Comparative Analysis of Technical Frameworks and Tools Supporting Decision Making

Karim Hamza[1] and Sehl Mellouli[2]

[1]Free University Brussels, Belgium, [2]University Laval, Canada

Abstract

Policy makers are the persons who take decisions for the well-being of their communities. In order to take good decisions, technical frameworks support policy makers in developing models that help explain phenomena of the policy context and therewith support the decision making process. This chapter provides an overview of different tools and technologies to develop these models. This chapter does not claim to be exhaustive by identifying all tools and technologies, but can serve as a basis for any developer who intends to develop a simulation model for policy-makers.

1. Introduction

To support policy making, public policy scholars and policy scientists have developed scientific theories, frameworks, conceptual and simulation models as well as tools to better understand policy and policy-making processes (see e.g., Clark, 2002; Sabatier, 1991; Sabatier, 2007, McCool, 1995; Kraft & Furlong, 2007; Smith & Larimer, 2009; Birkland, 2010; Sabatier, 2007).

A Framework refers to concepts to support structured and systematic analysis, design, implementation and assessment/evaluation of a solution. Specifically, a framework identifies elements, identifies the relationships between these elements, and provides a general set of variables that can be used to analyse the proposed solution. On the other hand, models are defined as a set of practical supports provided to policy-makers in order to take their decisions (Ostrom, 2011). Models are more precise than frameworks since a model uses specific assumptions about a limited set of variables, identified in the frameworks, to derive precise solutions when combining these variables (Ostrom, 2011). For more details on policy models, and in particular on examples of distinct policy models, the reader is referred to the white paper on comparing simulation models (see (Majstorovic, et al., 2014) in Annex II.3 to D4.2).

Each developed model uses a set of tools. These tools vary from implementation tools like programming languages, to the graphical user interfaces, to the adopted technologies such as multi-agent systems. This chapter will focus on tools by categorizing the different tools used to develop models and frameworks. This chapter will serve as a basis for any developer who will implement models for policy-makers to know the different used tools and their categories. Developers will have better insights to choose their tools. Models studied are described in Annex II.3 to D4.2, see (Majstorovic, et al., 2014).

The chapter is organised as follows. The next section describes and categorizes the different tools and technologies used to develop the models. Subsequently, Discussion and Conclusion reflects these tools and technologies.

2. Tools and Technologies

The aim of this section is to determine the different tools and technologies that supported the development or the use of the models described earlier. The tools and technologies are grouped into 4 categories: social network analysis, simulation tools, and GUI modelling. As already stressed in the introduction the collection here is a first analysis and is not complete. Further categories and tools will be analysed to complement the current overview.

2.1. Social network analysis

The first set of tools and technologies is related to Social network analysis. The main objective of Social Network Analysis (SNA) is to study the structure of social relationships in a group in order to cover all formal connections between people(Hanneman and Riddle 2001). These relationships can be grouped into four categories: communication, awareness, trust, and decision-making. SNA can be used by organisations as a tool to better understand the connections between their employees (Tichy, Tushman et al. 1979). It can help organisations evaluate the interactions and business outcomes of their employees such as job performance, job satisfaction, adoption of new ideas or technologies, or the creation of new ideas. In other words, SNA allows mapping and measuring formal and informal relationships between people to better understand the knowledge flows that bind interacting units, who knows whom, and who shares what information and knowledge with whom by what communication media.

As described earlier, there are different tools for social network analysis that have been used such as Gephi, Ucinet, or Nodexl. Gephi is an open source network exploration and manipulation software(Scott and Carrington 2011). Its goal is to help data analysts to make hypothesis about relationships, intuitively discover new communication patterns, or isolate structure singularities or faults during data sourcing. The network is represented as a graph that the user of Gephi has to explore. It is an interactive visualization and exploration platform for all kinds of networks and complex systems.

The second tool is Ucinet (Borgatti and Chase 2006). It is a Windows based application. It allows it users to explore relationships between people by calculating different measures such as centrality, cohesion, or brokerage. It includes also a module for testing hypothesis. Ucinet is part of a complete package that contains three basic programs: Ucinet, Spreadsheet and NetDraw. Each program fulfils different and complementary roles:

- Ucinet, as stated, is a program that calculates the indicators of social networks' analysis.
- Spreadsheet is a worksheet that allows to capture relational data in the form of adjacent or attribute matrixes. This tool is used prior to the calculation of indicators and graphic analysis.

- NetDraw is a graphic program with the purpose to visualize social networks. It allows observing the different actors of a network in the form of a 2 dimensional graph.

Finally, NodeXL is a free, open-source template that runs for Microsoft Excel 2007 and 2010 (Smith, Shneiderman et al. 2009). Its objective is to make it easy to explore network graphs. NodeXL generates a graph from a network that its edges were entered in a worksheet. NodeXL combines the power of excel with new and easy-to-use interactive network visualisation and analysis tool, The tool supports multiple social network data providers that import graph data (nodes and edge lists) into the Excel spreadsheet.

