The Microsimulation Tool of the MicMac-Project

Jutta Gampe, Sabine Zinn
Max Planck Institute for Demographic Research
Rostock, Germany

Frans Willekens, Nicole van der Gaag, Joop de Beer
Netherlands Interdisciplinary Demographic Institute
The Hague, The Netherlands

Jan Himmelspach, Adelinde Uhrmacher
Modelling and Simulation Research Group
University of Rostock, Germany

Paper for the 2nd *General Conference of the*
International Microsimulation Association, Ottawa, June 8–10, 2009

Draft version: May 8, 2009, 19:52

Abstract

We describe the microsimulation tool MicCore that was developed in the project "MicMac – bridging the micro-macro gap in population forecasting". Based on a generic continuous-time multistate model with age- and calendar time-specific transition rates the MicCore was implemented as a plug-in to JAMES II, a general modelling and simulation environment. We outline the general software considerations and the specific implementation steps, illustrate the tool by an example, and discuss future extensions of the MicCore.

1 The MicMac-Project

In this paper we describe the microsimulation software, which was developped in the project "MicMac – Bridging the micro-macro gap in population forecasting". This project was funded by the European Commission under the 6th Framework Programme from 2005–2009. It is an interdisciplinary effort of a consortium of eight European research institutes under the leadership of the Netherlands Interdisciplinary Demographic Institute (NIDI), including not only demographers, but also economists, sociologists, health experts and statisticians.[1]

1.1 Aims of MicMac

The aim of the MicMac-project was to offer a bridge between aggregate projections of cohorts (Mac) and projections of the life courses of individual cohort members (Mic). Despite

[1] The other members of the consortium are: Vienna Institute of Demography, Institut National d'Études Démographiques (INED), Bocconi University, Erasmus Medical Centre Rotterdam, Max Planck Institute for Demographic Research, International Institute for Applied Systems Analysis, and University of Rostock.

the availability of more sophisticated methodology and constantly increasing computing power demographic projections still are mostly confined to populations disaggregated by age and sex. Extensions to multistate projection models, in which people move among functional states, are also considered, however, intra-cohort variation of individuals is commonly not in the focus of population forecasts.

Demographic change, which in developed countries mainly implies ageing populations, increases the demand for adequate health care services, the adjustment of pension systems and other social protection systems. The sustainability of high-quality health care and pension systems is influenced to a considerable extent by demographic change and by the way people live their lives (lifestyle and life course). Therefore, an adequate monitoring and forecasting of demographic change and of the lifestyle and life courses of individuals in the population are a conditio sine qua non for the provision of health and social security to the people of Europe.

This requires a methodology that moves beyond conventional projections of the population by age and sex. What is needed is a methodology that complements the demographic projections with projections of the way people live their lives. Microsimulation lends itself naturally to this objective. One aim of MicMac was to develop such methodology and to provide tools, which are flexible and user-friendly, to facilitate the use of this methodology in the everyday practise of producing population projections.

Besides the development of methodology and software, the MicMac-project also included work packages on morbidity and mortality, fertility and living arrangements, and on uncertainty and expert knowledge in assumption making. These work packages collected data and performed data analyses to determine age profiles of relevant life course events, and produced a number of argument-based scenarios on future development of these profiles. The results of these work packages will finally allow case studies for two European countries (Italy and the Netherlands). In this paper we will restrict ourselves to the underlying multistate model, the design and implementation of the microsimulation software, and will illustrate the tool by an example. More details on the MicMac-project can be found at `www.micmac-projections.org`.

1.2 The MicMac Model

Both Mic and Mac are based on a multistate model with transition rates as parameters. While Mac focuses on state occupancies by age and sex, Mic addresses demographic events and other life transitions at the individual level.

The MicMac microsimulation is based on a continuous-time multistate model (Willekens, 2006; Andersen and Keiding, 2002). Each individual occupies one of I potential states, which are collected in the state-space $S = \{1, \ldots, I\}$. This state-space is determined by the problem to be studied and will contain the relevant demographic states that need to be considered. In Mic we usually include death as one of the potential transitions, hence include ‘dead’ as one (absorbing) state in the model. The model may also include in- and out-migration to and from the population. In this case we define another absorbent state ‘emigrated’.

Mic's multistate model has two key time scales, which both are considered as continuous, namely the age x of an individual and calendar time t. The third time-scale, the time since entry into the current state, is not yet considered in MicMac's multistate model. An extension to this setting poses no particular problem to the simulation approach though and is currently work in progress.

Whether and when an individual experiences an event is embodied in the transition rates of the multistate model. The lifecourse of an individual is formalized as a trajectory of a semi-Markov process $Z_t = J_{N(t)}$, $t \geq 0$. The associated two-dimensional stochastic process $(J_n, T_n)_{n \in \mathbb{N}_0}$ describes the evolution in time of an individual life course. Directly linked to this process is the non-homogeneous Markov Chain $(J_n)_{n \in \mathbb{N}_0}$ that maps the states visited by an individual. The subprocess $(T_n)_{n \in \mathbb{N}_0}$ gives the sequence of consecutive transition times of the process $(J_n, T_n)_{n \in \mathbb{N}_0}$. The random number $S_n = T_{n+1} - T_n$ is the waiting time in the state J_n.

The transition rate (or intensity or hazard rate) that describes the propensity for a transition to state j, given that the individual occupies state i at calendar time t and age x is defined as

$$\lambda_{ij}(\tau \mid t, x) = \lim_{h \downarrow 0} \frac{1}{h} P\big[J_{n+1} = j, S_n \in (\tau, \tau + h] \mid J_n = i, C(T_n) = t, A(T_n) = x, S_n > \tau\big].$$

Here we include $A(T_n)$ and $C(T_n)$ to express the age- and time-dependence of the process. The function $A(T_n)$ maps the (biological) age at T_n and the function $C(T_n)$ the calendar time at T_n (Wolf, 1986). If the transition rates of the process are known, the distribution functions of the sojourn times can be derived.

