A Continuous-Time Microsimulation and First Steps Towards a Multi-Level Approach in Demography

Dissertation

zur

Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

der Fakultät für Informatik and Elektrotechnik

der Universität Rostock

vorgelegt von
Sabine Zinn, geb. am 07.11.1979 in Meiningen
aus Rostock

Rostock, 31.05.2011

urn:nbn:de:gbv:28-diss2011-0187-1

Principal Advisor:	Prof. Dr. rer. nat. habil. Adelinde M. Uhrmacher *University of Rostock, Germany*
External Reviewers:	Prof. dr. ir. Frans Willekens (Emeritus professor) *University of Groningen, The Netherlands*
	Dr.-Ing. Jutta Gampe *Max Planck Institute for Demographic Research,* *Rostock, Germany*
Date of Defense:	August 12, 2011

Acknowlegments

This dissertation would never have been completed without the support of many people. I am indebted to my colleagues and the staff of the Max Planck-Institute for Demographic Research and the Modelling and Simulation Group of the Faculty of Computer Science and Electrical Engineering of the University Rostock. Particular thanks go to Jutta Gampe for her fruitful supervision. Jutta was an encouraging tutor: always inspiring and constructive in her comments. I am very grateful to Adelinde Uhrmacher who made me a part of her working group. She is full of ideas, always had time for my myriad of questions, and encouraged me to learn and develop my knowledge in the field of Modelling and Simulation. This thesis would not exist without Jan Himmelspach. I have to express my deep gratitude to him for his endless patience and encouragement. Jan widened my knowledge concerning software design immensely, and he helped me extremely with his fundamental assistance. I would like to acknowledge my debt to Frans Willekens for his valuable suggestions and his fast replies to my emails, often at the dead of night. Thanks to him, I learned that curiosity is one of the most important attitudes to have in order to become a good researcher. Furthermore, I want to thank Nicole van der Gaag for providing all the data I fit into my microsimulation drafts. Special thanks go to Ekaterina Ogurtsova for preparing various data input files for my microsimulation examples and to Jim Oeppen who helped me several times with general advice concerning microsimulation concepts (especially if SOCSIM was concerned). I would like to thank Nico Keilman for providing helpful comments on my mate-matching algorithm. Finally, I wish to express my acknowledgments to all members of the Laboratory of the Statistical Demography. Many thanks go to Lisa Berger who patiently took care of all organizational issues.

I have to acknowledge my gratitude to my friends for their patience and for always being close to me. I am extremely grateful to Sandra Krapf for her tremendous emotional support and scientific advice. She read and commented countless drafts of parts of my thesis. I want to acknowledge Giancarlo Camarda for always finding a sympathetic ear and a word of comfort. Many thanks go to Roland Ewald for answering all my nasty questions about computers, and to Svenja Weise who sent me good vibrations from Turkey. I thank my friends from Thuringia for always being sympathetic when I brought my work home and even to parties and festivals. I want to thank Steffi Schneider particularly for being “person number one” in my life, and also my mother for waiting patiently. Finally, many thanks go to my boyfriend Bernd, for having steadily been with me while living at the other end of Germany.

Abstract

To face the challenges of aging societies, policy makers need reliable information about future population compositions. Microsimulation is a methodology that lends itself to this objective, as it very closely mimics life-course dynamics. In this thesis, we describe the development of the demographic microsimulation with a continuous time scale that we have realized in the context of the project *MicMac - Bridging the micro-macro gap in population forecasting.* Furthermore, we detail extensions that we have added to the initial version of the MicMac microsimulation.

To avoid any redevelopments and to make use of up-to-date modelling and simulation (M&S) technology, we used the DEVS formalism to specify the microsimulation model, and we employed the M&S framework JAMES II to realize it. Due to its modular and state-based concept, the DEVS formalism is well suited to formulate population dynamics in the intended way. The usage of JAMES II allowed us to profit from already implemented M&S functionality and to validate the final microsimulation product.

In a second step, to overcome the unrealistic assumptions of independent life-courses, we worked out a method to match individuals. In our example, we defined a match when individuals experienced the onset of a marriage or cohabitation. Here we tried to mimic human partner choice and to aim at assortative mating. Testing the mate-matching method shows that it produces feasible results that are in accordance with observed mating behavior.

To model life-course dynamics after spouses are matched, we designed an approach that considers couples as entities and therefore describes the life-course dynamics of spouses jointly. We set up a ML-DEVS model to specify a microsimulation model that considers couple dynamics. The macro-DEVS model comprises two types of micro components: individuals and pairs. The micro components handle the life-course dynamics of unlinked individuals and the dynamics of married or cohabiting couples, while the macro-DEVS model guides the onset of partnerships (marriages or cohabitations), i.e., it performs mate matching and instructs the creation of pair models. We conducted a small case study to illustrate the capabilities of the microsimulation approach designed. Based on a synthetic population, we could show that it performs well.

Keywords: microsimulation, demographic application, discrete event simulation, DEVS, population projection, mate-matching, linked-lives

Zusammenfassung

Um auf die Herausforderungen alternder Gesellschaften einzugehen, benötigen Politiker zuverlässige Informationen über zukünftige Bevölkerungsstrukturen und -entwicklungen. Mikrosimulation ist eine Prognosetechnik, die genau auf solche Anforderungen zugeschnitten ist. Sie bildet individuelle Lebensläufe ab und eignet sich somit hervorragend, um Bevölkerungsdynamik realitätsnah abzubilden. In dieser Dissertation beschreiben wir die Entwicklung einer demografischen Mikrosimulation, die wir im Rahmen des Projektes *MicMac - Bridging the micro-macro gap in population forecasting* erstellt haben. Zudem erläutern wir Erweiterungen, die wir an der ursprünglichen MicMac Mikrosimulation vorgenommen haben. Um von den aktuellsten Modellierungs- und Simulationstechniken zu profitieren, haben wir die MicMac Mikrosimulation mit Hilfe des DEVS Formalismus spezifiziert und unter Verwendung der Modellierungs- und Simulationsumgebung JAMES II implementiert. Durch seine modulare und zustandsbasierte Konzeptualisierung eignet sich der DEVS Formalismus sehr gut um Bevölkerungsdynamik in der gewünschten Form zu beschreiben. Die Verwendung von JAMES II ermöglicht es auf bereits implementierte Funktionalitäten zurückzugreifen und das finale Mikrosimulationsprodukt zu validieren. In ihrer initialen Version setzt die MicMac Mikrosimulation die Annahme, dass individuelle Lebensläufe unabhängig voneinander sind. Dies ist eine unrealistische Annahme, die vermieden werden sollte. In dieser Dissertation haben wir eine Methode entwickelt, um zweigeschlechtliche Paarbeziehungen zu modellieren und simulieren. Bei der Konstruktion von Paaren versuchen wir menschliches Partnersuchverhalten nachzuahmen. Tests mit dem neu entwickelten Algorithmus zeigen, dass seine Resultate mit beobachtbarem Verhalten in Einklang stehen. In einem letzten Schritt haben wir einen Modellierungsansatz entwickelt, um die Lebensläufe von Eheleuten oder Lebensgefährten realitätsnah zu modellieren. Dieser Ansatz spezifiziert Paare als Entitäten mit eigener Dynamik. Um neben individuellem Verhalten Partnerverhalten zu spezifizieren, haben wir ein ML-DEVS Modell entwickelt. Die Makrokomponente des ML-DEVS Modells umfasst zwei Arten von Mikrokomponenten: Individuen und Paare. Die Mikrokomponenten bilden die Lebensläufe von Individuen und Paaren ab, während die Makrokomponente die Konstruktion und Auflösung von Ehen oder Lebensgemeinschaften handhabt. Anwendungsbeispiele unterstreichen das Potential der entwickelten Mikrosimulation.

Schlagwörter: Mikrosimulation, demografische Anwendung, Simulation diskreter Ereignisse, DEVS, Populationsvorhersage, Darstellung von zweigeschlechtlichen Paarbeziehungen

Contents

Appendices

List of Figures

List of Tables

Acronyms

DEVS	discrete event specification
dynDEVS	dynamic DEVS
dynPDEVS	parallel version of dynDEVS
FFS_NL 2003	Family and Fertility Survey of the Netherlands, conducted between February 2003 and May 2003
GUI	graphical user interface
LCG	linear congruental generator
mlDEVS	multi-level DEVS
M&S	modelling and simulation
NKPS	Netherlands Kinship Panel Study
PRNG	pseudo random number generator
tonie	time of next internal event

Glossary

Discrete event model

In discrete-time models time advances in discrete steps, i.e., the time axis is discretized. At each step, attributes and behavior of the studied unit are updated. In a discrete event model, no changes take place in between subsequent events, and only a finite number of events can occur in a finite time.

Discrete event simulation

A discrete event simulation executes a discrete event model.

JAMES II

JAMES II is an open modelling and simulation framework based on a flexible, plug-in-based architecture written in JAVA.

Linked lives

The principle of linked lives implies "that people in salient relationships with each other, such as parents and children, occupy mutually influential interlocking development trajectories that extend throughout their lives." (Greenfiled and Marks, 2006, p. 443)

Mate-matching

Mate-matching means building synthetic (married or cohabiting) couples.

MicMac project

The project MicMac "Bridging the micro-macro gap in population forecasting" was funded by the European Commission under the 6th Framework Programme from 2005 - 2009. 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) (NIDI, 2006).

Microsimulation

"A microsimulation consists of drawing a sample of realizations of a prespecified stochastic process."(Wolf, 2001, p. 315)

Model

"The facility or process of interest is usually called a system, and in order to study it scientifically we often have to make a set of assumptions about how it works. These assumptions, which usually take the form of mathematical or logical relationships, constitute a model that is used to try to gain some understanding of how the corresponding system behaves." (Law and Kelton, 2000, p. 1)

Markov process

A Markov process is a process where the "probability of any future behaviour of the process, when its present state is known exactly, is not altered by additional knowledge concerning its past" (Karlin and Taylor, 1975, p. 29).

Semi-Markov process

A semi-Markov process generalizes a Markov process such that the probability of undertaking a transition does not only depend on the present state, but also on the time already spent in that state.

Simulation

"A simulation is an experiment performed on a model." (Cellier, 1991, p. 6)

Simulator

A simulator describes the general execution semantics of a model.

The statistical tool R

R is a free and open-source software environment for statistical computing and graphics that is equipped with up-to-date statistical methodology and high-quality plot options.

1. Introduction: Realizing a Demographic Microsimulation

> "All demographic projection models are simplified, quantitative descriptions of the processes that determine population structures. They are simplified in the sense that not all variables affecting population structures are included in the model (they are also simplified in terms of functional form). They are quantitative in the sense that one set of numbers goes in and another set of numbers comes out."
>
> van Imhoff and Post (1998)

Demographic change will be one of the most challenging issues for Europe in the decades to come. Due to longer lives, low birth rates, changing family structures, and migration, the structure of European societies is significantly changing. To respond to the challenges that aging societies are facing, policy makers have to react: sustainable public finances have to be ensured to guarantee adequate pensions, health care, and long-term care. Due to a shift of the labor force to older ages and due to increasing immigration flows new, employment policies have to be designed and approved. Conditions for families need to be improved, and reconciliation of work and family life have to be warranted. Finally, for migrants, integration programs need to be realized. Policy makers, however, need reliable information about future population composition, and, in particular, answers to the questions of how different policies might affect individual behavior. For example, today highly topical is whether policies can significantly affect childbearing decision, e.g., by implementing a parental leave benefit for both men and women. Adequate monitoring and forecasting of the direction and dimension of the demographic change and of the lifestyle and life-courses of individuals is a conditio sine qua non for this purpose.

Demographic microsimulation Still standard demographic projections are mostly confined to populations disaggregated by age and sex. Although multi-state projection models, in which people move between demographically relevant states, increasingly substitute such projection models (Keyfitz and Caswell, 2005; van Imhoff, 1990), they do not commonly consider individual demographic behavior and intra-cohort variation (van Imhoff and Post, 1998; Willekens, 2005a). Demographic microsimulation is a projection method that accounts for the latter.

With respect to social science applications, microsimulation can be described as a technique to model and simulate the behavior of micro units over time, such as individuals or households. Wolf (2001, p. 315) defines it as follows:

> "Microsimulation consists of drawing a sample of realizations of a prespecified stochastic process. Microsimulation thus entails the generation of data (a set of realizations). [...] the model (the prespecified stochastic process) must be known in advance. The generated data will look like 'real data', and can, therefore, be analyzed and summarized just like real data [...]."

In the 1950s, Orcutt (1957) had already developed this approach to overcome shortcomings associated with macro models based on aggregate data. Macro models usually operate on averages, like the mean population age or the total fertility rate, and they neglect the heterogeneity of individuals in the population studied. Important insights about population composition and development are disregarded and, due to aggregation, information is lost.

The central device of microsimulations in demography is the life-course of an individual, which is defined by the sequence of states that the individual visits over time, and by the waiting times between these state transitions. Modelling and simulating the life-courses of a representative share of population members allows mapping population dynamics on a very detailed scale.

In demographic microsimulations, life-courses usually evolve along two time scales: individual age and calendar time. A possible third time scale is the time that an individual has already spent in his/her current demographic state, e.g., the time that has elapsed since the individual's wedding. All time scales can either be discrete (usually in units of years) or continuous (Galler, 1997; Wolf, 2001). In discrete-time models, time advances in discrete steps, i.e., the time axis is discretized. At each step, individual attributes and behavior are updated. In contrast, a continuous-time microsimulation model features a continuous time scale along which events occur, i.e., an event can occur at any instant in time. A continuous-time microsimulation model can appropriately be described by a discrete event model. Discrete event models feature a continuous time scale, and state variables can be either continuous or discrete (Fujimoto, 2000, Chapter 2). In a discrete event model, no changes take place in between subsequent events, and only a finite number of events can occur in a finite time span (Cellier, 1991, Section 1.9).[1] By assuming equidistant time steps, a discrete-event model can emulate a discrete-time model. If we define the step size of a discrete-time model as being the greatest common divisor among all event times in a discrete-event model, we can also use a discrete-time model to describe a discrete-event model (Fujimoto, 2000, Chapter 2). In practice, however, such processing might be very inefficient, because many time steps are presumably free of events.

Compared to a discrete-time microsimulation where at each time step all attributes of all individuals need to be updated, the processing of a continuous-time microsimulation can be

---

[1] That is opposed to classical continuous-time models where within finite time spans infinitely many events might occur.

very efficient (Satyabudhi and Onggo, 2008; van Imhoff and Post, 1998). This is due to the fact that in a continuous-time microsimulation, individual attributes are only updated when an event occurs. In demography, however, continuous-time microsimulation is applied less frequently than its discrete-time counterparts (Zinn et al., 2010, 2009), mainly because many modelers deem continuous-time models to be too cumbersome to handle (Galler, 1997). For example, Scott et al. (2003, p. 5) state that “continuous-time simulation has certain theoretical advantages, but implementation is more difficult, and model operation less transparent to the user, than for discrete-time”. Nevertheless, various tools exist that ease the implementation of a discrete event simulation, e.g., the library μsik (Perumalla, 2005), or the general modelling and simulation framework JAMES II (`www.jamesii.org`). Generally, for a precise description of population dynamics, continuous-time models are the optimal theoretical choice, as a continuous approach most closely mirrors life-course development (Willekens, 2005a). Depending on the definition of the time scale, the stochastic model of a microsimulation is parameterized either with transition probabilities (discrete-time model) or with incidence rates (continuous-time model). Both are assumed to vary with age and also with calendar time (e.g., decreasing mortality or increasing divorce rates). Commonly, for their estimation statistical, methods of event history analysis are applied to retrospective or prospective life histories reconstructed from longitudinal data and/or vital statistics. Assumptions about future rates/probabilites then define the projection scenarios.

Context of the thesis: The MicMac project In the context of the MicMac project, the author has realized a software tool for a demographic microsimulation with a continuous-time scale. The aim of the project *MicMac - Bridging the micro-macro gap in population forecasting* has been “to develop a methodology and the associated software that offers a bridge between aggregate projections of cohorts (Mac) and projections of the life-courses of individual cohort members (Mic).” (NIDI, 2006), see also `www.micmac-projections.org`. Both Mic and Mac are multi-state models. The output of Mac consists of cohort biographies, while Mic produces individual biographies. The idea of the MicMac project has been to run Mic and Mac in tandem, and to adjust microsimulation output relying on macroprojection output and vice versa (Willekens, 2005a). The project was funded by the European Commission under the 6th Framework Programme, between 2005 and 2009. It was an interdisciplinary effort of a consortium of eight European research institutes under the leadership of the Netherlands Interdisciplinary Demographic Institute (NIDI), including demographers, economists, sociologists, health experts, and statisticians.[2]

Structure of the thesis The model, the design, and the realization of the MicMac microsimulation is topic of the first part of this thesis. From the beginning on, it was clear that the microsimulation software should rely on computationally efficient simulation algorithms, while

---

[2] The other members of the consortium were: 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.

avoiding any re-development of already existing M&S (Modelling and Simulation) techniques. Following these requirements, the author has chosen two alternative approaches to implement the MicMac microsimulation: first, we have specified a DEVS (Discrete EVent Specification language) model for the MicMac microsimulation. DEVS is a traditional formalism of discrete event systems (Zeigler et al., 2000) which is supported by several software tools, e.g., DEVSJava, JDEVS, CD++, DEVS variants in JAMES II. Second, using the M&S framework JAMES II as a library, we have translated the microsimulation model directly into source code. Both approaches allow us to profit from already implemented, well-proven, and efficient up-to-date simulation methodology. The latter approach is well suited for the implementation of the MicMac microsimulation. Nonetheless, it hampers the realization of model extensions, such as the inclusion of linked lives: To realistically describe individual behavior, the effect of inter-individual interaction has to be considered. The second and third part of this thesis deal with this model extension. In the second part, we propose a mate-matching algorithm for continuous-time microsimulations; and in the third part, we suggest an extension of the MicMac microsimulation that additionally incorporates inter-dependencies between the life-courses of two linked individuals. Here, we rely on the DEVS model that we have developed for the MicMac microsimulation.

Part I.

A SIMPLE MODEL WITHOUT INTERACTION

The first part of this thesis is structured as follows: In the next chapter, we describe the model of the MicMac microsimulation. Herein, individual life-courses evolve along age and calendar time. We also describe a model extension that, besides age and calendar time, considers a third time scale, namely the time that an individual has already stayed in his/her current state. Subsequently, we list data types and methods suitable for estimating and constructing the input data for the microsimulation models. In Chapter 4, we present a DEVS model that formalizes the microsimulation model, and in Chapter 5, we describe the microsimulation tool that has been created using the M&S framework JAMES II as a library. We illustrate the capability of the MicMac microsimulation in Chapter 6 by forecasting a synthetic population based on the actual population of the Netherlands. In Chapter 7, we assess and validate the designed microsimulation and its outcomes.

2. Model Description

Generally, a demographic microsimulation model comprises a stochastic model of individual behavior and a virtual population (Willekens, 2007). The virtual population consists of individuals that are distributed over the attribute categories of the model (e.g., sex, marital status, educational attainment) and across ages (commonly given in single years of age). Usually this population resembles a real population structure. This model population evolves over time: individuals experience demographic events based on the stochastic model over their life times. Despite considerable regularity of demographic behavior the order and age-specific incidence of demographic events vary between individuals and cannot be fully explained by observable characteristics. Therefore, individual life-courses are appropriately modelled by stochastic models (Mayer and Tuma, 1990).

2.1. The state space

A standard approach to describe individual behavior is a continuous-time multi-state model (Andersen and Keiding, 2002; Bartholomew, 1973; Gampe and Zinn, 2007). A multi-state model is a stochastic process that at any point in time occupies one out of a set of discrete states (Hougaard, 1999). These states summarize the demographically relevant categories an individual can belong to. Generally, the state space is determined by the problem to be studied, but commonly it will at least comprise the elementary demographic characteristics of sex and marital status. One element always present in the state space is "dead," a risk to which each individual is always exposed to.[1] In the terminology we use here an individual's state usually is a combined characteristic, given by the combination of several attributes. To simplify the description we define so called state variables. All unique combinations of values of these state variables define the state space. The state space is denoted by Ψ.

To give an example, we describe the state space considered in the MicMac-project (Willekens et al., 2007). The state variables and their corresponding values are given in Table 2.1.

For example, *[female, married, living with partner, non-disabled, current smoker, no child, highly educated]* would be one potential state in this state space. Exceptions of this rule are the attributes "alive" and "native" which are not explicitly listed for obvious reasons, and the attributes "dead" and "emigrated," which denote states themselves. "Dead" and "emigrated" are both absorbing states that, once they have been entered, will never be left again. All

---

[1] In case one really does not want to incorporate transitions to "dead," this can be achieved by setting the risk to zero.

State Variable	Possible Values
gender	female, male
marital status	never married, married, divorced, widowed
living arrangement	child in parental home, living without partner, living with partner, living with other person(s), living in institution
disability status	disabled, non-disabled
smoking	never smoker, ever smoker, current smoker
fertility status	no child, first child, second child, $\cdots$
education	low (primary education only), medium (lower secondary school), high (upper secondary or tertiary education)
mortality	alive, dead
migration	native, immigrated, emigrated

Table 2.1.: State space considered in the MicMac-project: state variables and their corresponding values.

other states are non-absorbent and are called transient states; they can, at least potentially, be left again.

2.2. The virtual population

The virtual population consists of all individuals that are considered during simulation. It can be regarded as a longitudinal sample from a synthetically constructed population that resembles a real one.

Each individual enters the virtual population in a state that is an element of the state space. Commonly, three different possibilities exist to become part of the virtual population:

1. by being a member of the initial population;
2. by birth; or:
3. by immigration (if the model includes immigration into the population).

The initial population gives the size and structure of the virtual population at simulation starting time. That is, the individuals whose life-courses should be modelled and simulated have to be defined. Therefore, the user has to provide information about the distribution of individuals according to age and to the states that they occupy.

Newborns also have to be assigned to the state space. Obviously, most states do not apply, as newborns are always childless and not partnered. To assign those states that actually can serve as initial states of newborns (denoted as “states-at-birth”), occurrence probabilities are defined. Newborns are randomly assigned to these states according to their probabilities. Usually, simulation models that allow the creation of newborns employ an empirical sex ratio (the number of girls relative to the number of all newborns) to determine a newborn’s sex. This has to be taken into account when setting the occurrence probabilities of the possible “states-at-birth”.

Also, immigrants have to be assigned to the state space when entering the virtual population. Furthermore, his/her age at immigration has to be given.

Individuals can leave the population either by death or by emigration (if emigration is allowed). If the model allows migration, the virtual population is a so-called *open population*, otherwise it is denoted as *closed population*.

2.3. Stochastic model of individual behavior

A demographic event implies a change in the state of an individual. For example, an unmarried woman with no children, who holds a university degree and lives with her partner, changes her state when giving birth to her first child. Likewise, other events such as marriage or end of partnership would imply a change of state. It should be emphasized that age runs parallel to the process time in the model, and therefore birthdays, i.e., completion of another year of life, is not an event in itself. In a continuous-time multi-state model each individual life-course is specified by a sequence of state transitions (events) and the time spans between these transitions. Figure 2.1 shows a part of such a synthetic life-course.

A common way to characterize an individual life-course is via a trajectory of a stochastic process[2] $Z(t)$, $t \in \mathbb{R}_0^+$, from the family of Markovian processes (Kijima, 1997), where the process time t maps the time span over which we "observe" an individual life-course. The time t is set to zero when an individual enters the virtual population. As long as the individual is part of the virtual population, t evolves along the individual life time. The process $Z(t)$ is fully defined by the two-dimension process $(J_n, T_n)_{n\in\mathbb{N}_0}$ where $(J_n)_{n\in\mathbb{N}_0}$ is a Markov Chain that maps all states that an individual occupies, $J_n \in \Psi$, and $(T_n)_{n\in\mathbb{N}_0}$ is the sequence of the

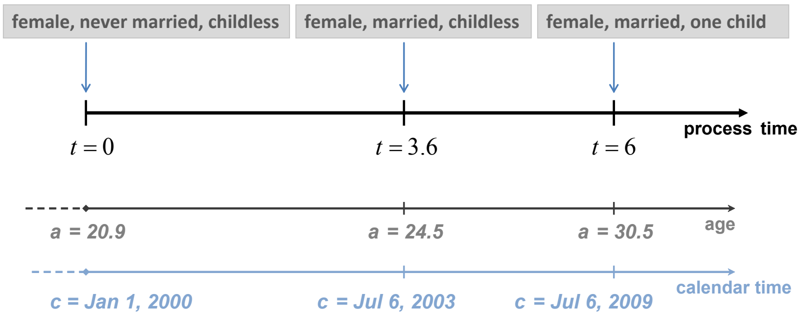


Figure 2.1.: Sketch of a life-course: a female who is part of the initial population on January 1, 2000. At this time she is $a = 20.9$ years old, never married and childless. On July 6, 2009, six years after she enters the virtual population, she experiences the birth of her first child.

---

[2] A stochastic process is a family of random variables $\{Z(t) \mid t \in \Theta\}$ defined on a given probability space $(\Omega, \mathcal{A}, P)$, indexed by the time variable t, where t varies over an index set Θ. A stochastic process maps $\Omega \times \Theta$ on $\mathbb{R}$. Therefore, alternatively we could write $Z(t)$ as $Z(t, \omega)$, $\omega \in \Omega$. The index set of a stochastic process that evolves along a continuous-time scale is $\Theta = \mathbb{R}_0^+$. For every fixed ω we call the function $t \to Z(t, \omega)$ a trajectory (or sample path) of the random process $Z(t)$ (Müller, 1975).

corresponding transition times along process time t, $T_n \in \mathbb{R}_0^+$, $T_0 = 0$. The random waiting time, for which an individual remains in state J_n, is determined by $W_n = T_{n+1} - T_n$.

Distinct from the process time T_n are calendar time and age (Wolf, 1986). Depending on the application at hand, process time (indicated by the transition times of the process) can be mapped into age and calendar time: The function $A(T_n)$ maps the age at T_n and the function $C(T_n)$ the calendar time at T_n. Both functions are translations. At the time of an individual's birth $A(\cdot)$ takes the value zero and $C(\cdot)$ is zero at 01-01-1970 00:00:00.[3]

In the MicMac microsimulation life-courses evolve along individual age and calendar time. For many applications such a design suffices, and suitable population forecasts can be derived (Willekens, 2005a). Some applications, however, require the consideration of a third time scale, besides age and calendar time, namely: the time that an individual has already spent in the current state.[4] For example, the divorce risk strongly depends on the time that has elapsed since wedding (Becker et al., 1977; Diekmann and Schmidheiny, 2008), and the propensity of giving birth to a second child depends on the time elapsed since the delivery of the first child (Neyer and Andersson, 2008). Therefore, an extension of the MicMac microsimulation to include this third time scale is very valuable.

Our next steps are as follows: first, we detail the stochastic model of individual behavior used by the MicMac microsimulation; then, we present an extension of this model that explicitly considers "the time elapsed since the last event" as a third time scale. For both stochastic models, we describe the simulation of synthetic life-courses.

2.3.1. Modelling and simulating life-courses over age and calendar time

In the MicMac microsimulation model, life-courses are considered along age and calendar time. That is, $Z(t)$ can be described as a non-homogeneous continuous-time Markov chain. The transition rates (also denoted as hazard rates or intensities) of a Markovian process are its key quantities. Once they are known, one can compute the distribution functions of the sojourn times. The transition intensities of a non-homogeneous continuous-time Markov chain are defined as follows:

$$\lambda_{s_j,s_k}(t) = \lim_{h \downarrow 0} \frac{1}{h} P[J_{n+1} = s_k, T_{n+1} \in (t, t+h] \mid J_n = s_j, T_{n+1} \geq t], \tag{2.1}$$

where s_j and s_k are elements of the state space, and t, $t \in \mathbb{R}_0^+$, describes the process time. That is, the transition intensity $\lambda_{s_j,s_k}(t)$ describes the propensity of an individual to undergo a transition from state s_j to s_k at process time t. Accounting for the age and calendar time dependence of the process, we obtain:

$$\begin{aligned}\lambda_{s_j,s_k}(c,a) = \lim_{h \downarrow 0} \frac{1}{h} P[J_{n+1} = s_k, C(T_{n+1}) \in (c, c+h], A(T_{n+1}) \in (a, a+h] \\ \mid J_n = s_j, A(T_{n+1}) \geq a, C(T_{n+1}) \geq c].\end{aligned}$$

---

[3] In computer science, 01-01-1970 00:00:00 commonly serves as a reference point from which time is measured.

[4] In the field of modelling and simulation (M&S), this third time scale is usually denoted as the "time elapsed."

If the transition rates of the process are known, the distribution function of the waiting times in the distinct states of the state space can be derived. The corresponding steps are detailed in the Appendix on page 179f. The distribution function of leaving state s_j for moving on to state s_k is:

$$F(w_{s_j,s_k}, c, a) = 1 - \exp\{-\Lambda_{s_j,s_k}(w_{s_j,s_k}, c, a)\},$$

where $s_j, s_k \in \Psi, s_j \neq s_k$, and a is the individual age at the last transition, c the corresponding calendar time, w_{s_j,s_k} the waiting time in s_j until moving on to s_k, and:

$$\Lambda_{s_j,s_k}(w_{s_j,s_k}, c, a) = \int_0^{w_{s_j,s_k}} \lambda_{s_j,s_k}(c+\nu, a+\nu)\ d\nu. \tag{2.2}$$

Alternative to the specification as a sequence of events and waiting times between these events, a life-course can be described as a sequence of waiting times to next events. Following this approach, to construct synthetic life-courses, we generate sequences of waiting times.

If an individual enters at calendar time c and age a a state s_j that is not the death or the emigration state, then the individual will eventually experience a further transition in the future. The individual might not only be exposed to one single event, but to a set of possible events. For example, a childless unmarried woman might experience childbirth, marriage, or death. We call such a situation a competing risk setting (Blossfeld et al., 1989, Section 3.4). In a competing risk setting, the waiting time function in a state s_j can be written as (Galler, 1997; Wolf, 1986):

$$S(w_{s_j}, c, a) = \prod_{k=1}^{K} S(w_{s_j,s_k}, c, a) = \prod_{k=1}^{K} \exp\{-\Lambda_{s_j,s_k}(w_{s_j,s_k}, c, a)\}, \tag{2.3}$$

where w_{s_j} is the waiting time in s_j and K is the number of possible next destination states (competing risks). This representation implies stochastic independence of the waiting times corresponding to the distinct competing risks (Galler, 1997). Thus, we can describe the waiting time function in s_j until leaving to another state by decomposing the waiting time function (2.3) into a set of independent waiting time functions, one for each possible destination state. The different waiting time functions are linked by the event that occurs first and censors all other events. Relying on this property, we determine the next event by computing random waiting times for all possible destination states s_k, and then by selecting the shortest one.[5] The waiting time in state s_j until moving to another state s_k can be simulated by the inversion theorem (Rubinstein and Kroese, 2008). If we denote by u a standard uniformly distributed random number, we yield a random waiting time w_{s_j,s_k} from the correct distribution by:

$$w_{s_j,s_k} = \Lambda^{-1}_{s_j,s_k}(w_{s_j,s_k}, c, a)[-\ln(1-u)]. \tag{2.4}$$

Let[6] $s_{k^\star} = \text{argmin}_{s_k \in \Psi, s_k \neq s_j} w_{s_j,s_k}$ and $w_{s_j,s_{k^\star}} = \min_k w_{s_j,s_k}$, then the individual under consid-

---

[5] In M&S this procedure is also known as “stochastic race.”

[6] $\text{argmin}_{a \in A} f(a)$ gives the argument (element of A) that minimizes the value of function $f(\cdot)$.

eration experiences his/her next state transition to $s_{k^\star}$ at age $a + w_{s_j,s_{k^\star}}$ and calendar time $c + w_{s_j,s_{k^\star}}$. This computation of the "shortest" waiting times is repeated for each individual of the virtual population until the simulation stop time has been reached, or until all individuals have left the population (due to death or emigration).

In the literature, an alternative procedure to simulate individual life-courses than the one presented above has been described. Instead of computing waiting times to all potential destination states and choosing the shortest one, Wolf (1986) suggests to first determine the waiting time w_{s_j} in s_j, and then to specify the destination state. In the appendix (see page 181ff.), we present a simple simulation study that underpins the equivalence of Wolf's (1986) approach and the method that we describe here.

In equation (2.4), the inversion of the integrated hazard $\Lambda_{s_j,s_k}(w_{s_j,s_k}, c, a)$ is required. Generally, for some basic distributions the inverse of the integrated hazard can be computed analytically (Willekens, 2009); for others this can only be done numerically (Law and Kelton, 2000). Assuming constant transition rates over age and calendar time intervals (mostly years) is a standard approach in demography that eases the derivation of process implications (Gill and Keilman, 1990). Integrated hazards become piecewise linear, and waiting time distributions piecewise exponential. It has been shown that the application of piecewise linear integrated hazards generally is a suitable approximation to $\Lambda^{-1}_{s_j,s_k}(w_{s_j,s_k}, c, a)$, and clearly eases its computation (Gampe and Zinn, 2007).

