Assessment of IT solutions used in the Hungarian income tax microsimulation system

To cite this article: I. Molnar and S. Hardhienata 2017 IOP Conf. Ser.: Mater. Sci. Eng. 166 012010

View the article online for updates and enhancements.
Assessment of IT solutions used in the Hungarian income tax microsimulation system

I. Molnar\textsuperscript{1} and S. Hardhienata\textsuperscript{2}

\textsuperscript{1}Retired Faculty, 4990 Sentinel Dr., Bethesda, MD, USA.
\textsuperscript{2}Department of Computer Science and Doctoral Program of Pakuan University, INDONESIA.

E-mails: molnar01@aol.com, s-hardh@indo.net.id

Abstract. This paper focuses on the use of information technology (IT) in diverse microsimulation studies and presents state-of-the-art solutions in the traditional application field of personal income tax simulation. The aim of the paper is to promote solutions, which can improve the efficiency and quality of microsimulation model implementation, assess their applicability and help to shift attention from microsimulation model implementation and data analysis towards experiment design and model use. First, the authors shortly discuss the relevant characteristics of the microsimulation application field and the managerial decision-making problem. After examination of the salient problems, advanced IT solutions, such as meta-database and service-oriented architecture are presented. The authors show how selected technologies can be applied to support both data- and behavior-driven and even agent-based personal income tax microsimulation model development. Finally, examples are presented and references made to the Hungarian Income Tax Simulator (HITS) models and their results. The paper concludes with a summary of the IT assessment and application-related author remarks dedicated to an Indonesian Income Tax Microsimulation Model.

1. Introduction

Microsimulation is a method that is able to handle the high complexity of the socioeconomic problems by creating and studying a model that makes intensive use of the statistical data of the observed objects, also called \textit{micro-units} of the socioeconomic system, e.g., person, family or household. The microsimulation models use simulation techniques in order to study the behavior of micro level units. To receive the best possible set of performance indicators of the socioeconomic system, the microsimulation model can be optimized using multi-objective nonlinear optimization algorithms. Microsimulation models can be integrated in e-government environments and provide support for decision making for government administrators.

Microsimulation models have the following main elements: simulation data, simulation model, simulation parameters and simulation results. The initial model data, intermediate and/or final simulation result data, are stored for further analysis. The simulation model consists of algorithms, which describe the behavior of micro-units and represent their environment. Simulating the micro-units’ state and behavior (the changes of state over time) and the change of micro-units as a result of policy changes is often called “aging the data”. Static aging refers to the adjustment by re-weighting the used sample based on some aggregate control variable (e.g., the composition of the sample by gender), while dynamic aging refers to adjusting each micro-unit attribute (e.g., income)
by recalculating them at each time period, one period at a time. Special care is taken to do the data analysis and the estimation of simulation model parameters, the latter of which are stored together with the simulation model. Given the model responses of micro-units at unit level, 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. The microsimulation model is working in an experimental framework in order to study the effects of policy changes on all micro-units, and on the microsimulation model behavior itself. Model calculations are executed by a digital computer.

Microsimulation models are generally accepted and frequently used to prepare political decisions in Australia (e.g., [7], [15], [25]), Europe (e.g., [3], [4], [10], [11], [12], [14], [28]), Canada and the USA (e.g., [2], [5], [9]. Economies in transition also use microsimulation models to study problems in demography, pension systems, healthcare and taxation.

In this paper the authors pursue an IT-oriented approach and assess the impact of new technology in enabling a wider spectrum of microsimulation model applications worldwide. The technology was developed during the past decade along with the applications (e.g., HITS) and it is essential to summarize all the lessons learned during the process. As will be clearly demonstrated using the example of the HITS, the presented technologies are able to satisfy all relevant user requirements. Moreover, by using this state-of-the-art IT, government policy decisions are transparently connected with the use of underlying microsimulation models and their implementation. With this, the foundation for the microsimulation model integration into an e-government application framework is established. The general applicability of the technology makes it easy to implement national income tax microsimulation models; a project plan for creating an Indonesian Income Tax Microsimulation System is shortly presented.

2. Personal Income Tax Microsimulation
2.1 Macro- and Micro-economic Considerations
Tax is a financial charge or levy imposed by a state or administrative authority on a taxpayer to finance various public financial obligations. Taxes are enforced by the tax authorities and tax evasion is punished by law. When state expenditures (e.g., expenditures on economic infrastructure, education, health care system, military, scientific research, culture, public works and government operations) exceed tax revenues, the state accumulates debt.