If an individual enters a non-absorbing state i at age x and time t, then the individual will eventually move to one of the remaining $I-1$ states of S. The waiting times in state i until a move to a potential destination state j, s_{ij}, can be simulated using the well-known inversion theorem (Rubinstein and Kroese, 2008, pp. 53f.). If we denote by u a standard uniformly distributed random number, i.e., $u \sim U_{[0,1]}$, we can create a random time s_{ij} from the correct distribution by

$$s_{ij} = \Lambda_{ij}^{-1}(s_{ij} \mid x, t) \left[-\ln(1-u) \right], \qquad j \neq i, j \in S. \tag{1}$$

Here

$$\Lambda_{ij}(s_{ij} \mid x, t) = \int_0^{s_{ij}} \lambda_{ij}(u \mid x, t)\, du$$

is the so called integrated (or cumulated) hazard. According a competing risk setting, we pick the shortest waiting time $s_{ij^\star}$ to state $j^\star$. Then the individual under consideration experiences a transition to $j^\star$ at age $x + s_{ij^\star}$ and calendar time $t + s_{ij^\star}$. This computation of 'shortest' waiting times is repeated for each individual of the virtual population until the pre-set simulation end-time is reached or the individual enters an absorbing state.

Equation (1) implies that we have to be able to invert the integrated hazards $\Lambda_{ij}(s_{ij}|x,t)$. For some basic distributions the inverse of the integrated hazard can be computed analytically, for others this can only be done numerically. A suitable approximation to

$\Lambda_{ij}^{-1}(s_{ij} \mid x, t)$ is the application of piecewise linear cumulative hazards (Gampe and Zinn, 2007). As the distribution of the sojourn times S_{ij} depends on the shape of the transition rates $\lambda_{ij}(\tau|t, x)$, which mainly are empirically estimated from data, we rarely can pick commonly used parametric waiting time distributions anyway.

1.3 Software Considerations

The MicMac-project is to provide methodology that is not linked to a specific problem, but that can be used for applications in a very general multistate framework. This flexibility needs to be anticipated in the software design. Simulating individual life courses in accordance with the stochastic model described in section 1.2 obviously is the core operation of the microsimulation. Naturally, the software shall be computationally efficient to handle the potentially large number of individual life-courses with state-spaces of considerable size.

However, to run the simulation successfully, data have to be collected, statistical models have to be estimated and assumptions on the future development of demographic processes have to be made, so that appropriate transition rates can be provided to simulate from the model. Additionally, the base population may have to be adjusted for the state space considered (Wissen and Ekamper, 2008). Furthermore, to allow proper interpretation of results, the output of a microsimulation, i.e., the individual life courses generated for the members of the virtual population, will have to be summarized in tables or key indicators and illustrated by appropriate figures.

To make the final product attractive for the prospective users these additional tasks, which precede and follow the actual simulation job, will also have to be covered by the software. The development of the MicMac-software was guided by the following principles:

> – The use of the MicMac-software should imply no additional costs. That is, free software without license restrictions should be used. Also, the software should be available for as many current operating systems as possible. No particular commercial product should be needed to run the MicMac-software.
> – The MicMac-project supports the idea of open source software and therefore the code for all software will be available to the users.
> – The MicMac-project intends to provide tools for the users, without forcing them to employ the solutions we suggest.

To meet as many criteria as possible we decided on the following software structure (see Figure 1):

1. A so called pre-processor to prepare transition rates and starting population will be provided as a suite of `R` functions (see below).

2. The MicCore will perform the actual simulation. This part of the software is described in detail in section 2.

3. A post-processor, again implemented in `R`, provides tools for summary and illustration of results.

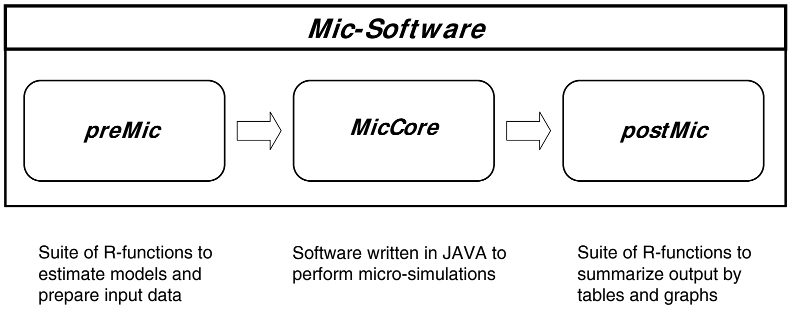
Mic-Software
preMic
MicCore
postMic
Suite of R-functions to estimate models and prepare input data
Software written in JAVA to perform micro-simulations
Suite of R-functions to summarize output by tables and graphs

Figure 1: Structure of the Mic-software

Using R as software tool for implementing the pre- and post-processor was motivated by its capacity as statistical analysis and graphics environment. R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License (R Development Core Team, 2007) and it runs on UNIX platforms, Windows and MacOS. The combination of up-to-date statistical methodology and well-designed publication-quality plots make it an ideal candidate for the pre- and post- processor. Nevertheless, we are aware that R may not be familiar to many prospective users. Therefore, even though we encourage its use, we designed our software so that users can continue with their preferred tools and will only have to provide the input rates in a specified format. The same applies to the simulated life-courses, which are simply stored in text files for further evaluation. The MicCore, which carries out the microsimulation and which will be described in the following section, requests no other interaction by the user than the provision of the transition rates in a well-defined format and a simulation frame, basically providing the starting population and the time horizon for the microsimulation.

2 The MicCore

Based on the model described in section 1.2 the corresponding software was intended as a dynamic microsimulation tool with the two continuous time scales age and calendar time (Willekens, 2006; Gampe and Zinn, 2007). As the development of new software is always a time consuming and error prone task, the option of either reusing or extending already existing solutions is a rewarding alternative, as long as no restrictions to the modelling scope have to be accepted.