2.3.2. Modelling and simulating life-courses over age, calendar time, and waiting time

We extend the MicMac microsimulation model by considering, besides age and calendar time, the 'time that has elapsed since the last event' as a third time scale. The resulting model uses, instead of non-homogeneous continuous-time Markov chains, non-homogeneous semi-Markov processes (Monteiro et al., 2006) to describe life-course trajectories. The transition intensities $\lambda_{s_j,s_k}(w,t)$ of a semi-Markov process can be written as:

$$\lambda_{s_j,s_k}(w,t) = \lim_{h\downarrow 0} \frac{1}{h} P[J_{n+1} = s_k, W_n \in (w, w+h] \mid J_n = s_j, T_n = t, W_n \geq w],$$

i.e., $\lambda_{s_j,s_k}(w,t)$ describes the propensity of an individual to experience a transition to state s_k at process time $t+w$, after a waiting time of w in s_j. Here, unlike in the previous section, the parameter t does not indicate the process time, but the time at entry into a state. Opposed to a non-homogeneous continuous-time Markov chain, the transition intensities of a semi-Markov process depend on the waiting time spent in a state. Accounting for the age and calendar dependencies of life-course trajectories, we obtain:

$$\lambda_{s_j,s_k}(w,c,a) = \lim_{h\downarrow 0} \frac{1}{h} P[J_{n+1} = s_k, W_n \in (w, w+h] \mid J_n = s_j, C(T_n) = c, A(T_n) = a, W_n \geq w]. \tag{2.5}$$

Once the transition rates $\lambda_{s_j,s_k}(w,c,a)$ are known, the distribution functions of the waiting times in the various states of the state space can be derived. The related processing is equivalent to the one presented for non-homogeneous continuous-time Markov chains, see Appendix on page 179. The distribution function $F(w_{s_j,s_k},c,a)$ of leaving the current state s_j after a waiting time of w_{s_j,s_k} for moving to s_k, after having entered s_j at age a and calendar time c, is:

$$F(w_{s_j,s_k},c,a) = 1 - \exp\{-\Lambda_{s_i,s_j}(w_{s_j,s_k},c,a)\},$$

where $s_j, s_k \in \Psi, s_j \neq s_k$ and

$$\Lambda_{s_j,s_k}(w_{s_j,s_k},c,a) = \int_0^{w_{s_j,s_k}} \lambda_{s_j,s_k}(\nu,c,a)\ d\nu. \tag{2.6}$$

To construct synthetic life-courses, we generate sequences of random waiting times. The processing is equivalent to the one elaborated in Section 2.3.1: we assume that an individual has just entered state s_j at age a and calendar time c. Then we generate waiting time variates to all possible destination states s_k, $s_j, s_k \in \Psi, s_j \neq s_k$. For this purpose, we employ the inversion theorem. We obtain random waiting times w_{s_j,s_k} in state s_j until moving to state s_k using the formula:

$$w_{s_j,s_k} = \Lambda^{-1}_{s_j,s_k}(w_{s_j,s_k},c,a)[-\ln(1-u)], \tag{2.7}$$

where u denotes a random number from a standard uniform distribution. Of all generated random waiting times, we pick the shortest one; suppose it is $w_{s_j,s_{k^\star}}$ for a move to state $s_{k^\star}$. The individual under consideration then experiences his/her next state transition to $s_{k^\star}$ at age $a + w_{s_j,s_{k^\star}}$ at calendar time $c + w_{s_j,s_{k^\star}}$, after having spent $w_{s_j,s_{k^\star}}$ time units in s_j. The computation of these 'shortest' waiting times is repeated for each individual of the virtual population until the simulation stop-time has been reached, or until all individuals have left the population (due to death or emigration). Piecewise, constant transition intensities ease the inversion of integrated hazard. This is also true of the integrated hazard $\Lambda^{-1}_{s_j,s_k}(w_{s_j,s_k},w,c,a)$ that is part of equation 2.7.

The presented procedure to simulate life-courses is very similar to the one described for the MicMac microsimulation model, with the only difference that the semi-Markov approach employs transition intensities that, besides age and calendar time, also depend on the time that an individual has already spent in his/her current state.

2.4. Specific settings for particular transitions

A few details of the microsimulation model need special treatment, and we will describe these specific settings in the following paragraphs.

2.4.1. Facing improper waiting time distributions

When simulating life-course events, it can happen that some waiting time distributions are improper, i.e., transitions are only possible up to a certain age, and if individuals will not have left a state by this age, they will stay there "forever". Technically, this implies that the intensity drops down to zero, and the integrated hazard $\Lambda(\cdot)$ does not increase from this age on. Consequently, the inversion (see equation 2.4 and equation 2.7) can fail. As a result, generated random variate can be infinitely long.

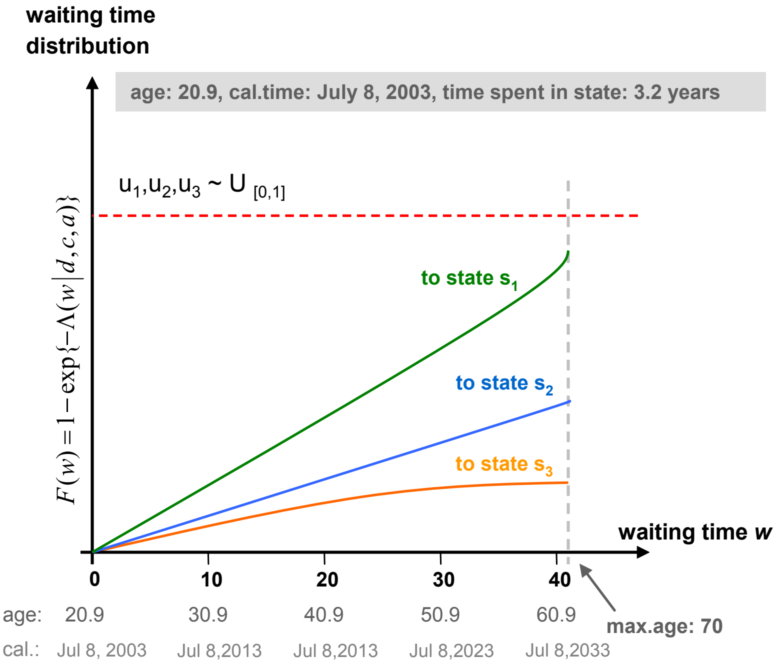


Figure 2.2.: The following is an example situation of infinite waiting times: At the age of 20.9, on July 8, 2003, after having already spent 3.2 years in the current state, a new waiting time until the next event has to be computed for the individual. Possible next states are s_1, s_2, and s_3. The corresponding cumulative distribution function derived from the integrated hazards are depicted in green, blue, and orange. The maximal age of 70 restricts the range of possible waiting times to $[0, 40.1]$ (years). Three uniformly distributed random numbers u_1, u_2, and u_3 are drawn to generate waiting times to transitions to s_1, s_2 and s_3. u_1, u_2 and u_3 correspond to waiting times that are out of the possible range of $[0, 40.1]$ (years), i.e., no finite waiting times can be determined.

As death is always a competing risk in our model, at least the transition to death is certain. Problems can arise if the user sets a too-restrictive upper bound for the maximum age until which life-courses are considered. In this case a finite waiting time to death might not be derived, cf. Figure 2.2. Infinite waiting times indicate that some individuals survive beyond the maximum age. To deal with such situations, two alternative strategies have been conceived: if an individual's age at simulation stopping time is smaller than the set maximum age, he/she remains in his/her current state until the simulation stopping time is reached. Otherwise, the individual stays in his/her current state until he/she reaches the maximum age, and then the individual becomes part of a special open age-category; indicating that no further information on survivorship to higher ages is available.

Similar problems of undefined waiting times can also arise due to the limitation of the projection period T_c. Rates beyond T_c are usually not provided, so in some cases finite waiting times cannot be derived. For example, on December 31 of the last year of simulation the probability of not finding a finite waiting time to a next event is relatively high. This situation is resolved in the same way as the one that is applied for the maximum age, as described above.

2.4.2. Infertile period between subsequent births

As already discussed before, the MicMac microsimulation model only includes age and calendar time. That is, the propensity of a woman to experience a birth only depends on age and calendar time, and not on the duration already spent in the current state. Thus, the model would allow women to undergo transitions to a higher parity within nine months after a previous birth. To overcome this inconsistency we explicitly set gap-times between consecutive births (i.e., the waiting times between consecutive births). We define an infertile period of nine months (= 0.75 years) after delivery, and within this period a woman cannot get another child. In this way, we modify the stochastic process used to describe a woman's life-course by adding "dead times." This leads to distorted outcomes in the sense that the input fertility rates do not correspond exactly to the simulated output. We illustrate this problem by a simple simulation study: we simulate childbirth events for $100,000$ childless women who initially are 15 years old, over the course of 40 years. The age-specific fertility rates we used for this purpose are depicted in Figure 2.3, and these rates are held constant over calendar time. Including a period of infertility leads to a lower proportion of second births as compared to the proportion without considering gap-times (see Figure 2.4).

To overcome this problem, we suggest to use a correction formula that has been already used in *SOCSIM*. This formula has its origin in the so-called dead time problem of nuclear instruments (Müller, 1973). It corrects the empirical rates $\lambda(c,a)$ for births of higher order to $\hat{\lambda}(c,a)$ by:

$$\hat{\lambda}(c,a) = \frac{\lambda(c,a)}{1 - 0.75\lambda(c,a)},$$

where a is age and c calendar time.

For the extended microsimulation model of Section 2.3.2, such a correction is not needed. Because of the semi-Markov model, the transition rates during gap-times can be simply set to zero.

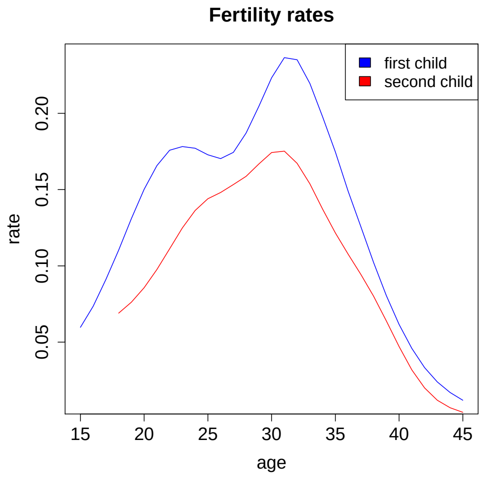


Figure 2.3.: Input fertility rates, not corrected for gap-times.

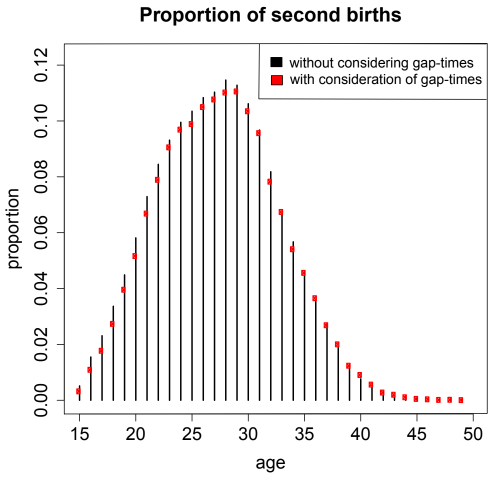


Figure 2.4.: Proportion of women who give birth to a second child among the women who are exposed to the risk of a second-birth event, with and without allowing for gap-times.

2.4.3. Entry into primary school

If the model state space includes educational achievement, it may be necessary to model the school enrolment of children, if individuals of the relevant age are simulated. Usually, we do not have transition rates for school-children because this is mostly regulated by the (externally set) start of the school year. To include entry into school-career nevertheless, we directly implement it: Enrolment into 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, is then set to be on September 1 of the corresponding year.

3. Preparation of Required Input Data

The type of continuous-time microsimulation that we consider in this thesis is parameterized by transition rates and an initial population. A set of transition rates has to be provided for each transition that individuals might experience during simulation. The rates have to cover the complete population projection horizon and the full age range. If we assume that, besides calendar time and age, rates depend also on the time that an individual has spent in his/her current state, then rates also have to be given according to the time spent in a state.

At simulation starting time, an initial population specifies all those individuals whose life-courses should be considered. Attributes of individuals, such as sex, marital status, educational level, etc., have to be provided as states of the model state space. Furthermore, the age of individuals at simulation starting time has to be given.

Transition rates are commonly estimated from data, and in this chapter, we first analyze different types of individual data concerning their suitability to estimate transition rates and in order to specify an initial population. Table 3.1 summarizes our results. Subsequently, we detail the estimation of transition rates based on occurrence-exposure rates, which is a standard method in demography. We describe this estimation method for rates depending only on age and calendar time, and for rates that additionally depend on the time that an individual has already spent in a state. Apart from past and present individual behavior, population projections demand transition rates concerning the future. In Section 3.2.2, we give some insights how assumptions about future rates might define projection scenarios. Subsequently, we describe the construction of an initial population.

The proposed demographic microsimulation is conceptualized as an open model (i.e., to allow in- and out-migration). The consideration of migration, however, requires data on immigrants and emigrants. In Section 3.4, we present data sources that can be used to get information about migration numbers and demographic behavior of immigrants.

3.1. Individual data

Individual data provide information for single individuals, and can be discriminated in prospective data, retrospective data, and cross-sectional data. Prospective data is gathered by following/observing individuals over time. Retrospective data contains individual information gathered at once and 'in retrospect'. Cross-sectional data measures information of a cross section of people at the same point in time.

Prospective data can result from civil registration systems, from follow-up studies, or can

Data types	**Transition rates**	**Initial population**
prospective data:		
vital register data	✓	✓
follow-up studies	✓	✗
longitudinal panel data	(✓)	✓
retrospective data	✓	✓
cross-sectional data	✗	✓

Table 3.1.: Suitability of individual data to estimate transition rates or to construct an initial population. The mark ✓denotes (usually) yes, and ✗ denotes (usually) not; the mark (✓) denotes that the data has to be edited before.

rely on periodic questionnaires (longitudinal panel data). Civil registration is primarily conducted “for the purpose of establishing the legal documents provided for by law” (United Nations, 2001, p. 50). It keeps for each inhabitant a continuous record of vital events[1] (i.e., it preserves exact event types, transition times, and waiting times between events for a complete population). For the estimation of transition rates or the construction of an initial population, such data are highly convenient. National population registration systems gathering vital events continuously have, however, been set up only in recent decades and by only a few nations, such as Denmark and Finland. Furthermore, because of data security reasons, data from registers are not easy to access for researchers. Therefore, for many applications, population register data might be unavailable.

In a follow-up study, changes of particular individual attributes are observed from a certain time point or from a particular event on (e.g., starting from January 1, 2000, or starting with the diagnosis of a certain disease). In this way, transitions between different states are reported exactly in time, and hence such data are generally appropriate for the estimation of transition rates. Unfortunately, follow-up studies are costly and time-consuming, which is the main reason why they are mainly conducted for epidemiological purposes with comparably small sample sizes (e.g., to analyze the risk of relapse after having recovered from cancer). That is, data from follow-up studies are often restricted to medical research.

Longitudinal panel studies collect individual data by periodically asking[2] a sample of citizens about their demographics, income, attitudes, etc. During an interview people are usually asked about their current attributes, and not necessarily about the events that they have experienced since the last interview. For example, since her last interview, a woman may have divorced and remarried, but during the interview she only reports that she is married, neglecting that she has meanwhile been divorced and has then married again. Consequently, to estimate transition rates from longitudinal panel data, it may be necessary to determine a strategy how to deal with events that possibly have not been reported. Two approaches have been proposed to handle this kind of missing information: if a change of attribute between two

---

[1] According to the United Nations vital events are live birth, foetal death, death, marriage, divorce, annulment, judicial separation, adoption, legitimation, and recognition (United Nations, 2001).

[2] Mostly, in periods of one, two or five years.

interviews is reported, one strategy is to assume that only one event has happened: namely, the one that directly leads to the change in attribute. This approach will underestimate the number of intermediate transitions, and the bias that is possibly introduced depends on the level of the underlying transition rates. A second option is to base the estimation on the underlying stochastic process, operating in either continuous-time (i.e., a Markov process) or discrete-time (i.e., an embedded Markov chain) to assess the real number of events and the corresponding event times (Kalbfleisch and Lawless, 1985; Laditka and Wolf, 1998; Wolf, 1988) and to derive a correct likelihood for panel data. The second approach demands a pretty complex estimation procedure, and its performance strongly depends on the definition of the underlying embedded process (e.g., the length of time steps if an embedded Markov chain is used). Nevertheless, if properly specified, it promises highly reliable results. Wolf and Gill (2009) compared both approaches for their performance. For this purpose, they estimated ‘true’ transition rates using a longitudinal panel study conducted monthly on disabled people in the U.S. Subsequently, from this panel study they created 12- and 24-month interval data, disregarding items from the monthly sequences. Based on these modified data, they estimated rates employing both approaches, and then compared them with the “true” rates. They found that both approaches performed poorly, while the second approach showed slightly better results.

As longitudinal panel data reflect the composition of real populations at specific points in time (possibly weighted by some sampling scheme) such data can be used for the construction of an initial population.

Retrospective data report individual life histories that people reconstruct from memory during interviews. As persons are directly asked for past events, longitudinal retrospective data can be adequate in principle to estimate transition rates. People may, however, forget about events: particularly about events that happened further in the past (Blossfeld and Rohwer, 2002, Chapter 1.1.3). For example, Matsuo and Willekens (2003) find in the (retrospective) Dutch Family and Fertility Survey inconsistencies both in the timing of reported events and also in missing events (e.g., several people reported a second marriage, but no first one). Matsuo and Willekens (2003) suggest approaches to correct for missing data and misreported events. Assuming a retrospective sample is corrected for data inconsistencies and is representative for a whole population, such data can then be taken for the construction of an initial population.

Cross-sectional data comprise observations of all members of a population or a representative subset for different interview times (e.g., for every year). If a representative subset of people is interviewed, for each interview a new sample of individuals is selected. A cross-sectional sample allows for construction of an initial population, but the sample design makes it impossible to track individuals over time. Therefore, prima facie, the estimation of transition rates seems to be impossible. To estimate rates, it is nonetheless common practice to construct synthetical “event histories” by matching individuals with similar attributes from different waves. A matching with respect to all reported attributes is unfeasible, as the probability of finding individuals with identical attributes in different waves is presumably very low. As a solution, matches are derived only according to few attributes, like sex, cohort, and maybe marital status, or according to so called propensity scores[3] (Sekhon, 2007). Such a processing means to potentially mix distinct behavioral patterns, and might consequently cause biased estimation results (Deaton, 1985). Another approach is to assume that individuals move between demographic states according to a continuous-time Markov process: then the derivation of statistical methodology for estimating transition rates from cross-sectional data is straightforward (Kalbfleisch and Lawless, 1985; Kalbfleisch et al., 1983).

Currently, the gross of available individual data is still cross-sectional data, most commonly census data. This solely is the reason for the interest in techniques to estimate time-varying factors and individual attributes from cross-sectional data, see e.g., Deaton (1985); Kasprzyk et al. (1989); Sekhon (2007).

Other data sources that can be used to gather information about demographics are tax lists, military muster and conscription rolls, budgetary surveys, wills and bequests, and insurance data. A common strategy to achieve microsimulation input data is, for example, to combine survey data and insurance data of national pension systems (see e.g., Geyer and Steiner (2010)).

3.2. Estimation of transition rates and construction of scenarios

In Chapter 2, we have described a continuous-time microsimulation model that requires as input transition rates in the form of:

$$\lambda_{s_j,s_k}(c,a) \text{ or} \tag{3.1}$$

$$\lambda_{s_j,s_k}(w,c,a). \tag{3.2}$$

Here, s_j indicates the state of origin, s_k: the state of destination; a: the age of an individual; c: the calendar time; and w: the time that has been elapsed since the last event. Rates of the form (3.1) correspond to non-homogeneous continuous-time Markov chains, and rates of the form (3.2) correspond to non-homogeneous semi-Markov processes (cf. Section 2.3). In the MicMac microsimulation, we employ rates of the form (3.1). Assuming that transition rates do not depend on w, but only on age and calendar time, is a widely made assumption in demography. Accordingly, standard estimation procedures mostly deal with age- and period-dependent hazard rates. Several authors extend existing approaches to also deal with duration-dependent transition rates, see e.g., Dabrowska et al. (1994); McClean and Montgomery (1999); Monteiro et al. (2006).

In Section 3.2.1, we focus on the estimation of transition rates based on occurrence-exposure rates. We describe this method for rates of the form (3.1) and for rates of the form (3.2).

---

[3] A propensity score is the conditional probability of an individual being assigned to a particular subpopulation given a set of covariates.

For the estimation of transition rates using different approaches we refer to the literature, see e.g. Hougaard (2000, Chapter 5, 6) and Blossfeld and Rohwer (2002, Chapter 5, 6, 9) for the estimation of transition rates of non-homogeneous Markov Chains, and Dabrowska et al. (1994); McClean and Montgomery (1999); McClean et al. (1998) for the estimation of transition rates of non-homogeneous semi-Markov processes. Afterwards, in Section 3.2.2, we sketch some ideas for the construction of scenarios for the prospective development of transition rates.

3.2.1. Occurrence-exposure rates

In demography, it is a common practice to specify transition rates as occurrence-exposure rates. These represent ratios of the number of occurrences to the total time that persons are exposed to the risk of experiencing a given type of transition during a time interval. Commonly, a time interval refers to a year-age category, but the addition of a duration category is straightforward. Instead of measuring occurrences and exposure times within two-dimensional year-age intervals, they are measured within three-dimensional year-age-duration intervals.

The application of occurrence-exposure rates puts only one implicit constraint on the shape of hazard rates: they are assumed to be piecewise constant (which leads to the so-called piecewise exponential model; see e.g., Blossfeld and Rohwer (2002, Chapter 5)). It is known that for short intervals occurrence-exposure rates are the maximum likelihood estimators of piecewise constant transition rates of continuous-time Markov Chains (Hoem, 1976) and of non-homogeneous semi-Markov processes (Monteiro et al., 2006).

Experience shows that population dynamics and individual behavior mainly follow smooth transition patterns, and that transition rates change continuously over time. Nonetheless, because of pragmatic reasons in demography, we often approximate transition rates by piecewise constant hazard rates. We generally differentiate between three types of piecewise constant hazard rates: age-period rates, cohort-period rates and age-cohort rates (see for instance Pressat (1972)). Age-period rates are held constant over age intervals and calendar years; they refer to two adjacent birth cohorts. Cohort-period rates are assumed to be constant for single birth cohorts within one year. Finally, age-cohort rates are constant for age intervals of birth cohorts, and these rates specify whether individuals experience a specific event between two birthdays; a corresponding interval covers two adjacent calendar years. Figure 3.1 illustrates the differences between the three types of piecewise constant rates. Which rate type is most appropriate depends on the application (Willekens et al., 2007): if it focuses on the development of life-courses of cohorts or single cohort members, age-cohort rates are adequate. For demographic projections studying the dynamics of a whole population, a period-cohort perspective is most appropriate. Generally, one rate type can always be approximated by another one (i.e., period-cohort rates can be approximated by averaging two subsequent age-period rates), and age-period rates and age-cohort rates are often assumed to be (almost) identical. The concept of the three rates types can be extended to account for an additional duration dependency: rates that are held constant within age, cohort, and period categories are additionally held constant within uniform duration categories.

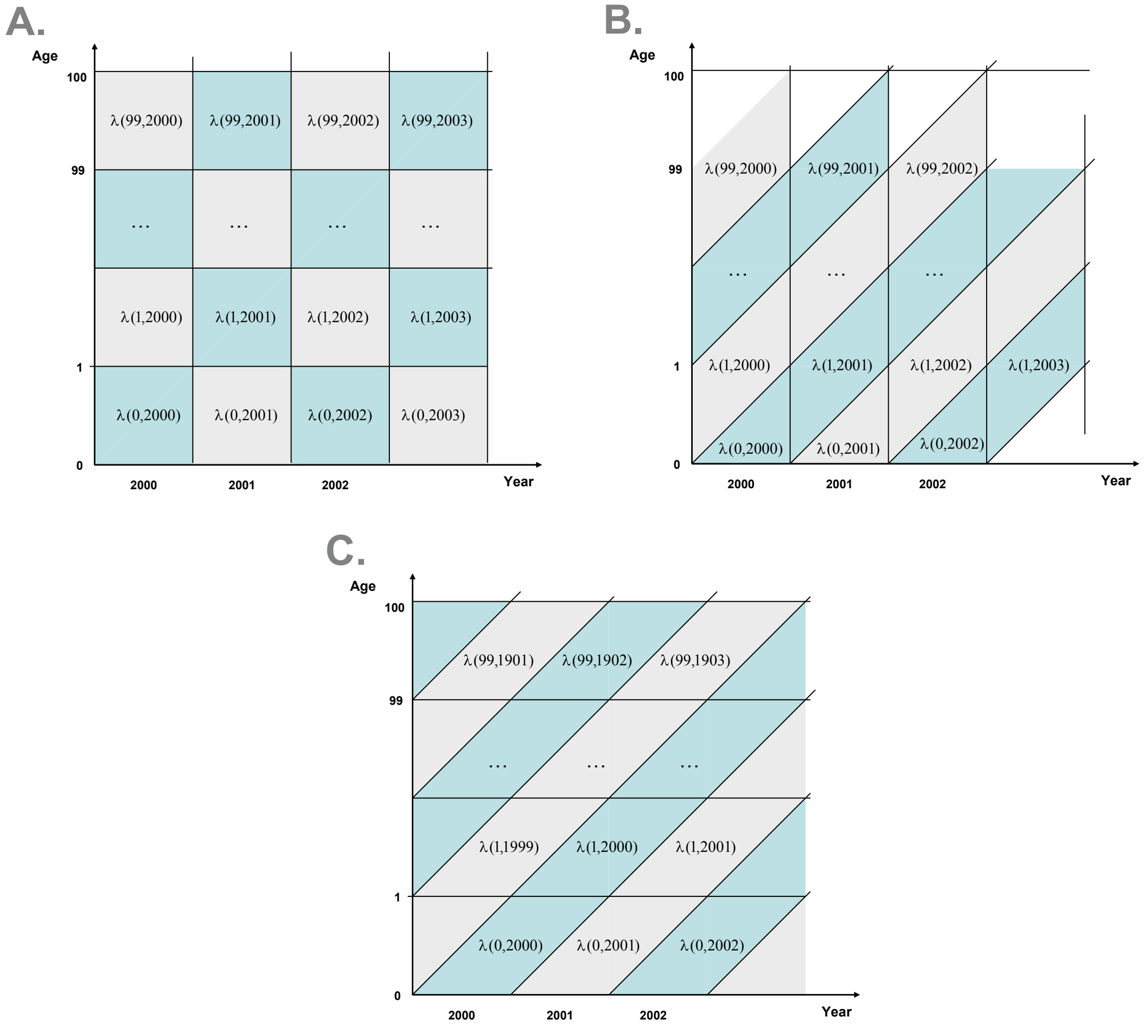


Figure 3.1.: Different kinds of piecewise-constant rates applied in demography: Lexis diagram A. depicts age-period rates, Lexis diagram B. cohort-period rates, and Lexis diagram C. shows age-cohort rates.

Estimating occurrence-exposure rates from prospective and retrospective data is straightforward because they include timing and event information. Thus, for age-year categories and for age-year-duration categories occurrences and exposure times can easily be derived. The situation differs if only cross-sectional data are available. As already described above, the difficulty with cross-sections is that individuals are not followed up, and therefore neither events nor exposure times can easily be identified. One possibility to estimate occurrence-exposure rates nonetheless is to aggregate cross-sectional data over individuals and events, and to apply methods that have specifically been developed for such purposes; e.g., Gill and Keilman (1990); Kalbfleisch et al. (1983).

Commonly, occurrence-exposure rates are estimated for each time interval separately; though such processing very likely causes erratic rates profiles, particulary if only few individuals have been observed. As a result, pointwise estimated occurrence-exposure rates do not often depict the actual profiles, and smoothing is advisable. A way to achieve smooth rates profiles is to integrate smoothing into the estimation procedure; i.e., the development of occurrence and exposure is captured by a (parametric) model that assumes smoothness as a requisite.

Assuming constant hazard rates over short time intervals implies that the total number of events D_T over a specific time interval T follows a Poisson distribution with with mean $\lambda_T E_T$. Here, λ_T maps the occurrence-exposure rate within T and E_T is the corresponding exposure time:

$$D_T \sim \text{Poisson}(\lambda_T E_T) \;\Rightarrow\; \mathbb{E}(\ln D_T) = \ln \lambda_T + \ln E_T.$$

We obtain smooth rates profiles by employing a regression model for D_T:

$$\mathbb{E}(\ln D_T) = f(\mathbf{x}) + \ln E_T, \tag{3.3}$$

where $\mathbf{x}$ is a vector of predictor variables and f is a smooth function. For example, generalized additive models (Hasti and Tibshirani, 1990) with a response that follows a Poisson distribution represent such regression models. Smoothness can be specified differently, e.g., Currie et al. (2006) suggest to specify f using B- and P-splines. Alternatively, Impicciatore and Billari (2007) describe a model called MAPLE (Method for Age Profile Longitudinal Estimation) where they employ piecewise cubic splines.

3.2.2. The construction of scenarios

For population projections, assumptions about (possible) prospective developments have to be made. One way to describe such developments is the use of *if-then scenarios*, which allow us to demonstrate the consequences of hypothetical conditions, such as different policies. To obtain a comprehensive picture of possible future population dynamics, a microsimulation has to be parameterized accordingly; i.e., for each envisaged "if-then scenario" a set of prospective transition rates has to be prepared. For this purpose, either experts have to be asked for their assessments or already estimated models have to be modified or extrapolated. A combination of both approaches would even be preferable. It is extremely challenging, however, to formulate options about changes in the future as changes of particular model parameters. Traditional population projections, therefore, still predominantly rely on formal models extrapolating historical and current trends. Integrating experts' assessment about likely future trends into a prediction model appears to be challenging (Lutz, 2009): on the one hand, substantive knowledge has to be incorporated into formal models, and on the other hand, the use of substantive knowledge has to be formalized. The few existing methods that combine argument-based approaches and hazard models relate to engineering problems; e.g., Wang et al. (2000). Further research will move to prove whether such methods can be adopted/-modified for the purposes of population projection scenarios. Within the scope of this thesis, traditional approaches might suffice to construct transition rates for different scenarios.

3.3. Constructing an initial population

An initial population gives the number of individuals in each state of the state space, and also in one-year age categories at the starting time of the simulation. Hence, an initial population can always be described by a multivariate distribution with discrete values - namely, the considered age groups and the values of the (discrete) state variables. If semi-Markov processes are used to describe individual life-courses, then, besides age and state of occupancy, an initial population also has to indicate the time that individuals have already spent in their current states when the simulation starts.

In this section we first analyze how appropriate different data types are for specifying an initial population, which is disaggregated by age and state of occupancy. Then, we present the iterative proportional fitting (IPF) algorithm as a method to construct such an initial population. Thereafter, we discuss the problem of how to impose on an initial population, which is disaggregated only by age and state of occupancy, and by the information about the duration that individuals have already spent in their current states. This latter issue is relevant if we additionally want to incorporate duration dependence in a semi-Markov context.

3.3.1. Constructing an initial population disaggregated by age and state of occupancy

Usable Data

Each type of data described in Section 3.1 is theoretically suitable to construct an initial population. Only data from follow-up studies may cause problems because of sample sizes that are too small. The minimal requirement to estimate an initial population disaggregated by age and state of occupancy is as follows: the age structure of the studied population in one-year age groups and information about the marginal distributions of the state variables. In principle, individual data meet all these requirements. Often, however, problems arise because the data at hand do not refer to the simulation starting time - but to an earlier date; or the data at hand corresponds to a different population. That is, a reweighting of the underlying multivariate population distribution is necessary. This requires further information about the population structure at simulation starting time. In addition, the more complex the state space of an application is, the more unlikely is the availability of a data set from which the multivariate structure of the model population is entirely derivable. Therefore, it might be necessary to combine information from different data sources. Available data sources provide different kinds of (partial) information that might be useful for our purposes. We categorize these as follows:

- marginal distributions of some or all state variables,

- the age distribution of the population under study,
- marginal information about population totals of certain state variables (e.g., one thousand women in the age group from 30 to 35 are childless),
- values for specific population groups (e.g., before age sixteen children are certainly in school),
- inequalities pertaining to certain population characteristics (e.g., women earn on average less than men).

Iterative proportional fitting

The problem of constructing an initial population – one that is disaggregated by age and state of occupancy – can be formalized as follows: we assume that the model state space comprises d state variables s_j, of which each can take L_j discrete values, $j = 1, \cdots, d$. Combined with age (categorized into L_{d+1} age groups), we obtain a $d+1$-dimensional distribution. We denote the values of this distribution by $c_{i_1,\cdots,i_{d+1}}$, $i_j = 1, \cdots, L_j, j = 1, \cdots, d+1$. A $d+1$-dimensional array $\mathcal{A}$ comprising all the $\prod_{j=1}^{d+1} L_j$ values $c_{i_1,\cdots,i_{d+1}}$ can be used to specify the $d+1$-dimensional distribution at hand. We subsequently assume that we know (of all state variables and of all one-year age groups) the values m_{j,i_j} of their respective marginal distributions. That is, m_{j,i_j} is the number of persons who feature value c_{i_j} of the state variable s_j.

To ease the subsequent description we introduce some notation:

$$c_{i_j,\bullet} = \sum_{i_j=1}^{L_1} \cdots \sum_{i_{j-1}=1}^{L_{j-1}} \sum_{i_{j+1}=1}^{L_{j+1}} \cdots \sum_{i_{d+1}=1}^{L_{d+1}} c_{i_1,\cdots,i_{d+1}};$$

i.e., the bullet "•" indicates the summation over those state variables that are not denoted in the subscript. The specification of an initial population of size N requires the determination of values for all cells of $\mathcal{A}$. They have to be determined such that for all s_j the following constraint is fulfilled:

$$m_{j,i_j} = c_{i_j,\bullet};$$

i.e., the estimated cell frequencies have to match the known marginal totals.