The tax system is also used by the government to redistribute resources among individuals or groups of individuals. According to the definitions created by the Organization for Economic Cooperation and Development (OECD), income tax is imposed on income of individuals or business entities. Individual or personal income tax is determined on the basis of the taxable income according to the tax law. Tax rates may vary (e.g., progressive or regressive) or are constant (flat).

Progressive tax is considered to be reducing economic inequality. The re-distributive property of the tax system, especially that of the personal income tax (along with the different type of property taxes), plays an important role in controlling long term inequality. Recent results of Th. Piketty [23] attracted additional research efforts and as a consequence, extended personal income tax models (e.g., tax-benefit models) are being introduced, with which an “optimal” set of policy control parameters (e.g., a construct of tax rates and tax benefits) should be determined (see [10] and [28]).

The Indonesian economy uses a progressive income tax system (see [29] and [30]) and has a vital interest to establish a tax system, which satisfies related economic and social requirements, while supporting and motivating the economic growth and social transition of the country. The provided IT solutions enable the creation and study of complex personal income tax microsimulation models and the quick adaptation of the technology itself. The present paper does
not discuss in further details the macro- and micro-economic issues related to the income tax modeling.

2.2 Data Collection, Storage, Retrieval and Analysis

One of the major components of microsimulation models (and therefore of the whole microsimulation modelling environment) are the micro units’ data: these are initial simulation model data, as well as intermediate and/or final simulation data. Initial data of microsimulation models are usually collected from cross-sectional and/or longitudinal surveys. Cross-sectional surveys collect data about a sample population for a single period of time (e.g., a survey of tourist 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 (see e.g., [16], [17], [24], [26]). In addition, the microsimulation model consists of a series of economic indicators and model parameters, which are also stored for simulation and used for further analysis and computations.

The model behavior in microsimulation models is defined by algorithms, which, among others, reflect the micro units’ behavioral rules, describe economic processes and represent their impact. By using this methodology, special care is taken to perform data analysis and estimate the 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. 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 model results in order to make statements about the various characteristics of the population as a whole and helping to determine “winners” and “losers” (see e.g., [14]).

The modelling process and numerical computation of simulation results can be riddled 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 rounding and/or method-related errors. Validation checks the modelling 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 whether it is/was executed properly and whether the results were 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 (see [16]).

Taking the development and the application environment of microsimulation models into account, one can say that data are collected, stored, retrieved and processed in a distributed way in different databases at different locations, moreover, they are maintained by various, 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 practical 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 modelling 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 social science systems, which are also very “messy” (see [8]).

Despite the data-related problems, to the best of the author’s 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 most of the recently published studies focus on platform dependent, non-portable microsimulation applications (e.g., PC-oriented or supercomputer-based) and provide related data management solutions (see e.g., [1] and [28]). Little attention has been
paid to the management of large data sets for distributed microsimulation databases and distributed applications. Portable and architecture neutral network-oriented technologies have been largely ignored. The authors are convinced that the development of networked multi-platform microsimulation applications is nowadays not just necessary but also technically possible (see [18], [19] and [27]).

2.3 Model classes and general software requirements

The authors call data systems “model-oriented”, when those were designed and used to support applications based on a class of mathematical models. In most cases presented in the literature, data systems have been designed and used for a specific microsimulation model. Because data represented in the microsimulation modelling environment originate in different organizations (Central Statistical Office, Central Bank, Ministry of Finance, etc.) and are generated by different sources (i.e., measured, synthetic, simulated data), the simultaneous use of data by simulation models can be considered a special type of data-driven application (see [27]).

Behavior-driven models, as opposed to data-driven models, reflect the behavioral patterns of micro-units and the impact of policy change on them. Behavior driven models are frequently realized using agent-based modelling (e.g., [20]). In agent-based models, an initial configuration of multiple interacting agents is provided and then the model is driven by agent interactions, which are observed over time without further intervention. Similar to ASPEN (see [1]), several model objects can act as an agent (e.g., individuals, governments, markets, firms, banks, etc.). Based on the agents’ characteristics, agents can be grouped, thus creating agent classes.

Despite the different modelling approaches, both model classes, data-driven and behavior-driven models, must handle model data and methods in the same way. In both cases large amount of data must be analyzed and processed in a way that model independence and data independence are provided. Therefore, general IT and a series of particular microsimulation software development environment requirements are applied. 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.