Screening available tools hence is a sensible approach before starting from scratch. We did a thorough review of existing solutions, which is reported in Zinn et al. (2009). As a result it was decided to implement the MicCore as a plug-in to JAMES II, which is a general modelling and simulation framework, and is being developed at the University of Rostock since 2003. JAMES II is an open-source project that runs on a JAVA platform and is available at `www.jamesii.org`. It can be employed on all systems, for which a Java Virtual Machine (JVM) is available. JAMES II is endowed with up-to-date m&s technology, thereby allowing high computational efficiency, which is important for large-scale simulations as we expect to be run by the MicCore.

2.1 Plug'n Simulate: JAMES II

JAMES II is based on a 'Plug'n simulate' concept (Himmelspach and Uhrmacher, 2007). This concept enables the developer to integrate novel components (as plug-ins) into an existing framework. JAMES II basically consists of a lean core, which provides a plug-in management system, pre-defined, commonly used extension points, general basic functionality, etc. The core does not determine any modelling formalism though and does not contain any simulation algorithms. Concrete functionality can be added by the use of plug-ins.

For each type of plug-in JAMES II has to provide a specific extension point. Once such a point exists any kind of corresponding plug-in can be installed. For example, once an extension point for data sinks is given, each type of data sink plug-in (e.g., addressing databases or files) can be added. In this way JAMES II allows any extension of already existing features by simply appending appropriate plug-ins. Many plug-ins are already available for JAMES II (with related extension points): random number generators, random distributions, data sinks, different simulation algorithms, and several modelling formalisms.

In order to create specialized applications JAMES II offers two possibilities:

1. the framework could be extended directly (by re-using functionality provided by JAMES II), or
2. concrete applications could be build on the top of it by using JAMES II as a "m&s service".

The MicCore has been created using the second option. In this context, we have integrated JAMES II into the MicCore by re-using its not visible functionality, like event queues and data sinks, and adding plug-ins (see section 2.2). The minimal requirements of JAMES II, when creating an own specialized modelling and simulation application (like the MicCore), are

- a modelling formalism, and
- at least one simulation algorithm.

To execute any model in JAMES II, the following workflow is pursued (Himmelspach et al., 2008):

- First an experiment definition in JAMES II has to be created which defines the parameter settings for all simulation runs to be executed for a study.
- Subsequently, the experiment instance instructs for each simulation run the generation of a new model instance which is read from a given location and instantiated using a set of parameters (which might be read from an external location as well). The corresponding model representation is determined by a plug-in.

- Furthermore, the experiment instance instructs the allocation of a proper simulation algorithm for the simulation run.
- Subsequently, after a completed simulation run, simulated quantities are stored for further analysis.

A schematic illustration of the execution workflow in JAMES II is shown in Figure 2.

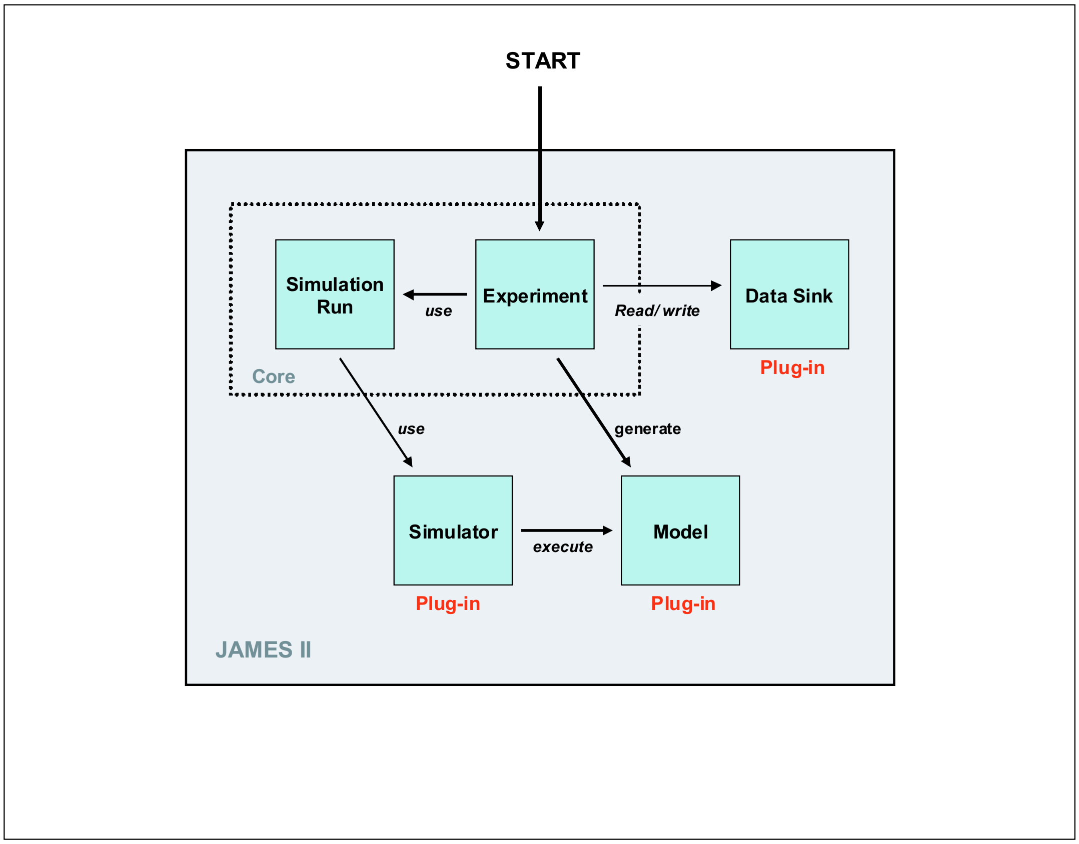

Figure 2: Workflow of JAMES II

2.2 The Implementation of the MicCore

Following the requirements of JAMES II we have developed two plug-ins for the MicCore:

1. a modelling plug-in, and
2. a simulator plug-in.