Several methods exist that can be used to estimate an initial population categorized according to age and state of occupancy, e.g., iterative proportional fitting (IPF), log-linear models, and entropy maximization. Which method suits the construction of an initial population best depends on the available data and on the imposed population structure. This means that there is no standard method to estimate cell frequencies. Here, we present the IPF algorithm. Regarding other methods applying to this task, we refer to the literature; see e.g., van Wissen and Ekamper (2009); Willekens (1999).

An iterative proportional fitting (IPF) procedure conducts matrix balancing or matrix scaling. That is, cell values are determined by multiplying the margin values by a seed array to yield a balanced array. We aim at balancing $\mathcal{A}$, which implies that the summation of cell values along the layers of $\mathcal{A}$ results in the corresponding marginal totals. In the initialization phase of an IPF approach, each cell of the array has to be filled with a "seed" value; e.g., $c^0_{i_1,\cdots,i_{d+1}} = 1/\prod_{j=1}^{d+1} L_j$, or values that contain a priori knowledge about the population structure. It is advisable to set cells to zero, of which it is ex ante known that they are zero (as an IPF procedure does not change zero values). There is no unique IPF algorithm, but a palette of different procedures, which basically differ in the manner of scaling. If only two dimensions are considered, the IPF is also referred to as the Fratar method, the Furness method, or as RAS optimization. In the following, we describe an extension of the RAS algorithm. For more sophisticated methods, we refer to the literature; see e.g., Ahuja et al. (1995) for a comprehensive description of network optimization algorithms. Our RAS extension is based on the description of the bi-proportional RAS algorithm given by Schneider and Zenios (1990). It conducts a sequential scaling along the distinct layers of $\mathcal{A}$. The corresponding procedure is as follows:

Step 0 (Initialization)

Define a stop criterion, initialize the iteration count $r = 0$, pick the first state variable $j = 1$, and $\mathcal{A}^0 = (c^0_{i_1,\cdots,i_{d+1}})_{i_j=1,\cdots,L_j, j=1,\cdots,d+1}$.

Step 1 (Scaling layer j)

For $i_j = 1,\cdots,L_j$ define

$$\beta^r_{j,i_j} = \frac{m_{j,i_j}}{c^r_{i_j,\bullet}},$$

and update $\mathcal{A}^r$ by $c^{r+1}_{i_1,\cdots,i_{d+1}} = \beta^r_{j,i_j} c^r_{i_1,\cdots,i_{d+1}}$.

Step 2 If $j < d+1$, go to the next layer: $j = j+1$ (return to Step 1); otherwise check whether the stop criterion is fulfilled. If it is not fulfilled, increment the iteration counter: $r = r+1$, set $j = 1$ and return to Step 1.

A useful stopping criterion of the RAS algorithm is an upper bound for the sum of deviations between calculated margins and the given margins. Schneider and Zenios (1990) describe a set of (weak) conditions under which the RAS algorithm converges. The basic idea is to construct first the transportation problem[4] associated to $\mathcal{A}$, and to check then its feasibility.

The RAS algorithm is an old method that has already been described 1937 under the name *method of twin factors* (Kruithof, 1937). Deming and Stephan (1940), who named the approach "method of iterative proportions," employed the method to estimate cell frequencies of a contingency table with known marginal. The method has been extended in many ways to cope with problems of different degrees of complexity. E.g., Fienberg (1970) describes an IPF procedure that allows for interactions between cell estimates, and that also provides for statistical inference in log-linear models. The relationships between the IPF and log-linear models is extensively discussed, among others, by Willekens (1980, 1982).

---

[4] The transportation problem is the problem of how to allocate amounts of goods from a group of "sources" to a group of destinations (Nemhauser et al., 1989).

3.3.2. Imposing on an initial population waiting times in initial states

So far, we have described data and methods to derive an initial population that resembles a population at a specific date, with a specific age structure and a specific multivariate distribution of individual attributes. To run the MicMac microsimulation, an initial population structured like this suffices. If we use non-homogeneous semi-Markov processes to describe individual life-courses, however, the situation differs: then we need additional information about the time that individuals have already spent in their initial states. If an initial population comes form register or longitudinal data, such information might be available.

When building an initial population, we know of no method that produces or estimates waiting times in a state: neglecting the duration that individuals have already spent in their current states when entering the virtual population is generally incorrect. In doing so, we would distort duration effects when determining the first event of members of the initial population. Liming et al. (2006) suggest a stochastic expectation-maximization (EM) algorithm for left-censored observations to iteratively impute unobserved duration times. A similar approach could also apply to the estimation of the waiting times in initial states. However, the design of such a method is not within the scope of this thesis. For simplicity reasons, we therefore suggest to assume waiting time distributions to describe the time that individuals have already spent in their initial states. These distributions can be used to randomly assign to each member of the initial population a waiting time variate. For example, we could employ the Hadwiger model (Hadwiger, 1940) and the Hadwiger mixture model (Chandola et al., 1999) to specify the time to first and second birth, or the Sickle model to describe the duration between marriage and divorce (Diekmann, 1990).

3.4. Migration data

To include migration into the microsimulation model, we need to know how many individuals enter the virtual population during simulation time. Respective numbers have to be given according to immigration year, age, sex, and state (= attributes) at immigration. If we additionally consider "duration-dependence," for each immigrant also the time that he/she has already spent in his/her state of immigration has to be provided. Furthermore, transition rates are needed that describe the propensity of immigrants to experience certain demographic events. Finally, emigration rates for individuals who are part of the virtual population are required.

Numbers of immigrants and their attributes

Each year European national statistical bureaus usually publish how many people have entered the country. These numbers are commonly provided (at least) according to nationality, age, and sex. As a superordinate body, EuroStat collects and publishes this data for a large number of European countries (EuroStat, 2010). That is, at least on a yearly base, immigration numbers are usually available. Information on other individual attributes of immigrants is not usually readily available. Data will have to be supplemented from social surveys, the national register system (if available), other statistical sources, or they may, at least partly, have to be based on estimates or assumptions. Similarly, assumptions about future immigration will have to be based on expert opinions or extrapolation of current trends.

Transition rates for immigrants and emigration rates

For administration purposes, all Western European countries collect time series of individual migrant data. Unfortunately, for research purposes this data is usually not accessible - at least not in its individual form. If it is made available, then it is mostly in the form of data aggregated over individuals. This complicates the estimation of transition propensities.

Almost a decade for now, new surveys have been created and existing ones have been extended to get deeper insights into the behavior and attitude of migrants. For example, in 2005, the German Mikrozensus got extended by a questionnaire directly addressing immigrants. Another example is the Netherlands Kinship Panel Survey (NKPS), which contains a sample that consists only of members of the four largest foreign ethnic groups of the Netherlands. If immigration data is available in form of individual data, transition rates can be estimated; e.g., based on occurrence-exposure rates (see Section 3.2.1). Small sample sizes often hinder a reliable study of the behavior of immigrants (Siegert, 2006). This implies that to estimate transition rates, we might have to set strong assumptions - which might remarkably bias the model outcome. Also in this case using register data can pose a remedy; as aforementioned for many applications, however, register data is out of range. To build scenarios of the future behavior of immigrants, current trends could be extrapolated, and experts might be asked.

For a gross of European countries emigration numbers are yearly published by EuroStat. Together with information provided in vital statistics and social surveys, this data can be used to determine emigration rates. Also in this case, expert opinions might serve as the best basis for the prospective development of emigration propensities.

4. Microsimulation Model as Dynamic System

A continuous-time microsimulation model specifies individual life-courses as sequences of discrete events that occur along a continuous time line. For implementation purposes, a model formalism that supports discrete event simulation seems therefore to be advantageous: on the one hand, already existing tools can be used for its implementation; i.e., no software has to be developed from scratch. Furthermore, we can benefit from well-proven and efficient up-to-date simulation methodology.

Possible candidates to realize a continuous-time microsimulation are the traditional formalisms of discrete event systems: DEVS (Zeigler et al., 2000), stochastic Petri Nets (Murata, 1989), and stochastic pi-calculus (Priami, 1995). Whilst both of the latter approaches rely on homogeneous continuous-time Markov chains, DEVS does not ask for a specific stochastic model to explain system behavior. Within this thesis, we deemed it therefore best to formulate our continuous-time microsimulation as a DEVS model.

The DEVS formalism offers all the functionalities that are necessary to formulate population dynamics in the requested way (cf. Chapter 2). Due to its modular and state-based concept, the formalism is well-suited to specify the kind of multi-state model that we use to describe individual life-courses. The DEVS approach is originated in systems theory and aims at a modular definition of hierarchical models. It defines real systems as composites of sub-models, which are either atomic or coupled. The formalism is proven, tested, and widely established (Labiche and Wainer, 2005; Uhrmacher et al., 2010; Zeigler, 2000). Several DEVS based tools exist (e.g., DEVSJava, JDEVS, CD++, DEVS variants in JAMES II) that support implementation.

To ease the modelling of problems and systems, various DEVS variants have been developed (Uhrmacher et al., 2010; Zeigler et al., 2000). For the demographic microsimulation without inter-individual relationships, we employ two different DEVS model variants: first, we use a classical atomic DEVS model to formulate population dynamics over time. To capture time-varying population sizes in a more sophisticated manner and to improve model performance, we have additionally developed a DYNDEVS network that allows a variable number of individual model components.

Essentially, the classical DEVS approach entirely suffices to describe our microsimulation model and its dynamics. Any DEVS extensions just conduces to the elegancy of the model specification. Generally, different DEVS specifications might suit to describe our microsimulation model. To ensure equivalence though, all of them have to be isomorph.[1] Uhrmacher (2001) has proven that this relationship holds between the original DEVS formalism and DYNDEVS.

The DEVS specifications detailed here rely on the work presented in Zinn et al. (2010). In this article we present a DEVS model that allows to specify a microsimulation model of the MicMac type, i.e., a model in which individual transition propensities do depend on age and calendar time, but not on the time that has been elapsed since a last transition. Within this thesis, we elaborate an extension of this DEVS model that additionally allows the consideration of waiting time dependencies of individual transition propensities.

4.1. An atomic DEVS model for demographic microsimulation

We formulate the microsimulation model described in Chapter 2 as an atomic DEVS model **Pop**$^+$. Its state is defined by the attributes of individual population members. Population dynamics are caused by state transitions and the arrival of immigrants. The process time t, to which we refer in the following, maps the time horizon $T = [0, t_e]$ of our population projection, i.e., the time span over which we observe the model population. **Pop**$^+$ is given by the structure:

$$\mathbf{Pop}^+ = \langle\ X_{IM},\ Y_{EM},\ \Theta,\ \omega_0,\ \delta_{\text{int}},\ \delta_{\text{ext}},\ \lambda\ , \text{ta}\ \rangle$$

where

$$X_{IM} = \begin{cases} [\psi_1^0, \cdots, \psi_{\text{im}}^0] & \text{if } im \text{ immigrants enter,} \\ \varnothing & \text{otherwise,} \end{cases}$$

ψ_i^0 $(i = 1, \cdots, \text{im})$ comprises the state s_0 and the age of an immigrant when he/she enters the population, as well as his/her birth date (s_0 is the initial state of that process $Z(t)$ that refers to an immigrant's life-course),

Y_{EM} is a output port for emigrants to leave the population,

$\Theta = \times_{k=1}^n \Psi_k \times \mathbb{R}_0^+$ is the state space of **Pop**$^+$, where n is the number of all individuals $\mathbf{I} = \{I_1, \cdots, I_n\}$ that ever belong to **Pop**$^+$ along model time t,

an element ϑ of Θ comprises the demographic characteristics $\times_{k=1}^n \Psi_k$ of all individuals of the virtual population at time t, and also the calendar time $C(t)$ at t.

$$\Psi_k = \begin{cases} \Psi \times \Psi \times \mathbb{R}_0^+ \times \mathbb{R}_0^+ \times \mathbb{R}_0^+ & \text{if } I_k \text{ is member of population at } t \\ 0 & \text{otherwise,} \end{cases}$$

Ψ is the state space of the process $Z(t)$ that maps the individual life-course

---

[1] The isomorphism between DEVS variants is called bi-simulation.

$\psi_k = [s_l, s_c, b, a, w]$, $\psi_k \in \Psi_k$, describes the last state s_l, the current state s_c $(s_l, s_c \in \Psi)$, the birth date b, and the age a of individual $I_k \in \mathbf{I}$, and the time w that I_k has already spent in s_c $(b, a, w \in \mathbb{R}_0^+)$, ($s_l$ and s_c are states of the process $Z(t)$ that refers to I_k's life-course) or indicates its non-existence: $\psi_k = 0$ $(k = 1, \cdots, n)$

$\psi_k^0 = [s_0, s_0, b, a_0 - w_0, w_0]$ gives I_k's initial state s_0, his/her birth date b, and his/her age a_0 at model initialization, and the time w_0 already spent in s_0, i.e., $a_0 - w_0$ is the age at which I_k entered s_0 (s_0 is the initial state of that process $Z(t)$ that refers to I_k's life-course)

$\vartheta = [\psi_1, \cdots, \psi_n]$ is the state of $\mathbf{Pop}^+$, where $\vartheta = [\psi_1, \cdots, \psi_{n_t}, 0^{\ (n-n_t)}]$, and n_t is the number of individuals that belong or have ever belonged to population until model time t,

$\vartheta_0 = [\psi_1^0, \cdots, \psi_n^0]$ is the initial state of $\mathbf{Pop}^+$ (ψ_k^0, $k = 1, \cdots, n$, is determined by initial population of the microsimulation),

$I_{k^\star}$ is the individual with the shortest waiting time:

$I_{k^\star} = \text{argmin}_{\ \{I_k \in \mathbf{I} | \psi_k \neq 0\}} \left(\min_{s_j \in \Psi \setminus \{s_c\}} (w_{s_c, s_j}), s_c \in \Psi \right)$

$s_{j^\star} = \text{argmin}_{\ s_j \in \Psi \setminus \{s_c\}} (w_{s_c, s_j}), s_c \in \Psi$, is the next state of $I_{k^\star}$

$\tau = w_{s_c, s_{j^\star}}, s_c \in \Psi$ is the related waiting time that from t onwards $I_{k^\star}$ remains in s_c until moving on to $s_{j^\star}$,

$\psi_{k^\star} = (s_c, s_{j^\star}, b, a + \tau, 0)$ gives $I_{k^\star}$'s next state and the corresponding age at transition,

$be : \Psi \times \Psi \to \{\varnothing, n_c\ newborns, emigration\}$ identifies birth and emigration events (birth events imply $n_c \in \mathbb{N}$ newborns),

$\psi_r = (s_l, s_c, b, a, w + \tau)$ updates the waiting time of individuals whose state is not affected by $\mathbf{Pop}^+$'s transition and who are part of the population at time t, $r = 1, \cdots, n_t, r \neq k^\star, \psi_r \neq 0$,

$\delta_{\text{int}} : \Theta \to \Theta$ is the internal state transition function of $\mathbf{Pop}^+$:

$$\delta_{\text{int}}(\vartheta) = \begin{cases} [\psi_1, \cdots, \psi_{k^\star}, \cdots, \psi_{n_t}, 0^{\ (n-n_t)}], \\ \qquad \text{if } bde(s_l^\star, s_{j^\star}) = \varnothing, \\ [\psi_1, \cdots, \psi_{k^\star}, \cdots, \psi_{n_t}, \psi_1^0, \cdots, \psi_{n_c}^0, 0^{(n-n_t-n_c)}], \\ \qquad \text{if } be(s_l^\star, s_{j^\star}) = \{n_c\ newborns\}, \\ \text{not defined, otherwise,} \end{cases}$$

where $\vartheta \in \Theta$, $s_l^\star$ describes the (now) last state of $I_{k^\star}$, and $[\psi_1^0, \cdots, \psi_{n_c}^0]$ comprises the birth dates and the states of newborns.

$\delta_{\text{ext}} : \Theta \times \mathbb{R}_0^+ \times X_{IM} \to \Theta$ is the external state transition function of $\mathbf{Pop}^+$:

$$\delta_{\text{ext}}(\vartheta, e, im) = \begin{cases} [\psi_1, \cdots, \psi_{n_t}, \text{IM}, 0^{\ (n-n_t-\text{im})}], & \text{if } im \text{ immigrants enter } \mathbf{Pop}^+ \\ \text{not defined, otherwise,} \end{cases}$$

where $\vartheta \in \Theta$, e gives the time elapsed since the last transition, $e \in (0, \tau)$, and $\text{IM} = [\psi_1^0, \cdots, \psi_{\text{im}}^0]$ comprises the birth dates of the immigrants, their states and ages at immigration,

$ta : \Theta \to \mathbb{R}_0^+ \cup \{\infty\}$ is the time advanced function: $\text{ta}(\vartheta) = \tau$

$\lambda : \Theta \to Y_{EM}$ is the output function, $\vartheta \in \Theta$:

$$\lambda(\vartheta) = \begin{cases} \psi_{k^\star} & \text{if } be(s_c, s_{j^\star}) = emigration, \\ \varnothing & \text{otherwise.} \end{cases}$$

Immigrants will be generated by the model *OUTSIDE*, to which the population model is coupled (cf. Figure 4.1). *OUTSIDE* maps all those populations that are not described $\mathbf{Pop}^+$. The generation of immigration events relies on empirical data about immigration dates and number of immigrants. The *OUTSIDE*'s outputs serve as an input to our population model. Several immigrants may enter the population simultaneously; e.g. family members or couples. Via $\mathbf{Pop}^+$'s output port Y_{EM}, information about emigrants are sent to the *OUTSIDE* model. Modelling migration in this way allows extending the model to study the migration behavior between different populations. For this purpose, different population models would have to be coupled. However, this is not within the scope of this thesis.

Attributes and behavior of individuals are captured by $\mathbf{Pop}^+$'s state ϑ. Each of its components refers to an individual who was, already is, or will at some time be part of the virtual population. Components of ϑ that refer to individuals who at time t have already left the population (due to death or emigration) are set to their last state, i.e., 'dead' or 'emigrated'. A zero value is assigned to individuals that are not yet born or immigrated.

Changes in individual attributes (provoked by state transitions of the corresponding stochastic processes) result in an internal event of $\mathbf{Pop}^+$. Here the rule applies that the next event is triggered by the earliest state change of an individual. Using the algorithm given in Chapter 2, from t onwards waiting times are computed for all individuals I_k, $k = 1, \cdots, n_t$, that are part of the virtual population.[2] To clarify this approach, we detail the related processing: at process time t, the state ψ_k of the individual I_k shows that he/she is a years old, and he/she stays already w time units in his/her current state s_c. Using I_k's birth date b and his/her current age a we can compute the calendar time c that refers to the current process time t: $c = b + a$. By means of these quantities, based on equation (2.7) (on page 15), we can determine random waiting times to all possible destination states s_j of I_k, $s_j \in \Psi_k$.[3] Among these waiting times,

---

[2] Individuals who are dead or who have emigrated from the virtual population are assumed to never leave the 'dead' or the 'being emigrated' state, i.e., to them we assign infinite waiting times.

[3] If we apply instead of non-homogenous semi-Markov processes non-homogenous continuous-time Markov chains for the description of individual behavior, we use equation (2.4) on page 13 to determine random waiting times.

the shortest one is chosen. The same computation is repeated for all individuals that are part of **Pop**$^+$ at t. Subsequently, among all shortest waiting times, the individual with the shortest waiting time is identified and the associated transition is performed. Let $I_{k^\star}$ denote this individual. If $I_{k^\star}$'s state change does not imply that he/she will leave the population nor that (new) individuals will enter it (because $I_{k^\star}$ experiences a childbirth event), just the affected component of ϑ will be replaced by $I_{k^\star}$'s new state and the respective age at transition. If a (female) individual experiences a birth event, we add as many children to the population as the woman gives birth to. The state of **Pop**$^+$ is updated accordingly by assigning initial attributes to the components that refer to the newborns.

To keep track of the time that has been elapsed since the last transition of an individual, at each state transition of **Pop**$^+$ the waiting time and the age of individuals who are not affected by the current event have to be updated.

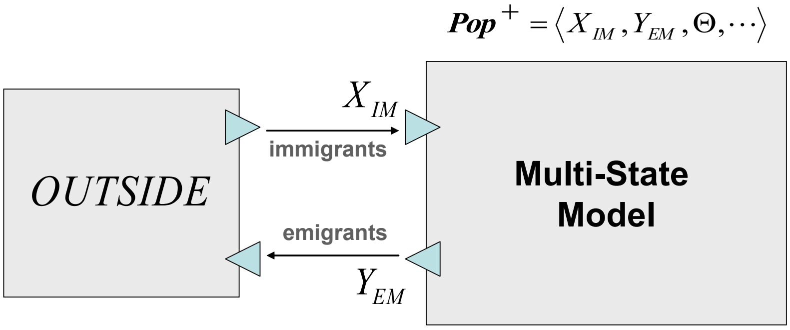


Figure 4.1.: The atomic model **Pop**$^+$ comprises the multi-state model described in Chapter 2. The model *OUTSIDE* maps the 'world outside'. It creates the attributes of immigrants that enter the population. Via **Pop**$^+$'s output port Y_{EM} the attributes of emigrants are forwarded to *OUTSIDE*.

Whereas the DEVS atomic model allows a one to one translation of the stochastic models of Chapter 2 (see also Zinn et al. (2010)) and provides additional structure by distinguishing nicely between immigration and the internal dynamics of our population, certain problems also remain. As individuals might enter and leave the population (due to migration, birth or death), the number of population members changes over time. The classical DEVS approach requires, however, a state space that is time invariant. For the **Pop**$^+$ model, at the time of model initialization the number of all individuals who might join the population up to time t_e needs to be assessed. As childbirth, emigration, or death rely on random processes, this is rather difficult. Even worse, immigrants might enter the population at any time during simulation. In the above vector representation, the problem is being addressed by setting n (the amount of potential population members) to a large number that is unlikely to be exceeded during the simulation up to time t_e. This is nevertheless neither a general nor a (space-) efficient solution.

In addition, each simulation step requires an update of the time that an individual has already spent in his/her current state. This is very inefficient: it would completely suffice to compute a new waiting time only if an individual has just experienced an event. Using instead of an atomic model a DEVS network model facilitates such a processing.

Besides this, in the current model, no relationships between individuals are considered, implicitly assuming that the individual life-courses can be calculated independently from each other. This a very strong assumption that might lead to distorted simulation outcome. Examples are simulations of marital patterns or kinship networks (Murphy, 2003; Ruggles, 1993). A generic approach like our microsimulation model should provide suitable means to take interrelations between individual life-courses into account. This is not the case, if a population is described as one monolithic atomic DEVS model, as we have proposed in this section. Using an atomic DEVS model for each individual would enable us to describe the relation between individuals by explicit couplings between atomic models. However, neither disappearing and appearing atomic models nor changing interactions between models are supported in classical DEVS. Both observations ask for a different approach.

4.2. A dynPDEVS network for demographic microsimulation

DYNDEVS has been developed to support variable structure models, i.e., models that change their own composition, interaction, and behavior pattern (Uhrmacher, 2001). In DYNDEVS, a model's description entails the possibility to change its own state and behavior pattern. Models are interpreted as a set of models successively generating themselves by model transitions that map the current state of a model into a set of models that the model belongs to. A sequence of model incarnations is thereby produced. In the following, we employ a revised and parallel version of DYNDEVS, DYNPDEVS, (Uhrmacher et al., 2010) to describe the virtual population. The network or coupled model represents the population. Associated network components represent a population's individual members. In doing so, we account for both the discrete event conception of the microsimulation and for the temporal variation in the corresponding population. Individual behavior – and therefore population dynamics – emerge due to state transitions of individual models.

The dynamic DYNPDEVS network **Pop** describes the virtual population of the microsimulation described in Chapter 2. Its atomic components are 'individual models' denoted by I. At process time t, $t \geq 0$, the population consists of n_t, $n_t \in \mathbb{N}$, individuals. The set $\mathbf{I}$ comprises all individual models of **Pop**.

The special structure of our microsimulation model implies that not the entire functionality of the original DYNPDEVS formalism is engrossed; e.g., we do not employ any couplings between micro models. Therefore, to be clear, we adapt the original DYNPDEVS approach such that we let out functionality that we do not demand.
We formulate the **Pop** network as structure:

$$\mathbf{Pop} = \langle X_{sc}, Y_{sc}, \mathrm{pop}_{\mathrm{init}},\ \Gamma(\mathrm{pop}_{\mathrm{init}})\rangle$$

where

$$X_{sc} = \begin{cases} [\psi_1^0, \cdots, \psi_{\text{im}}^0] & \text{if } im \text{ immigrants enter,} \\ \varnothing & \text{otherwise,} \end{cases}$$

ψ_i^0 ($i = 1, \cdots, \text{im}$) comprises the state s_0 and the age of an immigrant when he/she enters the population, as well as his/her birth date (s_0 is the initial state of that process $Z(t)$ that refers to an immigrant's life-course),

Y_{sc} is a output port for emigrants to leave the population,

$\text{pop}_{\text{init}} \in \Gamma(\text{pop}_{\text{init}})$ describes the initial population model,

$\Gamma(\text{pop}_{\text{init}})$ is the set of all possible incarnations of the population model $\langle \mathbf{I}, \rho_{\text{pop}} \rangle$,

$\mathbf{I}$ is the set of all individual DYNPDEVS models $\langle Y_{\text{sc}}^{bde}, I_0, I \rangle$,

Y_{sc}^{bde} is the structural output port via which the structural changes like death, emigration, or birth are signalized to the network,

I_0 is the initial individual model,

I has the structure $\langle \Psi_I, \psi_0, \delta_{\text{int}}, \rho_\lambda, \text{ta} \rangle$, where

$\Psi_I = \Psi \times \Psi \times \mathbb{R}_0^+ \times \mathbb{R}_0^+ \times \mathbb{R}_0^+$ set of possible states that I can occupy,

Ψ is the state space of the stochastic process that maps the individual life-course,

$\psi = [s_l, s_c, b, a, w]$, $\psi \in \Psi_I$, is a tuple that comprises I's last state s_l, his/her current state s_c ($s_l, s_c \in \Psi$), the birth date b and the age a of I, and the time w that I has already spent in s_c ($b, a, w \in \mathbb{R}_0^+$),

$\psi_0 = [s_0, s_0, b, a_0 - w_0, w_0]$, comprises I's initial state s_0, his/her birth date b, his/her age a_0 at model initialization, and the time w_0 already spent in s_0, i.e., $a_0 - w_0$ is the age at which I entered s_0 (s_0 is the initial state of the stochastic process that refers to I's life-course)

$\delta_{\text{int}}(s_l, s_c, b, a, w) = (s_c, s_{j^\star}, b, a + \tau, 0)$, $\delta_{\text{int}} : \Psi_I \to \Psi_I$, is the internal state transition function of I, where $s_{j^\star} = \text{argmin}_{\, s_j \in \Psi / \{s_c\}}(w_{s_c, s_j})$ is the next state of I, and $\tau = w_{s_c, s_{j^\star}} = \min\{w_{s_c, s_j}\}$ is the (remaining) waiting time of I in s_c until going on to $s_{j^\star}$

$\rho_\lambda : \Psi_I \to Y_{\text{sc}}^{bde}$ forwards structural model changes of I:

$$\rho_\lambda(\psi) = \begin{cases} n_c, & \text{if } bde(s_l, s_c) = n_c \; newborns, \\ true, & \text{if } bde(s_l, s_c) = death, \\ \psi, & \text{if } bde(s_l, s_c) = emigration, \\ \varnothing, & \text{otherwise,} \end{cases}$$

with $bde : \Psi \times \Psi \to \{\varnothing, n_c \; newborns, death, emigration\}$,

$ta : \Psi_I \to \mathbb{R}_0^+ \cup \{\infty\}$ is the time advanced function of I: $ta(\psi) = \tau$,

$\rho_{\text{pop}} : \times_{i \in \mathbf{I}} \oplus_{i \in \mathbf{I}} Y_{\text{sc}}^i \times X_{sc} \to \Gamma(\text{pop}_{\text{init}})$ is the network transition function of **Pop**,

ρ_{pop} takes into account

(i) the arrival of immigrants: $X_{sc} \neq \varnothing$ (causes the creation of new individual models),

(ii) the announcement of childbirth(s) (and therefore the creation of n_c new individual model(s), $n_c \in \mathbb{N}$),

(iii) the announcement of the extinction of a model (due to emigration or death).

The **Pop** network exhibits one input port X_{sc} for receiving information about the arrival of immigrants. It further possesses an output port Y_{sc} to let emigrants leave the population. Such a modelling ensures that migration flows between different populations can directly be modelled by coupling population networks. The **Pop** network does not exhibit any ordinary input ports X or output ports Y.

Pop comprises a set Γ of all possible model incarnations, including the initial populations pop_{init} of the model.

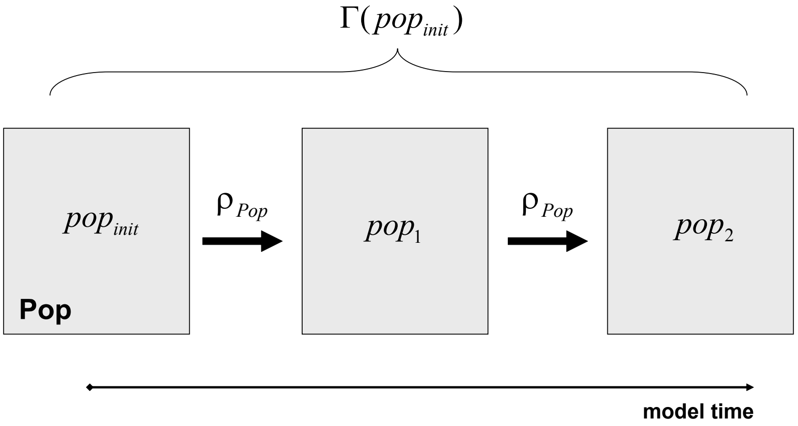


Figure 4.2.: The set Γ contains all possible model incarnations of the population network **Pop**. Network transitions are handled by the function ρ_{pop}. Beginning with the initial population model pop_{init}, every time a network transition is caused, a new network incarnation is created.

Each individual model is equipped with an internal transition function δ_{int} that handles all of its state transitions. Furthermore, it incorporates the time advance function ta that determines the corresponding series of waiting times in states. If and when events occur during an individual life-course is specified by the stochastic model that is described in Chapter 2. When determining next events, a difference is made between the initialization phase and subsequent transitions: in the initialization phase, an internal event is computed for an individual for the first time. This implies that the time that an individual has already spent in his/her current (initial) state s_0 is usually bigger than zero, i.e., $w > 0$. After having initialized all individuals, new waiting times to next events are computed exactly when an event has been executed; i.e., in this moment the waiting time variable is always $w = 0$.

The execution semantics of DYNPDEVS ensures that among all possible next state transitions that one always is realized for which the shortest waiting time has been determined.

If an upcoming transition of an individual model is related to a birth event that causes n_c newborns, function ρ_λ signalizes to output port Y_{sc}^{bde} the implied structural network changes. Likewise, ρ_λ forwards to Y_{sc}^{bde} an imminent death or emigration event.

Subsequent changes in the structure of **Pop** and, accordingly, the creation of new model components and the deletion of existing ones is executed by the model transition function ρ_{pop}. The network transition function ρ_{pop} is activated for eliminating or adding model components.

Whereas the elimination of components is only caused by information received via individuals' structural output ports (death and emigration events), the adding of components is triggered by both information received via individuals' output ports (birth events) and external events (arrival of immigrants). The network transition function accordingly possesses two arguments: (1) information received via individuals' structural output ports and (2) the immigrants that enter the population. Figure 4.2 illustrates the role of ρ_{pop}. An example situation is depicted in Figure 4.3.

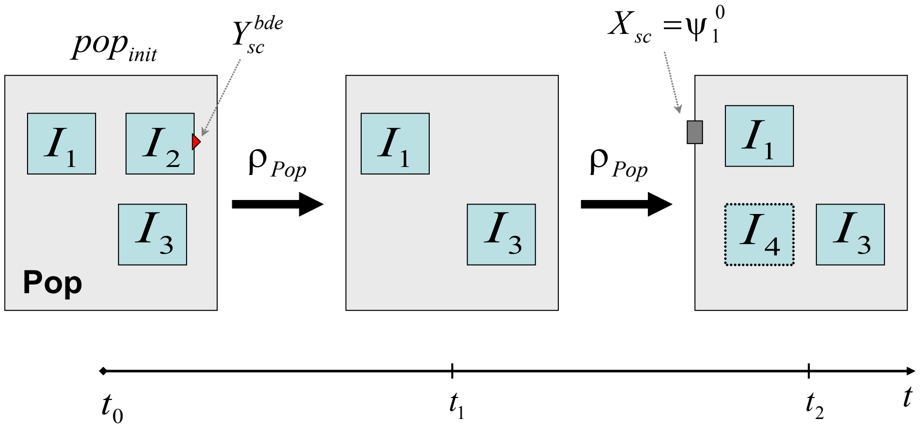


Figure 4.3.: Function ρ_{pop} allows for modelling a time-varying number of individuals included in **Pop**. Of the initial population pop_{init} individual I_2 dies at time t_1. The individual model informs via its output port Y_{sc}^{bde} the population **Pop**. Subsequently, function ρ_{pop} performs the deletion of individual I_2. Given that an immigrant arrives at time t_2, ρ_{pop} triggers the creation of model I_4 for specifying the newly arrived individual.

