is defined by *algorithms*, which, among others, reflect the micro units' behavior rules, related probabilities, furthermore, describe economic processes and represent their impact. By using this methodology, special care is taken to do the *data analysis* and the *estimation of microsimulation model parameters.*

The microsimulation model is working in an *experimental framework* in order to study the effects of

policy changes on the microsimulation model behavior. Given the model responses of micro-units at unit level, the microsimulation models can estimate aggregate effects and aggregate changes by grouping, creating distributions, tabulating or summing up unit-level individual model results in order to make statements about the various characteristics of the population as a whole, helping to determine "winners" and "losers".

The modeling process and numerical computation of simulation results are loaded with different kind of errors; e.g., the model is never a perfect representation of the system, parameter-estimations are rarely perfect, while the numerical computation might also contain round-off and/or method-related errors. *Validation* checks the modeling process and its final result, the conceptual model, and whether the model was able to represent the studied system in a satisfactory way. *Verification* checks the computer model, especially whether the computer model is executed properly and the results calculated correctly. It is the responsibility of the modeler to make a final decision about the model and the acceptance/rejection of the model results. Therefore, model verification and validation also use various sophisticated statistical methods and techniques, the results of which can be desirable to store, retrieve and analyze.

Taking the development and the application environment of microsimulation models into account, one can state that *data are collected, stored, retrieved and processed in a distributed way in different databases at different locations,* moreover, *they are maintained by different, mainly government authorities*. Most of the data are available in the form of different time series, in such a way that data content is hard to define, it might change with time and data integrity and accuracy are difficult to maintain. One of the biggest problems could be the management of the same data content under different names and different data content under the same name. Some of the simulation modeling problems are traditionally solved by using synthetic data sets (e.g., merging, imputing), which means that artificial data sources (and methods) are applied instead of traditional system data sampling. In addition to the "real system's" data, different types of simulation data are regularly stored and retrieved. These characteristics of microsimulation models' "messy" data sets are reflections of the characteristics of very large scale systems of the social sciences which are also very "messy".

2.2. Mathematical models used

The stored data of microsimulation models can be: initial model data, intermediate and final simulation data. These data are stored for further analysis. The description of model behavior in microsimulation models is based on algorithms, which describe the behavior of the micro units and their environment.

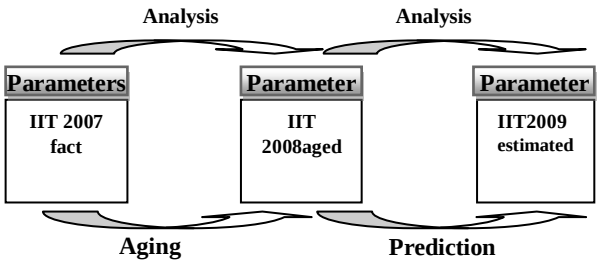

Figure 1: The Individual Income Tax simulation (IIT) process

The estimation of simulation model parameters, the general data analysis in order to determine and/or estimate behavioral rules are important steps of the modeling process. The microsimulation model is embedded in an experimental framework, which provides an environment to study the effects of policy changes on the microsimulation model behavior.

All available data are carefully analyzed using special techniques, which have been developed to improve microsimulation data quality (e.g., imputing, merging, synthetic data). For solving special model verification and validation problems, different methods and techniques were developed, which are not discussed further.

2.3. Computer and Information technology applied

The historically developed major model classes must maintain a significant amount of data and use methods for processing and analyzing these data. Data sources were historically integrated into microsimulation models in one of the following three different ways:

- File processing approach
- Database-oriented approach
- Agent-oriented approach

These approaches are not architecture neutral or network-oriented. Two tier databases are used at most. About a decade ago computer science started to develop new network-oriented technologies usable also for model-based applications, which support the use of heterogeneous hardware and/or software platforms. These network-based technologies became more widespread recently, when technical development allowed for networked multiplatform applications and beyond the networked data access also for distributed computing.

Despite the data-related problems, to the best of the authors' knowledge as of now there are no significant efforts to use standardized database content, data structure and access and data retrieval for microsimulation models. It is clear that even most recently published studies focus on platform dependent, non-portable microsimulation applications (e.g., PC-oriented or supercomputer-based) and provide related data management solutions. Little attention has been paid to the management of large data sets for distributed microsimulation databases and distributed applications,

moreover, portable and architecture neutral network-oriented technologies have been largely ignored (see Flory and Stöwhase 2011; Sutherland 2011).