2.2.1 The modelling plug-in of the MicCore

The MicCore model is defined by a population that contains individuals and by rate matrices. Before performing a microsimulation run, the model population is initialized by either using a concrete observed population structure or by using an estimated synthetic base population. The initial population characteristics are provided in an ASCII file. The pre-processor of the MicMac-project offers tools to set up the base population in the format needed by the MicCore.

In a similar way, rate matrices have to be provided via an ASCII file. These rate matrices can be built up by any software tool. MicMac's pre-processor again provides functions that create these input rates in the appropriate format. All input files are read by a model reader mechanism that has been developed for the MicCore. After the MicCore model is set, a simulation run using the MicCore simulator plug-in can be executed.

2.2.2 The Simulator of the MicCore

The simulator of the MicCore contains the simulation algorithm which is used to simulate the life courses of individuals in the synthetic population. We employ a discrete event simulation (DES) engine to execute the model. This implies that all individuals are simulated simultaneously according to a continuous simulation time line. This line represents the calendar time scale of the MicMac model. Transitions between states are modelled in discrete time steps. More precisely, simulation time evolves discretely from one event to the next. Event queues are common data structures to implement sequential DES engines (Theodoropoulos et al., 2009). JAMES II provides several event queue plug-ins that can be used to collect events provoked by DES engines. For the MicMac model we selected the calendar queue (Brown, 1988) plug-in.

In the initialization phase of the simulation all individuals are en-queued in the calendar queue. For each individual of the initial population the next transition time and the corresponding destination state is determined (by simulating sojourn times and selecting the minimal waiting time, see section 1.2). All individuals are en-queued according to their transition times, i.e., the individual with the earliest transition time is en-queued at the first position of the queue, the individual with the second smallest transition time follows, and so on. Each element (i.e., individual) in the event queue has a logical timestamp to indicate the next transition time. During the simulation a loop iterates over the queue. In each iteration step the individual with the lowest timestamp is de-queued and the following steps are performed:

1. the de-queued individual experiences the pre-scheduled state transition,
2. the next transition time and destination state is determined and
3. the individual is en-queued again at the right position (according to the new timestamp).

This procedure is triggered for each simulated individual during the simulation time along his/her complete life-course. Hereby, at each point in simulation time only one prescheduled event per individual is possible.

Individuals can enter (newborns or immigrants) or leave (emigrants or deaths) the model population during the simulation. Once newborns or immigrants enter the population, they are treated like other individuals: the next transition to be experienced is computed and they are en-queued at the right position accordingly. If an individual exits the population, i.e., dies or emigrates, it is not en-queued again.

2.2.3 Workflow of the MicCore

For the actual simulation run the MicCore requests input files containing the state space, a set of corresponding transition rates and the initial population. The pre-processor of the MicMac software can assist in the preparation of these files. Furthermore, a time horizon for the simulation has to be set. By using a simple GUI, which has been implemented for the MicCore, the user enters all required instructions to run a microsimulation. Subsequently, after a completed simulation run, the synthetic lifecourses are stored in two ASCII files (using a data sink mechanism provided by JAMES II):

1. One file contains the birth dates of all simulated individuals.

2. The other file contains all simulated event times and the corresponding states of destination. It also contains the times, when simulated individuals entered the synthetic population. This is either at the starting time of the simulation, at date of immigration, or at birth date for newborns.

The MicMac's post-processor provides an extensive palette of functions to summarize and illustrate the output of a MicCore run. The processing of the MicCore is shown in Figure 3 (cf. also Figure 2).

2.2.4 Special Features of the MicCore

Although the model of MicMac is a generic multi-state model, which is not limited to particular applications, there are a few peculiarities of population projections that have been implemented in the MicCore directly.

The first concerns transitions to higher parity. As the model currently only includes age and calendar time, we have to explicitly model gap-times between consecutive births. Specifically, if fertility events are considered in an application, fertility rates related to higher parities are corrected for the inclusion of gestation gaps. We define an infertile period of 9 month (= 0.75 years) after delivery. The correction formula we apply for this purpose has been already used in *SOCSIM*, and has its origin in the so called dead time problem of nuclear instruments (Müller, 1973). The correction formula that we use is:

$$\hat{\lambda}(x,t) = \frac{\lambda(x,t)}{1 - 0.75\lambda(x,t)}$$

where $\lambda(x,t)$ is an empirical fertility rate that refers to a higher order birth event at age x at time t and $\hat{\lambda}(x,t)$ is the corresponding corrected rate.

The second issue concerns applications that include educational achievement in the state-space. Usually, we do not have transition rates for school-children, when to enter primary school, because this is mostly regulated by the start of the school year. Nevertheless the entry into the schooling-career needs to be modelled. In the MicCore enrolment into