Looking at the individuals, we see simplifications of the typical atomic DYNPDEVS model. As the dynamics of each individual is currently solely driven by its internal dynamics, some functionality of DYNPDEVS is not exploited: in particular, X, Y, X_{sc}, λ and δ_{ext} are not required so far, and are therefore not defined. Of course those definitions are needed, as soon as linkages between individuals are included; cf. Part III of this thesis. In addition, it is not foreseen that the individuals themselves change their own behavior. Thus, the set of incarnations (that typically represents an atomic DYNPDEVS model) boils down to one individual. Consequently, the model transition function ρ_α of the original DYNPDEVS becomes obsolete.

4.3. Execution semantics of the dynPDEVS network

We deem the DYNPDEVS network model specification appropriate for our purpose. Therefore, we subsequently rely on this model, and do not further elaborate on the atomic DEVS model described in Section 4.1.

The realization of a discrete event simulation requires the employment of an event queue, which is a list of events sorted by their scheduled event times. The event queue of a continuous-time microsimulation holds the events that are scheduled for the individuals of the virtual population (at most one event per individual). In each step, the event with the minimal time stamp is dequeued; if the event is not death or emigration for the related individual, a new event is computed and enqueued. Therefore, roughly speaking, an event queue organizes the scheduling of upcoming events. An efficient model execution demands an event queue that is optimal for the application at hand. In Section 7.1.3 we test different types of event queues to identify those which suit best to a demographic microsimulation.

For the processing of events, executable simulation code has to be derived; i.e., simulation semantics has to be specified. The simulation semantics of DEVS models is described by *simulators* and *coordinators*. Simulators correspond to atomic models and coordinators to coupled models/network model; cf. Figure 4.4. A coordinator implements the event queue algorithm while guaranteeing the correct synchronization of its components. It holds an event queue for the next event times of its component simulators. Coordinators are responsible for the correct synchronization of the component simulators and also for the handling of external events (in our case: the arrival of immigrants). To perform this task, a coordinator uses communication protocols: it waits for protocols sent by its subordinates and transmits them to its parent; if a model consists of only one coupled model, as in the case of **Pop**, protocols are directly forwarded to the root-coordinator. The root-coordinator guides the overall simulation processing. It initializes a new simulation and instructs the model execution until some termination criteria is met. To each DEVS model, a corresponding processor tree can be given that directly maps the hierarchical structure of coupled model on the architecture of simulators and coordinators. Figure 4.4 displays the processor tree of the population network **Pop**.

4.3.1. An abstract simulator for dynPDEVS

Each DEVS variant features its own *abstract simulator* that describes its general execution semantics (Zeigler et al., 2000). We subsequently detail the abstract simulator of the DYNPDEVS model while relying on the specification given in Himmelspach (2007, Chapter 4).

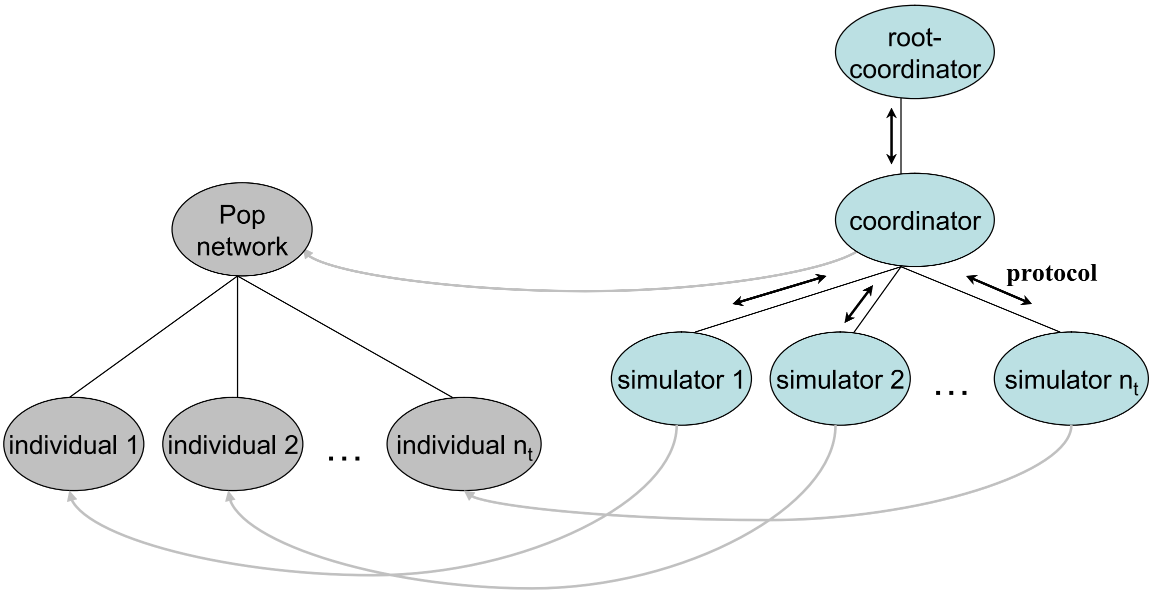


Figure 4.4.: Each DEVS model can be mapped into a processor tree: to each individual model of the DYNPDEVS population network **Pop** corresponds a simulator, and to the network a coordinator. The root-coordinator monitors the overall execution of the simulation. The communication between the different levels of the tree is conducted based on protocols.

Notation	***Message Type***
$\star$-message	activation message, message indicating that an internal event is due
y-message	output message
x-message	input message
sc-message	message about structural change(s)
done-message	termination message

Table 4.1.: DYNPDEVS message notation.

Our description is adapted to the structure of the population network **Pop**; concerning general information about simulation schemes of DYNPDEVS, we refer to Himmelspach (2007, Chapter 4) and Uhrmacher (2001).

To ensure consistency within a simulation step, messages are processed in a well-defined order (Table 4.1 summarizes the different types of messages): if a $\star$-message activates a model, a y-message with output data is forwarded. Afterwards, the simulator waits for a x-message comprising input information. Structural model changes provoked by a $\star$-messages or a x-message are forwarded via a *sc*-message to the parent coordinator. In response, the parent coordinator might likewise instruct structural changes to the model (via sending a *sc*-message). Finally, a *done*-message signalizes the completion of a simulation step. In Figure 4.5 the corresponding communication protocol is illustrated.

The abstract simulator of a DYNPDEVS model comprises three algorithms: the algorithm

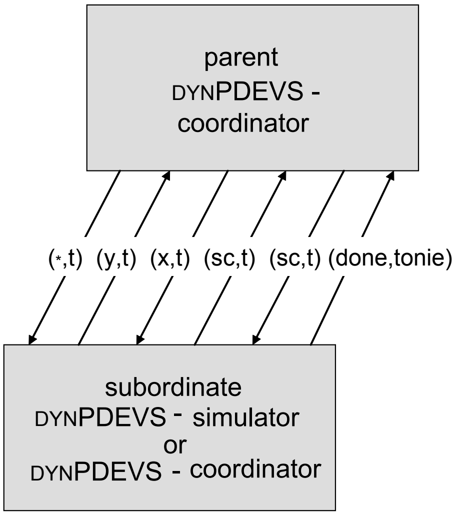


Figure 4.5.: DYNPDEVS communication protocol. The ⋆-, y-, x-, and *sc*-message comprise beside their regular information also the actual simulation time t, the *done* message comprises the time *tonie* of the next internal event.

that describes the processing of the root-coordinator (cf. Algorithm 1), the algorithm that specifies the processing of a coordinator (cf. Algorithm 2), and the algorithm corresponding to the simulator of a DYNPDEVS model (cf. Algorithm 3).

The root-coordinator initializes a simulation by sending a ⋆-message to all subordinated coordinators and simulators. It waits afterwards for a y-message (output information of subordinated models). Subsequently, the root-coordinator sends an empty x-message (input information) to its child. An exchange of information about structural changes (via *sc*-messages) follows. Finally, the root-coordinator receives the time of the next simulation step (*tonie*). As long as the end of simulation is not reached (i.e., eos is true), the root-coordinator instructs further simulation steps. In the case of **Pop**, individual models do not cause any output; i.e., in our setting only empty x- and y-messages are forwarded.

Algorithm 1 Pseudocode of the root-coordinator of DYNPDEVS.

```
while not eos do
  send $^\star$ message to child
  wait for receive $y$ message from child
  send $x$ message to child
  wait for receive sc message from child
  send sc message to child
  wait for receive done message including tonie from child
end while
```

The DYNPDEVS coordinator instructs the processing of coupled models/network models, cf. Algorithm 2. Guided by a ⋆-message, it activates the simulators of those subordinate models that undergo internal events (individual state transition) at actual simulation time. Such models are called ‘imminent’. A coordinator can also be activated by receiving a x-

message sent by another imminent model which is not subordinated. We call models that are reached by a x-message 'influencees'.

After having prodded its imminent children into action, a coordinator waits for their y-messages (i.e., output information). Having received them, the coordinator then forwards the y-messages to its superordinate node. As soon as the superordinate node holds the information, it forwards x-messages to all affected coordinators and simulators. The coordinator waits for the x-message of its superordinate node and forwards it to all influenced or imminent children. The latter receive an empty x-message. Now, the coordinator waits for the *sc*-messages (i.e., information about structural changes) of its influenced and imminent children, and forwards these to its parent. The parent responds by sending a *sc*-message about structural changes to the coordinator which the coordinator in turn forwards to all influenced, imminent, or targeted children.

Subsequently, the coordinator waits for the *done*-message of all activated children, including their time to the next event (*tonie*). Then it processes structural changes. In a final step, the coordinator computes the minimal time to next event (*tonie*) and sends it to its superordinate knot. As in **Pop** individual models do not generally receive any input or produce any output, x and y messages are always empty. The processing of the DYNDEVS coordinator breaks down to the execution of internal events and the handling of structural changes.

To avoid logical problems when processing structural messages, Himmelspach and Uhrmacher (2004) recommend that they be previously arranged. They suggest ordering them according to: (1) creation of models, (2) creation of ports, (3) creation of couplings, (4) deletion of couplings, (5) deletion of ports, and (6) deletion of models.

Algorithm 2 Pseudocode of the coordinator of DYNPDEVS.

```
when receive $\star$ or $x$ or $sc$ message
if message is $\star$ or $x$ message then
  if message is $\star$ message then
    send $\star$ message to imminent children
    wait for receive $y$ messages from the imminent children
    send $y$ message to parent
    wait for receive $x$ message from parent
  end if
  send $x$ messages to all influenced or imminent children
  wait for receive $sc$ message from all influenced or imminent children
  send $sc$-message to parent
  wait for receive $sc$ message from parent
end if
send $sc$-message to all influenced, imminent or $sc$ targeted children
wait for receive done messages including tonie from all influenced, imminent or $sc$ targeted children process $sc$ message update tonie
send done-message including tonie
end when
```

Each atomic DYNPDEVS model can be invoked by a $\star$-message, a x-message, or a *sc*-

message. The first case concerns situations when the event time of a model is due, the second case indicates the arrival of some input information, and the third case is raised if structural changes are instructed. If a model is activated by a $\star$-message, the output function of the model is executed and output information is sent. In our case, an individual model does generally not produce any output, i.e., output streams are always empty. After having sent the output information, the DYNPDEVS model waits for a x-message. If it receives an empty message, it conducts the internal transition function and the confluent transition function, respectively. Otherwise, if the x-message that a DYNPDEVS model receives is not empty, its external transition function is triggered. Structural changes that might result from state transitions are forwarded to the superordinate knot, and instructions implying own structural changes are awaited. If the simulator of a DYNPDEVS model receives thereafter a non-empty *sc*-message, structural model changes are conducted based on the content of the *sc*-message. After having executed all state transitions and structural changes, the time to the next event (*tonie*) is computed, and a *done*-message is sent to the superordinate coordinator. For **Pop**, neither a confluent transition function is defined (as the probability of concurrent internal and external events is practically zero) nor an external transition function is needed (an individual model does not receive any input messages), that is, after an (empty) x-message always follows the execution of the internal transition function. Structural changes provoked by an individual model are either birth events, or emigration or death events. The latter always cause the deletion of the respective model, and the first the creation of new models. Algorithm 2 describes the processing of the simulator of an atomic DYNPDEVS model.

Algorithm 3 Pseudocode of the simulator of DYNPDEVS.

```
when receive $\star$ or $x$ or $sc$ message
if message is $\star$ or $x$ message then
  if message is $\star$ message then
    execute model.lambda
    send $y$ message to parent
    wait for receive $x$ message
    if isEmpty ($x$-message) then
      execute model.deltaInt
    else
      execute model.deltaCon
    end if
  else
    execute model.deltaExt
  end if
  execute model.timeAdvance
  send $sc$ message to parent
  wait for receive $sc$ message
end if
process $sc$ message
send done message including tonie
end when
```

4.3.2. Limitations of a direct implementation of the abstract simulator

Normally, a direct realization of the abstract simulator of a DEVS variant is used as a processing scheme (Zeigler et al., 2000). Himmelspach and Uhrmacher (2006) elaborate problems of such an approach and discuss ideas to overcome limitations. Their critique mainly concerns two aspects: first they deplore the number of messages that coordinators and simulators exchange for communication purposes. Depending on the depth of the processor tree and the number of models, passing messages through the processor tree can result in a remarkable overload. The second critique concerns the resources that are needed to execute a simulation algorithm like the one specified in Algorithm 3. Depending on the used computing environment, executing a simulator in this way would at least demand one process or one thread per model scheme.[4] In our case, this implies that with **Pop** meaningful population projections would be intractable. In more detail: to achieve reliable population projections our microsimulation should at least include one percent of the mimicked population. In the case of the contemporary Netherlands, one percent of the population constitutes around $130,000$ individuals; i.e., to execute our model we need at least $130,000$ processes/threads. A common system is, however, usually not able to manage such an enormous amount of workload. For example, standard implementations of the Java Virtual Machine (JVM) can handle at the utmost a few thousand threads simultaneously. That is, a direct implementation of the abstract simulator is not necessarily the most efficient way to compute a model. To speed up the model execution, Himmelspach and Uhrmacher (2006) propose for the parallel version of DEVS, PDEVS, a family of simulation algorithms which sequentially compute (parts of) the processor tree. An extension to dynDEVS is straightforward: nothing but the sending/receiving cycle and the processing of *sc*-messages have to be added.

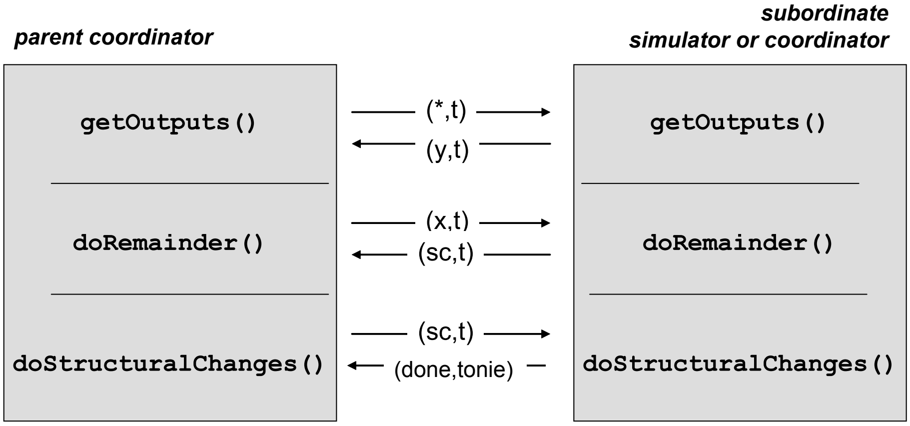


Figure 4.6.: Processing scheme of the sequential abstract simulator of dynPDEVS.

---

[4] In computer science, a process is the actual execution of the instructions given in a computer program. By definition a thread is a light weight process, or more clearly: within a program a thread is a sequentially executed stream of instructions.

Algorithm 4 Pseudocode of the coordinator of the 'sequential abstract simulator for DYNPDEVS'.

```
function Messages getOutputs()
    for each child in imminents do
        allMsgs = union (allMsgs, child.getOutputs())
    influencees = propagateInputCouplingMessages(allMsgs)
    allMsgs = getOutputCouplingMessages(allMsgs)
    return allMsgs
end function

function StructuralMessages doRemainder(msgs)
    influencees = union (influencees, propagateInputOutputCouplingMessages(msgs))
    for each child in union (influencees, imminents) do
        strMsgs = union (strMsgs, child.doStructuralChanges())
    return strMsgs
end function

function double doStructuralChanges(strMsgs)
    influencees = union (influencees, propagateStrMessages(strMsgs))
    for each child in influencees do
        setTonie(child, child.doRemainder())
    return getMinTonie
end function
```

4.3.3. A sequential abstract simulator for dynPDEVS

In this section, we present an extension of one of the sequential simulation algorithms developed by Himmelspach and Uhrmacher (2006) for PDEVS. We have worked out an algorithm for DYNPDEVS that relies on a processing scheme that they have called "sequential abstract simulator."

The sequential abstract simulator for DYNPDEVS maintains the basic concept of the classical DEVS coordinators and simulators, inclusive their communication scheme. It modifies the simulation processing, however, by splitting it into three subsequent steps. These steps are implemented in form of the three methods `getOutputs`, `doRemainder`, and `doStructuralChanges`; cf. Figure 4.6. During simulation processing, these methods are successively called.

The methods `getOutputs`, `doRemainder`, and `doStructuralChanges` are differently realized by coordinators (cf. Algorithm 5) and simulators (cf. Algorithm 4). By `getOutputs`, a coordinator activates its imminent children and propagates their outcome messages according to the momentary coupling structure. By `doRemainder`, it informs imminent or influenced children about external events and waits for structural changes communicated by them. Structural changes are then forwarded to the parent. Finally, by `doStructuralChanges`, a coordinator processes structural changes to its subordinates and determines the system's next event time.

A simulator employs `getOutputs` to call the model's output function and to forward the respective output information. By `doRemainder`, a simulator computes (internal or external) events and communicates resulting structural changes to its superordinate coordinator. By `doStructuralChanges`, it handles structural changes that are instructed by the superordinate coordinator. Afterwards, by `doStructuralChanges`, a simulator updates the time advanced function of the model and forwards its next event time to its superordinate coordinator.

The presented processing scheme requires one thread/process per coupled model/network model. Therefore, it is alternatively called a non-threaded version of the (classical) abstract simulator (Himmelspach and Uhrmacher, 2006). Using the sequential abstract simulator to execute **Pop** means that independently of the population size we only need two threads/processes (one for the network and one for the root-coordinator). That is, opposed to a direct implementation of the abstract simulator of the DYNPDEVS, the sequential abstract simulator allows us to conduct population projections of meaningful size.

Algorithm 5 Pseudocode of the simulator of 'the sequential abstract simulator for DYNPDEVS.'

```
function Messages getOutputs()
    execute model.lambda
    return getMsgs(model)
end function

function StructuralMessages doRemainder(msgs)
    strMsgs = model.stateTransition(msgs)
    return strMsgs
end function

function double doStructuralChanges(strMsgs)
    model = model.processStructuralChanges(strMsgs)
    execute model.timeAdvance
    return $tonie$
end function
```

5. Simple Implementation Using JAMES II

The microsimulation model described in Chapter 2 can also be directly translated into source code, without specifying it beforehand as a discrete event system. Such an immediate implementation, however, poses two main drawbacks: on the one hand, a lot of functionality has to be implemented from scratch, and well-proven and widely established simulation methodology is not accessible without further ado. On the other hand, because the simulation model and its execution semantics are directly translated into source code, model extensions (e.g., the inclusion of inter-individual relationships) are hindered. We address the first problem by implementing the microsimulation using the M&S framework JAMES II as a library. The second problem could be addressed by describing a simulation model using a model specification formalism. A model extension means then to either extend the original model specification or to move on to a variant of the formalism. In any case, this way we circumvent messing up source code by subsequently adding new functionality, and we avoid any redevelopments. For our microsimulation application, we deem the DEVS language suitable to such a purpose.

5.1. JAMES II: general overview

The most important idea while developing JAMES II (available at `www.jamesii.org`, accessed December 2010) was to create a framework that allows us to create more specialized applications by extending it. JAMES II has been created at the University of Rostock from 2003 on. It is an open source project, and thus is available at no costs. It runs on top of the Java platform (and therefore on all machines for which a Java Virtual Machine (JVM) exists), and it is not bound to any modelling or simulation execution paradigm, nor to any formalism or language. As a framework, it can be "transformed" into specialized applications meeting the needs of certain use cases. Any product created on top of the framework can be easily extended later on; e.g., by adding a new simulation algorithm to make use of multi-core CPUs – without any need to modify the application created.

JAMES II is a M&S framework based on the "Plug'n simulate" concept Himmelspach and Uhrmacher (2007b). It exhibits a highly comprehensive work bench, it holds different types of data sinks (e.g., files, data base), event queues, random number generators, high quality up-to-date M&S functionality, like techniques for distributed computation; it supports M&S workflow requirements, and additionally offers tools for optimization, and statistical analysis. To borrow functionality from other software, JAMES II permits interfaces; e.g., to the statistical environment R (see for more details `http://cran.r-project.org`, accessed December 2010). The default GUI (Graphical User Interface) of JAMES II can be used as a framework for the creation of a more specialized GUI, or a own GUI can be created from scratch.

JAMES II shall ease the creation of specialized M&S applications. These applications can thereby either be created by extending the framework directly or by building the application on top of the framework; i.e., by using the framework as a "M&S service".

5.2. Implementation in JAMES II

To implement the simulation engine of the microsimulation of the MicMac project, we have used JAMES II as a library of methods. Our implementation consists of two plug-ins: a modelling plug-in and a simulator plug-in. A model based on the modelling plug-in is being composed of two entity types:

1. the *population*, which comprises a set of *individuals*, and:
2. the *statespace*, which holds the structure of the model state space.

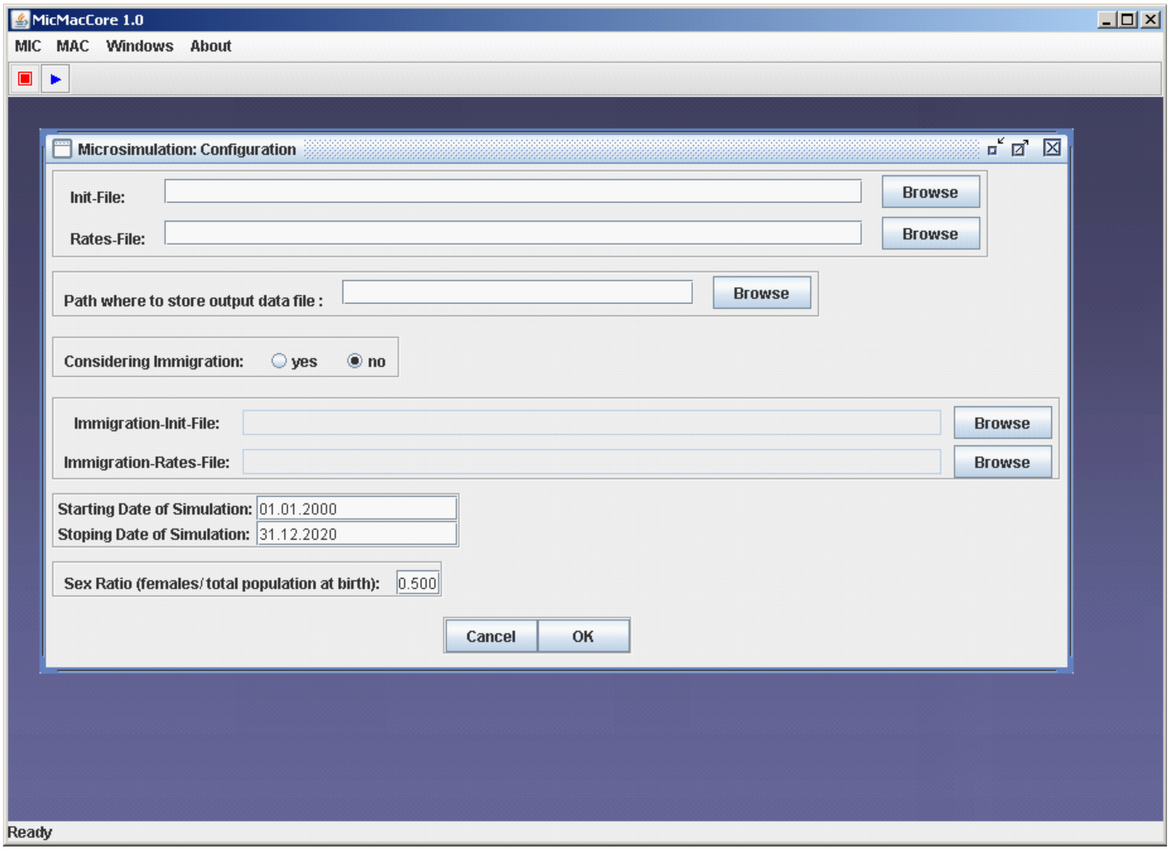
Models are not completely implemented in source code (only the two basic classes are coded): models are described by creating rate matrices and an initial population that can stem from any source (such a source might be the statistical package R), which are then read by an instance of the model reading mechanism of JAMES II.

The simulator plug-in is based on the classical "hold" loop. In an event queue, all scheduled events for the individuals are held (at most one event per individual). In each step, we have to dequeue the event with the minimal time stamp, compute the state transition according to the model's state transitions, and enqueue a new event for this individual. The template for simulation algorithms in JAMES II requires to implement this functionality in the `nextStep` method. Everything else (i.e., run control, looping, and so on) is automatically handled by corresponding classes in JAMES II. For our application, the algorithm is simple:

```
public void nextStep() {
  Event event = eventQueue.dequeue();
  List newEvents = computeTransition(event);
  for each newEvent in newEvents do
    eventQueue.enqueue(newEvent);
  }
}
```

Events denote state transitions happening to specific individuals. The `computeTransition` method determines the new events to be scheduled. It incorporates the simulation algorithm described in Chapter 2. Please note that individuals might be added or be removed from the population, and thus, if individuals are added, more than one new event might be returned;

or none, if an individual is removed. Subsequently, after a completed simulation run, the simulation output is stored using an adequate data sink plug-in that is provided by JAMES II. More details on this implementation of the (simulation engine of the) MicMac microsimulation is given in Zinn et al. (2009). The source code is attached to this thesis.



Considering Immigration: ○ yes ● no

Starting Date of Simulation: 01.01.2000
Stoping Date of Simulation: 31.12.2020

Sex Ratio (females/total population at birth): 0.500

Figure 5.1.: Using the GUI of the MicMac software, the user can enter all instructions that are needed to run a microsimulation.

5.3. Workflow of the MicMac microsimulation

Before performing the actual microsimulation, the model has to be specified and parameters (i.e., transition rates and an initial population) need to be estimated. Furthermore, the analysis of the simulation output should be feasible as well. Both the modeling and the analysis should be done in a sophisticated manner. The statistical tool R is well suited for implementing data preparation functions and summary functions for the simulation output. R comprises a wide range of statistical models and corresponding estimation procedures. As R features its own programming language, self-developed functionality can be easily added. Furthermore, due to its package structure, R-users can profit from developments made by others. We use R for the simulation to derive the input data from empirical data. After data preparation, the so-called “Pre-Processor” of the MicMac software builds up two ASCII based files in a well-defined format: one for the initial population and another one for the state space states and the related rates (Ogurtsova et al., 2009). For the actual simulation run, the MicMac microsimulation requests nothing more than the two input files[1] prepared by the Pre-Processor and a time horizon for the simulation. We have implemented a simple GUI (Graphical User Interface) through which all instructions required to run a microsimulation can be entered (cp. Figure 5.1). After a simulation run is completed, information on the simulated life-courses is provided in two files Zinn and Gampe (2011):

1. an ASCII file containing the birth dates of all simulated individuals, and
2. an ASCII file containing the dates of transitions and the corresponding destination states for all simulated individuals.

These files have a well-defined format, which can be accessed and managed further by arbitrary tools. We have implemented in R a comprehensive palette of instruments in order to evaluate and illustrate the output of a microsimulation run. We call this palette the "Post-Processor". Among other featured elements, the post-processor comprises the following (Zinn and Gampe, 2010):

- transformation of microsimulation output into a life history format,
- frequency tables of the states occupied at specific dates,
- population pyramids at specific dates,
- frequency distributions of the states occupied on January 1 of each year during the simulation period,
- analysis of first transition,
- analysis of origin and destination states, and
- the identification of the most frequent ("typical") life-course.

Finally, it should be pointed out that the Pre-Processor, the simulation engine of the MicMac microsimulation, and the Post-Processor have been designed has stand-alone tools. That is, they need not be used in combination. Alternative statistical software tools can be applied for preparing the simulation input data and for analyzing the generated output data as well. The engine of the MicMac microsimulation can also be replaced.

---

[1] The consideration of immigration flows demands two further input files: one file comprising the numbers of immigrants given according to immigration date, age at immigration, and attributes of immigrants. The other file contains transition rates describing the behavior of immigrants once they have entered the population. Details about the structure and the content of these files are given in Zinn and Gampe (2011).

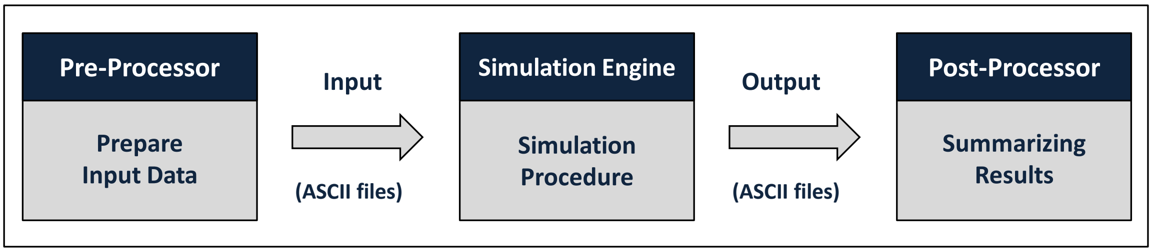


Figure 5.2.: Workflow of MicMac microsimulation.

An alternative implementation is for example the DYNPDEVS population network presented in Section 4.2. A detailed description of the general structure of the input files (containing rates matrices and the initial population) and the output files can be found in the corresponding manual Zinn and Gampe (2011). In Figure 5.2 the workflow of the MicMac microsimulation is depicted: the data flows between the software components of the MicMac software are illustrated in the graphic.

6. Application: The Netherlands

6.1. The synthetic population

To study the capabilities of the MicMac model for forecasting purposes, we simulate a synthetic population resembling the population of the Netherlands. The MicMac microsimulation model does not incorporate the time that has elapsed since a last event in the transition rates. Although this simplification might conflict with out purpose of illustrating actual individual behavior, the chosen setting may suffice. The assumption may be more daring for some events (e.g., divorce) than for others.

The example will consider changes in marital status and living arrangement, birth of children (for women) and transitions in educational attainments. Mortality is included as well. The state space consists of the following elements[1]:

- gender: female (fem); male (male)
- marital status and living arrangement: living at parental home and never married (PH); married for the first time, but never lived in a union before (nMA); married for the first time and cohabiting before (pcMA); remarried (pmMA); living alone and never lived in a union before (nSI); living alone but cohabiting before (pcSI); living alone and married before (pmSI); first cohabitation (nCO); higher order cohabitation but never married before (pcCO); cohabitation and married before (pmCO)
- fertility: childless (noChild); one child (firstChild); two children (secondChild); three or more children (thirdChild, fouthChild, ...)
- educational attainment: no education or primary education only (lowEdu); lower secondary school (medEdu); upper secondary or tertiary education (highEdu)
- mortality: dead (dead); alive

The particular choice of the values for the state variable ‘marital status and living arrangement’ allows us to provide some information about an individual’s partnership history in the current partnership status. As several studies find strong evidence that the partnership history influences prospective partnership behavior, this strategy seems to be advisable and allows to maintain a first order Markov model. For example, Boyle et al. (2008) show for various

---

[1] For each variable we list its values after the colon, separated by semi-colons. The abbreviations of the values which are used in the code are given in parenthesis.

countries that cohabitation before marriage significantly affects the divorce risk, and Lichter and Qian (2008) find that higher order cohabitations are less likely to end in marriage than first order cohabitations.

Simulations are run over 17 years, starting on January 1, 2004 up to December 31, 2020. In our study we are interested in partnership dynamics and fertility behavior and not in mortality at older ages. For this reason, during simulation we only focus on individuals aged between 0 and 63. The initial population consists of 139,048 males and 134,910 females (which corresponds to 2% of the actual Dutch population aged 0 to 63 on January 1, 2004). During simulation, individuals can experience the following events: giving birth (for females), leaving the parental home, launching a cohabitation, marrying, getting divorced or separated, changing their educational level, and death.

To obtain transition rates for the example, we have made use of different European data sources. Death rates were taken from the EuroStat2004 and EuroStat2008 projections for the Netherlands (baseline scenario)[2]. They vary with age and calendar time. Figure 6.1 depicts log-mortality rates by sex.