3. NEW APPROACH TO MICROSIMULATION DATA MANAGEMENT

3.1. Requirements towards data stream management environments

Based on widely acknowledged simulation and information technology principles, a series of requirements were developed and implemented for the general microsimulation software development environment. These requirements help to establish a broad framework for different microsimulation applications:

- Platform-independent hardware and software solutions based on open standards.
- Data and network security.
- User friendliness, standardized user interfaces.
- Network-oriented data and model access using a database management system.
- Distributed model development, model execution and data analysis.
- Efficiency of software development during the whole software life cycle.

The implementation of the major principles listed, helped to manage the current difficulties of the microsimulation model development process. The implementation uses meta-database and Service-Oriented Technology (SOA), both of which are considered as key technologies for microsimulation. The technological solutions helped to expand the scope of existing research and to introduce two significant additional changes, which are going to be implemented:

- Use only data sources, which were selected based on the principle "straight from the reliable source".
- The overall efficiency of the microsimulation application can be best supported using role-based and workflow supported application.

3.2. Use of Object-oriented Data Modeling, Meta Databases and Data Warehouses

Large and complex data sets can be best managed using a database management system, which consists of crucial information about all individual microsimulation data available in the data system; i.e., a meta-database, which consists of data about microsimulation data. Based on the applied microsimulation models, the microsimulation data meta-database must reflect the supported microsimulation model classes and therefore is rather model-oriented (see Molnár and Sinka 2006).

The major advantage of using a meta-database for microsimulation data consists of, among others, increased data quality and overall cost-efficiency. Other criteria listed, like network-oriented data access and data analysis, as well as the efficiency of software development require different technologies, which are sometimes integrated with the core database management system (e.g., Oracle's Business Intelligence solutions support data warehousing, data analysis, data mining and report generation).

When using a meta-database for microsimulation data, all used data can be best referred to by using the appropriate data references of the meta-database. Because all methods, which can be executed on the microsimulation database data, can also be considered as data, it is evident that the most efficient solution to store the methods themselves is to store them in database(s), i.e., by creating appropriate method-database(s). Having method-databases requires the same data management practice as using a meta-database; i.e., a meta-database for microsimulation methods must be created.

The following information technology solutions are used:

- *Meta database*: a database of data about microsimulation data (how microsimulation data are collected/generated, accessed, processed, etc.).
- *Data warehouse*: data are stored in a special structure based on data-related dimensions (e.g., time, collected by, data content). Data are pre-processed before being stored (e.g., filter, extract, transform, classify, aggregate, summarize).
- *Object-oriented data modeling*: a data modeling paradigm, which applies object-oriented approach and data modeling.
- *Object oriented programming*: a programming paradigm.
- *Service Oriented Architecture*: applications implemented as Web-based components, which offer certain functionality to clients via the Internet.

Meta database and the related object-oriented database technologies help to manage the complex data sets and clean up the "messy" data.

3.3. Use of Service Oriented Architecture and Web services

SOA provides methods for systems development and integration in a standardized environment, where different software system functionalities are offered as independent and interoperable services. Web service architecture is built on open standards and vendor-neutral specifications and it is a way to implement a SOA.

This approach is a shift from a single application usage to a multi-component distributed application running on different platforms using open standard based network communication. Parallel to the authors' research and the implementation of the microsimulation software development environment, this technology became the foyer of cloud computing.

From the point of view of microsimulation software development, the major emphasis is on model development, use and re-use. Microsimulation algorithms can be developed using high level programming languages (e.g., Java); data analysis and

parameter estimation can be prepared by special mathematical and statistical software tools (e.g., SAS, SPSS). Both components are supported by Web services. Networked DBMS functionality can also be accessed using the high level programming languages provided by the Web service framework.

The Web service components have well-defined interfaces. Once deployed, they can be discovered, used/and reused by consumers (clients, other services or applications) as building blocks. Web service architecture is built on open standards and vendor-neutral specifications.

A significant advantage of this approach is that most elements of the technology are platform independent, and widely available (in part as free or open source software). Further advantages include but are not restricted to the following:

- Legacy systems can be integrated, existing codes re-used,
- Software development and maintenance costs, furthermore operational costs can be reduced,
- New business models can be established and new revenue can be generated, while interfaces with customers and integration with business partners can be improved.

The IT/IS industry is moving rapidly towards SOA and cloud computing in general, therefore the current concerns related to the use of Web services (e.g., there is no network performance guarantee, some standards are still missing) will soon disappear.

SOA is able to satisfy several application requirements listed, such as distributed data processing and model computations, security, platform independency and overall software development efficiency through the whole software life cycle. At the same time, SOA enables a move to cloud computing and the application of data-driven simulation techniques.

3.4. Enabling the use of data-driven simulation

Dynamic Data-Driven Application Simulation (D3AS) is a new paradigm, which enables the application (or simulation) and measurements to become a control system, in which data dynamically control almost all aspects of the long term simulation. (See Figure 2 based on Molnár and Sinka 2010, and NSF 2011.)