2.2. Simulation tools

Agent based modelling simulation tools are powerful tools to simulate the impact of adopted policies. There are mainly two tools: Repast, Netlogo, JAMSIM, and AnyLogic.

Repast is an open source library that provides a collection of tools and structures that are useful for simulation (North, Collier et al. 2006). It is a set of classes for creating, running, displaying and collecting data from agent based simulations. It is created by the University of Chicago's Social Science Research Computing division. The Kosice example as presented in (Majstorovic, et al., 2014) uses DRAMS – a Declarative Rule-based Agent Modelling system (Lotzmann and Meyer 2011) – as a simulation engine. DRAMS is based on Repast and supports declarative agent modelling.

NetLogo is a multi-agent based tool to develop simulations of natural and social phenomena (Sklar, 2011). The tool was developed by Uri Wilensky in 1999 and has been in continuous development ever since at the Center for Connected Learning and Computer-Based Modeling. NetLogo is particularly well suited for modelling complex systems. It has the potential to make operating independently hundreds or thousands of "agents" in order to explore the connections between the micro-level behaviour of individuals and the macro-level patterns that emerge from their interactions. The SKIN example as analysed in (Majstorovic, et al., 2014) (see Annex II.3 to D4.2) uses NetLogo for the simulation.

Finally, AnyLogic is a multi-agent simulation tool that is Process-centric and that supports the methodology of System Dynamics (Kirkwood 1998). It provides powerful tools to capture the complexity and heterogeneity of business, economic and social systems to any desired level of detail. The VirSim (Fasth et al. 2010) example as analysed in (Majstorovic, et al., 2014) (see Annex II.3 to D4.2) uses AnyLogic as a tool for the simulation.

JAMSIM is a framework for creating micro-simulation models in Java (Manion et al., 2012). It consists of ASCAPE, JAVA and R. Micro-simulations are used to make social simulations. It provides code and packages for common features of micro-simulation models for end users. It is a means to model real life events by simulating the actions of the individual units that make up the system where the events occur (Brown and Harding 2002), and as : computer-simulation of a society in which the population is represented by a large sample of its individual members and their behaviours (Spielauer 2011). JAMSIM models run as standalone programs with a friendly GUI without requiring technical expertise. MEL-C is described as an example of micro-simulation in (Majstorovic, et al., 2014).

2.3. Tools for GUI modelling

The third family of tools is the Graphical User Modelling tools. Any simulation model provides a graphical user interface. The tools that will be presented are: Eclipse, Netbeans, and GUI design studio:

- Eclipse is a multi-language Integrated development environment (IDE) comprising a base workspace and an extensible plug-in system for customizing the environment (Jensen 1989). It supports different programming languages such as Java, C++, C#, or Php. It is a WSWG tool that help developers develop their applications.
- NetBeans IDE is provided to Java developers (Boudreau, Glick et al. 2002). Its main characteristic is that it supports the newest Java technologies and latest Java enhancements before other IDEs. It allows developers to design their graphical user interfaces quickly and smoothly by dragging and positioning GUI components from a palette into the NetBeans Editor.
- GUI Design Studio (White and Almezen 2000) is a specialised software design tool for anyone involved in application user interface design, including User Experience Designers, Business Analysts, Developers, Project Managers and Consultants.

3. Discussion and Conclusion

This chapter provided a general overview of five different models that were implemented to support policy-makers in their decisions and the related used technologies to develop these models. The technologies were grouped into three categories: social network analysis tools, agent based simulation tools, and graphical user interface modelling tools.

This chapter doesn't claim to be exhaustive by identifying all models and technologies, but can serve as a basis for any developer who intends to develop a simulation tool for policy-makers. The chapter presented seven models, and grouped the tools and technologies into three families.

From this overview, we can come-up with the three following insights:

1. The models are not related to a single technology or tool. To this end, a developer has to make choices to see for example if he's going to adopt multi-agent technology or if he's going to develop its application in Java. To take this decision, the developer needs more investigation in comparisons between technologies or between tools.
2. The multi-agent technology is a predominant technology that has been used in the majority of the proposed models. However, this doesn't claim that it is the best technology for any solution to be developed. Each developed solution has its own characteristic and its own context. Developers must be aware of these realities when choosing their technology.
3. The presented models are generally supported by frameworks. A Framework refers to concepts in systems design to support structured and systematic analysis, design, implementation and assessment/evaluation. Hence, developers should investigate the different existing frameworks to see the one that will support their model. The framework can impact the tools and technologies that can be adopted since it defines different rules of analysis, design, implementation, and assessment/evaluation.

The development of a model requires at least three skills. First, the developer should be aware about the process that policy-makers follow in order to take their decisions. Second, the developer has to be aware of the different existing tools and technologies that can be used to help policy-makers. Finally, software engineering techniques and tools must be applied in order to manage the development of these models.

