

A user interface for large-scale demographic simulation

Cristina Montañola-Sales, Josep Casanovas-Garcia
Departament d'Estadística i Investigació Operativa
Universitat Politècnica de Catalunya – BarcelonaTech
Barcelona Supercomputing Center
Barcelona, Spain
cristina.montanola@upc.edu, josepk@fib.upc.edu

Bhakti S. S. Onggo
Management Science Department
Lancaster University
Lancaster, UK
s.onggo@lancs.ac.uk

Zengxiang Li
Institute of High Performance Computing
Singapore
liz@ihpc.a-star.edu.sg

Abstract— Agent-based modeling is one of the promising modeling tools that can be used in the study of population dynamics. Two of the main obstacles hindering the use of agent-based simulation in practice are its scalability when the analysis requires large-scale models as in policy research, and its ease-of-use especially for users with no programming experience. While there has been a significant work on the scalability issue, ease-of-use aspect has not been addressed in the same intensity. This paper presents a graphical user interface designed for a simulation tool which allows modelers with no programming background to specify agent-based demographic models and run them on parallel environments. The interface eases the definition of models to describe individual and group dynamics processes with both qualitative and quantitative data. The main advantage is to allow users to transparently run the models on high performance computing infrastructures.

Keywords—agent-based modeling; demographic simulation; parallel simulation;

I. INTRODUCTION

Agent-based modeling (ABM) is a useful approach to deep on the understanding of population dynamics through simulation. The main reason is that the object of study in these disciplines, human society present or past, is difficult to analyze through classical analytical techniques due to the unpredictable and changing (dynamic) nature. ABM has already been applied to demographic simulation [1]. ABM is commonly used for small scenarios because the number of agents and interactions between them can be extremely large in some of case studies, thus forcing the scientist to limit its number in order to execute the simulation in a standard computer. However, complex policy models may require a significant amount of computing power. Parallel simulation techniques might pay an important role in the future of social simulation, supporting the management of large simulations.

Currently, there are many tools to support the development and execution of generic ABM. However,

typically desktop ABM tools do not scale well to what is required for extremely large applications. Indeed, the performance analysis is a difficult task in parallel and distributed simulation [2], especially when the system is as dynamic as human populations. Scalability needs to be addressed, though there is no consensus on dealing with the difficulties it encounters on agent-based models [3], [4].

We have developed a tool that is designed to allow demographic ABM be run in parallel environments. To make our tool accessible to experts interested in population projections, we have designed a graphical user interface (GUI). This interface allows modelers to specify demographic ABMs and to transparently run the models on a parallel computing infrastructure. The rest of the paper is organized as follows. Section 2 discusses Human Computer Interfaces (HCI) for simulation. Section 3 summarizes the architecture of the tool. We present the GUI design in section 4. Finally, section 5 presents our concluding.

II. HCI FOR SIMULATION

From the point of view of users, the interface of a simulation tool is the tool itself [5], since it is the part of the system with which they interact. In complex systems easy-to-use interfaces are especially important since models are difficult enough for non-experts to understand [6]. HCI simplifies the development of models and relieve social scientists of self-development simulation features [7].

In [8] issues that affect usability of simulation systems are examined. Characteristics such as user degree of computer literacy and knowledge or domain knowledge should be taken into account in the interface design [9]. A good interface should be easy to use, efficient, supportive and satisfy user needs [10]. Among the different kinds, web-based user interfaces for simulation take advantage of a browser to support the interaction of simulation graphical interfaces with users. Indeed web-based solutions not only serve as an operating system but also as a distribution channel for use

[11]. Integrating the Web with the field of simulation has several advantages, such as the ease-of-use, wide availability, cross-platform capability, collaboration features and control access [12]. However, it encounters also some disadvantages, including loss in speed, graphical interface limitations, security vulnerability, application stability and license restrictions. Despite these drawbacks, the current tendency is to increase the development of web user [13], [14]. However, areas such as user interface ease-of-use should be further research [15], particularly in the area of simulation where the number of works in this direction is scarce.