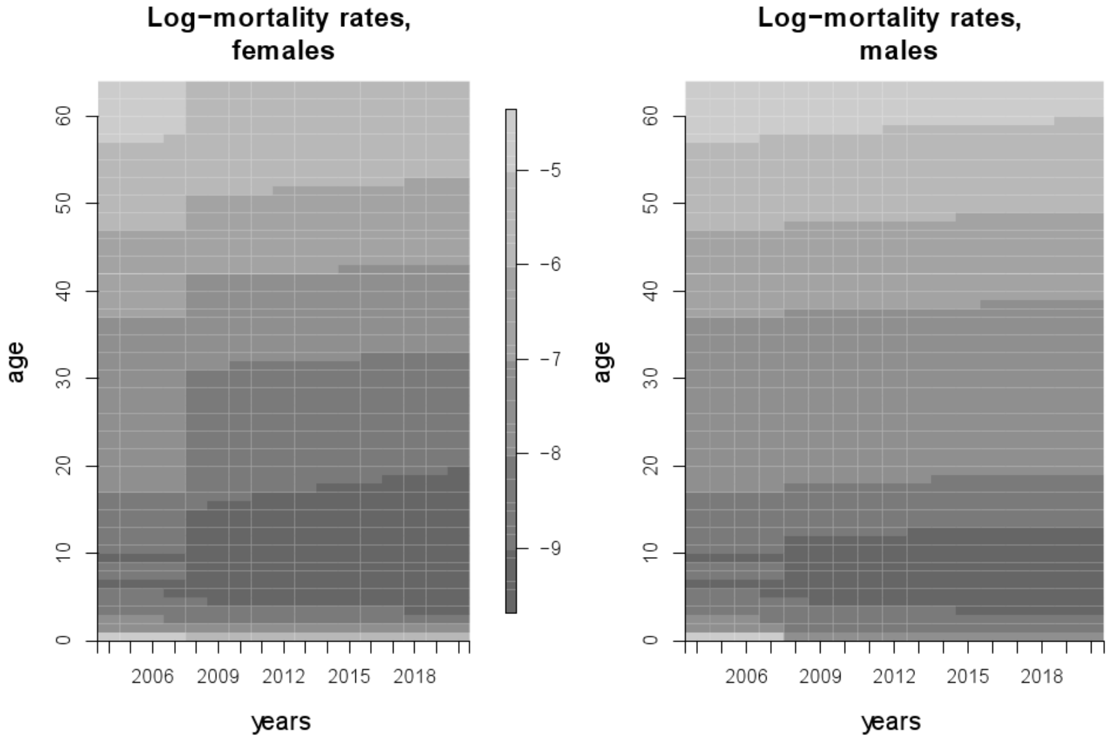


Figure 6.1.: Image plot of log-mortality for females and males for the period 2004 - 2020 and ages from 0 to 63.

To estimate transition rates related to fertility of changing the marital status and the living arrangement, we applied a slightly modified version of MAPLES (Impicciatore and Billari, 2007). This method estimates age profiles from longitudinal survey data using a generalized

---

[2] Detailed data on EUROPOP 2004 and 2008 mortality were kindly provided by Eurostat.

additive model and cubic splines (cf. Section 3.2.1). The Family and Fertility Survey of the Netherlands conducted between February 2003 and May 2003 (FFS_NL 2003) has served as a data basis.[3] It contains micro-information on fertility behavior and changes in marital status. The age-specific rates (for fertility, marital status/living arrangement) that we have estimated from the FFS_NL 2003 were held constant over calendar time. Figure 6.2 shows the transition pattern for changes in marital status and living arrangement. The age-profiles of all transition rates are given in the appendix (see page 185ff).

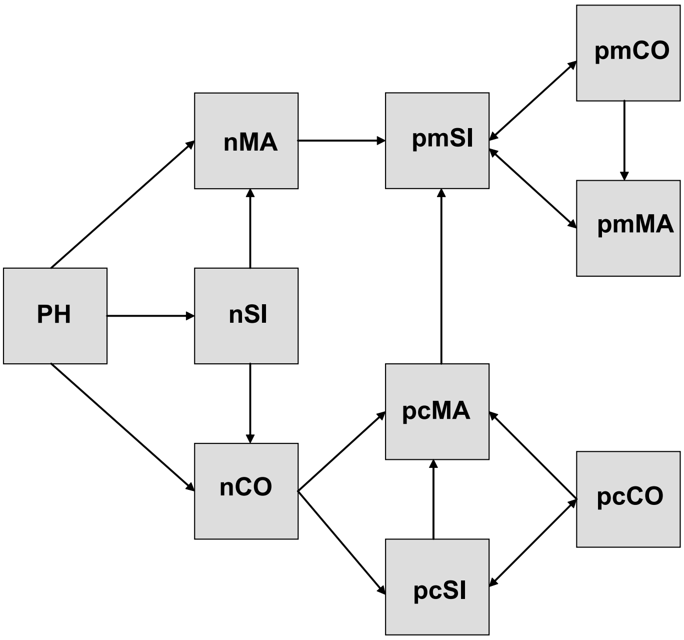


Figure 6.2.: Transition pattern concerning changes in the marital status.

To quantify an individual's propensity of changing his/her educational attainment, we have used data provided in Goujon (2008). The corresponding numbers are also given in the appendix (see page 185ff). We have constructed the initial population using the method of iterative proportional fitting; cf. Section 3.3.1. Figure 6.3 on page 26 shows the initial population cross-classified according to age, sex, and fertility status (left), according to age, sex, and marital status (graph in the middle), according to age, sex, and educational attainment (middle), and age, sex, and educational attainment (right). The marginal frequency distributions necessary to estimate these numbers have been taken from EuroStat (EuroStat, 2010) and the FFS_NL 2003 survey. The MicMac microsimulation allows to incorporate migration, although for the sake of simplicity, we only consider persons who were born in the Netherlands. Emigration is not taken into account either.

---

[3] The Centraal Bureau voor Statistiek in the Netherlands has provided the data of the Fertility and Family Survey for the Netherlands (FFS_NL 2003).

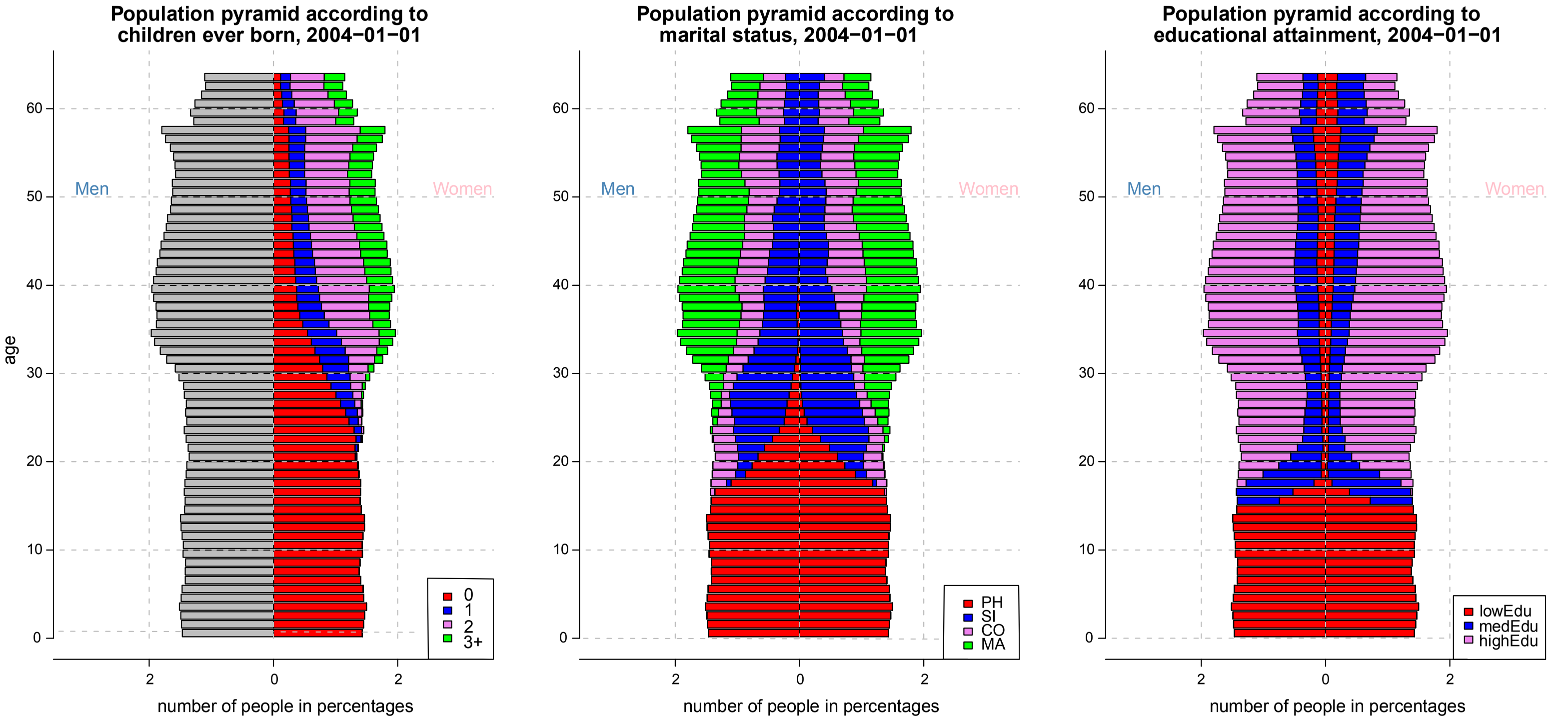
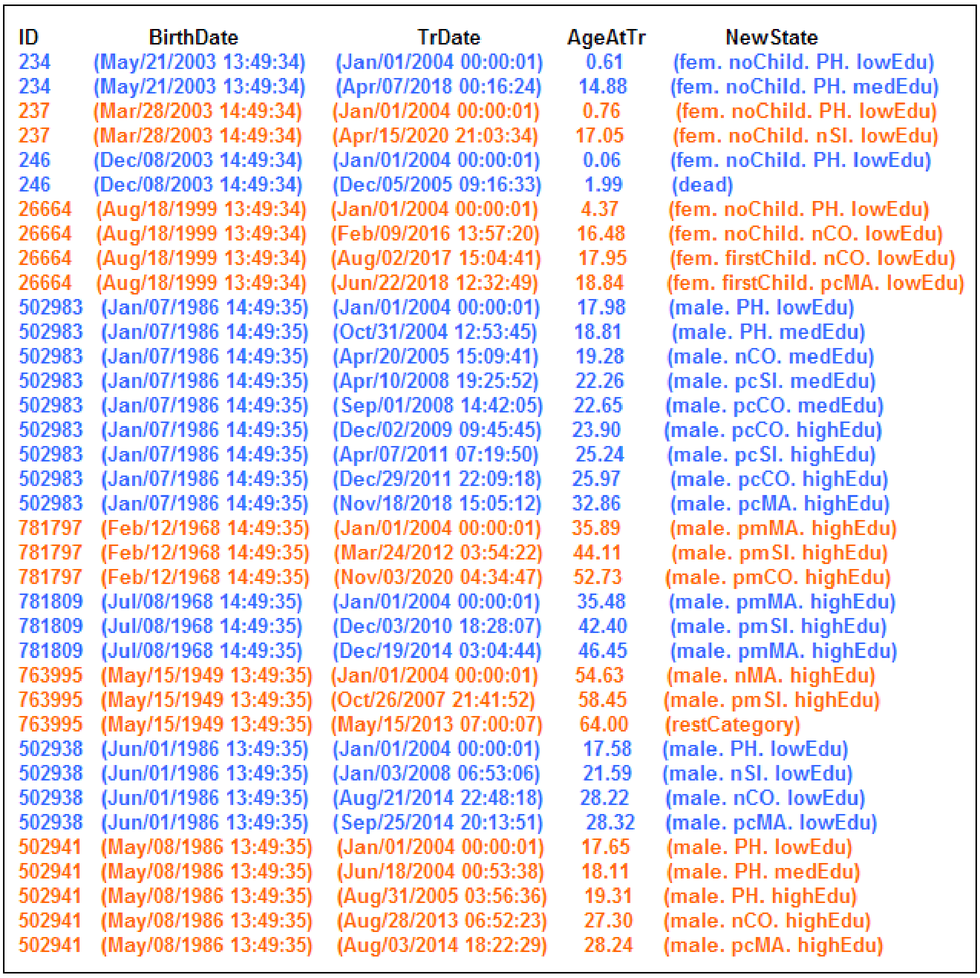


Figure 6.3.: The initial population refers to January 1, 2004. The left figure shows the female population, categorized according to fertility status: one child; two children; three or more children. In the middle individuals according to marital status/living arrangement are given; coding: PH (living at parental home), SI (living alone), CO (cohabiting), MA (being married). The right figure displays the initial population according to educational attainment; coding: lowEdu (no education or only primary school), medEdu (lower secondary school), highEdu (upper secondary or tertiary education).

6.2. Results

We use the Post-Processor of the MicMac software (Zinn and Gampe, 2010) to produce descriptive statistics of the output results of the microsimulation example described in the previous section. Subsequently, we present some of our findings. (More results are given in the appendix, see page 197ff.) Our analysis is far from being exhaustive. The purpose is to give a flavor of the variety of questions and implications that can be studied by means of a demographic microsimulation.

We performed the example on an Intel(R) Core(TM) 2 Duo CPU with a 3.16GHz equipped with 3GB memory 10 simulation runs. The runs last between 39.86 and 44.32 seconds. During simulation, all demographic events (births and deaths, and state transitions of individuals) are tracked. The average number of simulated events is 416, 399 and the average number of newborns is 40, 346.



ID	BirthDate	TrDate	AgeAtTr	NewState
234	(May/21/2003 13:49:34)	(Jan/01/2004 00:00:01)	0.61	(fem. noChild. PH. lowEdu)
234	(May/21/2003 13:49:34)	(Apr/07/2018 00:16:24)	14.88	(fem. noChild. PH. medEdu)
237	(Mar/28/2003 14:49:34)	(Jan/01/2004 00:00:01)	0.76	(fem. noChild. PH. lowEdu)
237	(Mar/28/2003 14:49:34)	(Apr/15/2020 21:03:34)	17.05	(fem. noChild. nSI. lowEdu)
246	(Dec/08/2003 14:49:34)	(Jan/01/2004 00:00:01)	0.06	(fem. noChild. PH. lowEdu)
246	(Dec/08/2003 14:49:34)	(Dec/05/2005 09:16:33)	1.99	(dead)
26664	(Aug/18/1999 13:49:34)	(Jan/01/2004 00:00:01)	4.37	(fem. noChild. PH. lowEdu)
26664	(Aug/18/1999 13:49:34)	(Feb/09/2016 13:57:20)	16.48	(fem. noChild. nCO. lowEdu)
26664	(Aug/18/1999 13:49:34)	(Aug/02/2017 15:04:41)	17.95	(fem. firstChild. nCO. lowEdu)
26664	(Aug/18/1999 13:49:34)	(Jun/22/2018 12:32:49)	18.84	(fem. firstChild. pcMA. lowEdu)
502983	(Jan/07/1986 14:49:35)	(Jan/01/2004 00:00:01)	17.98	(male. PH. lowEdu)
502983	(Jan/07/1986 14:49:35)	(Oct/31/2004 12:53:45)	18.81	(male. PH. medEdu)
502983	(Jan/07/1986 14:49:35)	(Apr/20/2005 15:09:41)	19.28	(male. nCO. medEdu)
502983	(Jan/07/1986 14:49:35)	(Apr/10/2008 19:25:52)	22.26	(male. pcSI. medEdu)
502983	(Jan/07/1986 14:49:35)	(Sep/01/2008 14:42:05)	22.65	(male. pcCO. medEdu)
502983	(Jan/07/1986 14:49:35)	(Dec/02/2009 09:45:45)	23.90	(male. pcCO. highEdu)
502983	(Jan/07/1986 14:49:35)	(Apr/07/2011 07:19:50)	25.24	(male. pcSI. highEdu)
502983	(Jan/07/1986 14:49:35)	(Dec/29/2011 22:09:18)	25.97	(male. pcCO. highEdu)
502983	(Jan/07/1986 14:49:35)	(Nov/18/2018 15:05:12)	32.86	(male. pcMA. highEdu)
781797	(Feb/12/1968 14:49:35)	(Jan/01/2004 00:00:01)	35.89	(male. pmMA. highEdu)
781797	(Feb/12/1968 14:49:35)	(Mar/24/2012 03:54:22)	44.11	(male. pmSI. highEdu)
781797	(Feb/12/1968 14:49:35)	(Nov/03/2020 04:34:47)	52.73	(male. pmCO. highEdu)
781809	(Jul/08/1968 14:49:35)	(Jan/01/2004 00:00:01)	35.48	(male. pmMA. highEdu)
781809	(Jul/08/1968 14:49:35)	(Dec/03/2010 18:28:07)	42.40	(male. pmSI. highEdu)
781809	(Jul/08/1968 14:49:35)	(Dec/19/2014 03:04:44)	46.45	(male. pmMA. highEdu)
763995	(May/15/1949 13:49:35)	(Jan/01/2004 00:00:01)	54.63	(male. nMA. highEdu)
763995	(May/15/1949 13:49:35)	(Oct/26/2007 21:41:52)	58.45	(male. pmSI. highEdu)
763995	(May/15/1949 13:49:35)	(May/15/2013 07:00:07)	64.00	(restCategory)
502938	(Jun/01/1986 13:49:35)	(Jan/01/2004 00:00:01)	17.58	(male. PH. lowEdu)
502938	(Jun/01/1986 13:49:35)	(Jan/03/2008 06:53:06)	21.59	(male. nSI. lowEdu)
502938	(Jun/01/1986 13:49:35)	(Aug/21/2014 22:48:18)	28.22	(male. nCO. lowEdu)
502938	(Jun/01/1986 13:49:35)	(Sep/25/2014 20:13:51)	28.32	(male. pcMA. lowEdu)
502941	(May/08/1986 13:49:35)	(Jan/01/2004 00:00:01)	17.65	(male. PH. lowEdu)
502941	(May/08/1986 13:49:35)	(Jun/18/2004 00:53:38)	18.11	(male. PH. medEdu)
502941	(May/08/1986 13:49:35)	(Aug/31/2005 03:56:36)	19.31	(male. PH. highEdu)
502941	(May/08/1986 13:49:35)	(Aug/28/2013 06:52:23)	27.30	(male. nCO. highEdu)
502941	(May/08/1986 13:49:35)	(Aug/03/2014 18:22:29)	28.24	(male. pcMA. highEdu)

Figure 6.4.: The simulated life-courses of ten individuals. The column “BirthDate” gives the birth dates of the individuals, “TrDate” contains the transition dates, and “AgeAtTr” the corresponding transition ages. “NewState” gives the states that individuals enter when they undergo an event.

As a first step the Post-Processor, converts the microsimulation output into a format resembling event history data. This format eases further computation. In Figure 6.4, typical life-courses of ten simulated individuals are given, already transformed by the post-processor. Each record describes one event that an individual has experienced during simulation. The record gives the individual's ID and the birth time ("BirthDate") of the individual, the transition date ("TrDate"), the transition age ("AgeAtTr"), and the state to which the individual has moved ("NewState"). The first transition date of individuals who are part of the initial population, corresponds to the simulation starting time, and the "NewState" in this case is the initial state. Likewise, the first transition time of a newborn corresponds to his/her birth date and the associated "NewState" to the state that he/she occupies at birth.[4]

The age of 63 is a rather low value for the maximum age until which life-courses are simulated. That is, many individuals will survive beyond this age. The microsimulation software collects these individuals in an open age category called "(restCategory)."

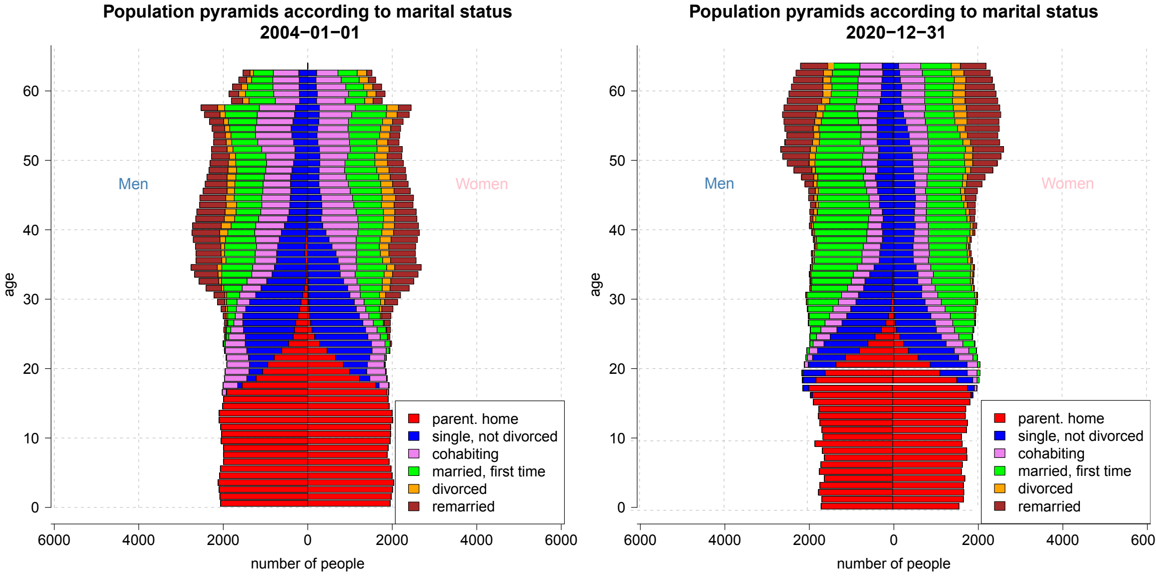


Figure 6.5.: Comparison of the age-sex structure of the virtual population at January 1, 2004 and December 31, 2020. Numbers are given according to the marital status and living arrangement of people.

Figures 6.5 and 6.6 show the age-sex-composition of the initial population at 2004-01-01 and of the simulated population at 2020-12-31. Population aging is evident. The small number of newborns is striking. While in the first year of the simulation less children are born than

---

[4] In Section 2.2 we describe how we assign newborns to the state space.

in the year before, in the following years the number of newborns remains stable. Figure 6.5 shows population numbers according to age, sex, and marital status/living arrangement. Over the simulation period for both women and men, the share of first marriages and remarriages increases remarkably. We observe further that – compared to 2004-01-01 – in the end of 2020, more young people aged between 18 and 23 live by their own - contrasted by less singles aged between 25 and 35. Moreover, we find that in the end of 2020 few individuals aged from 25 to 40 are divorced. The same applies to remarriages. Such large numbers of married people accompanied with such small number of divorced people are not as expected, and they indicate a problem in the input transition rates. That is, at this point we have to question the correctness of the input transition rates of changing the marital status.

Figure 6.6 depicts the number of women in each age class and their respective numbers of children ever born. It is obvious that along the simulation horizon the number of women with children decreases. This pattern is evident over all parities. In fact, in the end of 2020 women under 40 with three or more children are a threatened population group.

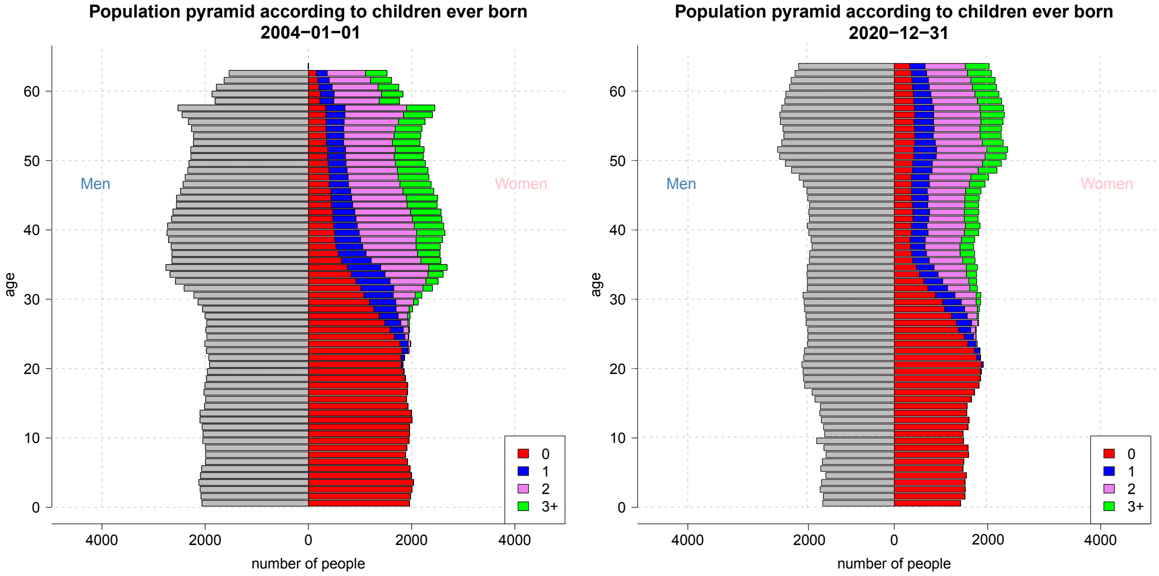


Figure 6.6.: Comparison of the age-sex structure of the virtual population at January 1, 2004 and December 31, 2020. For women the numbers are classified by the number of children ever born.

Besides the population distribution according to the current state-occupancies, we can also analyze the ages at which some events happen. If we compute, for example, the median age at first remarriage, we see that on average women tend to remarry younger than men. Figure 6.7 depicts the development of the median age of first remarriage over time for both women and men, with 95% confidence intervals added. While the median age of males remarrying for the first time ranges from 44 to 48, the corresponding median age for females varies between 37 and 45. As we have assumed time-constant remarriage rates, we do not find any crucial changes in the median age of first remarriage over time.

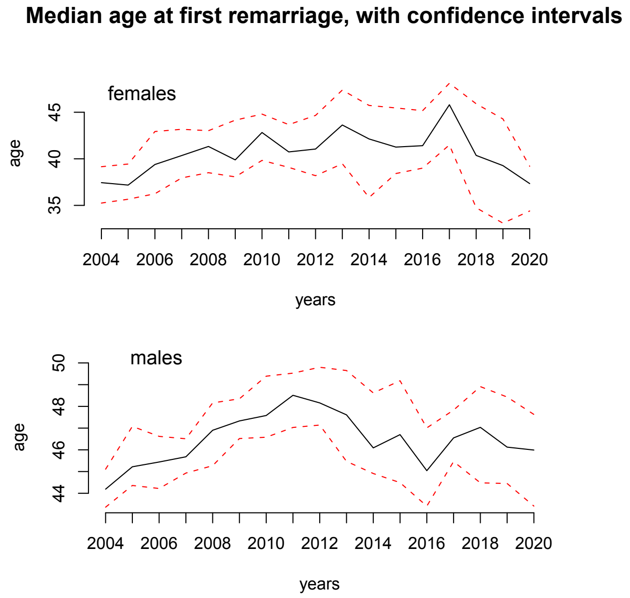


Figure 6.7.: Median age at first remarriage over time for both females and males, inclusive 95% confidence intervals.

Figure 6.8 shows the those event sequences that the simulated females and males have undergone most frequently (top three). For every event, the median transition age is given (the corresponding standard derivations are added in parentheses). In the studied case, the most frequently simulated event sequences start with birth, i.e., at age zero. As education is part of our example state space, all simulated children go through a school career. Most teenagers graduate from lower secondary school around the age of 16. Many female teenagers then continue their educational career until reaching an upper secondary or tertiary graduation before leaving the parental home (7.4% around the age of 20). Around the age of 20, among individuals with primary or lower secondary education, males seem to be more prone than females to leave the parental home to live alone (7.1% males compared to 5.1% females). At the first glance, the frequencies of the most frequently simulated event sequences may appear very small; these small numbers, however, have to be put in relation to the huge number of possible event sequences in our example.

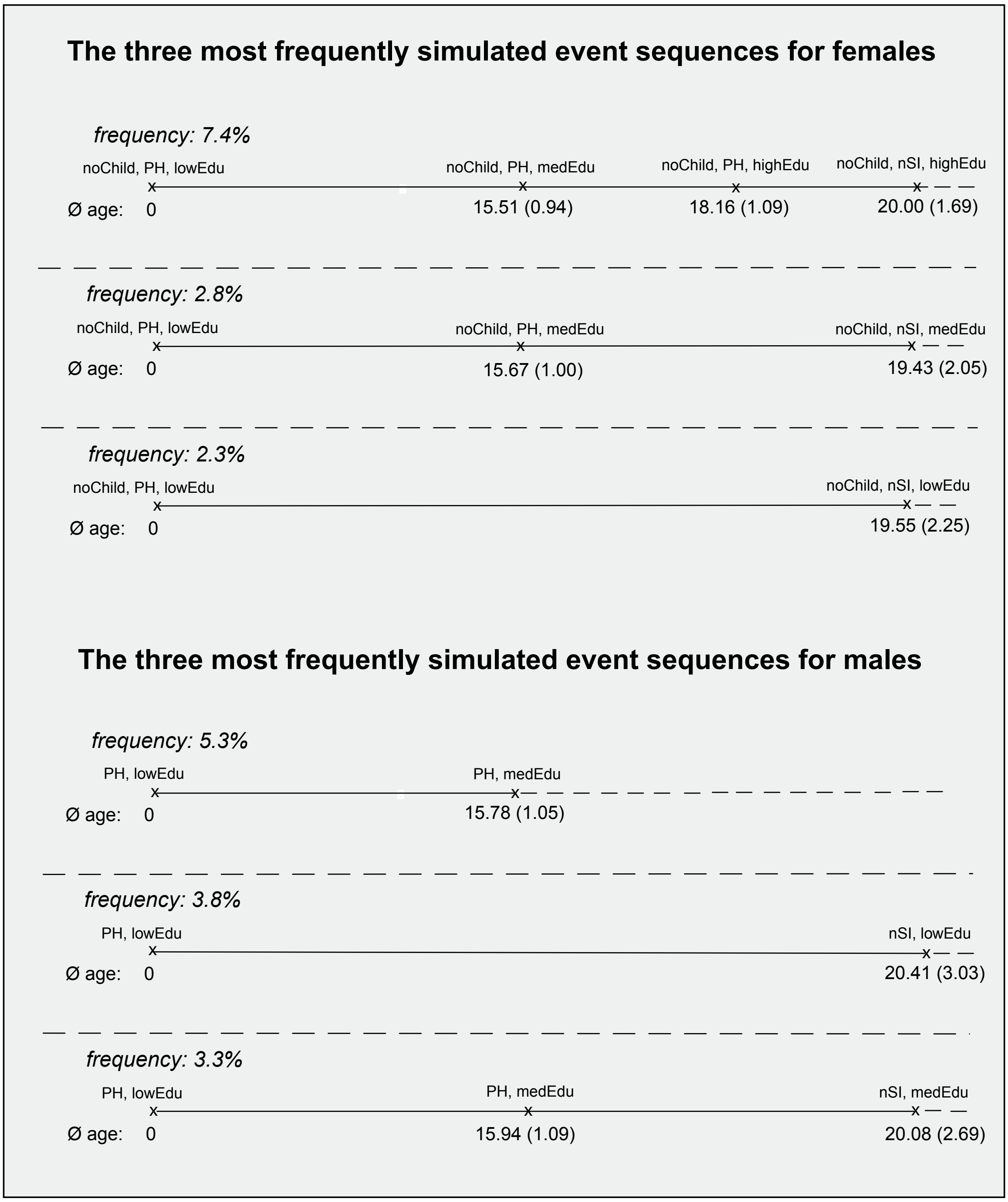


Figure 6.8.: The three most frequently simulated event sequences for females and males. The median age at each event is given below the respective event (the corresponding standard derivations are added in parentheses).

Figure 6.9 shows the frequencies of transitions into and out of the state of being single after having been married before (coding: 'pmSI'), for both women and men. We find that clearly more divorced men (12%) than divorced women (8%) break up a cohabitation (transition from 'pmCO' to 'pmSI'). Also, divorced men are less prone to cohabit than divorced women (transition from ‘pmSI’ to ‘pmCO’: 63% men compared to 70% women), which is in line with the previous result. As opposed to men, more women whose marriage was not preceded by a cohabitation are getting divorced (transition from ‘nMA’ to ‘pmSI’: 31% men and 38% women). In contrast, more men who were cohabiting before marriage undergo a divorce (transition from ‘pcMA’ to ‘pmSI’: 42% men and 36% women). Among dissolved marriages, the percentage of remarriages is relatively small (transition from ‘pmMA’ to ‘pmSI’: 18% for females and 15% for males).

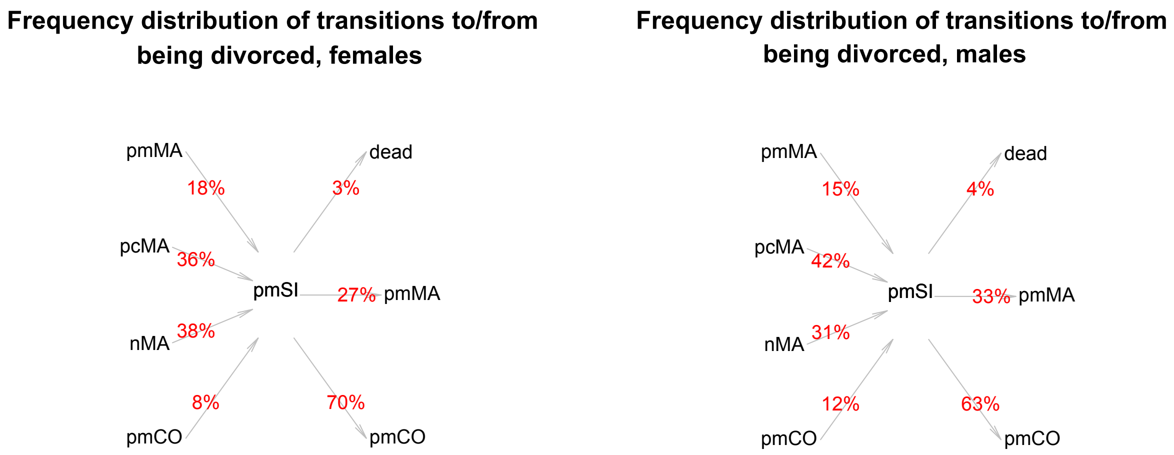


Figure 6.9.: Frequency distribution of transitions to and from being single after having been married before (‘pmSI’).