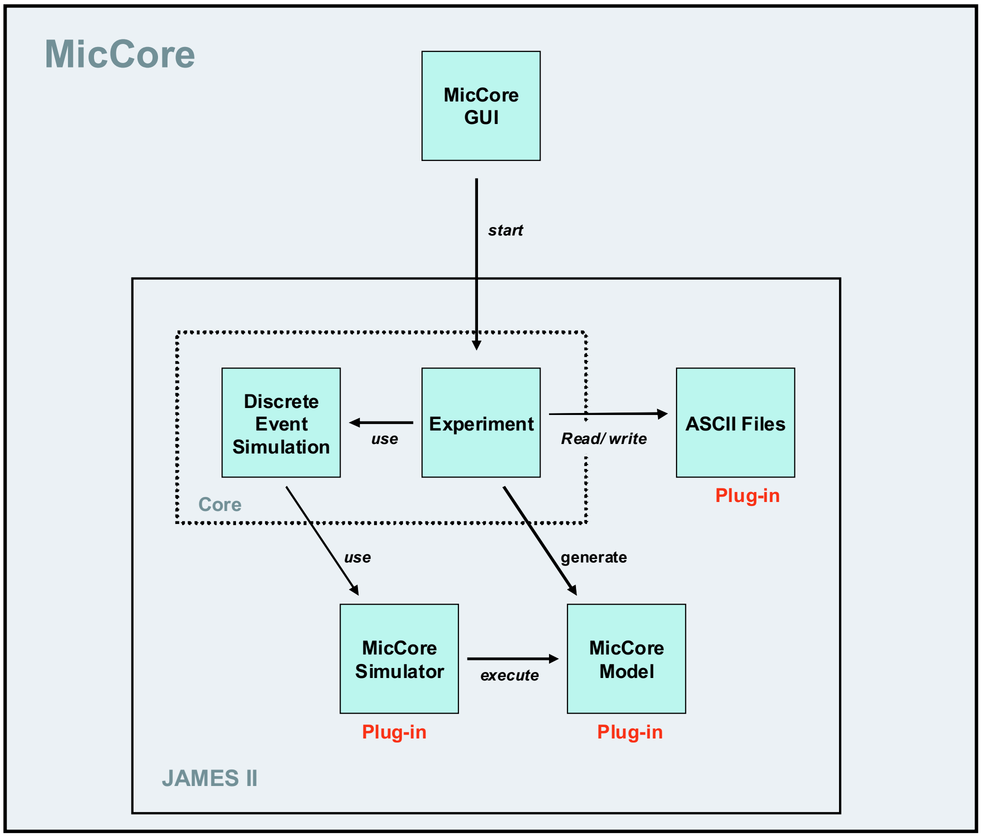

Figure 3: Workflow of the MicCore. JAMES II is embedded as back-end for simulation, the plug-ins developed are used.

primary school is triggered by a high transition rate into primary education at the age of six, which basically forces a transition at this age. The date for the transition, however, is then set to be on September 1 of the corresponding year.

The MicMac model and, consequently, the MicCore also offers the option to consider migration. In his case the numbers of immigrants according to their year of immigration and state occupied at immigration have to be provided. Furthermore, the MicCore offers the possibility that transition rates for immigrants differ from those applied to native members of the population. For emigration events, corresponding exiting rates have to be provided.

3 MicMac's Microsimulation in Practise

3.1 Example: Fertility and Mortality in the Netherlands

To perform an illustrative example on the capacity of the MicCore we apply the EUROPOP 2004 projections for the Netherlands (baseline scenario)[2] provided by EuroStat. This data set comprises for the years 2004 to 2050 information on Dutch mortality and fertility. At this, the female population is differentiated by children ever born. We are going to simulate from 2004-01-01 to 2050-12-31 lifecourses for the Dutch population.

[2] cp. http://epp.eurostat.ec.europa.eu/portal/page/portal/statistics/, accessed Mai 2009

In order to actually run a simulation, we need an initial population, a state space and related transition rates.

The initial population we employ comprises $\frac{1}{20}$ of the original Dutch population at the 2004-01-01.[3] It is categorized according to age, sex, and number of children ever born (cp. Figure 4). Summarized, the initial population consists of 410615 females and 402296 males. The individuals are aged in the interval from 0 to 99.

We assume that over the complete simulation time period 38952 females and 37569 males enter the population by immigration. The relative distribution (numbers referring to one year sum up to 1) of numbers of immigrants from 2004 to 2050 is displayed in Figure 5.

The state variables we consider are (Variable values are written in parenthesis after the variable names):

- Sex (male, female)
- Children ever born (up to 7)

Therefore, the only events individuals can experience during simulation are: giving birth or dying. The fertility and mortality rates, that we apply in our example, are age- and sex-specific and vary over calendar time. The rates are displayed in Figures 6 and 7. The sex ratio we use to set up the gender of the newborns is 0.485. Even though the MicCore allows to apply transition rates that refer only to immigrants, in this example we do not make use of this feature. In other words, in order to simulate lifecourses the same transition rates are used for natives and immigrants.

3.2 Running the MicCore and Output Enhancement

We performed the example on an Intel(R) Pentium(R) 4 CPU, with a 3.20GHz equipped with 3GB memory. We made 10 simulation runs. (We use a 'standard' machine as the user should be able to run a microsimulation on its own desktop computer.) The runs last between 615.78 and 630.02 seconds. The Mersenne Twister random number generator was used for the generation of the random waiting times.

Using the post-processor we are able to produce descriptive statistics of the output results. A few results are mentioned and illustrated shortly in the subsequent. The Figure 8 shows a comparison between the population structure that has been given in the initial population and the simulated population at 2050-12-31. In particular the graph depicts the number of women in each age class and their respective numbers of children It is obvious that along the simulation period the number of newborns decreases. Furthermore, there seems to be some evidence that less women have a second child and more have only one child. Figure 9 shows parts of three lifecourses which simulated females experienced. These sections of lifecourses are the three most frequently simulated ones (refereing to females). In Figure 10 two boxplots are given that describe some basic statistics related to

[3] This is due to two facts. Firstly, as the MicCore runs on normal desktop machines the capacity to execute a microsimulation that considers a complete population is (normally) not given. Secondly, the simulation of a sufficient high proportion of a population under study is enough to get all insights in demographic processes we are interested in.