III. YADES ARCHITECTURE

To approach agent-based techniques to modelers interested in population projections, we developed a tool (called Yades) which simulates the demographic evolution and interactions of individuals in a society [16]. It provides the placeholders for different demographic processes such as fertility, mortality, change in economic status, change in marital status, and migration with two different types of agents: family unit and regions. Yades has three components: a web user interface, a demographic simulation library and the simulation code generator. The web user interface is designed so that modelers can build their demographic ABMs for a set of group dynamics processes. The simulation code generator can produce the corresponding C++ code that is linked to the demographic simulation library which uses a scalable parallel discrete-event simulation engine. The generated code is ready for compilation using a target C++ compiler. The overall framework is shown in Fig. 1.

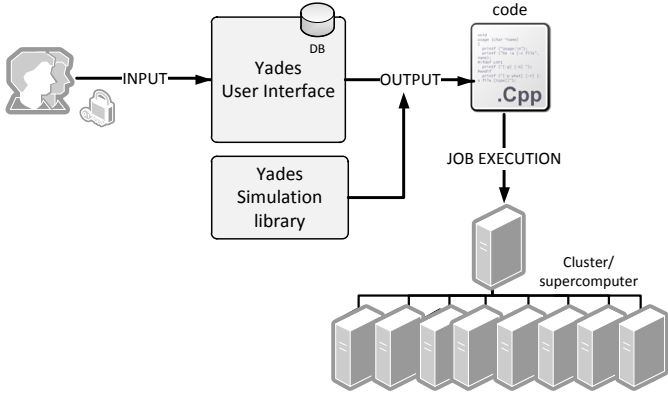


Fig. 1. Yades modeling and simulation framework

There are two types of agents in Yades: family unit and regions. A family unit is defined as a single independent individual or two independent individuals living together (as married, in civil-partnership, or in cohabitation) and any dependent individuals (children). A family unit may receive events that are related to five demographic components that may change the system states. Modelers can specify models for five demographic components: fertility, a change in economic status, a change in marital status, migration and mortality. The second agent type represents a region where a number of families live. This agent will handle domestic migrations, immigration, changes in simulation parameters

and periodic reports. Yades allows users to have regions with different population characteristics. Fig. 2 shows how the demographic scenario is mapped on a parallel architecture.

Yades simulation library is implemented using μ sik parallel simulation library [17]. μ sik is a parallel discrete-event simulation library that supports multiple synchronization algorithms such as look ahead-based conservative protocol and rollback-based optimistic protocol. This library adopts the process interaction world-view in which a simulation model is formed by a set of interacting (logical) processes. Logical processes (LPs) communicate through events with the standardized communications protocol Message Passing Interface (MPI). Multiple LPs can be mapped onto a physical process that is run on top of a processing element (PE). A machine can have more than one PE (e.g., in multi-core architecture). Yades uses LP to implement agents and μ sik kernel for simulator instances, using an optimistic synchronization with a state-saving mechanism and a time window of 12 months. The detailed implementation has been reported in [16] and [18].

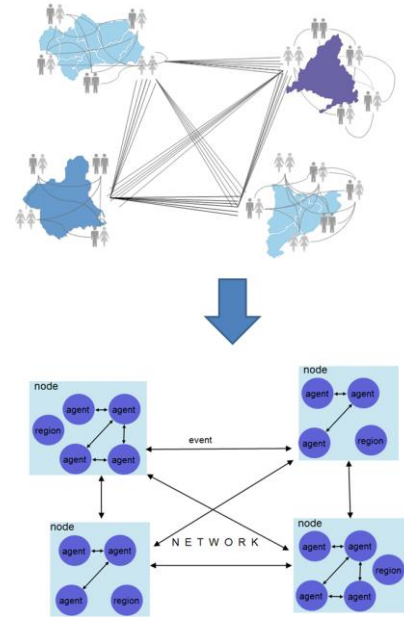


Fig. 2. From a demographic model to the agent-based simulation model

We now present a web user interface aim to support the development of agent-based models with Yades simulation library which is accessible at <http://yades.fib.upc.edu>.