Figure 6.10 depicts the age-at-first-divorce distribution of females (left hand side) and males (right hand side), classified by level of education. We find that lowly educated persons tend to undergo divorce events at younger ages. This finding can be explained to a large extent by the fact that higher educated people tend to marry late, and if these people experience divorce events, they do so later in life. The marriage rates that we use in our example reflect this behavior (see transition rates depicted in the appendix on page 185), and thus we also find such behavior in our simulation results (see Table 6.1 and Figure 6.10).

Educational attainment *Gender*	lowEdu	medEdu	highEdu
female	27.23	31.29	35.78
male	29.60	32.37	35.47

Table 6.1.: Median ages at first marriage, according to gender and educational attainment.

The model specification chosen here does certainly not resemble real behavior - it is naive to assume constant fertility behavior, and constant transition rates for the educational attainment as well as for changes in the marital status/living arrangement. Furthermore, especially

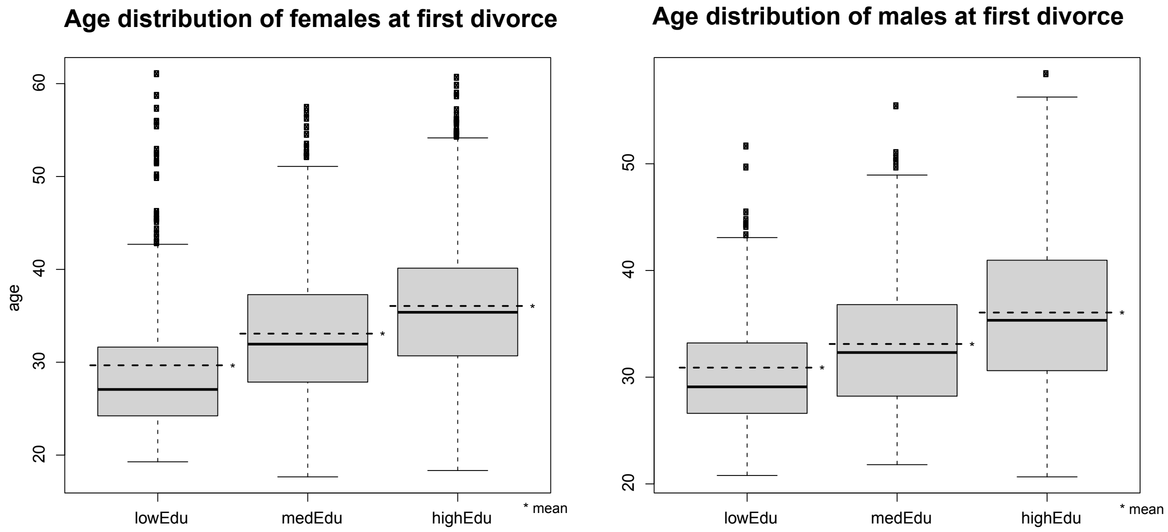


Figure 6.10.: Age-at-first-divorce distribution of females and males, according to educational attainment.

in the Netherlands, the role of migration should not be neglected. We return to this issue in Section 7.2.2.

As mentioned before, the presented data example does not show the full potential of the MicMac microsimulation. Because of its generic model the MicMac microsimulation allows to produce population projections relying on various future scenarios. This facilitates, for example, comparison of the effects of different prospective policies and/or migration assumptions.

7. Performance and Validation

7.1. Conducting experiments

In their technical report about SWARM, Minar et al. (1996) state that “Unfortunately, computer modelling frequently turns good scientists into bad programmers. Most scientists are not trained as software engineers.”. One of the reasons that they have identified is the lack of standard laboratory equipment for doing experiments with models. Experiments ease the detection of faulty settings. That is, experiments facilitate testing the suitability of simulation algorithms and their implementation.

The modelling and simulation framework JAMES II features, besides many up-to-date modelling and simulation techniques, an experimentation layer, which allows to setup and to control experiments (Himmelspach et al., 2008). In Chapter 5, we have described an implementation of the MicMac microsimulation that uses JAMES II as a M&S library. Profiting from the experimentation layer of JAMES II we can evaluate the performance and suitability of this implementation. Subsequently, we describe two corresponding experiments: The first experiment explores whether the random number generator that was used changes the results, and the second experiment deals with performance issues on computing the microsimulation model.

7.1.1. Data and scenario

To conduct experiments, we use the data example described in Chapter 6 in a shortened and slightly modified form. We use data of a synthetic population that resembles the Dutch population. Starting on January 1, 2008, life-courses of five cohorts are generated: individuals born in 1960, 1970, 1980, 1990, and 2000. The corresponding numbers are:

born in *gender*	*2000*	*1990*	*1980*	*1970*	*1960*	*total*
females	100946	99892	100843	129992	123748	555421
males	105151	105077	101407	131719	125923	569277
total	206097	204969	202250	261711	249671	1124698

We simulate synthetic cohorts that correspond to one, two, twenty, and one hundred percent of the number of actual individuals. The simulation starting time is January, 1, 2008, and the simulation stops on December 31, 2050. At simulation starting time, individuals who belong to the cohort born in 1960 are 47 years old, and individuals born in 2000 are 7 years old. We focus on fertility behavior, and changes in the marital status/living arrangement. As at older ages the marital and fertility history is usually completed, we consider life-courses only until age 63.

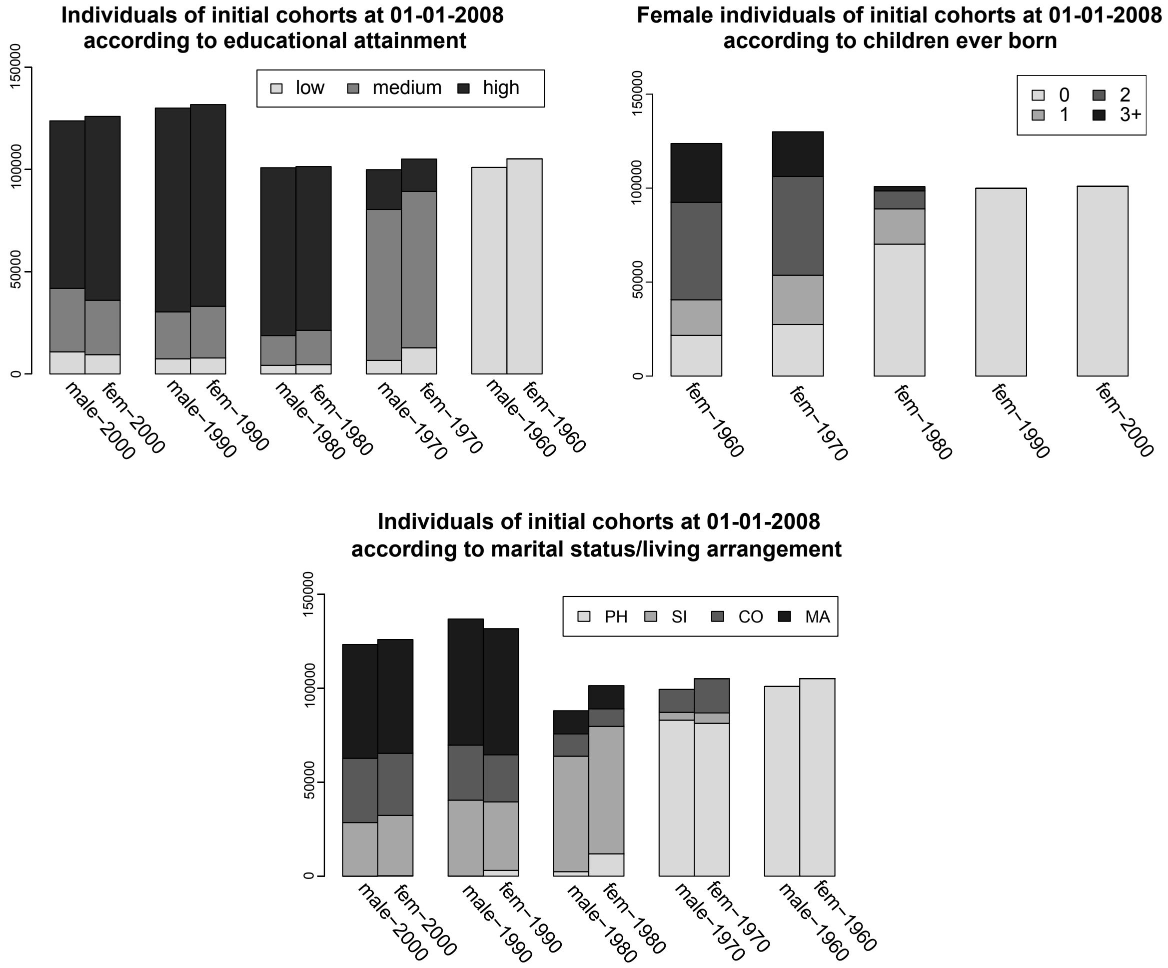


Figure 7.1.: We consider five cohorts: individuals born in 1960, 1970, 1980, 1990, and 2000. The graph on the top left shows the initial cohorts at January 1, 2008 according to educational attainment; coding: low (only primary school), medium (lower secondary school), high (upper secondary or tertiary education). The top right graph shows the females of the initial cohorts on January 1, 2008 according to children ever born. The initial cohorts on January 1, 2008 according to marital status/living arrangement are given in the lower graph; coding: PH (living at parental home), SI (living alone), CO (cohabiting), MA (being married).

The state space, the assumed transition patterns, and the transition rates are as described in Chapter 6, including the assumption that birth rates and rates of changing the marital status/living arrangement vary with age, but are held constant over calendar time. Figure 7.1 shows the distribution of the five initial cohorts according sex, educational attainment, marital status/living arrangement, and fertility status. For the period from 2008 to 2050 mortality rates have been taken from the EuroStat2008 projections for the Netherlands (baseline scenario), see Figure 7.2.

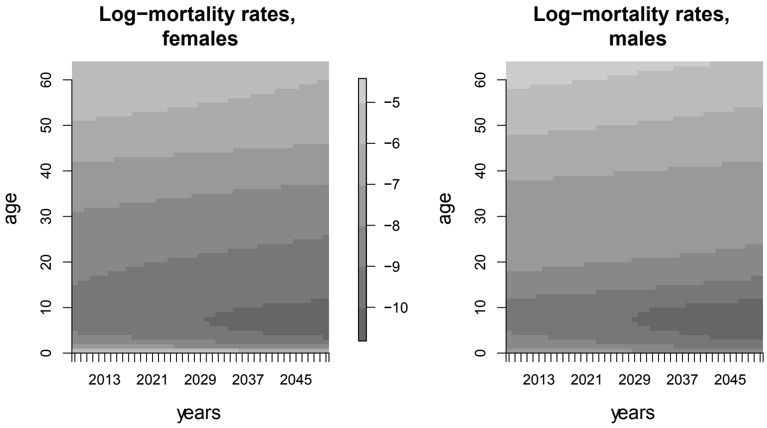


Figure 7.2.: Image plot of log-mortality rates of females and males, period from 2008 to 2050 and age range from 0 to 63.

7.1.2. Testing different pseudo random number generators

The replication of simulation results on different machines and with different implementations has been recognized as sine qua non for the reliability of simulation models (Edmonds and Hales, 2003). If the model to be computed or the computation algorithm contains stochastics, one has to take special care of the pseudo random number generator (PRNG). Starting from a prespecified seed, a PRNG produces always the same sequence of pseudo random numbers. The period is the maximum length of that sequence before the PRNG begins to repeat numbers. Therefore, the period length ρ of a PRNG is crucial for its applicability. Many different PRNG have been invented so far and often they have been considered to be faulty after a while, e.g., because of unwanted correlation patterns between random number variates. For example, the linear congruental generators (LCG) cause serial correlation between successive random number variates, and it is recommended to no longer use them (Press et al., 2007). Despite their deficiencies, LCG are still rather popular these days because they are very efficient in terms of speed, easy to implement, and, correspondingly, portable code, parameters, and test toolkits are available. Many versions and enhancements have been done to counter the unwanted features of LCGs: The order of recursion has been increased from order one to a higher order, the length of periods has been enlarged, different types of PRNG have been combined, etc.

To generally avoid any regularities in the pattern of the employed pseudo random numbers, L'Ecuyer and Hellekalek (1998) suggest to draw less than $\sqrt{\rho}$ pseudo random numbers. That is, if an application requires to draw many more random numbers than the square root of the period of the used PRNG, then simulation results might be distorted. Commonly, however, the suitability of a PRNG depends on the structure of a particular problem. Minding this fact, we have checked whether different PRNGs have any influence on the results computed in the described use case. We have tested the following three PRNGs:

- RANDU which is an outdated LCG with a period of 2^{31} ($\sqrt{\rho} \approx 46,341$),
- Java random number generator which is a LCG with a period of approximately 2^{48} ($\sqrt{\rho} \approx 16,777,216$),
- Mersenne Twister which is based on a matrix linear recurrence over a finite binary field, and in its commonly used variant, MT19937, it has a period of $2^{19937} - 1$ ($2^{9968} < \sqrt{\rho} < 2^{9969}$).

To facilitate comparability each PRNG has been initialized with the same (arbitrarily chosen) seed. Considering our example, the average number of random numbers that have to be drawn can be computed as the product of the average number of individuals n that are considered during simulation, the average number of events e that they will experience over their life-course, and the average number c of competing risks per state transition. For different values of n, e and c, the table below gives an estimation of the amount of random numbers needed:

Size of initial population	$c = 3, e = 3$	$c = 3, e = 5$
100% : $n_0 = 1,124,698$	10,122,282	16,870,470
10% : $n_0 = 112,470$	1,012,230	1,687,050
1% : $n_0 = 11,247$	101,223	168,705

In the table the size of the initial population n_0 has been chosen as a lower bound of the average population size, i.e., individuals that enter the population during simulation are not counted in.

All the numbers in the table are remarkably higher than the square root of the period of RANDU. On the contrary, the square roots of the periods of the java random number generator and of the Mersenne Twister exceed in all cases the required amount of random number variates. We are aware of the fact that taking $n_0 = 1,124,698$ and $c = 3, e = 5$ yields an amount of needed random variates that is close to the square root of the period of the java random number generator. Notwithstanding, for our example, we expect overall no significant differences between the results that are obtained using the java random and the Mersenne Twister generator. The situation differs in case of RANDU: here a tuple of three in series drawn random number variates is always highly correlated Hellekalek (1998); Marsaglia (1968), and this defect might bias the simulation results.

All the listed generators are available as plug-ins in JAMES II; thus corresponding experiments can easily be done. The random number generator to be used is simply handed over to the computation from the experimentation layer.

We run the Netherlands example using ten percent of the total size of the five cohorts and three different random number generators. Each experiment has been executed three times. As seeds we have (arbitrarily) chosen 213,423, 5,689, and 628,164. That is, in total we have conducted nine simulation runs: one run per seed and studied PRNG. The average of random number variates produced per run was $1,817,524$ numbers. Between different runs we could not make any significant differences in the amount of computed random numbers.

To find out whether potential regularities in the pattern of drawn random variates bias the simulation outcome, we have analyzed the empirical distributions of all events that have occurred within the simulation period. Empirical event distributions have been computed for each random number generator and each selected seed separately. Figure 7.3 shows the results. In summary, a comparison of the empirical event distributions achieved using the three different PRNGs does not reveal any remarkable differences in the simulation results.

Figure 7.3.: Comparison of empirical event time distributions.

7.1.3. Efficiency of different event queues

The performance of M&S software depends on a variety of factors. In this section we focus on the impact of a single sub data structure used by the computation algorithm. Primarily the model is crucial: depending on its structure different data structures or algorithms perform better than others. Subsequently, for our example setting we analyze how well event handling is managed by different data structures. The model described above is a discrete event model and as such its computation can be based on an event queue. In the used microsimulation implementation the event queue is integrated into the computation algorithm using the plug-in mechanism of JAMES II. Therefore, it can easily be exchanged for each new computation without the need to modify any code. JAMES II comprises several distinct event queue realizations (Himmelspach and Uhrmacher, 2007a). As in our application, events have only to be enqueued into and dequeued from an event queue, our review does not include any event queue extensions that allow requeue operations (to update the time of an already-enqueued event). For our experiments, we have selected the following four implementations of event queues from JAMES II:

- calendar queue,
- mlist,
- heap based queue, and
- simple queue.

The calendar queue, the mlist, and the heap queue are well-known event queues and are comprehensively described in literature (Brown, 1988; Goh and Thng, 2003; van Emde Boas et al., 1977). The simple event queue is an outmoded data structure that employs an unsorted list to maintain events. In our study, it serves as a reference for the goodness of performance of the other queue implementations considered. By experimental evaluation supported by JAMES II, we have analyzed which of the listed queues performs best for the example scenario. This knowledge helps to speed up the microsimulation execution.

We have executed the experiments on a desktop workstation equipped with two Xeon quad core CPUs, activated hyper threading, deactivated automated CPU overclocking, equipped with 32GB of RAM achieving a Windows Experience index of 7.8 for CPU and memory and Java Sci mark (using Java VM 1.6.0 (64 bit)). We have run each experiment four times to eliminate the impact of further concurrent processes on the machine. In our microsimulation, implementation data collection is done using a file based storage. Any outcome data is kept in memory, and to avoid any side effects, written after a simulation run has been finished. In our experiments, we have used the Mersenne Twister as PRNG, initialized with seed 10991.

In Table 7.1 the run times of our experiment setup are given. The times clearly show that the event queue implementation has a considerable impact on the overall runtime. That is, using the “wrong” queue results in a remarkable loss of performance. In our setting, the simple event queue performs particularly poorly. For a continuous-time microsimulation, we deem the mlist event queue as being the most convenient choice.

Population size *Event Queue*	1%	2%	20%	100%
MList	4.06 (0.17)	5.17 (0.27)	29.50 (2.59)	134.32 (4.53)
HeapEventQueue	4.12 (0.30)	5.32 (0.47)	30.25 (2.08)	145.24 (6.23)
CalendarQueue	4.40 (0.31)	6.42 (0.48)	135.66 (1.92)	4101.41 (329.99)
SimpleEventQueue	12.60(0.72)	2210.34 (142.96)	Na	Na

Table 7.1.: Mean execution times (in seconds). Each experiment has been executed four times. In parentheses the standard deviation is given. All values have been rounded to two digits after the decimal comma. The 'Na' notation shows that corresponding runs have been interrupted because of enormously long run times.

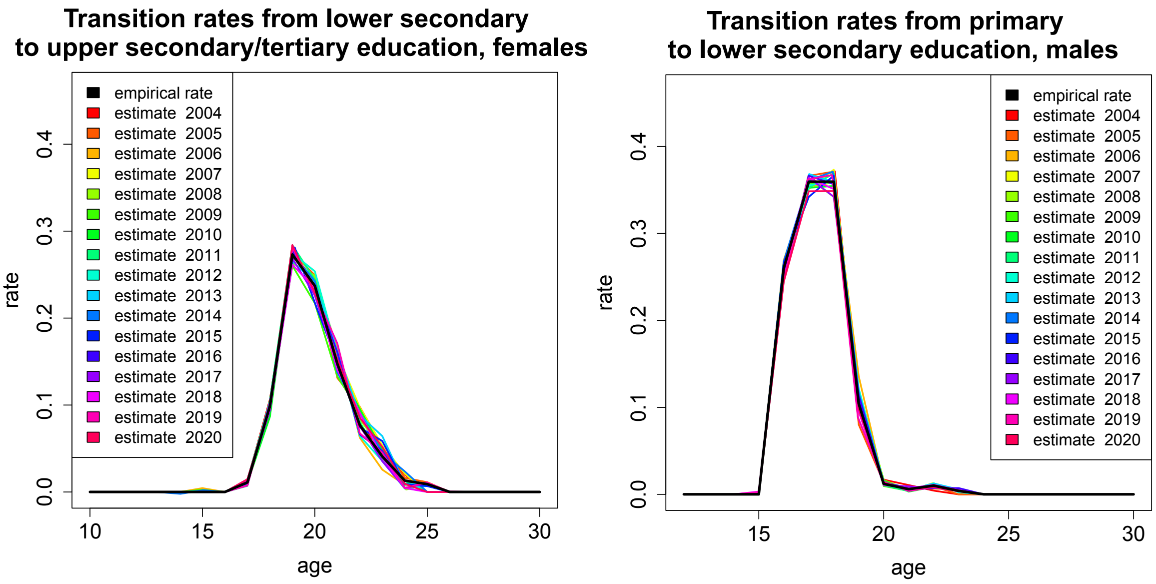


Figure 7.4.: Re-estimation of transition rates. The graph on the left hand side shows the empirical and the re-estimated rates of females with a lower secondary education to graduate from a higher secondary/tertiary school. The graph on the right hand side shows the empirical and the re-estimated rates of males to graduate from lower secondary school.

7.2. Verification and validation

Model verification and validation is crucial for the reliability of every model, and for the corresponding simulation, hereby, verification is a pre-requisite for validation. It is concerned with whether the model was correctly built (Balci, 2003), i.e., verifying a model means checking the model specification and implementation. In our application, the re-estimation of the empirical transition rates that we have used as input is the most basic verification step. We subsequently elaborate the respective processing and show some results. Thereafter, we propose three methods to validate the microsimulation output. The first method requires us to assess the errors inherent in the data and the errors induced by the statistical model that is used to estimate input parameters. The second method relies on 'typical human face validation' and implies qualitative evaluation of the suitability of simulation outputs. Finally, the third method means to compare the microsimulation results to the outcome of an alternative model. For this purpose, we exploit the relationship between the Mic and the Mac model of the MicMac project.

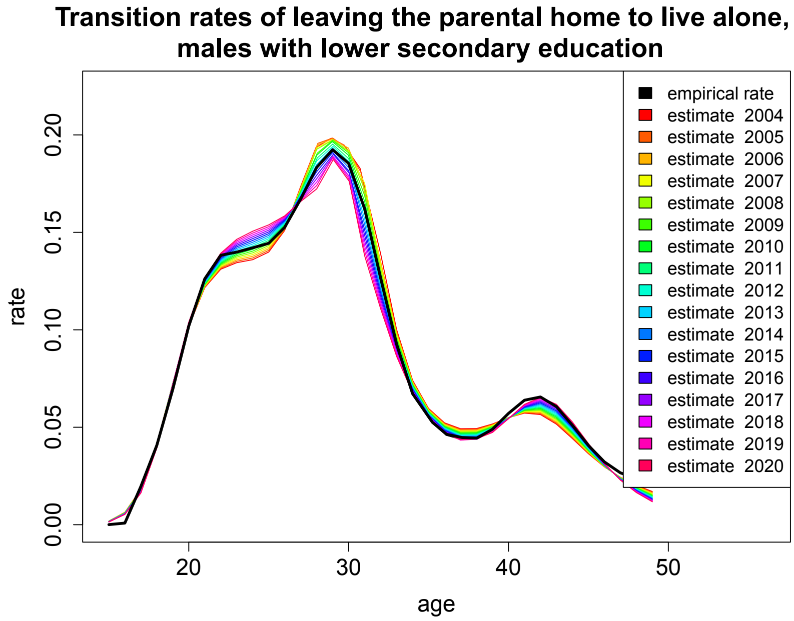


Figure 7.5.: Re-estimated transition rates. Empirical and re-estimated transition rates of males with a lower secondary education to leave the parental home for living alone.

7.2.1. Verification of the implementation

By definition, a microsimulation allows to derive the distribution of each considered life-course event and of the waiting times between these events. Hence, using the microsimulation output, we can compute transition rates that give the propensities of simulated individuals to experience certain demographic events. Comparing these rates with the empirical transition rates, which we have used as input, shows whether we have correctly implemented the simulation model and the simulation algorithm. In general, the re-estimation of the empirical transition rates is a very basic verification step, however, also the most obvious one. To verify our microsimulation implementation we have estimated occurrence-exposure rates (cf. Section 3.2.1) from the results of the microsimulation illustration described in Chapter 6. For smoothing these rates we employ an associated two-dimensional P-Splines methodology (Currie et al., 2006) that has been implemented in an R package named `MortalitySmooth` (Camarda, 2009). In the setting considered, the re-estimation of rates shows that the simulation produces consistent output. Some results are plotted in the Figures 7.4, 7.5, and 7.6.

In our example only death rates vary with age and with calendar time. All other transition rates are only age-dependent. Correspondingly, during simulation to all events apart from dying rates apply that only depend on age. In all presented figures the respective age-specific curves of the empirical rates are depicted as black curves. For all years within the simulation horizon, re-estimated age-specific rates are given in colors passing through the palette of rainbow colors.

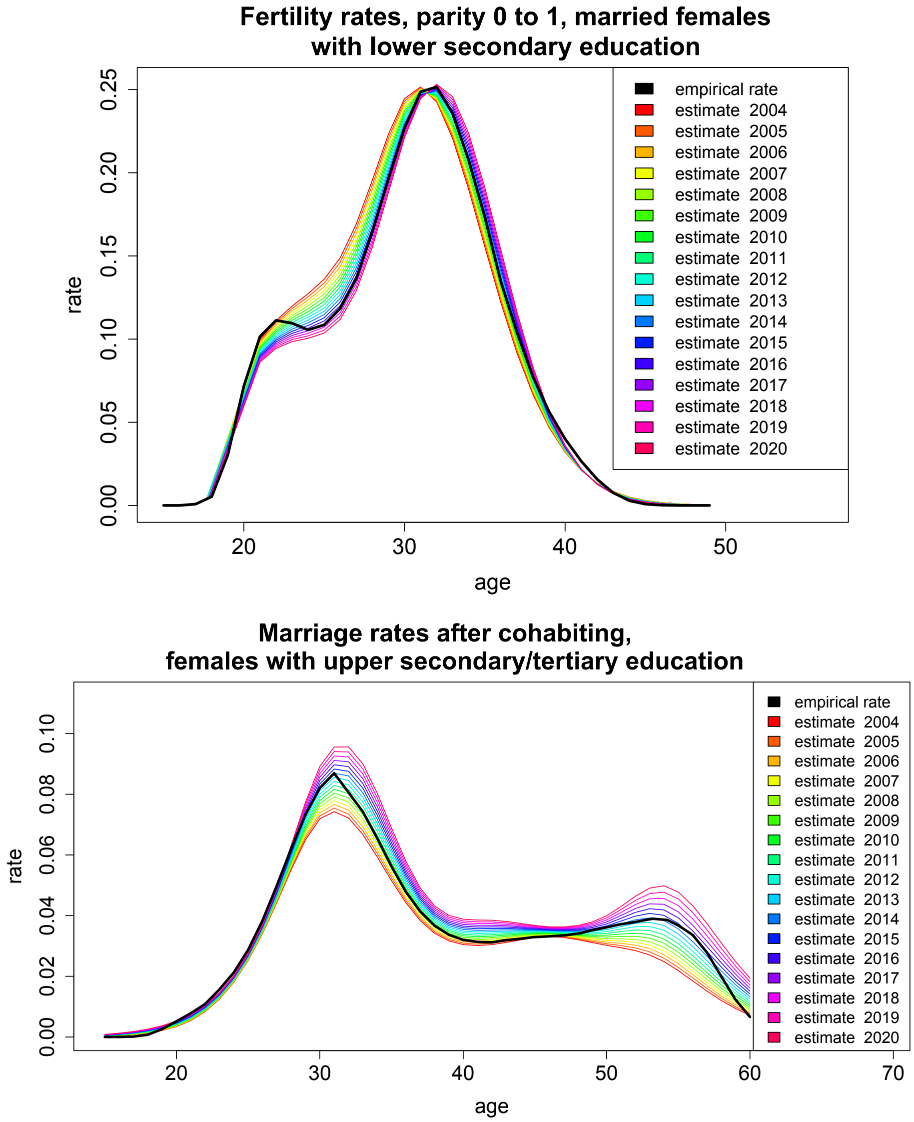


Figure 7.6.: Re-estimated transition rates. The upper graph depicts the empirical and the re-estimated first-birth rates of married females. Lower graph shows the marriage rates of cohabiting women with high education.

Figure 7.4 shows on the left hand side the empirical and the re-estimated rates of females graduating from upper secondary/tertiary school/university. The graph on the right hand side depicts the empirical and the re-estimated rates of males finishing lower secondary school. Apparently, the simulation algorithm manages to closely mimic the angular shape of the empirical transition rates.

Figure 7.5 shows the transition rates of males leaving the parental home for living alone.

The depicted rates correspond to males with a lower secondary education. We find that the re-estimated curves are very similar to their empirical counterparts.

Figure 7.6 displays the empirical and the re-estimated first-birth rates of married females with a lower secondary education (upper graph). The lower graph shows the marriage rates of cohabiting females with an upper secondary/tertiary education. In both cases, the overall pattern of the re-estimated rates nicely resembles the development of the empirical age-specific transition rates. Nonetheless, we find - especially in the lower graph - that the re-estimated rates are less curved than the the empirical rates. The comparably flat curves of the re-estimated rates might be the result of smoothing rates that are estimated based on a relatively low number of simulated events. One would expect that considering a larger share of Dutch people in the initial population (not only 2%) leads to more pronounced transition patterns.

Summing up, we deem the close resemblance between re-estimated and empirical transition rates a strong indicator for the correctness of the low-end implementation and the specification of the MicMac microsimulation described in Chapter 5.

7.2.2. Model Validation

Validating a simulation model is a good and useful practise, and necessary for the credibility of the achieved results. For simulation systems dealing with human societies Bharathy and Silverman (2010) discuss various validation techniques. For our microsimulation, we subsequently describe three of these techniques: methodological, qualitative, and external validation.

Methodological validation

Methodological validation requires assessing the quality of the data that are used to parameterize models and the estimated simulation input. Here low quality data and inadequate statistical methods certainly contribute considerably to distorted and misleading simulation results. Concerning the preparation of input data for a continuous-time microsimulation we refer to Chapter 3. There we describe data types and methods that suit the estimation of transition rates and the construction of an initial population. Concerning more general information about the methodological validation of microsimulations we refer to the work of Wolf (2001) and Klevmarken (2002).

Qualitative validation

Qualitative validation means to perform ‘typical human face validation’ and to assess the suitability of simulation outputs. Counterintuitive results and results that contradict common knowledge point to a model misspecification. We use transition rates to describe individual behavior. Therefore, the dynamics of the virtual population is mainly determined by these rates. That is, counterintuitive results point to inconsistencies in the transition rates. For example, we know that (in Western societies) higher educated women tend to have their first child later in life. Thus, used input fertility rates should reflect such a pattern.

In Chapter 6 we present a case study dealing with the contemporary Netherlands. For the most part we deem the used transition rates being consistent with common knowledge. Assuming constant rates for fertility transitions, however, changes in the education attainment and the marital status/living arrangement definitely oversimplifies reality and certainly causes blurred population projections. Moreover, although our microsimulation model allows the consideration of migration flows, we have neglected them. This definitely biases our population projections. We are no experts in the field of Dutch fertility, mortality, education, and migration. Therefore, it is not advisable to let us solely decide on the parameterization and the evaluation of a Dutch population model. For this purpose, experts should be asked.

Cross-model validation

External validation means to test simulation output against population projections from other sources, like national population projections conducted by statistical offices, or against the outcome of alternative models, so called cross-model validation. We can employ the core idea of the MicMac project to cross-model validate our microsimulation: a microsimulation (Mic) and a macroprojection (Mac) are run in tandem, and the microsimulation output and the macroprojection output can be compared. Here, running both models in tandem means that the microsimulation and the macroprojection are based on the same multi-state model and are consistently parameterized (Willekens, 2005b).

Opposed to a microsimulation, a (conventional) macroprojection does not comprise any stochastics. It conducts population projections using a demographic standard methodology called cohort component model (Willekens, 2006). A cohort component model divides a population into subpopulations that share certain attributes like age, sex, marital status, and education attainment. The state space of the underlying multi-state model contains the combination of all considered attribute values. A state vector comprising the number of individuals in each subgroup defines the studied population at equidistant time points (e.g., every year). To each subpopulation for each period transition rates are assigned. These quantify the individual propensities to undergo transitions between the different states of the state space. The so-called transition matrix combines them. Based on the transition matrix from period to period subpopulations numbers are computed. Within the MicMac project, we use for this purpose a method that relies on so-called Kolmogorov forward equation. The method is comprehensively explained by van Imhoff (1990). Unlike microsimulations, macroprojections do not produce event histories, but are restricted to mean population and event numbers. Nonetheless, macroprojections are widely-used and well established. Reasons might be that the corresponding methodology is easy to understand, and software tools and implementation guidance are available. Within the MicMac project, we have implemented a macroprojection software (Zinn and Gampe, 2011) relying on the methods described in van Imhoff and Keilman (1991).