2.4 Applied IT Solutions

Some of the listed requirements can be realized by using Service Oriented Architectures (SOA) (e.g., web-service technologies), as suggested in [19]. The large and complex data systems 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 meta-database of microsimulation data must reflect the supported microsimulation model classes. The major advantage of using a meta-database for microsimulation data consist of, among others, increased data quality and overall cost-efficiency. The other listed criteria, 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 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 a database(s), i.e., by creating appropriate method-database(s). Having method-databases requires the same data management practice as using a meta-database of data; i.e., a meta-database for microsimulation methods must be created.
The model-oriented, data-driven microsimulation architecture, presented in [27], uses the following IT solutions:

- **Meta-database**, 
- **Data warehouse**, 
- **Object-oriented data modelling (OODM)** see [13], 
- **Object oriented programming (OOP)** see [6], 
- **Service Oriented Architecture (SOA)** see [19] and [20].

The authors consider the SOA and the meta-database as *key technologies* for microsimulation modelling; therefore these technologies will be discussed subsequently. Meta-database and the related object-oriented database technologies help to manage the complex data sets related to microsimulation applications. Service-oriented architectures are able to satisfy several application requirements, such as distributed data processing and model computations, security, platform independence and overall software development efficiency through the whole software life cycle.

2.5 Use of Microsimulation Meta-Database

The author considers a fundamental principle of the microsimulation application and development environment that two types of meta-databases are to be developed, one for the microsimulation data and another for the microsimulation model methods (see also in [27]). The advantage of this solution is that large amount of “messy” data can be managed comfortably along with the microsimulation model development results accumulated over a long period of time and substantiated in different model versions.

2.5.1 Meta-database for microsimulation data

The meta-database stores data about each microsimulation data into four groups:

- **Data content**: used for identification and usability decisions. Among others, the following data are stored: unique id, mnemonic, description, validation, version, usability role, and accessibility.
- **Data access**: access-related data are stored in specific order, depending on the storage medium, the location, the type of access (e.g., local or networked), the applied protocol (e.g., http, ftp, soap), etc.
- **Data processing**: most of the data are used for further calculations; therefore, algorithms and/or methods used for processing these data must be identified. The full specification of each algorithm and/or method is given as data content in two different ways: using direct statements (all processing steps of the algorithm are described directly, there are no loosely coupled method calls) and using indirect statements (the algorithm consists of loosely coupled method calls; e.g., Java RMI, Microsoft COM).
- **Other data**: technical data might also be stored e.g., action log, access log, error log.

The architecture with extended meta-database for microsimulation data is presented in Figure 1.

---

**Figure 1.** The architecture with extended meta-database.
2.5.2 Meta-database for microsimulation methods

A meta-database for microsimulation methods can be considered as a separate meta-database. In this case all meta-database data groups, except the data processing, can be used for the method definition. (In special cases, the meta-database for microsimulation methods can be directly integrated into the data processing data group of the meta-database for microsimulation data). The architecture with extended meta-database for microsimulation methods is presented in Figure 2.

![Figure 2. The architecture with extended meta-database for microsimulation methods.](image)

2.6 Other software components

As presented above, the meta-databases for microsimulation are characterized by networked remote access and therefore both methods and data are accessed using information stored in meta-databases. Methods can produce both data-driven and behavior-driven models and therefore, agent-based modelling can be realized by not requiring separate software components. If special agent-based modelling tools or software components must be used, they can be integrated into this environment and invoked upon request. Microsimulation systems described here can be realized using the following major software components (see Figure 3).

![Figure 3. Major Software Components](image)

- **RunTime System (RTS):** runs all software components. The JBoss Application Server has been used, which is an Open Source J2EE implementation.
- **FrameWork System (FWS):** synchronizes the components’ run, working as the centre of the architecture communicating with the components. Implemented in J2EE.
- **Remote Execution System (RES):** runs all loosely coupled methods (for both remote data access and remote method execution). Implemented in J2EE, uses PHP running in Apache WebServer.
- **Local Execution System (LES):** runs local methods. Implemented in J2EE.
- **Messaging System (MES):** manages communication between/among the components. The JBoss AS JMS has been used.
- **Data Base Access System (DBAS):** provides access to all local databases (incl. meta-databases). EIS Connection Pool, which is implemented in the Application Server.
- **Data Base Management System (DBMS):** manages all local databases (incl. meta-databases). MySQL Server 5 open source RDBMS has been used.