IV. GUI DESIGN

In this section, we explain the GUI design to build agent-based demographic simulation models with Yades. We have followed a participatory design with the anthropologist who co-authors this paper and who is interested in the use of ABS. Participatory design is especially positive when working on complex systems [9]. The discussion suggested that we need to provide four types of interfaces to allow modelers specify the behavior of individuals in relation to the fertility, mortality, marital status, economic status and migration. They are state-transition diagram, multiple regression, logic rules, and

standard theoretical distributions. The GUI is implemented in PHP and we used HTML5, CSS and JavaScript for the front-end layer. We will now introduce the need of a GUI for a demographic ABS and describe GUI components involved.

A. The need of a specialized

To confront the difficulty that social scientists and policy modelers are not familiar with computer programs/codes, we need a specialized GUI which could allow users to easily model different behaviors using familiar methods such as regression or intuitive diagrams such as a state-transition diagram. In that way, we ensure the software design answers the needs and capabilities of the users for whom they are intended [19].

Social modelers often use statistical estimation techniques to derive into agent decision rules and behaviors. These rules are simple models that relate certain situations with some actions. Key variables that govern agent behaviors have to be identified and statistical relationships have to be obtained [20]. To summarize, the requirements for the design of the GUI are as follows.

- To design a simple user interface for users who are not familiar with programming and parallel computer environment
- To exploit the potential power offered by parallel computers transparently
- To provide different types of input data: theoretical distributions, multiple regression, logical rules and state-transition diagram.
- To produce/show outputs at micro and macro levels

The GUI presented in this paper follows a user-centered design approach which involves users throughout the design and development process [19]. To achieve this, we include a user during the requirement analysis and design. The crucial part, however, is to also involve the user in the evaluation of the system during the whole design and development process.

B. GUI components

The demographic ABM is made up of five agent behaviors that shape the population dynamics: fertility, mortality, marital status, economic status and migrations. In this subsection, we will explain the GUI design for each part. Before running the simulation, modelers will also need to specify the initial population and simulation settings. The interface features each area can have diverse initial populations. Fig. 3 shows the internal structure and components of the GUI.

1) *Initial population specification*: The user interface allows users to specify typical family structures in the region. It includes information on the population distribution by gender and age groups, distribution of different types of family units (parents with/without children, single parents, single individuals, etc.), the age distribution of children in the family, the distribution of different economic statuses, and the distribution of different marital statuses. The GUI allows the information to be entered manually or to import it from a CSV file.

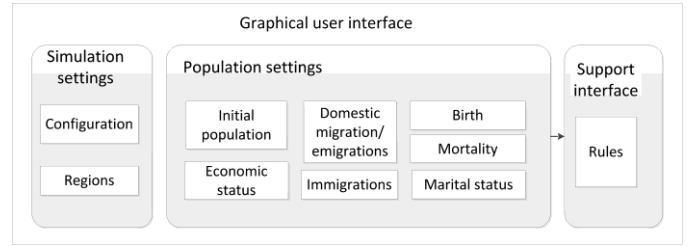


Fig. 3. Internal components of the GUI

2) *Fertility*: Common fertility models to describe the propensity of the women to bear children include age-specific fertility, parity-specific fertility, birth spacing and their combinations. Age-specific fertility uses age to determine the probability of having a child. Parity-specific fertility takes into account the number of children a female person has already had. The birth spacing model focuses more on the time between each birth. This model is rather complicated because it includes factors such as birth control, abstinence period, economic status, etc. The user interface is designed to allow users to specify these models. They will need to indicate the fertility age interval and a model to generate the time to birth. Most of these models can be specified using a theoretical distribution function, a multiple regression function or logical rules based on individual characteristics.

3) *Mortality*: The most commonly used mortality models are the life table (and its variants) and survival analysis. The life table summarizes the variation of mortality by age and gender. Survival analysis uses a distribution function to sample an individual's lifetime at birth. The user interface allows users to specify these two commonly used models through entering the life table manually or importing it from a CSV file.