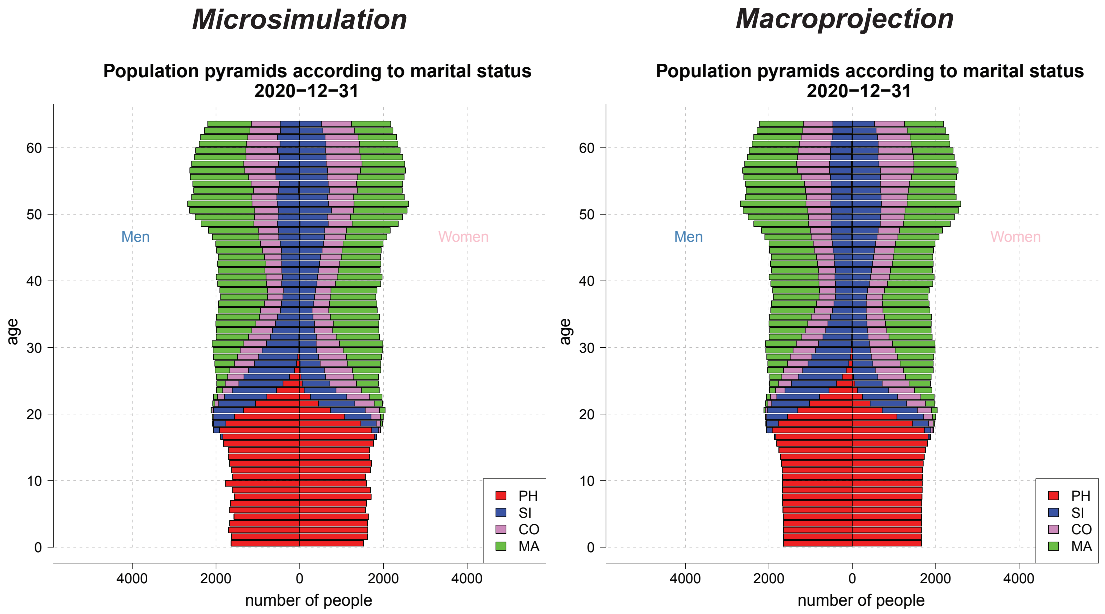


Figure 7.7.: Comparison of microsimulation and macroprojection results: population pyramids according to marital status; coding: living at parental home (PH), being single (SI), cohabiting (CO), being married (MA).

A major drawback of macroprojections is that a large number of individual attributes included the model hampers its execution (van Imhoff and Post, 1998): With each additional attribute the size of the transition matrices increases exponentially.[1] Oversized transition matrices entail mainly two problems: First, they require a lot of storage. Second, a large, complex example state space usually contains many transitions that are impossible (e.g., from 'second child' to 'first child'). Consequently, the related transition matrices are sparse matrices. A macroprojection model, however, requires the inversion of the transition matrices and, as it is well known, the inversion of sparse matrices is pretty problematic (Brayton et al., 1970). That is, a common macroprojection cannot easily cope with such comprehensive state spaces like the one we have considered in our microsimulation application; cf. Chapter 3. As a consequence, we restrict ourselves to a macroprojection that disaggregates the model population by age, sex, number of children, and a marital status/living arrangement variable with four possible values (instead of ten). The respective values are living at parental home, living alone, cohabiting, and being married.

Running the MicMac microsimulation and macroprojection that are consistently parame-

---

[1] We consider a state space that comprises d state variables each of which can take L_j discrete values, $j = 1, \cdots, d$. Combined with age (usually categorized into 101 age groups), we obtain per period a state vector with $L_1 \cdot L_2 \ldots L_d \cdot 101$ components, one for each subpopulation. The corresponding transition matrix consists then of $(L_1 \cdot L_2 \ldots L_d \cdot 101)$ rows and of $(L_1 \cdot L_2 \ldots L_d \cdot 101)$ columns. For example, we assume an application with 4 state variables each with 4 categories, then per period we yield a transition matrix of the dimension $25,856 \times 25,856$ $(4 \cdot 4 \cdot 4 \cdot 4 \cdot 101 = 25,856)$. Adding a further state variable with 4 categories would lead to transition matrices of the dimension $103,424 \times 103,424$.

terized, we find that – apart from slight variations[2] – the aggregated microsimulation output corresponds to the macroprojection results, see Figure 7.7 and Figure 7.8. This indicates the validity of both models.

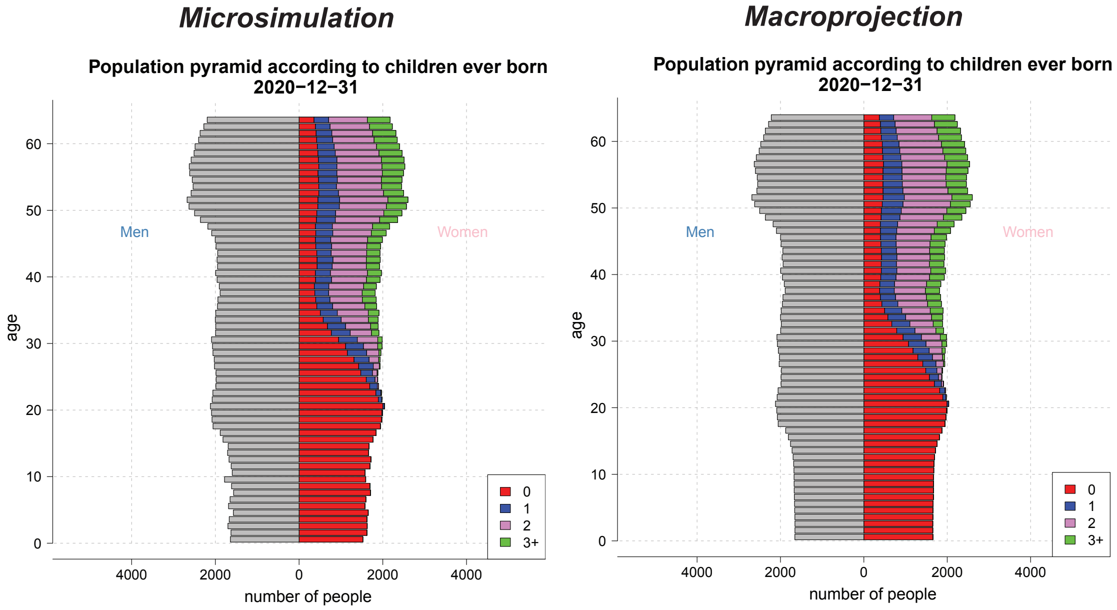


Figure 7.8.: Comparison of microsimulation and macroprojection results: population pyramids according to children ever born; coding: childless (0), one child (1), two children (2), three or more children (3+).

---

[2] Most of the variation is presumingly Monte Carlo variation caused by sampling in the microsimulation.

8. Microsimulation without Interaction: Discussion and Outlook

The first part of this thesis deals with the demographic microsimulation that the author has designed within the project *MicMac Bridging the micro-macro gap in population forecasting.* In Chapter 2, we described its stochastic model of individual behavior. Furthermore, we presented an extension of the stochastic model used in MicMac. While in the MicMac model individual life-courses evolve along two time scales, age and calendar time, the extended model additionally considers a third time scale; the time that an individual has already spent in his/her current demographic state. Chapter 3 was concerned with the preparation of the input data of a continuous-time microsimulation. We concentrate here on the presented model extension; i.e., on the model that specifies life-courses along age, calendar time, and time already spent in a state. We also discussed data types and methods that can be used for estimating transition rates and an initial population. Furthermore, we pointed to the necessity of building reasonable scenarios. Chapter 3 closed with some remarks about the preparation of migration input data for the presented microsimulation.

Implementing the MicMac microsimulation and its extension using a formalism that supports discrete event simulation allows us to properly specify the kind of multi-state model that we use to describe individual life-courses. Due to its modular and state-based concept, the DEVS formalism is well-suited to formulate population dynamics exactly in the intended way. The formalism is proven, tested, and widely established; and in addition there are a couple of DEVS based tools around (like the DEVS variants in JAMES II) that support the implementation of DEVS models. Using such a tool assures that we do not create a complete application from scratch. In Chapter 4 we presented two DEVS model specifications of the MicMac microsimulation and its extension. We described an atomic DEVS model and a DYNPDEVS model. The atomic DEVS model features strengths, but also limitations. The limitations are addressed by exploiting a DEVS variant that supports variable structures (i.e., DYNPDEVS) and by modelling individuals as atomic models and the population as a network model. Subsequently, we detailed the semantics that have originally been developed for the execution of a DYNPDEVS model. In Chapter 5 we described a low-end implementation of the MicMac microsimulation: we have directly translated the MicMac model into source code, using the modelling and simulation framework JAMES II as a library. Proceeding like this poses some flaws, for instance, the microsimulation cannot easily be extended (e.g., to include linked lives). However, as prototype this implementation suffices. Using its low-end implementation, we illustrated the capabilities of the MicMac microsimulation in Chapter 6. We conducted population projections for a population resembling the Dutch population. Thereafter, in Chapter 7, we carried out performance experiments to find an efficient data structure to maintain event lists, and to study the behavior of different pseudo random number generators. Finally, we verified and validated the microsimulation model, its simulation algorithm, and its implementation. For this purpose we re-estimated empirical transition rates, assessed the suitability of our simulation results, and cross-model validated the microsimulation using the MicMac macroprojection. In summary, we deem the MicMac microsimulation suitable for computing population projections at a very detailed level.

Comparisons with other microsimulation models and corresponding software, that allow the use of the same stochastic process model, would also be desirable. Naturally, only continuous-time microsimulations tools would be candidates for comparison. LifePath (Statistics Canada, 2011a) and PENSIM (Holmer et al., 2009) are two products that come close to the features of MicMac. Thorough and in-depth comparisons are, however, hampered by several obstacles. For example, the simulation procedure of LifePath is hidden from the user.[1] In general, microsimulation tools are tailored for specific applications (such as a national tax or pension system) and are consequently not developed with the intention to make it a tool for general applications, based on a generic model. As a consequence, technical information is rarely available, source code is not open and the data, on which the simulations are based, are part of the model and often cannot be accessed or changed. Therefore a direct comparison of performance of other continuous-time microsimulation products is not part of this thesis.

Despite its capabilities, the MicMac microsimulation has limitations that should be relaxed to allow more realistic descriptions of individual behavior and hence population processes. Currently, individual life-courses are simulated independently. Hence the concept of linked lives is not yet available in the MicMac microsimulation. This includes the inheritance of characteristics of parents (as adults) to newborns. Also, we currently do not model possible changes in transition rates of one person due, for example, to 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. In the following parts of this thesis we will tackle two of the listed limitations: First, we propose a technique to include a marriage market into a continuous-time microsimulation. Then, we present a microsimulation model, inclusive simulation algorithm, that involves both independent individuals and married/cohabiting couples. Other extensions, such as the consideration of the inheritance of behavioral traits, are not handled within this thesis and are subject of future work.

---

[1] LifePath has been designed using the generic microsimulation language “ModGen” which is a shortcut for Model Generator (Statistics Canada, 2011b). ModGen is a subset of the C++ programming language. It can be used to establish microsimulation models that are variants of LifePath. ModGen has, however, drawbacks: It requires Visual Studio 2008 which is a commercial and complex software product, and therefore implies additional expenses. The source code of the ModGen library is not accessible, and any new microsimulation applications necessarily has to be a variant of LifePath, with all its pre-set assumptions and variables. This makes a comparison on identical models and data basically impossible.

Part II.

MATCHING OF INDIVIDUALS

Simulating individual life-courses independently is the usual starting point for demographic microsimulations, but realistic population modelling cannot get away from considering so-called 'linked lives'. People cohabit or marry, have children and live in families; and this environment has an impact on their demographic behavior. This implies that some individual demographic events require that other individuals are linked to a person, and the relationship between linked individuals may modify their future behavior (i.e., the behavioral model describing concordant life-courses). Neglecting kinship and partnership relations in simulations will produce misleading outcomes. As a first step, individuals that will be linked together by an event have to be identified.

In this part of the thesis, we consider one particular problem, namely, the onset of partnership (cohabitation or marriage) and the matching of proper mates in a continuous-time microsimulation model. Continuous-time microsimulations pose some problems when forming relationships that discrete-time models can avoid.

In discrete-time models, which update information on demographic events at discrete points in time (commonly each year or each month), it is convenient to construct mating pools at equidistant time points; e.g., for every year. During simulation, individuals enter these mating pools, and they then undergo mate-matching. In continuous-time models events occur at exact time points and individuals will never practically experience partnership events at the same time. Therefore, a pool of potential partners cannot as easily be constructed as in discrete-time models.

A simple way to avoid this problem would be to use a so-called open model. In this model class, spouses are created as new individuals when needed, rather than selected from already existing members of the population. Although such 'external' partnership formations do happen in real populations, they constitute the minority of cases. Open mating models would therefore artificially increase the number of individuals, and these new individuals would still have to be supplied with individual attributes to allow realistic simulation of their remaining life-course.

We focus therefore on so called closed models, where appropriate spouses have to be identified from the current members of a population. Both discrete and continuous-time models need an algorithm to determine appropriate partners: Who should be linked to whom? Humans mate assortatively (Blossfeld and Timm, 2003; Kalmijn, 1998); that is, partners are not chosen randomly but are selected according to preferences. Spouses are typically similar with respect to certain attributes, and synthetic couples should reflect this habit.

We structure this part of the thesis as follows: in a first step we briefly describe behavioral models used in demography; we distinguish between models with implicit behavioral rules and models with explicit behavioral rules. In Chapter 10, we review the mate-matching algorithms that were already proposed for microsimulations. We also discuss which ideas and techniques of these approaches are useful for mate-matching in a continuous-time model. In Chapter 11, we propose a novel mate-matching procedure for continuous-time microsimulation and we critically evaluate its capabilities. We illustrate the approach by running a microsimulation for a synthetic population reflecting characteristics of the Dutch population. Results are given in Chapter 12. We conclude by validating the new procedure and by giving an outlook to future work.

9. Behavioral Models

Conventional demographic approaches to model partnership dynamics might provide some helpful ideas for the development of an effective and realistic micro level mate-matching strategy. To model observed partnership profiles, two main approaches can basically be distinguished: nuptiality models and behavioral models. The most important representatives of the former model class are two-sex macro mating models which employ a marriage function to describe observed marriage numbers. By contrast, in behavioral models the endeavor is made to mimic individual mating behavior. Here the modeler is confronted with the problem that mating rules are largely unobservable, and only their outcome can be observed. In this thesis, we restrict our review to behavioral models, and refer for more details concerning two-sex macro mating models to the review paper by Pollard and Höhn (1993). Behavioral models can generally be distinguished into models with implicit and explicit behavioral rules.

9.1. Behavioral models with implicit behavioral rules

In demography, behavioral models with implicit behavioral rules are a conventional way to describe empirically observed age-at-marriage patterns. The Hernes model (Hernes, 1972) and the Coale-McNeil model (Coale, 1971; Coale and McNeil, 1972) are important representatives of this model class. Both models map first-marriage age-patterns.

Although the Hernes and the Coale-McNeil model are macro models, they are built on some assumptions about individual behavior: the Hernes model introduces a threshold value for the acceptance of partners, which decreases with age. The Coale-McNeil model decomposes the complex process of marriage into several sub-processes. One process drives the entry into the marriage market, another one determines the waiting time until marriage.

In more detail: the Hernes (1972) model is a diffusion model in which an individual's propensity to marry is positively correlated with the proportion of already married individuals of the same cohort. The model builds on two main forces: "First, the pressure to marry increases with age because of the existence of social norms stating that 'who married late marries ill'. [...] Second, as time (and age) goes by the 'marriageability' of individuals is reduced, so while each individual may become more eager to marry, he or she becomes less able to secure a willing partner" (Todd et al., 2005, p. 562). The application of these forces to an artificial adolescent cohort results in a unimodal age distribution that resembles actual age profiles at first marriage. Detailed information about nuptiality diffusion models and their applications are given in Billari (2001) and Billari et al. (2007).

The Coale-McNeil marriage model is a latent-state model (Todd et al., 2005). The basic idea is that individuals experience a sequence of transitions between different latent stages until they enter a partnership. The Coale-McNeil model defines the age at first marriage as a sum of four unobserved random variables (Coale and McNeil, 1972): the age at which a person becomes marriageable (entry into marriage market), the waiting time until meeting a potential husband/wife, the waiting time between starting dating and engagement, and the waiting time between engagement and marriage. A normal distribution is used to describe the age at entry into the marriage market and three exponential distributions to describe the waiting times until marriage. In this case, the distribution of age-at-first-marriage follows a double exponential distribution (Coale and McNeil, 1972; Liang, 2000). For specific age profiles, e.g. standard schedules for countries, the parameters of a double exponential distribution have to be estimated. The Coale-McNeil model has been widely applied. Reviews are given in Liang (2000) and Kaneko (2003).

The basic assumptions of the Hernes model and the Coale-McNeil model about individual behavior will be useful building blocks in micro level modelling.

9.2. Behavioral models with explicit behavioral rules

The waiting time until partnership onset is the outcome of an interplay of the demand and searching strategies of individuals looking for mates and the availability of proper spouses. In macro models, as described above, individual mate-searching processes are not explicitly modelled. Recently several researchers employed agent-based models in order to reveal individual searching strategies that can lead to observed aggregate partnership patterns. The resulting model class is called behavioral models (Burch, 1995) because humans are modelled as agents with different attributes that interact to find a proper partner. Interaction is guided by a set of behavioral rules. During the search process, agents suffer from ‘bounded rationality’ (Simon, 1957): they face constraints regarding time and information. Individuals can only use their restricted period of “marriageability” to detect a proper spouse. Furthermore, individuals do not have perfect information about the composition of the marriage market so they will never know all potential spouses. Todd (2002, 1997) elaborates that, as a consequence of human bounded rationality, individuals apply ‘fast and frugal’ search heuristics in their decision processes. Todd and Billari (2003) and Todd et al. (2005) apply this finding in models of individual mate-choice processes. They design agent-based models in which a mate searcher sequentially encounters potential spouses. Decisions are made by individual aspiration-based heuristics. Each seeker has an aspiration level concerning the traits he/she appreciates in a spouse.[1] If the traits of a candidate meet or exceed the aspiration level, a partnership is formed. Otherwise, the seeker waits for the arrival of a new candidate. Simon (1990) calls this procedure a *satisficing* approach: different candidates are inspected until one is found that “meets the expectation.”

---

[1] The aspiration level could also be replaced by a utility level (Dagsvik et al., 2001).

One-sided and mutual decision processes have been employed to model mate-choice behavior. To study whether a one-sided decision process can result in an actual age-at-marriage distribution, Billari (2000) assigns a value to each candidate that maps his/her quality as a mate. Seekers sequentially encounter randomly chosen opposite-sex candidates. During a learning phase, they only adjust their aspiration levels, but no couples are formed. After a seeker has inspected a pre-set number of potential partners (= learning phase), he/she sets the aspiration level to the mate value of the highest quality candidate that he/she has met. The seeker then mates the first candidate whose mate value is at least as high as his/her aspiration level.[2] Only if heterogeneity in the length of the learning phase is allowed, Billari (2000) finds that this model can fit observed age-at-first-marriage profiles.

As a one-sided mating process is an unrealistic assumption in a modern society, Todd et al. (2005) suggest a more realistic two-sided mutual-choice situation. The adjustment of the individual aspiration level relies on offers and rejections received by others. During a normally distributed learning phase an individual raises the aspiration level if he/she receives an offer from a higher-valued member of the opposite sex, and lowers the level each time a lower-valued individual refuses an offer. As a result, individuals rapidly estimate their own mate value and mate with individuals with similar values. This mate search heuristic has also proved to be capable to reproduce the observed distribution of ages at first marriage.

French and Kus (2008) argue that in reality males usually show off in front of females, and females make the final mating decision. In line with this, they suggest to modify the latter heuristic by integrating a mate-choice asymmetry: in their model males make a proposal to females and females accept or reject it. Dating works according to the rules of a stable marriage algorithm (Gale and Shapley, 1962). In reality, people invest differently in encountering and mating potential mates. In their model, French and Kus (2008) account for this behavior by assigning to each individual a value that maps the level of energy that the individual is willing to put in dating and mating. They call the value “temperature”. For example, single females in their mid-thirties put more energy in the search than males at the same age. Therefore, to them a higher temperature is assigned. How many potential spouses a seeker will scan depends on his/her “temperature.” The higher the temperature, the more individuals are inspected. French and Kus (2008) find that their model outcome corresponds with actual partnership profiles as well.

In all these behavioral models, an individual passes through four stages: Entering the partnership market, meeting potential partners, making a decision whom to ask, and waiting for a reply. Simple microsimulation models, where events for an individual are generated by empirically determined transition rates, do not allow an immediate mapping of such behavior. Here the age and the calendar time at which a partnership starts is triggered by a mechanism external to the individual. Nevertheless, the mechanisms that drive the decision of partnering can be imitated in a mate-matching procedure of a microsimulation model. The essential

---

[2] This procedure is well known in the field of discrete optimization. It is called the secretary or dowry problem (Ferguson, 1989).

concepts of the reviewed behaviorial approaches, which permit realistic models, are:

- Searching individuals encounter potential spouses sequentially.
- The decision process with whom to mate is mutual.
- Spouses are *satisficing*. For the quality of a required match each seeker exhibits an aspiration level.
- Once two individuals meet, they take a decision and communicate it. There is no way of returning to an already encountered candidate.
- Until onset of a partnership each individual inspects a random number of potential spouses. How many candidates effectively are inspected depends both on the attributes of the seeker and of the available candidates.
- With each rejection an individual lowers his/her aspiration level.

10. Mating Models of Demographic Microsimulation Models

The ultimate goal in demographic mate-matching models is to determine synthetic couples, for which the partners resemble the characteristics that are found in real couples in important aspects. To achieve this, data from actual couples are necessary. Retrospective surveys that record the attributes of spouses before and after the onset of partnerships would be most appropriate. Surveys mostly gather, however, only very limited information on partnership relations (Huinink and Feldhaus, 2009). Problems also arise because actual mate-choice mechanisms are largely unobservable, and only the outcome of these mechanisms can be seen. Therefore assumptions on the process of mate-selection have to be made, however, the simulated outcomes can be contrasted with those actually observed (de Vos and Palloni, 1989). If in a two sex model the searching process is not explicitly modelled, like it is the case in most microsimulations, a further difficulty emerges: first an event that leads to the onset of a partnership is simulated and only then the appropriate spouse is determined. This is in contrast to the temporal progression in the real world, where marriages are formed only if partners have found each other.

Two main approaches are used to match individuals for partnerships: open mating models and closed mating models. In open models, appropriate spouses are created anew, when needed, while in closed models partners are identified among already existing individuals. In the following sections we review the existing approaches to mate-matching before introducing our own proposal in Section 11. For a broader overview of microsimulation models, including discussions of model purposes (policy issues, population projections, etc.), the empirical data base applied, sample size issues, and the modelling of time (discrete or continuous), see O'Donoghue (1999), Zaidi and Rake (2002) and Spielauer (2002).

10.1. Open mating model

In an open model, appropriate spouses are created 'ex nihilo'. Their attributes are generated in such a manner that the characteristics of the newly created couples resemble actual ones. Age and educational attainment are the attributes that are usually regarded as being essential in this context.

Open mating models have been developed for discrete and continuous-time microsimulation models. The discrete-time microsimulation CAMSIM (Smith, 1987), and the continuous-time microsimulations PENSIM (Holmer et al., 2009) and LifePaths (Statistics Canada, 2011a) are examples of open mating models.

As it is not necessary to identify a proper spouse from within the model population, the implementation of open models is straightforward. A further advantage of this approach is that simulations for individuals (and their immediate families) can be run independently of other individuals (O'Donoghue, 1999). Open models reveal, however, three major problems. First, as spouses are created when needed, the assumption of an unlimited supply of 'appropriate' spouses is made, which is not in line with observed shortages in partnership markets (Harknett, 2008; Warner et al., 2010). Consequently, the newly created individuals are not necessarily representative of the target population. Second, the interpretation of an open model is difficult. The purpose of a microsimulation is to model population dynamics realistically. It is not, however, realistic to pull an appropriate spouse 'out of the hat' when needed. Although nowadays individuals increasingly mate spouses from abroad, the majority of couples are established between persons that live in the same country or even region. Furthermore, as a mating process precedes each partnership formation, partners from abroad do not simply appear either. Finally, the retrospective life-course, that is the biography before partnership formation, of a newly created spouse is missing.

To address the latter issue, Smith (1987) describes a method to create spouses who are age-matched by mimicking age differences that are reported for real couples. This method is employed in CAMSIM. To address the problem of missing retrospective life-courses, in PENSIM (Holmer et al., 2009) complete retrospective life-courses of spouses are sampled from retrospective surveys. This approach presupposes the availability of extensive event history data, which can be a severe restriction in many applications.

10.2. Closed mating model

In a closed model, partners for marriage or cohabitations have to be found among existing individuals. Besides the time, when couples are formed, the following issues have to be addressed:

> How can we determine which of the individuals is in the (mating) pool or pool of eligible partners?
>
> Who matches whom?
>
> What are the mating rules?

All closed mating models so far were realized in a discrete-time microsimulation model and we consequently restrict this review to discrete-time models.

In a discrete microsimulation model, time changes in discrete steps. After each step, all individuals of the model population are inspected whether they will experience an event during the next interval and, if yes, which event this will be. In case an individual is scheduled to experience the onset of a partnership (marriage or cohabitation), he/she is marked as being searching for a spouse. At each time step, after every member of the population has been inspected, all searching individuals are collected in a partnership market. A ‘partnership market’ is a construct that is used to pool all those individuals who look for a spouse. Technically speaking, the partnership market is a sorted list of individuals. One important sorting criterion is age, but additional criteria may be added to mimic real assortative mating.

For constructing couples from the individuals in the pool an identical number of female and male candidates is mandatory. In demography this problem is known as the *two-sex problem*. Van Imhoff and Post (1998) discuss its relevance for microsimulation models. To assure an identical number of males and females two alternative strategies have been proposed in the literature.

1. If in a given period the number of males (females) exceeds the number of females (males), the excess males (females) remain unmarried. In the next period, they are either again at risk to experience a partnership event (Leblanc et al., 2009; Perese, 2002), or they are automatically members of the pool of potential spouses, i.e., they are pre-scheduled for a cohabitation or marriage event (Hammel et al., 1990).
2. Individuals are added or removed as needed to or from the pool of candidates (Leblanc et al., 2009). This implies an open mating model.

Both strategies imply that some of the outcomes of the stochastic model (i.e., a union formation) are disregarded. For each individual in the partnership market a marriage or cohabitation event has been simulated. Therefore, removing excess individuals from the partnership market means to ignore a simulated event. Likewise, adding individuals to the market means that some individuals, who were not simulated to experience a marriage or cohabitation event, get partnered.

In either case, only as many couples can be built as females and males, respectively, are pooled in the market. For constructing synthetic couples, two problems have to be solved: Who mates whom, and what data are needed to construct couples that resemble actual/observed ones? Both questions concern the mating rules that are applied to match individuals. Generally, two types of mating rules can be found in microsimulation models: stable and stochastic mating rules (Perese, 2002). All mating rules make use of a compatibility measure to determine the quality of potential pairings. Based on the attributes of the potential spouses this measure quantifies the quality of the respective pairing. First we discuss compatibility measures, and then we describe stochastic mating rules.

10.2.1. Compatibility measure

A compatibility measure transforms female and male attributes into a numeric index that quantifies how compatible a woman and a man are. Commonly, values between zero and one are used to express compatibility, with a large value indicating high compatibility. Likewise, a small value points to incompatibility. According to the theory of assortative mating, partners tend to have similar attributes like similar ages and levels of education (Kalmijn, 1998). That is, for mate-matching we use a compatibility measure that basically describes the similarity/dissimilarity between potential spouses based on certain important attributes.

We introduce some notation.

- At some point in time the partnership market comprises m women and n men.
- We denote the female attributes by w_i, $i = 1, \cdots, m$, and the male attributes by m_j, $j = 1, \cdots, n$. Typically, such attributes can be represented by a vector, and the different components of w_i and m_j, respectively, quantify characteristics such as age, educational attainment, etc.
- The set $\mathcal{F}$ comprises all w_i and the set $\mathcal{M}$ all m_j.

The compatibility measure C is defined as a following mapping:

$$\begin{aligned} C &: \mathcal{F} \times \mathcal{M} \to [0,1] \\ (w_i, m_j) &\to C(w_i, m_j) = c_{ij} \end{aligned}$$

If $C(w_i, m_j) > C(w_k, m_j)$ for two females i and k and the same male j, then the pairing of i and j shows a better agreement than the pairing of k and j. Commonly, the elements of w_i and m_j only give age and educational attainment of males and females. Bacon and Pennec (2007) provide an extensive review of attributes that have been employed in mating models.

Two different specifications of C are typically used: distance functions, and the likelihood of a union between potential pairings. Distance functions measure the discrepancy in the attributes of spouses; for example the following exponential distance function (Perese, 2002):

$$C(w_i, m_j) = \exp\big(-0.5\sqrt{(a_j - a_i)^2 + (e_j - e_i)^2}\ \big),$$

where a_i and e_i indicate the age and the educational level of a woman i, and a_j and e_j the respective values of a man j. As the argument of this exponential function is always negative, large age and education differences yield values close to zero, whilst small differences result in values close to one.

The likelihood of a union between potential pairs can be quantified by logit models (Bouffard et al., 2001; Perese, 2002). The model predicts the probability that two individuals, each with given attributes, form a partnership. Data on observed couples are used to estimate the coefficients of these models. Ideally, the estimated coefficients are in accordance with the theory of assortative mating (Bouffard et al., 2001; Leblanc et al., 2009).[1] In order to account

---

[1] Naturally, persons that are closely related show strong assortative characteristics. Therefore, in order to avoid incestuous pairings, the compatibility measure should be controlled by an incest taboo, e.g., by assigning a compatibility value of zero to brothers and sisters. The same is true for pairings between previous partners.

for different types of partnerships (cohabitations and marriages) and to differentiate between first and higher order partnerships[2], typically more than one logit model is applied.

10.2.2. Stable mating rules

Once compatibility measures are assigned to individual pairings, the actual unions have to be formed. To match individuals so called stable or stochastic rules can be employed. The problem of finding stable mating rules is equivalent to the *stable marriages problem.* Gale and Shapley (1962, p. 11) describe the problem as follows:

> “A certain community consists of n men and n women. Each person ranks those of the opposite-sex in accordance with his or her preferences for a marriage partner. We seek a satisfactory way of marrying off all members of the community. [...] we call a set of marriages unstable [...] if under it there are a man and a woman who are not married to each other but prefer each other to their actual mates.”

In other words, stable mating rules produce a set of stable matings. Gale and Shapley (1962) prove that, for any number n of men and women, it is always possible to solve the problem. They also developed an algorithm that produces a set of stable marriages.

The stable marriage approach requires that, each woman expresses her preference regarding each man and vice versa. It is based on a sequence of proposals from men to women. Each man proposes, in descending order, to the women (according to his preferences). A man is pausing when a woman agrees to consider his proposal. He is continuing if a proposal is immediately or subsequently rejected. When a woman receives a proposal, she rejects if she already holds a better proposal (relying on her preferences). Otherwise, she agrees to hold the proposal for consideration. In doing so, she rejects any poorer proposal that she may hold. This procedure assures that no man can have a better partner than he gets in this matching and no woman can have a worse one. Consequently, the Gale-Shapley algorithm produces marriages that greatly favor the men’s preferences.

Since 1962, several studies improved the algorithm (Gusfield and Irving, 1989; Ma, 1995; Teo et al., 2001). Researchers mainly worked on two drawbacks of the Gale-Shapley algorithm: The sex that makes the first proposal is favored, and the classical Gale-Shapley algorithm does not result in a unique set of stable marriages.

The degree of compatibility between a woman and a man can be taken as a proper measure of their revealed preferences for each other. From now on, we quantify this degree using the compatibility measure C that has been introduced in Section 10.2.1. Its usage for constructing synthetic couples results in a simplified stable mating problem (Bouffard et al., 2001). While the preference of a woman for a man may differ from the man’s preference for the woman, the compatibility measure is symmetric in its treatment of men and women.

---

[2] We call a second (third, etc.) marriage a higher order marriage and a second (third, etc.) cohabitation a higher order cohabitation.

Based on the compatibility measure C, a stable set of pairings is defined (Perese, 2002) in the following way: for all couples (w_i, m_j) and (w_k, m_l), $i \neq k, j \neq l$ the subsequent condition holds true:

$$C(w_i, m_j) \geq C(w_k, m_j) \quad \text{or} \quad C(w_i, m_j) \geq C(w_i, m_l),$$

i.e., either no man can make a better match than the one he gets (first inequality) or no woman can make a better one (second inequality). Only if both inequalities fail, a set of pairings is unstable.

To find a stable set of unions, the following algorithm can be used:

1. Men and women are split into separate sets, and the compatibility measure is computed for all potential pairs.
2. All pairings are ordered according to their compatibility measure (in descending order).
3. Those two individuals that have the highest degree of compatibility are matched, and all pairings that include one of the spouses of this new couple are removed from the list of potential couples.
4. The compatibility of the remaining individuals is re-ranked and the next most compatible couple is paired.
5. This procedure is repeated until all matches have been made.

If there are ties, an additional selection rule has to be defined. This algorithm has a time complexity of $O(n^2)$, where n is the number of pairs to be formed.

Implementations of the stable mating algorithm can be found in the Swedish spatial microsimulation model SVERIGE (Holm et al., 2002) and the U.S. policy microsimulation POLISIM (Bouffard et al., 2001; Caldwell, 1996; Caldwell et al., 1999; O'Harra and Sabelhaus, 2002). Although the stable mating approach is based on extensive research, is easy to understand and implement, and treats both sexes equally (Bouffard et al., 2001), it suffers from a considerable deficit. In the beginning, it produces couples that have high compatibility measures, but towards the end of the process only individuals who do not match well remain in the pool. Consequently, the algorithm creates matches that hardly have any counterparts in reality.