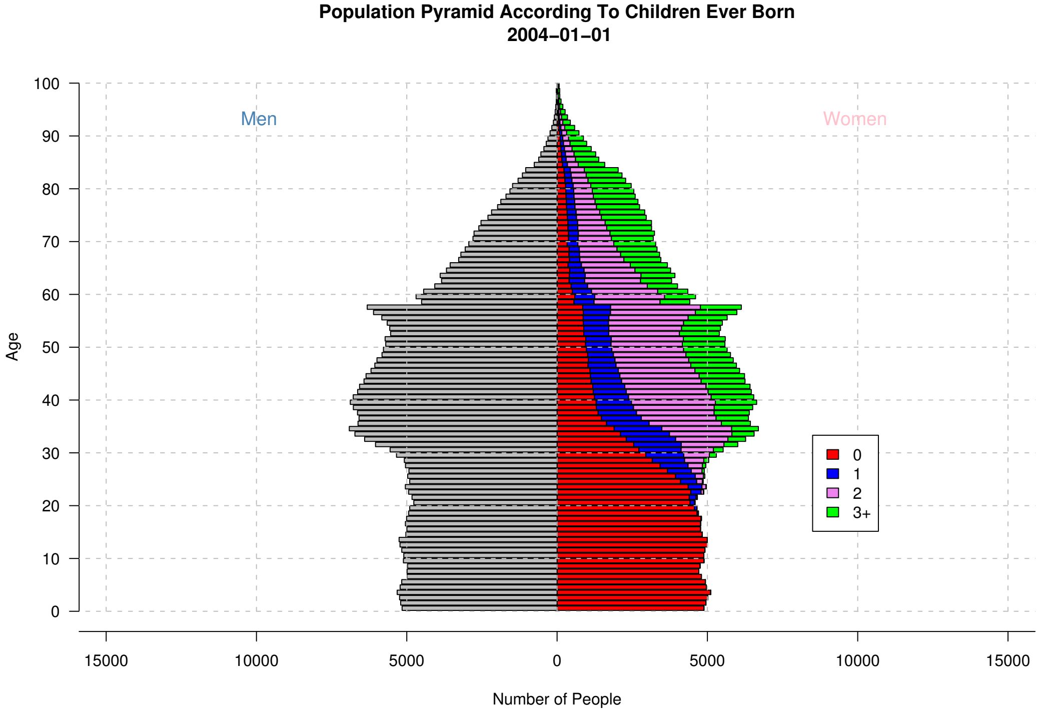

Figure 4: Structure of the initial population

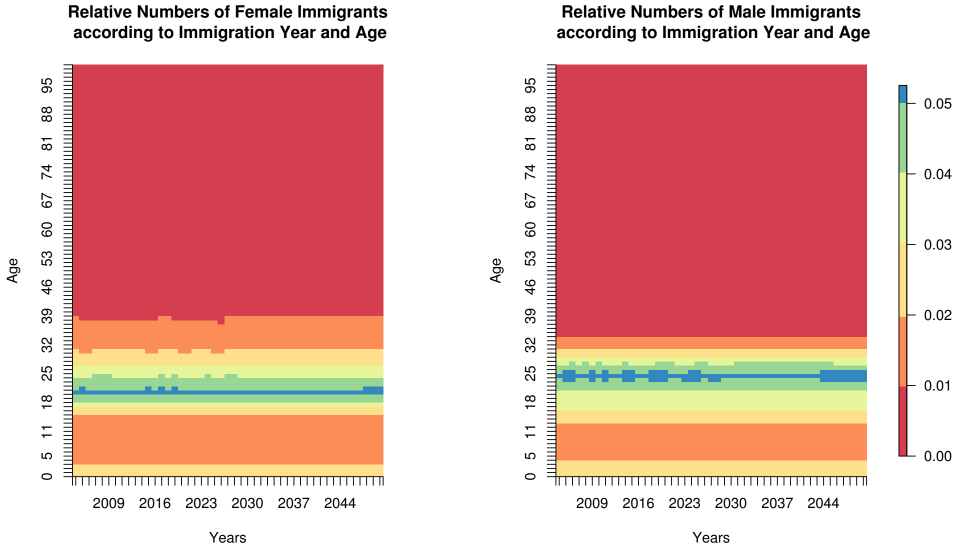

Figure 5: Relative distribution of immigration numbers

the transition from first to second child. Hereby the complete simulation period has been considered. The values that are given graphically in Figure 10 are additionally given as numbers in table 1. The image plot 11 displays the well known decrease of male mortality.