4) *Marital status*: In demography, we recognize a number of marital statuses such as single, married and divorced. The GUI is designed to allow users to specify the transition in a state-transition diagram, a graphical tool which helps on defining the multi-state modeling, a standard methodology in demography [21]. Allowing multi-state models enable users to represent different transition models, previously inferred. The difference between marriage and cohabitation has shown to be significant in terms of understanding households dynamics [22]. Users will then need to specify each transition and the time to the next state using distribution function, multiple regression or logical rules. Modelers will need to specify a "match maker" function [23], which is based in logic rules according to age, gender, economic status.

5) *Economic status*: The ability to track an individual's economic status is included in the model because there is a plan to build a policy model on top of it (for public policy planning). As well as marital status, economic status component is essential for this purpose. The GUI for the change in economic status is very similar to marital status model but in this case we do not need the match maker function.

6) *Migrations*: Modelers need to specify two types of migrations: domestic migration and international migration (either emigration or immigration). The user interface allows modelers to specify a migration model that determines whether a family unit is going to migrate. Mobility is specified using a

constant probability, logit regression or logical rules. For those who migrate, the destination area is determined by a migration matrix. Immigrant populations are likely to have different demographic characteristics to the populations they join [24]. For this reason, users need to specify the number of immigrants per month and their characteristics.

C. Yades GUI code generator

The tool is meant to be run either in a sequential or parallel environment. However, access to a parallel environment is usually restrictive. Moreover, we cannot produce an executable of the simulation because it depends on the architecture is going to be used. Therefore, we provide users the simulation code resulting from their specification of the model. In that way, users with access to a cluster or supercomputer can compile and execute the code. While we free users from the coding process of the model we cannot release them from managing a job in a High Performance Computing environment. Other ABM tools have similar functioning in leaving the management of job's execution to users, though there have been attempts to automatically generate parallel agent-based simulation code [25].

V. CONCLUSIONS

We have presented a user interface for our agent-based demographic simulation tool which is available for public and we expect to have more feedback from users who would use this tool for group analysis. We believe the two of the main issues in the wider adoption of parallel simulation are scalability and ease-of-use. This tool tackles the scalability problem in large scale and complex agent-based models by running the models on top of a parallel discrete-event simulation engine. The ease-of-use issue is tackled by providing an interface that allows modelers to describe personal behavior, such as fertility and change in marital status. We believe that this will allow users to concentrate on understanding the modeling process rather than in the simulation library it is being used. We believe this will help to remove a major barrier on using simulation although we are aware technical knowledge is necessary to execute scenarios in High Performance Computing facilities.

ACKNOWLEDGMENT

This research was supported by a PhD mobility grant from the Research and Innovation Ministry of the Spanish Government and by the Royal Society International Joint Project 2009/R2 grant number JP090402. We want to thank Jordi Gassó to support the development of the GUI and inLab FIB at UPC-BarcelonaTech for hosting it.