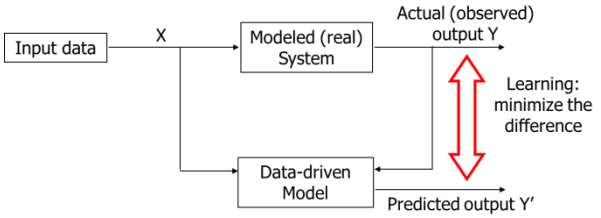

Figure 2: The Structure of Dynamic Data Driven Application

Traditional simulation models run many simulations using static data as initial condition. D3AS runs a very small number of simulations with additional data "injected" as they become available. Dynamic data are used to determine e.g., whether a "warm start" is needed, whether a rollback in time is required or whether the error of the simulation run is still acceptable. The D3AS approach to modeling focuses on the use of Machine Learning (ML) methods in building models to complement or replace "knowledge-driven" models, which describe the behavior of the system.

D3AS enables to create more accurate models of complex systems, which are adaptive to changing conditions and infer new knowledge (not determined by the initial state and parameters). Possible new application areas are business, engineering, and sciences (e.g., manufacturing process control, resource management, weather and climate prediction, traffic management, social and behavioral modeling).

4. HUNGARIAN INCOME TAX SIMULATOR

Elements of the presented methodology were implemented and the software applied for both a prototype model and the 2009 Hungarian Income Tax Simulator (HITS-2009). The HITS-2009 used real data to determine the possible fiscal impacts of the new linear income tax policy, planned to be introduced (see Molnár et al. 2009).

Based on the data source principles, anonymous 2008 income tax declaration data submitted to the Hungarian Tax Office (HTO) were used for the whole population and not the Household Statistical Panel (HSP) (i.e., no samples were applied, but about 4.5 million individual income tax declarations directly processed). The HITS-2009 is a classical tax policy simulator and as such, it is static, but able to predict the impact of different income tax policy changes, among others the fiscal impact of the generated income tax revenue component of the 2009 state budget.

In the same time, HITS possesses the capability to use updated data resources as a feedback control in a later point of simulation time and establish a major step towards conversion of the static microsimulation model into a dynamic one.

The HITS-2009 model could only have been assessed in June 2010, by which time the individual income tax declarations had already been processed. The authors of Molnár et al. 2009 found that model behavior was consistent with widely accepted results of the related economic theories. However, taking into account that the income tax model was created as a static microsimulation model (e.g., with no behavioral changes of the tax payers), model results must be interpreted and related to the reality with extreme caution. The standardized model output (e.g., additional statistics like the marginal tax rate) provides new opportunities for analysts to get more insights about income tax behavior.

The model results demonstrated clearly that both the methodology and technology are able to live up to the high expectations and result in increased model reliability and accuracy. To demonstrate the

standardized outputs of HITS-2009, Figure 3 and 4 are presented. The figures show the impact of different income tax rates to the budget and to the tax payers, respectively.

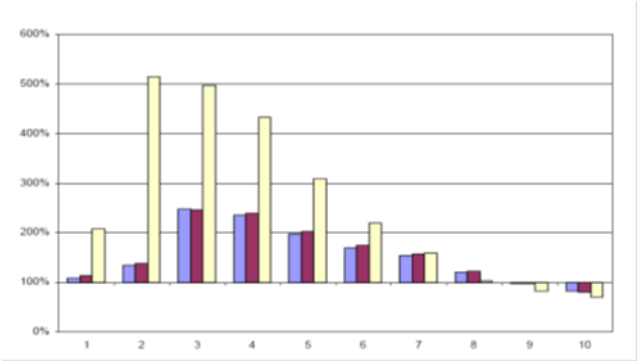

Figure 3: Average annual tax increase/decrease in different deciles in comparison to 2007

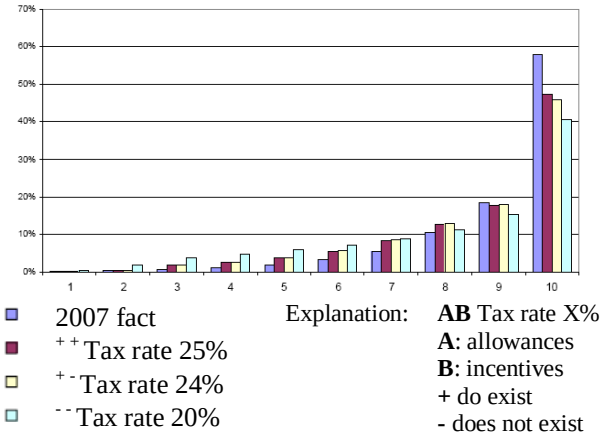

Figure 4: Contribution of income tax deciles to the budget (in case of different tax rates)