Using the carefully selected software components, the implementation of RTS, DBAS, DBMS and MES is a simple task of parameter setting. The implementation of FWS, LES, and RES, furthermore the population of the microsimulation meta-database, including the programming of agent-based microsimulation methods, require significant efforts.

All components of the HITS (incl. the operating system) were implemented as open source software. The extended LAMP ((Linux, Apache, MySQL, PHP), and JBoss provide robustness, scalability, reliability and data security. The applied web solutions and technologies also provide great organizational flexibility by separating developer competencies, which allows much higher life-cycle efficiency.

### 2.7 Data- and behavior-driven microsimulation models

The development of both data-driven and behavior-driven microsimulation models consists of two phases, which can be realized to a great extent independently from each other. In phase one the software system is implemented, while in phase two, the meta-database(s) and database(s) are populated. Both phases require different team compositions and professional background.

The HITS model implementation uses standard technologies, which are easy to adapt and modify, especially taking in consideration that the source code is fully available under GNU license. The Indonesian Income Tax Microsimulation System can build around these solid foundations; extensions and modifications are easy to implement. The HITS model scenarios were implemented using the multi-tier architecture:

- **Client tier** contains the microsimulation model and the client stub, which invokes the tax income forecast web service using SOAP on HTTP.
- **Application tier** contains the Oracle HTTP Server (actually an Apache), which receives request and sends response, furthermore the MySQL Application Server with MySQL Container for J2EE to store the web service, which performs the model calculations for income tax forecasting.
- **Information tier** runs MySQL database, which includes the microsimulation model data.

The software elements of the presented architecture were selected based on cost efficiency: Open Source, Free or Public Domain software have been used. The elements of the software architecture can be replaced by any other software-component of the same functionality, but additional resources are needed to ensure a smooth transition.

### 2.8 Assessment of the microsimulation model development environment

The authors consider the SOA and the meta-database as key technologies for the microsimulation modelling; therefore these technologies will be discussed below. Meta-database and the related object-oriented database technologies help to manage the complex data sets related to microsimulation applications. Service-oriented architectures are able to satisfy several application requirements, such as distributed data processing and model computations, security, platform independence and overall software development efficiency through the whole software life cycle (see [19] and [20]).

One of the most important features of the architecture and especially the meta-database concept is that there is a great flexibility in using various data-sets for the microsimulation models provided. Any databases (e.g., longitudinal, cross-sectional, survey-related, semi-synthetic, or synthetic) can be selected and used along with its interface to the simulation model. The choice assumes that the appropriate microsimulation model will also be developed (or has already been developed). In
order to show this flexibility, instead of household statistical survey data, HITS scenarios use anonymous national tax office data (about 4.5 million personal income tax records were processed). Because all data (not just sample data of a given sample size) are related to the tax paying population, a higher statistical accuracy is anticipated. A consequence of this is that a better quality and more accurate aging process is anticipated. These factors increase the quality of the modelling process (model parameters and global variables are estimated using the original [full] population) and the data quality, as well. The authors think that these quality improvements are significant. Later developments, especially some contributions of the IMA 2015 provide a positive feedback showing an increased use of administrative data-sets and the acceptance of publicly-accessible microsimulation models.

The interfaces developed for the databases make it possible to use a series of data analysis tools, like SAP, SPSS, R, or other individual tools. Separate databases can be defined also for macroeconomic data or for data analysis results (using the data warehouse concept). When taxpayers’ data are analyzed, individual and full population-related data are collected and stored. The “aging” process uses these data using the ‘component-based income prediction’ method. Predicting future personal income, the main components of the personal income is to be determined by using previously defined (calculating the parameters and variables) individual and group-population data. The personal tax is calculated and is always based on the current tax regulations applied to the individual and the individual’s calculated (predicted) income. Statistical methods are applied to accumulate statistics based e.g., on income group (location, sex, or any other individual-related data property), in order to create useful indicators for political decision makers. The applied methodology separates data sets and simulation models efficiently, while providing sufficient information for decision makers, including also the national income tax budget component.

In order to illustrate the basic principles, the HITS scenarios base-line model and flat income tax microsimulation will be presented based on [21]. Other scenarios are discussed in detail in [20] and [27]. The tax contribution of income deciles to the national budget (in case of different flat tax rate scenarios) are presented in Figure 4.

![Figure 4](image-url)

**Figure 4.** The tax contribution of income

3. Conclusion
The paper and the references present a generally applicable methodology and a microsimulation model development environment to create and use 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 managed by a multidimensional distributed database, which makes extensive use of meta-database data, which are also included 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 modelling 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 behavior-driven static and dynamic microsimulation models. The resulting architecture does not require, but allows and supports the use of specialized agent-based modelling software.