REFERENCES

- [1] F. C. Billari, F. Ongaro, and A. Prskawetz, *Agent-Based Computational Demography: Using Simulation to Improve Our Understanding of Demographic Behaviour*, vol. IX. Elsevier, 2003, p. 210.
- [2] S. L. Ferenci, K. S. Perumalla, and R. M. Fujimoto, "An approach for federating parallel simulators," in *Proceedings of the Fourteenth Workshop on Parallel and Distributed Simulation, PADS*, 2000, pp. 63–70.
- [3] M. Hybinette, E. Kraemer, Y. Xiong, G. Matthews, and J. Ahmed, "SASSY: A Design for a Scalable Agent- Based Simulation System Using a Distributed Discrete Event Infrastructure," in *Proceedings of the 36th Conference on Winter Simulation (Monterey, California)*, 2006, pp. 926–933.
- [4] L. Tesfatsion, "Agent-based computational economics: Growing economies from the bottom up," *Artif. Life*, vol. 8, no. 1, pp. 55–82, 2002.
- [5] D. Hix and H. R. Hartson, *Developing user interfaces: ensuring usability through product & process*. John Wiley & Sons, Inc., 1993.
- [6] J. Saw and M. Butler, "Exploring graphical user interfaces and interaction strategies in simulations," in *Hello! Where are you in the landscape of educational technology?* *Proceedings ascilite Melbourne* 2008.
- [7] R. Tobias and C. Hofmann, "Evaluation of free Java-libraries for social-scientific agent based simulation," *J. Artif. Soc. Soc. Simul.*, vol. 7, no. 1, 2004.
- [8] J. Kuljis, "HCI and simulation packages," in *Proceedings of the 28th conference on Winter simulation*, 1996, pp. 687–694.
- [9] B. Shneiderman and C. Plaisant, *Designing The User Interface: Strategies for Effective Human-Computer Interaction*, 5th ed. Pearson Education India, 2010.
- [10] A. Smith, *Human computer factors*. McGraw-Hill, 1997.
- [11] J. Kuljis and R. J. Paul, "Web-based discrete event simulation models: Current states and possible futures," *Simul. Gaming*, vol. 34, no. 1, pp. 39–53, 2003.
- [12] J. Byrne, C. Heavey, and P. J. Byrne, "A review of Web-based simulation and supporting tools," *Simul. Model. Pract. Theory*, vol. 18, no. 3, pp. 253–276, 2010.
- [13] B. Myers, S. E. Hudson, and R. Pausch, "Past, present, and future of user interface software tools," *ACM Trans. Comput. Interact.*, vol. 7, no. 1, pp. 3–28, 2000.
- [14] T. O'reilly, "What is Web 2.0: Design patterns and business models for the next generation of software," *Commun. Strateg.*, no. 1, p. 17, 2007.
- [15] S. Murugesan, Y. Deshpande, S. Hansen, and A. Ginige, "Web engineering: A new discipline for development of Web-based systems," *Lect. notes Comput. Sci.*, pp. 3–13, 2001.
- [16] B. S. S. Onggo, "Parallel discrete-event simulation of population dynamics," in *Proceedings of the 40th Conference on Winter Simulation*, 2008, pp. 1047–1054.
- [17] K. S. Perumalla, "μsik: A Micro-Kernel for Parallel/Distributed Simulation Systems," in *Proceedings of the 19th Workshop on Principles of Advanced and Distributed Simulation*, 2005, pp. 59–68.
- [18] B. Onggo, C. Montañola-Sales, and J. Casanovas-Garcia, "Performance Analysis of Parallel Demographic Simulation," in *Proceedings of the 24th European Simulation and Modelling Conference*, 25-27 October 2010, pp. 142–148, 2010.
- [19] D. L. Stone and D. Stone, *User interface design and evaluation*. Morgan Kaufmann, 2005.
- [20] C. M. Macal and M. J. North, "Tutorial on agent-based modeling and simulation," in *Proceedings of the 37th Conference on Winter Simulation*, 2005, pp. 2–15.
- [21] F. Willekens, "Biographic forecasting: Bridging the micro-macro gap in population forecasting," *New Zeal. Popul. Rev.*, vol. 31, no. 1, pp. 77–124, 2005.
- [22] D. de Vaus, "Diversity and change in Australian families: Statistical Profiles," *Melbourne, Aust. Aust. Inst. Fam. Stud.*, 2004.
- [23] F. C. Billari, A. Prskawetz, B. A. Diaz, and T. Fent, "The 'Wedding-Ring': An agent-based marriage model based on social interaction," *Demogr. Res.*, vol. 17, no. 3, pp. 59–82, 2007.
- [24] W. Haug, P. A. Compton, and Y. Courbage, *The Demographic Characteristics of Immigrant Populations*, no. 38. Council of Europe Publishing, 2002.
- [25] P. Richmond, D. Walker, S. Coakley, and D. Romano, "High performance cellular level agent-based simulation with FLAME for the GPU," *Brief. Bioinform.*, vol. 11, no. 3, pp. 334–347, 2010.