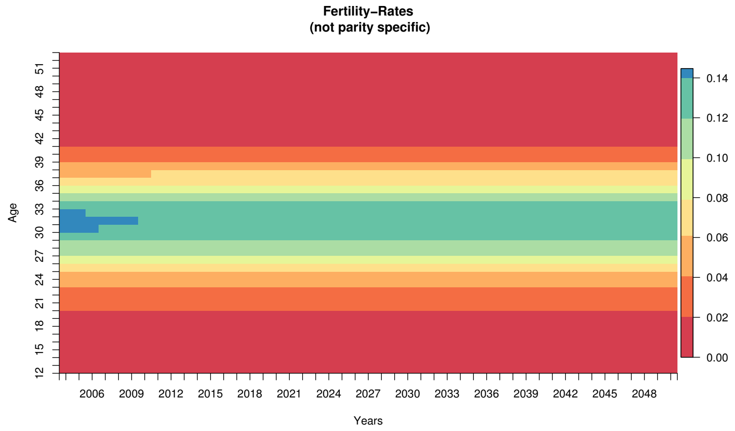

Figure 6: Non-parity specific fertility rates

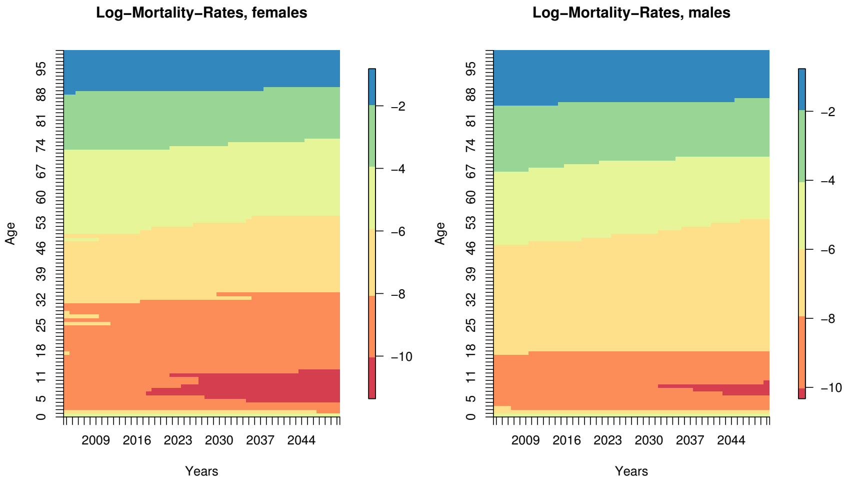

Figure 7: Log-mortality rates

	Average Duration	Average Age
Min	0.75	17.40
1st Quantile	2.23	29.07
Median	4.10	31.78
Mean	4.99	31.93
3rd Quantile	6.870	34.61
Maximum	27.94	50.09

Table 1: Basic statistics on the average duration until and average age of giving birth the the second child

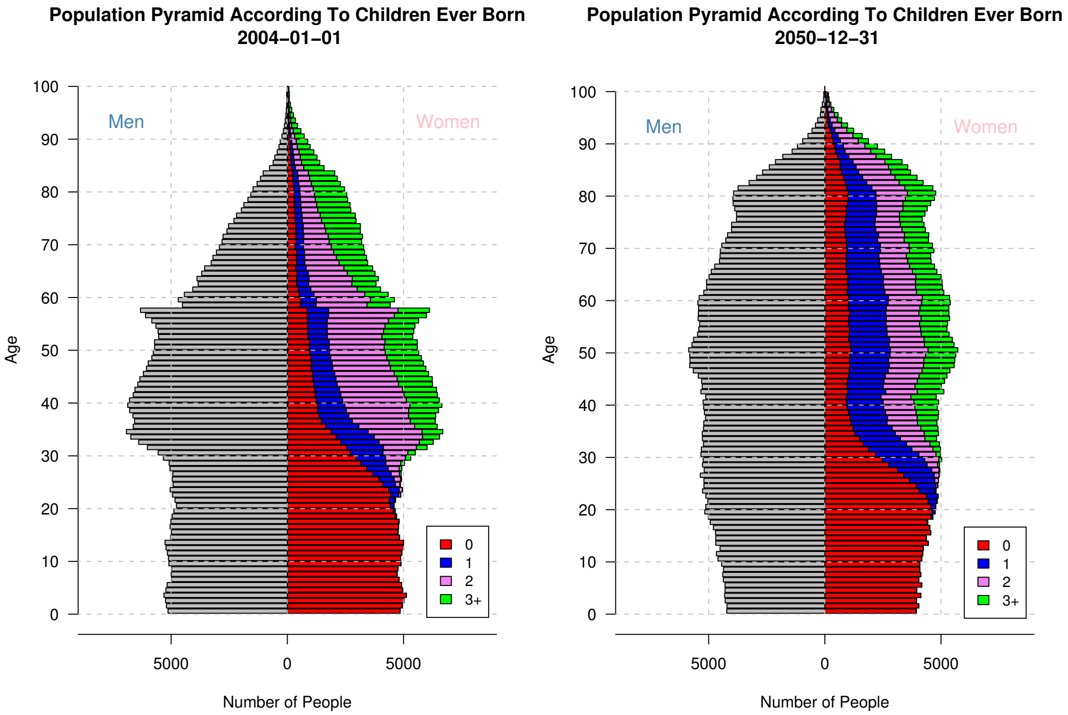

Figure 8: Comparison of population structure

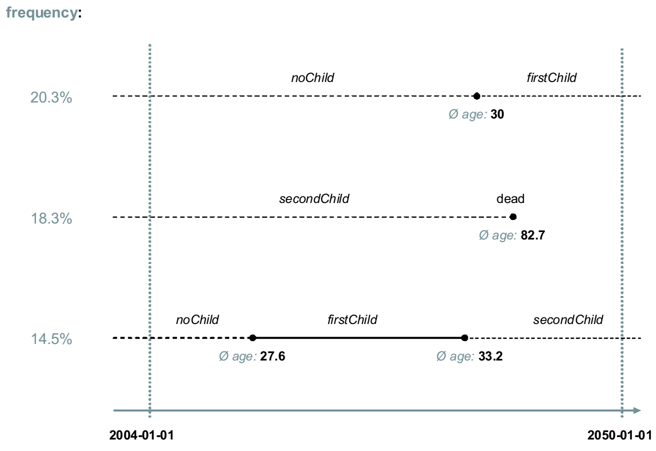

Figure 9: Most frequently simulated female lifecourses

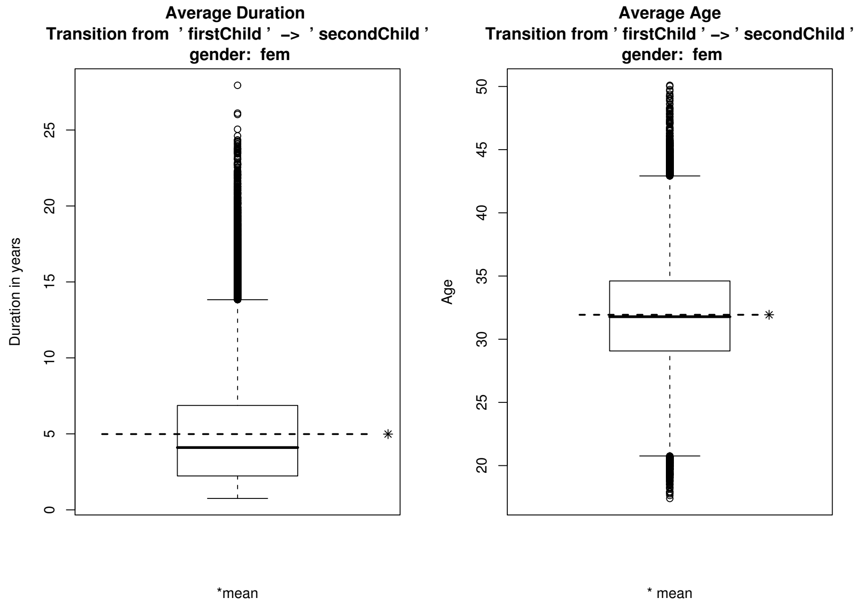

Figure 10: Basic statistics on the average duration until and average age of giving birth to the second child