Using the presented technologies, all relevant user requirements, listed as objectives, can be satisfied. The experiences with models of different sizes show results, on which one can build microsimulation applications used for administrative purposes. However, in case of large and complex microsimulation models, users need to handle a long sequence of data definitions and method calls, which might be a source of different errors. As a solution, the development of a modelling language or role-based further software developments might be appropriate.

With the increasing use of model-based policy decisions and the increased professional interest in microsimulation, the authors sincerely believe that the current practice of political decision-making will be radically changed by including a phase of impact analysis of policy changes before the decision is made. The importance of new technologies, especially the use of IT therefore cannot be overestimated. It is suggested to employ technological solutions which provide long-term, high quality and flexible solutions and use the available resources in the most efficient way. Advantages of scientific research and development can only be fully utilized if the political decision-makers and practical economists are aware of the new research results on the one hand, and the researchers are aware of the needs of practical decision makers, on the other hand. The technological solutions provided will redirect the attention to the simulation modelling of socio-economic systems and supporting the use of model results; the knowledge produced during the modelling process is substantiated in the final product, which could form the basis of a knowledge-based policy decision.

The conversion of a traditional microsimulation model into a web-based model application makes it possible to integrate legacy systems into 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. A rational final stage of the development is the integration of the microsimulation model-development environment into an e-government environment.

The HITS model applications underline the importance of a real world environment 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 for scientific research and state administration alike. With the separate management of data and methods, independent alternative model solutions can be developed parallel and compared in an easy way.

Unlike in the case of engineering models, socioeconomic models are not developed and used within the framework of a rational, often standardized process, but rather according to the policy makers’ decisions, which again are often missing the deep scientific arguments. In order to extend the range of applications and provide wide availability of the microsimulation models, a change of policy-makers’ world view, as well as changes in administrative regulations of data access must be implemented. In order to support this process, applied IT solutions will have to be made available in the near future.

The first considerations and results in the adaptation of the HITS to the Indonesian Income Tax System is an ongoing effort, which will be converted into a project in the near future. Beyond the assessment of ICT solutions used in the HITS, the research team will investigate the requirements of the Indonesian Income Tax Microsimulation System, the applicability of the use of a new non-linear optimization algorithm to determine optimal tax strategies, furthermore, determine the prerequisites of the practical use of the Indonesian Income Tax Microsimulation System and the international dissemination of its results.
References