5. CONCLUSION

The paper presents a generally applicable methodology and a microsimulation model development environment to create and use data- and/or model-driven microsimulation models within a platform-independent, distributed and networked software environment. The basic idea of the solution is that the data and model complexity is best managed by a multidimensional distributed database, which makes extensive use of meta-database data, which are also implemented in the very same distributed environment using Web-services. In order to increase the flexibility of data storage and data processing, the relationships stored in databases are described using object-oriented data modeling and programming concepts; the computation engine uses the stored data and methods from the same database. The resulting microsimulation model development environment is able to support the implementation of data-driven and model-driven static and dynamic microsimulation models. The resulting architecture does not require but enables and supports the use of specialized agent-based modeling software.

Using the presented technologies, all relevant user requirements, listed as objectives, can be satisfied. The experiences with simple models showed results upon which one can build real word microsimulation applications. The HITS-2009 model demonstrated the strength of the applied methodology and technology under "real-world" circumstances.

The conversion of a traditional microsimulation model into web-based model applications enables the integration of legacy systems into the new, integrated information systems. The new technology based on SOA provides real solutions for distributed and secure data access, which is a major concern in most government applications, and also ensures distributed model development and analysis. SOA also paves the way toward collaborative model building and use.

The pilot project application of methodology and technology demonstrate the potential of regular use and can be included into microsimulation model bases, forming the first elements of a microsimulation-based Decision Support System. These systems could be extremely useful in scientific research (e.g., study of artificial societies) and state administration (e.g., tax microsimulation).

In order to extend the range of applications and provide wide availability of the microsimulation models, administrative regulation of data access must also be implemented. The use of distributed databases and meta databases in microsimulation models sends a strong signal to the authorities responsible for national statistical data and raises questions related to the efficient and regulated use of these data. The answers to these questions will determine the use of microsimulation models for national policy decision-making and with that the future of distributed state administration applications, which will certainly have a significant impact on any EU-wide regulations.

REFERENCES

Flory, J. and Stöwhase, S. 2011. MIKMOD-ESt - A Static Microsimulation for Model the Evaluation of Personal Income Taxation in Germany.*3rd General Conference of the International Microsimulation Association: Microsimulation and Policy Design*, Stockholm, Sweden. 1-19.

Molnár, I. and Sinka, I. 2010. State-of-the-art information technology for microsimulation. *International Journal of Technology, Modeling and Management (IJTMM)*. Serials Publications. 1 (1), 41-62.

Molnár, I. and Sinka, I. 2006. Model-oriented, data-driven architecture for microsimulation. *Simulation News, Europe, Journal on Developments and Trends in Modeling and Simulation.* ISSN: 0929-2268, 16 (3), 37-40.

Molnár, I. ,Bátori, D., BelyóP., Sinka, I., Szathmáry, B.and Tóth,Cs. G. 2009. Adó mikroszimuláció (Egyes adónemek mikroszimulációja), ECOSTAT Gazdaságelemző és Informatikai Intézet, Időszaki Közlemények 37.szám, Budapest, ISBN: 978 963 88329 6 2, ISSN: 1418 7892., 1-59.

NSF (National Science Foundation). 2011. *DDDAS: Dynamic Data Driven Applications Systems*. Available from: http://www.nsf.gov/cise/cns/ /dddas/ [Accessed 06.06.2011]

Sutherland,H. 2011. Address: "EUROMOD: the state of play and a vision for the future" 3rd General Conference of the International Microsimulation Association: Microsimulation and Policy Design, Stockholm, Sweden.

AUTHORS BIOGRAPHY

István Molnár

Educated at the Corvinus University Budapest, Hungary, where he received his M.Sc. and Dr. Oec.(Ph.D.) degrees.He completed his postdoctoral studies in Darmstadt, Germany and took part in different research projects in Germany and Western Europe as Guest Scientist in the 1980s and 1990s. In 1996, he received his C.Sc. (Ph.D.) degree from the Hungarian Academy of Sciences. His main fields of interest are simulation, simulation optimization, software technology and education. He has been a member of different international scientific organizations and editorial boards of scientific journals, publishing houses. Currently, he is the Editor in Chief of IJTMM and a member of the Editorial Board of IJMLO.

György Lipovszki

Born in Miskolc, Hungary and finished his study at Budapest University of Technology and Economics, where he was graduated in 1975 in electronics sciences. Currently, he is an Associate Professor at the Department of Mechatronics, Optics and Engineering Informatics. His research field is the development of modeling and simulation framework systems in different programming environments. He is a member of the Editorial Board of IJSTL and IJTMM.