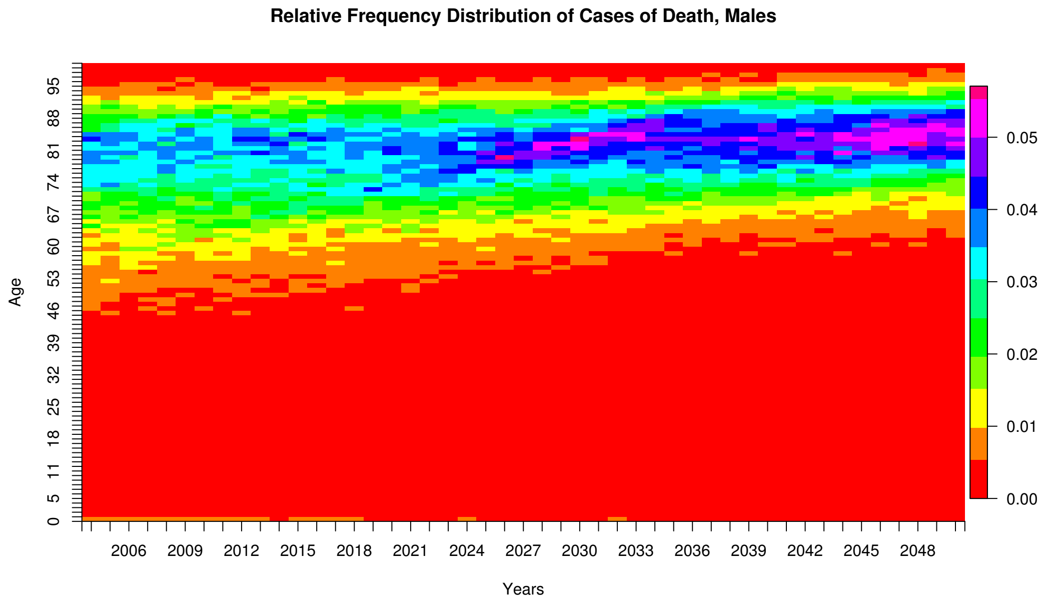

Figure 11: Relative distribution (numbers referring to one year sum up to 1) of Deaths

4 Discussion and Outlook

Despite the generic and rather flexible multistate model, MicMac has several limitations that should be relaxed in the future to allow more realistic descriptions of individual behaviour and hence population processes. Several extensions to the model described in section 1.2 are planned, some of these are already work in progress. Currently, individual life courses are simulated independently. Hence the concept of linked lives is not yet available in the MicCore. This includes the inheritance of characteristics of parents (as adults) to newborns. Currently newborn children simply enter into the same state-at-birth (no education, never married, non-smoker etc.) and will lead their lives according to the stochastic model only. Also, we currently do not model possible changes in transition rates of one person due to, for example, characteristics of the partner after union formation or marriage. Another related issue is the inclusion of a proper marriage market. Currently we do not match partners if the event of marriage is triggered by the simulation. A straightforward extension of the model will be the inclusion of a third time-scale in the stochastic model, besides age and calendar time, namely time since entry into current state. While implementing this further complexity in the MicCore will be straightforward, this will lead to considerable additional requirements for good micro-data to estimate (or at least reasonably provide) the corresponding empirical transition rates.

References

Andersen, P. and N. Keiding (2002). Multi-state models for event history analysis. *Statistical Methods in Medical Research 11*, 91–115.

Brown, R. (1988). Calendar queues: a fast 0(1) priority queue implementation for the simulation event set problem. *Commun. ACM 31*(10), 1220–1227.

Gampe, J. and S. Zinn (2007). Description of the microsimulation model. Technical report, MPIDR, Rostock.

Himmelspach, J., R. Ewald, and A. M. Uhrmacher (2008, December). A flexible and scalable experimentation layer for JAMES II. In S. Mason, R. Hill, L. Moench, and O. Rose (Eds.), *Proceedings of the Winter simulation conference*, pp. 827–835. IEEE Computer Society.

Himmelspach, J. and A. M. Uhrmacher (2007, March). Plug'n simulate. In *ANSS '07: Proceedings of the 40th Annual Simulation Symposium*, Washington, DC, USA, pp. 137–143. IEEE Computer Society.

Müller, J. (1973). Dead-time problems. *Nuclear Instruments and Methods 112*, 47–57.

Rubinstein, R. and D. Kroese (2008). *Simulation and the Monte Carlo Method* (2nd ed.). Wiley Sries in Probability and Statistics. Wiley.

Theodoropoulos, G., R. Minson, R. Ewald, and M. Lees (2009). *Simulation Engines for Multi-Agent Systems*, Chapter 3, pp. 1–29. Taylor and Francis.

Willekens, F. (2006, September). Description of the micro-simulation model (continuous time micro-simulation). Technical report, NIDI, The Hague.

Wissen, L. v. and P. Ekamper (2008, 9-12 July). Estimating a starting population for micmac. In *Paper presented at the European Population Conference*, Barcelona, Spain.

Wolf, D. (1986). Simulation methods for analyzing continuous-time event-history models. *Sociological Methoddology 16*, 283–308.

Zinn, S., J. Gampe, J. Himmelspach, and A. Uhrmacher (2009). MicCore: A tool for microsimulation. In *submitted to the Proceedings of the 2009 Winter Simulation Conference.*