[1] N. Basu, R.J. Pryor, T. Quint T. Arnold, ASPEN: A Microsimulation Model of the Economy (Sandia Report, Sandia National Laboratories, SAND96-2459 UC905, Albuquerque, New Mexico, 1996).
[2] W.W. Beach, R.A. Rector, R.S. Hederman, A.B. Goyburu, T. Kane The Candidates’ Tax Plans: Comparing the Economic and Fiscal Effects of the Bush and Kerry Tax Proposals. (The Heritage Center for Data Analysis, Washington, D.C., 2004).
[3] P. Belyó, I. Molnar, Introduction of Capital Income Taxation: a Microsimulation Study, Simulationstechnique 18th Symposium in Erlangen (Ed.: F. Huelsemann, M. Kowarschik, and U. Ruede) Forschritte in der Simulationstechnik FS15, ISBN 3-936150-41-9, SCS European Publishing House e. V., Erlangen, 2005, 688-693.
[4] P. Belyó, I. Molnar, Use of Microsimulation Models in Political Decision Making, Public Finance Quarterly, LI/3, HUISSN 0031-496-X, Budapest, Hungary, 353-365.
[5] L.E. Burman, The Individual Alternative Minimum Tax, President’s Advisory Panel on Federal Tax Reform, (The Urban Institute, The Tax Policy Center, Washington D.C., 2005)
[6] P. Coad, J. Nicola, Object –oriented Programming (New Jersey, Prentice Hall, second edition, 1993).
[7] J. Creedy, G. Kalb. Behavioural Microsimulation Modelling for Tax Policy, Analysis in Australia: Experience and Prospects (The University of Melbourne, Melbourne Institute Working Paper No. 2/05, ISSN 1328-4991 (Print),ISSN 1447-5863 (Online), 2005).
[8] P. Davidsson, Agent Based Social Simulation: A Computer Science View, Journal of Artificial Societies and Social Simulation 5(1), 2002, (http://jasss.soc.surrey.ac.uk /5/1/7.html), last visited: 4/30/2016).
[9] Ch. Edwards, A Proposal for a “Dual-Rate Income Tax” (Cato Institute, Tax & budget bulletin, No. 22, Washington D.C., 2005).
[10] O. Lelkes, Tax-Benefit Microsimulation Models In Eastern Europe, International Journal of Microsimulation, 1(1), 2007, 54-56.
[11] J. Flory, S. Stöwhase, MIKMOD-EST: A Static Microsimulation Model of Personal Income Taxation in Germany, International Journal of Microsimulation, 5(2), 2012, 66-73.
[12] W.G. Gale, J. Rohaly, Effects of Tax Simplification Options on Equity, Efficiency, and Simplicity: A Quantitative Analysis (Urban Institute, Brookings Institution, Tax Policy Center, Washington, D.C., 2002).
[13] P. Gray, K.G. Kulkarni. N.W. Paton, Object-Oriented Databases A Semantic Data Model Approach (New York, Prentice-Hall, 1992).
[14] P. Haan, V. Steiner, Distributional and Fiscal Effects of the German Tax Reform 2000. A Behavioral Microsimulation Analysis (German Institute for Economic Research, Discussion Papers #419, Berlin, 2004)
[15] A. Harding, N. Warren, An introduction to microsimulation models of tax reform (NATSEM. National Centre for Social and Economic Modelling, University of Canberra, 1998).
[16] J. Li, C. O’Donoghue, A survey of dynamic microsimulation models: uses, model structure and methodology. (International Journal of Microsimulation, 6(2), 2013, 3-55).
[17] R.J.A. Little, D.B. Rubin, Statistical analysis with missing data (New York, Wiley-Interscience, 2002)
[18] I. Molnar, Gy. Lipovszki, Data Stream Management in Income Tax Microsimulation Models, Proceedings of The 23rd European Modeling & Simulation Symposium (EMSS2011), Rome, Italy, 2011, 394-399.
[19] I. Molnar, Microsimulation Model Development Environments International Journal of Simulation Systems, Science and Technology, 6(5), ISSN:1473-804x [Online], ISSN: 1473-8031 [Print], 2005, 33-41.
[20] I. Molnar, I. Sinka, Toward Agent-Based Microsimulation -- Another Approach Proceedings of the First Asia international Conference on Modelling & Simulation. IEEE Computer Society, Washington, DC, 2007, 403-408.
[21] I. Molnar, I. Sinka, P. Belyo, Cs.G. Toth, D. Bathori, Hungarian individual income tax microsimulation. Proceedings of the seventh International Conference on Simulation in Industry and Services, Brussels, Belgium, 2009, 187-211.

[22] G. Orcutt, M. Greenberger, J. Korbel, and A. Rivlin, Microanalysis of socioeconomic systems: a simulation study (Harper & Brothers, New York, 1961).

[23] Th. Piketty, Capital in the twenty first century (The Belknap Press of Harvard University Press, Cambridge, Massachusetts, 2014).

[24] D.B. Rubin, Multiple Imputation for Nonresponse in Surveys (New York, John Wiley & Sons, 2004).

[25] R. Scutella, M. Harris, J. Creedy, A. Duncan, Microsimulation modelling of taxation and the labour market: the Melbourne Institute tax and transfer simulator (Cheltenham, United Kingdom, Edward Elgar Publishing, 2002).

[26] D. Schofield, J. Polette, A comparison of data merging methodologies for extending a microsimulation model (NATSEM STINMOD Technical Paper No. 11. National Centre for Social and Economic Modelling, University of Canberra, 1998).

[27] I. Sinka, I. Molnar 2006 “Modell-oriented Data-driven Architectures for Microsimulation” (Simulationstechnique 19th Symposium in Hannover. In: ASIM Forschritte in der Simulationstechnik (Ed.: M. Becker, H. Szczerbicka), SCS Publishing House e. V., ISBN 3-936150-49-4, Erlangen, 2006. 35-41).

[28] H. Sutherland, Final report: EOROMOD: an integrated European benefit-tax model (EUROMOD Working Paper No. EM9/01, 2001).

[29] - : Taxation and Investment in Indonesia 2014: Reach, Relevance, Reliability, Indonesian Highlight 2016, A Publication of Deloitte Touche Tohmatsu Limited, 2016.

[30] - : Indonesian Pocket Tax Book 2015, PWC Indonesia, www.pwc.com/id, 2015.