This chapter is a starting point for future research directions. First, policy models are generally based on policy frameworks. There are different policy frameworks that existing in the literature such as (Whitt 2003, Hussain and Hotel 2004). However, the move from the frameworks to the models is not clear. Hence, it becomes very important to identify methodological links to move from frameworks and models.

Second, models have to be evaluated. However, to this end, training data has to be collected in order to validate the results provided by the models and the real outcomes. Hence, the techniques to be adopted for data collection become an important challenge to the development of models.

Finally, agent based simulation tools require that specific design methodologies and languages can be used. As seen in the different presented models, even if the agent technology is adopted, but there is no reference to any specific design methodology or an agent programming language has been made. Consequently, it becomes very important to identify the methodology that can be adopted to develop the agent based models, such methodologies can be Gaia, FATMAS, etc. In addition, developing languages have to be specified, such languages can be JADE, Agent toolkit, etc.

4. References

Ahrweiler, P., Pyka, A., Gilbert, N. (2004), Simulating Knowledge Dynamics in Innovation Networks (SKIN), in: R. Leombruni, und M. Richiardi (eds.), The Agent-Based Computational Approach, World Scientific Press, Singapore, 284-296.

Borgatti, S. and R. Chase (2006). "UCINET: social network analysis software." Analytic Technologies--Social Network Analysis & Cultural Domain Analysis.

Boudreau, T., et al. (2002). NetBeans: the definitive guide, O'Reilly Media, Inc.

Brown, L. and A. Harding (2002). "Social modelling and public policy: application of microsimulation modelling in Australia." Journal of Artificial Societies and Social Simulation **5**(4).

Fasth T., Ihlar M., Brouwers L. (2010) VirSim – A Model to Support Pandemic Policy Making, PLOS Currents Influenza. 2010 Sep 22 [last modified: 2012 Mar 29]. Edition 1. doi: 10.1371/currents.RRN1181

Hanneman, R. A. and M. Riddle (2001). "Social network analysis." Riverside: University of California.

Hussain, A. and M. Hotel (2004). Peace, Poverty and Economic Growth in South Asia: A Policy Framwork. Institute of Regional Studies Conference in Islamabad. June.

Jensen, M. C. (1989). "Eclipse of the public corporation." Harvard Business Review **67**(5): 61-74.

Kirkwood, C. W. (1998). "System Dynamics Methods." College of Business Arizona State University USA.

Lotzmann, U. and R. Meyer (2011). DRAMS–a declarative rule-based agent modelling system. Proceedings of 25th European Conference on Modelling and Simulation.

Majstorovic, Dragana; Wimmer, A. Maria; Lay-Yee, Roy; Davis, Peter, Ahrweiler, Petra and Deljoo, Ameneh (2014). Comparative Analysis of Simulation Models. Annex II.3 In Synthesis Report of Knowledge Assests, including Visions.

Mannion O, Lay-Yee R, Wrapson W, Davis P, Pearson J. JAMSIM: A microsimulation modelling policy tool. Journal of Artificial Societies and Social Simulation 2012, Vol. 15 No. 1. http://jasss.soc.surrey.ac.uk/15/1/8.html

Milne, B. J., et al. (2014). "A collaborative approach to bridging the research-policy gap through the development of policy advice software." Evidence & Policy: A Journal of Research, Debate and Practice **10**(1): 127-136.

North, M. J., et al. (2006). "Experiences creating three implementations of the repast agent modeling toolkit." ACM Transactions on Modeling and Computer Simulation (TOMACS) **16**(1): 1-25.

Scott, J. and P. J. Carrington (2011). The SAGE handbook of social network analysis, SAGE publications.

Shariff, M. (2007). Alfresco enterprise content management implementation, Packt Publishing Ltd.

Sklar, E. (2011). NetLogo, a multi-agent simulation environment. Artificial Life, Vol. 13 No. 3, pp. 303-311.

Smith, M. A., et al. (2009). Analyzing (social media) networks with NodeXL. Proceedings of the fourth international conference on Communities and technologies, ACM.

Spielauer, M. (2011). "What is social science microsimulation?" Social Science Computer Review **29**(1): 9-20.

Tichy, N. M., et al. (1979). "Social network analysis for organizations." Academy of management review **4**(4): 507-519.

UrbanSim. (2014). http://en.wikipedia.org/wiki/UrbanSim. Visited February 18, 2014.

Waddell, P. (2002). UrbanSim: Modeling Urban Development for Land Use, Transportation and Environmental Planning. Journal of the American Planning Association, Vol. 68 No. 3, pp. 297-314.

White, L. and H. Almezen (2000). Generating test cases for GUI responsibilities using complete interaction sequences. Software Reliability Engineering, 2000. ISSRE 2000. Proceedings. 11th International Symposium on, IEEE.

Whitt, R. S. (2003). "Horizontal Leap Forward: Formulating a New Communications Public Policy Framwork Based on the Network Layers Model, A." Fed. Comm. LJ **56**: 587.