Bouffard et al. (2001) study the effects of this imbalance. Using the Canadian 1981 census they employed a logit model to measure compatibility. They found that the stable mating algorithm produced too many "extreme" pairings, such as couples with age differences of spouses greater than 20 years. To overcome this problem of a stable mating algorithm Bouffard et al. (2001) studied several modifications, none of which lead to a significant improvement of results though.

Leblanc et al. (2009) described an algorithm (ODD, order of decreasing difficulty) that first finds good matches for those individuals who show undesirable characteristics and then continues to construct pairs in decreasing order of difficulty. Unfortunately, this algorithm is also not capable of reproducing actual data well.

Randomly reducing the pool of prospective spouses is another approach that was suggested to solve the problem of "bad" matches (Cumpston, 2009; Leblanc et al., 2009). The corresponding procedure randomly draws an individual i from the pool of prospective spouses. Then a certain number p of opposite-sex individuals is randomly selected from the pool, and the one that shows the highest compatibility with i is selected as spouse, regardless of whether this value is high in absolute terms. This procedure is repeated until the marriage market is depleted.

Leblanc et al. (2009) discuss that this modified approach still suffers from the problems of the original stable mating algorithm. They find that "the algorithm generates far too many marriages with extreme age differences" (Leblanc et al., 2009, p. 18). Moreover, the constructed matchings show a distribution of compatibility measures that diverges significantly from the one estimated from real data.

In conclusion, the concept of a compatibility measure and the stable marriage algorithm are incompatible: For "any arbitrary pairing, the measure's value should be proportional to the probability that those persons end up marrying. The stable marriage algorithm, however, actively departs from this property because, in its quest for stability, it disproportionately favors for pairings with high compatibility values" (Bouffard et al., 2001, p. 15). Stochastic mating rules are an option to overcome this problem.

10.2.3. Stochastic mating rules

In a stochastic mating model, the compatibility measure between a woman and a man determines the probability of a respective match and the outcome of a stochastic experiment determines whether a match between two potential spouses occurs.

A stochastic matching procedure ensures that individuals with a low compatibility also have a chance to get matched. With regard to their compatibility, constructed couples are thus not necessarily optimal ones. As a result, the occurrence of "extreme" matchings is less likely, which is a big advantage over the stable mating algorithm.

In microsimulation models, three variants of stochastic mating are basically applied, depending on whether and which sex dominates the choice of spouses (male-, female- or mixed-dominant algorithms). Figure 10.1 shows the presented classification of mating models and mate-matching algorithms designed for microsimulation models.

In male-dominant mate-matching algorithms, men choose their spouses from a list of eligible women. In the 1970s, the DYNASIM team developed an efficient algorithm that produces acceptable results in linear time (Perese, 2002). The algorithm involves the following steps: Men and women, not necessarily of the same number, are put into separate lists. The elements in each list are put in random order. The first man on the male list is selected. He will 'see' a total number of K women, where $K = 10$ for men below age 35, and $K = 20$ otherwise. Therefore a vector of length K for the respective compatibility measures is initialized. The

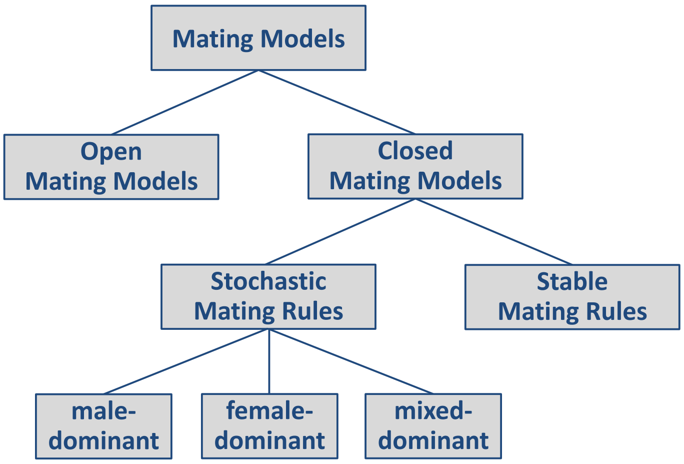


Figure 10.1.: Classification of mating models and mate-matching algorithms for microsimulation.

first woman from the female list is selected and the compatibility measure for this pair is calculated. If this compatibility measure is greater than a random number R, which is uniformly distributed between 0 and 1, then the match is made. Otherwise, the next woman from the female list is taken, the compatibility measure is computed and compared with a new random number R. Should no match have been made after all K women were 'presented', the one (out of the K) with the highest compatibility measure is chosen as spouse. The procedure then moves on to the next male, who again 'meets' women from the top of the female list, and it continues until either the male or the female list is empty.

DYNASIM originally used an exponential distance function (cp. Section 10.2.1) as compatibility measure. Perese (2002) replaced it by logit models, and found that, due to this replacement, many potential couples have very low compatibility measures. As a result, the probability of producing matches declines, and more iterations are needed to find a proper spouse. A significant increase in the algorithm's runtime is the consequence. A remedy to this problem is using normalized compatibility measures. Before the matching starts, for each man, the highest compatibility value he can achieve is determined. Subsequently, all compatibility measures that a man exhibits with potential spouses are divided by this highest value. The normalization ensures that a man shares a compatibility value of 1 at least with one woman. Perese (2002) argues that "this technique creates a more randomized process than the one employed in DYNASIM, which arbitrarily limits the search to 10 women for each man before a match is made with certainty" (Perese, 2002, p. 17). The modification, however, increases the complexity of the original algorithm, which is $O(n^2)$ instead of the original $O(n)$. In order to test the algorithm, Perese estimated logit models using SIPP (Survey of Income and Program Participation) survey data. Simulation runs showed that the algorithm closely replicated actual data.

In a female-dominant mate-matching algorithm, the roles of women and men are simply reversed: women choose from a list of potential spouses (of fixed or random length). Female-dominant mate-matching procedures are used in the first version of SOCSIM (Hammel et al., 1976, Chapter 9), and in DYNAMOD (Kelly, 2003; King et al., 1999).

In a mixed-dominant mate-matching procedure both sexes are treated equally. Two variants, 'the sequential approach' and 'the concurrent approach' can be distinguished. The sequential approach has been realized in the SOCSIM microsimulation (Hammel et al., 1990) and the concurrent approach in the U.S. CORSIM microsimulation and the Canadian DYNASIM microsimulation.

In the sequential approach, individuals "seek a [...] partner in random order from among the members of the opposite sex in accordance with their criteria of preference" (Wachter, 1995, p. 7). In the terminology used here, the preferences are quantified by a compatibility measure. The actual spouse is chosen at random from the opposite-sex candidates with the highest compatibility. In contrast, in the concurrent approach first the compatibility measure between all potential pairings is determined, and then couples are constructed. The concurrent approach of stochastic mate-matching goes back to the work of Vink and Easther (Bouffard et al., 2001). It had been developed originally for the CORSIM microsimulation model.
The sequential approach, as realized in SOCSIM, performs the following steps:

1. Sort in random order all searching individuals $w_1, \cdots, w_m$ and $m_1, \cdots, m_n$ into a list L_1, and initialize a counter $k = 1$.

2. Draw an individual i at random from L_1.

3. All opposite-sex individuals that meet some minimal criteria (such as no incest, no remarriage of previously divorced couples, and no extreme age differences between the spouses) are inserted into a second list L_2. The number of individuals in L_2 is denoted by l.

4. If $l > 0$, then

    the values c_p of the compatibility measures between i and all individuals in L_2 are computed, $p = 1, \cdots, l$. All individuals in L_2 are sorted in decreasing order according to their compatibility measures.[3]

    The first M members of the list are selected. Their respective compatibility measures are denoted by $\tilde{c}_1, \cdots, \tilde{c}_M$.

    A random number R which is uniformly distributed between 0 and $\sum_{p=1}^{M} \tilde{c}_p$ is chosen.

    i is linked to the individual h, for which $\sum_{p=1}^{h-1} \tilde{c}_p \leq R < \sum_{p=1}^{h} \tilde{c}_p$.

The counter k is incremented by 1, and steps 2 to 4 are repeated until $k = \min(n, m)$.

Unsuccessful suitors, that is, if $l = 0$ in step 4, remain unpaired. An arbitrary number between 1 and l can be set for M. If $M = 1$, the sequential algorithm and the revised stable mating algorithm in Section 10.2.2 are equivalent. If $M = \max(n, m)$, the sequential approach has time complexity of $O(n^2)$. It has linear complexity, if the minimal criteria (e.g., no incest, no remarriage of previously divorced couples, and no extreme age differences between the spouses, cp. step 3) are rather restrictive and L_2 has only few elements. Wachter (1995) and Wachter et al. (1998) find that for $M = \max(n, m)$ the algorithm produces satisfactory results.

Bouffard et al. (2001), who extensively tested the concurrent approach of stochastic mate-matching, found that it is capable of reproducing actual data.
The concurrent approach performs the following steps:

1. The compatibility measures $c_{i,j} = C(w_i, m_j)$ for all possible combinations of women $w_1, \cdots, w_m$ and men $m_1, \cdots, m_n$ in the partnership market are computed, and put into a mn-dimensional vector v:

$$v = [v_k, k = 1, \cdots, nm] = [c_{1,1}, \cdots, c_{1,n}, c_{2,1} \cdots, c_{m,n}].$$

2. Initialize two variables in order to map the effective dimension of v: $p = m, q = n$.

3. Calculate the total sum over all compatibility measures: $C_s = \sum_{k=1}^{pq} v_k$.

4. Draw a random number R uniformly distributed between 0 and C_s.

5. Match the couple whose the compatibility measure is mapped by that v_k for which $\sum_{k=1}^{K-1} v_k < R \leq \sum_{k=1}^{K} v_k$.

6. Remove those components from v that refer to either the woman or the man of the newly created couple. Consequently, the number of elements of v is decremented: $p := p - 1, q := q - 1$.

   Repeat steps 3 to 6 until $\min(n, m)$ couples are formed.

As compatibility values always have to be computed for all potential pairings, the algorithm has time complexity of $O(n^2)$.

In conclusion, we can sum up the following findings for mate-matching algorithms of existing microsimulations:

- Closed models are easier to interpret than open models, and they enable us to study the effects of mating processes on the population composition.

---

[3] In SOCSIM for the assessment of compatibility, a function has been employed that yields small values for high compatibility, and vice versa. We modify the algorithm of SOCSIM such that it fits to the definition of the compatibility measure given in Section 10.2.1.

- To measure the compatibility between two persons, computing the likelihood of a potential pairing is more appropriate than using a distance function. Each strategy that has been proposed so far to obtain the same numbers of women and men in the partnership market has drawbacks.
- Stochastic mate-matching procedures resemble actual data better than stable mating procedures, but the outcome of a stochastic mate-matching algorithm is not significantly affected by the chosen variant (male-, female-, or mixed-dominant).
- In the context of stochastic mate-matching, a "sequential approach" is on average more efficient than a "concurrent approach".

In conclusion, for our mate-matching algorithm we opt to implement a closed model that embodies a two-sided "sequential" stochastic mate-matching procedure, as well as learning and satisficing seekers. Using in the mate-matching process compatibility measures to decide on matches ensures assortative mating: the more similar a woman and a man are the more likely is a linkage. The details of the approach are described in the following section. Note that a sequential stochastic mate-matching procedure is equivalent to the matching mechanism in search theory developed in labor economics; see, e.g., Woodbury and Davidson (2003).

11. A Mutual Mate-Matching Procedure in Continuous-Time

In a continuous-time microsimulation, events and waiting times to events are simulated based on empirical rates (Gampe and Zinn, 2007). When we simulate the entry of an individual into the searching and mating phase, we therefore rely on observed behavior. As the mate-matching procedure that we propose mimics human mating as a decision process, matches that are created during simulation are the outcome of intended behavior. Consequently, not all individuals who undergo during simulation a mate searching phase will find a partner. Reasons for this are the competition with others, or simply a short supply of spouses with compatible characteristics. We discuss implications of simulating intended instead of observed behavior in Section 11.4.

11.1. Scheduling of partnership events

In discrete-time models, the state that individuals are in is updated at regular points in time. Therefore monthly or annual partnership markets are a convenient concept in discrete-time models. In continuous-time models the probability that two events will happen at exactly the same time point is zero and practically individuals will never experience partnership events at the same time. Due to this design, a pool of potential partners is hard to identify. A way to tackle this problem is to include into the mate-matching algorithm the scheduling of events and the construction of a partnership market that individuals can enter or leave over the complete simulation time range. This will be described in the following, considering cohabitation and marriage as two separate types of partnership.

A partnership (marriage or cohabitation) has to have a clearly defined formation time. How to derive this time from the simulated events experienced by two individuals is illustrated by an example: a woman I_1 experiences the onset of a partnership at time t_1, and a man I_2 at time t_2. Without loss of generality, we assume $t_2 < t_1$. One way to compute a formation time $\tilde{t}$ of a partnership between I_1 and I_2 is

$$\tilde{t} = t_2 + c \cdot (t_1 - t_2), \;\; c \in [0, 1].$$

Then, instead of t_1 and t_2, for both I_1 and I_2, the adjusted $\tilde{t}$ is used as starting time of a partnership. The new partnership formation time $\tilde{t}$ is located between t_1 and t_2. Whether $\tilde{t}$ is closer to t_1 or to t_2 is determined by the parameter c. A setting of $c = 0.5$ results in the mean of t_1 and t_2. Changing simulated event times in this way, means changing the outcome of the microsimulation model. To avoid significantly biased outcome, we have to assure that $t_1 - t_2$ is small. Accordingly we decide that I_1 and I_2 can only be regarded as potential spouses, if their simulated times t_1 and t_2 are close enough. We define t_1 and t_2 as close enough, if $t_1 \in \Gamma_2$ and $t_2 \in \Gamma_1$, where

$$\Gamma_1 = [\min(t_S, u_1, t_1 - B), \max(t_1 + B, t_E)] \text{ and } \Gamma_2 = [\min(t_S, u_2, t_2 - B), \max(t_2 + B, t_E)].$$

t_S is the simulation start time, t_E the simulation stop time, and u_i is the time of the event that I_i has experienced previous to the upcoming partnership formation event, $i = 1, 2$. B is an arbitrary time period, but which is commonly shorter than one year. Γ_i is called the "searching period" of I_i.[1] Γ_i starts soonest with I_i's entry into the partnership market. For consistency, at the earliest, the searching period starts at simulation start time and, at the latest, ends at simulation end time.

With respect to this definition, individuals can only date if their searching periods overlap. Subsequently, we use $B = 0.5$ years and $c = 0.5$.[2] The latter results in $\tilde{t} = 0.5 \cdot (t_1 + t_2)$. Figure 11.1 illustrates the adjustment of event times using an example. Section 12.3 deals with how shifting events in the suggested way changes the output of the stochastic microsimulation model.

11.2. Compatibility of individual characteristics

Even if the searching periods of mating willing individuals overlap, their characteristics might not match. Therefore, besides event times, individual characteristics also have to be checked for conformance. For this purpose, a compatibility measure is used such as introduced in section 10.2.1. As distance functions measure the quality of pairings very simplistically, logit models are employed to evaluate how well the characteristics of potential spouses agree.

Which covariates will enter the logit models depends on the state space of the actual application. In this paper a generic microsimulation model is assumed; i.e., the state space is not fixed. Only individual age and sex are mandatory attributes. Depending on the problem to be studied, different relevant demographic states will be considered. Obviously, only those covariates can be included in the logit models that are in the state space. If, for example, educational attainment, children ever born, or ethnicity are included in the state space, these attributes are natural candidates for covariates in the logit models. For a pair of individuals, the compatibility measure gives, conditioned on the considered attributes, the probability of a union.

---

[1] That is, we determine the length of the searching period exogenously. In doing so, we follow Billari's (2000) approach (cf. Chapter 9.2). However, opposed to Billari (2000) we do not consider heterogeneity in the length of the searching period.

[2] Our case study shows that this parameterization yields plausible outcomes (cp. Section 12.3).

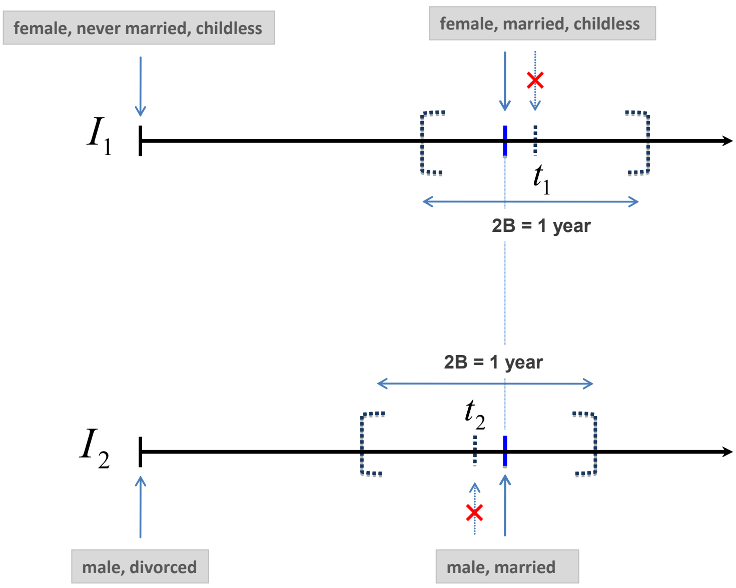


Figure 11.1.: Woman I_1 experiences a marriage event at time t_1. Man I_2 experiences a marriage event at time t_2. As $t_1 \in [t_2 - 0.5 \text{ years}, t_2 + 0.5 \text{ years}]$ and $t_2 \in [t_1 - 0.5 \text{ years}, t_1 + 0.5 \text{ years}]$ both individuals have overlapping searching periods and might meet during the mating process. Hence, they can be considered as potential spouses. Their formation time would be $\tilde{t} = 0.5 \cdot (t_1 + t_2)$, if they were actually linked in the mate-matching algorithm.

11.3. Partnership market and mate-matching procedure

We employ a partnership market to collect 'mating-minded' individuals (i.e., individual who want to marry or cohabitate). As soon as a marriage or cohabitation event has been simulated, an individual joins this market. He or she leaves the partnership market either after he/she has found a proper spouse or his/her searching period is expired. In contrast to partnership markets of discrete-time models, individuals can enter and leave the market over the complete simulation time range. We implement the partnership market using a so called marriage queue $\mathcal{M}$. The marriage queue consists of all unpaired individuals who look for a partner (because of a simulated partnership event). Each individual in the queue is equipped with a stamp that indicates the time of the upcoming partnership event. This setting is a direct consequence of the stochastic model that in a continuous-time microsimulation is used to assess events: empirical waiting times (derivable from empirical rates of mating) determine the occurrence and timing of partnership events. To be able to better explain the approach, we consider the following example: Simulation starts at time t_S when a woman I_1 is a_0 years old. She has never been married and is childless at this time. Conditional on her current state, her age and the current calendar time, we simulate a waiting time of $w = 3.6$ years to a marriage event. The woman I_1 in the example enters the market at time t_S and her waiting time to marriage is $w = 3.6$ years. Her searching period is $\Gamma_1 = [t_s + 3.6 - B \text{ years}, t_s + 3.6 + B \text{ years}]$, and with $B = 0.5$ it is $\Gamma_1 = [t_s + 3.1 \text{ years}, t_s + 4.1 \text{ years}]$. (Doubling B gives the length of the searching period, i.e., with $B = 0.5$ the length is one year.) An appropriate spouse for I_1 has to exhibit a searching period that overlaps Γ_1. Furthermore, the joint characteristics of I_1 and a potential spouse have to be in accordance with joint characteristics of actual couples.

We take into account that individuals have constraints regarding the social network size they can perceive. Humans are thought to be limited to social networks with approximately 150 members (Hill and Dunbar, 2003). Considering this fact, for each 'mating-minded' individual we restrict the number of potential spouses. We set an upper bound that follows a normal distribution with expectation $\mu = 120$ and standard deviation $\sigma = 30$.[3]

Furthermore, we assign to each individual a random value that captures his/her aspiration level regarding a partner. This aspiration level takes values between 0 and 1. If the compatibility measure between an individual and a potential spouse exceeds the aspiration level, he/she accepts the pairing. Thus mate-seekers are satisficing (Simon, 1990): different candidates are inspected until one is found that "meets the expectation". In the proposed algorithm, individuals reduce their aspiration level by δ_A, $\delta_A \in [0, 1]$, every time they are involved in an unsuccessful encounter. This way their chance to find a mate the next time is increased. The reduction of the aspiration level with each rejection corresponds to a strategy proposed by Billari (2000) and Todd et al. (2005). They suggest to use individual aspiration-based heuristics to model human mate-choice processes. The basic idea is that if the traits of a potential spouse meet or exceed an individual's aspiration level, a partnership is formed. Aspiration levels are adjusted according to offers and rejections received by others. Here this approach is adapted by reducing the aspiration levels of individuals who date but reject each other.

We assume that the individual aspiration levels follow a beta distribution. Based on the theory of initial parental investment, women are assumed to be choosier than men concerning their partners (Buss, 2006; Trivers, 1972). The degree of "choosiness" of females and males varies, however, with age. Women tend to decrease their requirements with declining fecundity. When they are in their early thirties they are less demanding than at younger ages. For single women older than 35 the "ticking of the biological clock" even leads to a considerably increased effort to date men (Pawlowskia and Dunbar, 1999). While after age 40, women tend to be more picky again (French and Kus, 2008). Although socio-economic factors play a role in this context , too, we only consider the age trajectory here.

Men, however, behave differently. When they are young, men are more involved in short-term relationships, and therefore are more interested in the number of sexual partners, not so much in the quality of a relationship (Buss, 2006). As a consequence, they are less selective concerning the traits of a partner. However, when men start to look for long-term relation-

---

[3] Considering this upper bound as the actual number of potential spouses that an individual can meet would be misleading. It should rather be interpreted as an upper bound for the maximal possible number of potential spouses. Naturally, we expect that the actual number of potential spouses is smaller than an individual's maximal social network size.

Description	**Parameter**	**Value**
Intersection of searching periods	B	0.5
Upper bound of number of potential spouses	N	normally distributed, $\mu = 120, \sigma = 30$
Individual aspiration level	a_i	beta distributed, gender- & age-dependent (cp. Figure 11.2)
Decrement of aspiration level in case of rejection	δ_A	0.1
Bound for small pool size	s_p	10
Decrement of aspiration level in case of small pool size	δ_B	0.3

Table 11.1.: Parameters and suggested parameter values for the sequential stochastic mate-matching algorithm.

ships, willing to establish a family and to invest in offspring, their behavior changes, and their level of "choosiness" increases. We choose 30 as the age when men start to intensively look for a long-term spouse. To account for the variability in the degree of aspiration we parameterize the beta distribution accordingly. The parameter values of the beta-distribution are gender-specific and vary with age, see Figure 11.2.

An important aspect is the size of the pool of potential spouses. If it is small, it is not reasonable to assume a very selective seeker. To increase the chance that each individual finds a partner, we assume that, if a seeker faces less than s_p potential partners, he/she reduces the aspiration level by δ_B, $\delta_B \in [0, 1]$, additionally to the reduction δ_A induced by failure.

All parameter values for the mate-matching procedure are given Table 11.1. For Western Europe, we deem the chosen parameterization reasonable as the results of the case study in Section 12.3 will illustrate.

To actually construct synthetic couples, we use a modified version of the sequential stochastic mate-matching procedure that was introduced in Section 10.2.3. If an individual I_i experiences the onset of a partnership, the following steps are performed:

1. The searching period Γ_i of I_i is determined, and his/her level of aspiration a_i is generated.
2. If the marriage queue $\mathcal{M}$ is empty, I_i is inserted into $\mathcal{M}$.

   Otherwise we

   > draw a random number N, normally distributed with expectation μ and standard deviation σ to define the size of the social network of I_i. If N is greater than the current number $N_{\mathcal{M}}$ of individuals in the marriage queue, we assign $N = N_{\mathcal{M}}$.
   >
   > We randomly take N individuals out of $\mathcal{M}$ whose searching periods have to overlap Γ_i, and we insert them into the so called working marriage queue $\mathcal{W}$.

Individuals of the same sex as I_i and individuals who do not meet some minimal criteria are removed from $\mathcal{W}$.

If $\mathcal{W}$ turns out be empty, we insert I_i into $\mathcal{M}$.

Otherwise we do the following:

(i) If $\mathcal{W}$ contains less than s_p individuals, we reduce the aspiration level of I_i to $a_i = \max(0, a_i - \delta_B)$, and

(ii) initialize $j = 1$.

(iii) We take the jth individual I_j of $\mathcal{W}$. We denote the aspiration level of I_j by a_j. We compute the compatibility measure $c_{ij} = C(w_i, m_j)$, or $c_{ij} = C(w_j, m_i)$, respectively, between I_i and I_j. If $a_i < c_{ij}$ and $a_j < c_{ij}$, the individuals I_i and I_j get paired, and I_j is removed from $\mathcal{M}$.

(iv) Otherwise, the aspiration level of I_i is reduced to $a_i = \max(0, a_i - \delta_A)$, the aspiration level of I_j to $a_j = \max(0, a_j - \delta_A)$, and j is incremented by 1.

We repeat steps (iii) and (iv) until either I_i is paired or all individuals of $\mathcal{W}$ have been inspected.

If no appropriate spouse can be found for I_i, he/she is enqueued into $\mathcal{M}$.

In other words, if I_i fails to find an appropriate spouse at the first try, he/she joins the marriage queue. Here I_i stays until a new individual enters the market, encounters I_i, and both agree to mate.

Both the terms “marriage queue” and “working marriage queue” used in the description have been introduced in SOCSIM by Hammel et al. (1990). To select individuals from the working marriage queue we use the following minimal criteria: no incest, no remarriage of previously divorced couples, and no extreme age differences between the spouses.

11.4. The difficulty of getting everybody matched

The presented mate-matching algorithm does not guarantee that each searching individual will be paired. Mate-matching fails if an individual is not able to find (within his/her searching period) a spouse with compatible characteristics. In order to be successful, each seeker has to have access to a pool of potential spouses. This can only be assured if the model population maps a large proportion of an actual population.

If the searching period of a “mating-minded” individual, who should find a partner but did not succeed, is expired, three options exist:

A. Extend the searching period. The individual remains in the marriage queue.

B. Return the individual to model population unpaired. The individual is removed from the marriage queue. He/ she is again at risk to experience a partnership event.

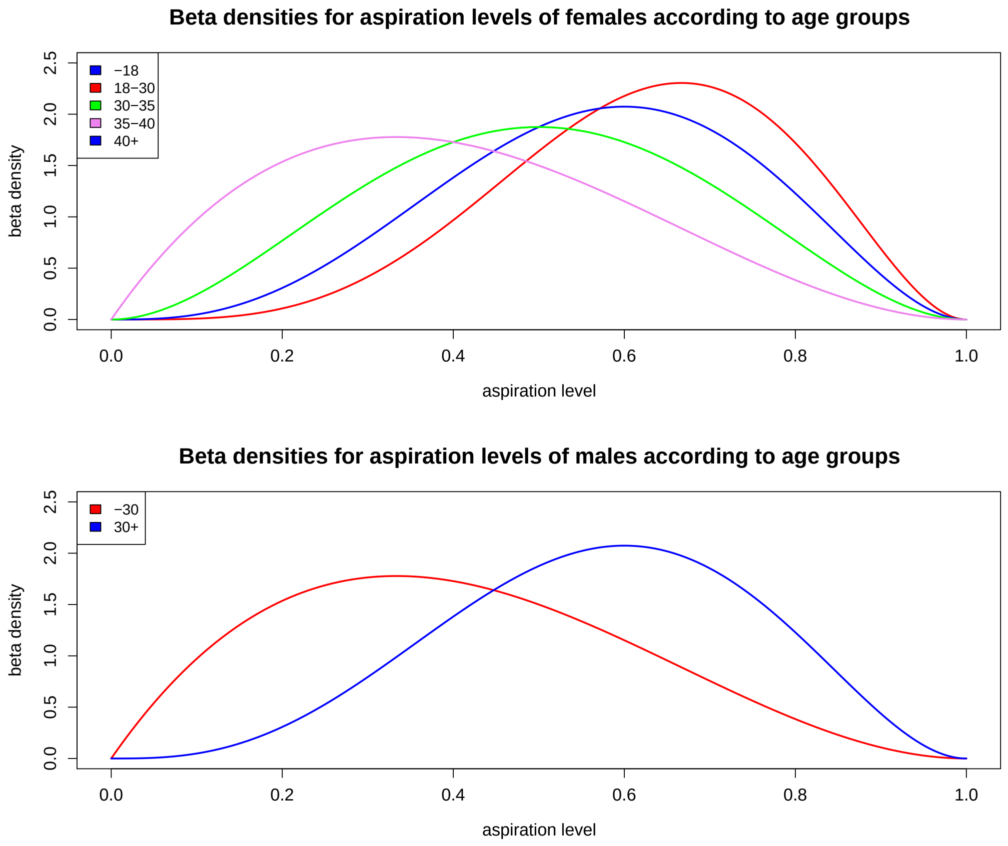


Figure 11.2.: Densities of the beta distributions that are used to determine aspiration levels regarding partners. The densities vary with gender and age. For females we apply four different curves: one below age 18, one for ages between 18 and 30, one between ages 30 and 35, and one after age 40. For males we apply two different curves: one for males younger than 30 and the other after age 30.

C. Let a proper spouse immigrate or the individual emigrate. The individual is removed from the marriage queue.

The last idea is borrowed from open models, as an appropriate spouse is created ‘ex nihilo’.

Each of these options entails a major difficulty. Extending the searching period means shifting the time of the partnership event (onset of marriage or cohabitation). Rejecting a seeker implies ignoring an already scheduled event. Allowing too many immigrated spouses spoils the representativeness of the model population. Consequently, in order to assure plausible outcomes, searching periods that expire without success should be an exception. This can be achieved by simulating a population that resembles an actual observed one in size, so that during the simulation period each individual can meet at least one potential spouse.

12. Mate-Matching in Practise: The Modern Netherlands

12.1. The synthetic population

To illustrate the developed algorithm, we use an example of a synthetic population that resembles the Dutch population. Starting at January 1, 2004, the life-courses of $139,048$ males and of $134,910$ females (which corresponds to 2% of the actual Dutch population aged 0 to 63 on January 1, 2004) are generated. The simulation evolves over 17 years, and stops at December 31, 2020. During simulation, women can become mothers, both men and women might increase their educational level, or experience changes in their marital status. Individuals can leave the parental home, marry, cohabitate, become single, etc., and all individuals are always exposed to death. In Section 6, the example setting and the employed data are described in detail.

The presented mate-matching algorithm has been included in the microsimulation implementation described in Chapter 5. To assure that each "mating-minded" individual is matched, we apply option A of Section 11.4: if somebody is not successful during his/her searching period, the timing of his/her marriage or cohabitation onset is shifted.

12.2. Computation of compatibility measures

To conduct the proposed mate-matching procedure, we require the computation of compatibility measures between potential spouses. For this purpose we used four logit models (see Section 10.2.1). Each model describes the probability to enter a specific partnership type from the perspective of the male spouse:

Model 1: entering first cohabitation

Model 2: entering higher order cohabitations

Model 3: entering first marriage

Model 4: entering higher order marriages

An individual who marries his/her common-law spouse, has already chosen him/her when entering the cohabitation. Therefore, in the two latter models, only such marriage events are considered that are not preceded by cohabitations. For estimating the models, we used the first wave of the Netherlands Kinship Panel Study (NKPS) that has been conducted in the period from 2002 to 2004 (Dykstra et al., 2005).[1] This study has included only partnerships that started in the years from 1990 to 2002. We have constructed a data set that contains for each observed couple a record that consists of:

- the age of the male spouse,
- the age difference between the female and the male spouse (in integer years),
- the levels of educational attainment for each spouse,
- an indicator whether the female spouse was married before, and
- the number of children that the spouses have with former partners.

This sample design is retrospective, i.e., in the data the attributes of female and male spouses were sampled conditional on being paired. To accurately estimate a retrospective regression model, case and control sampling fractions have to be consisted. However, in the present setting such data were not available because it cannot be observed who in reality did not mate. To nonetheless add information about controls, for each observed couple a synthetic couple was built by randomly assigning to each male spouse a female who was not his observed partner. The response variable was set to one in the case a couple had been observed. Otherwise, the response variable was set to zero. The constructed data set resembles the retrospective data design of a case-control study with an identical number of cases (observed couples) and controls (synthetic couples). Unfortunately, when conducting mate-matching we are confronted with a prospective problem: for measuring the compatibility of a pairing we need the likelihood that two individuals with certain attributes mate. A mandatory condition of a prospective model is that case and control fractions are made up by the source population, and not by the experiment designer; i.e., the sampling has to be random. Hence, for a prospective problem we have constructed a case-control data set that fits to a retrospective regression model. A solution to that problem can be found in Prentice and Pyke (1979). They show that all non-intercept parameters of a prospective logit regression model are asymptotically correct under case-control sampling. As a prospective model does not allow deterministic case and control sampling fractions, its application to a case-control data set causes a biased estimator of the intercept. In our mate matching procedure, compatibility between two potential spouses is measured on a relative scale, depending only on the attributes of two individuals, and not on the composition of the pool of available candidates. Therefore, for our purposes the estimation of a prospective logit model is suitable.

---

[1] The Netherland Kinship Panel Study (NKPS) is funded by grant 480-10-009 from Major Investments Fund of the Netherlands Organization for Scientific Research (NWO), and by the Netherlands Interdisciplinary Demographic Institute (NIDI), Utrecht University, the University of Amsterdam and Tilburg University.

Model 1

Variable	**Coefficient**	**p-value**
Age of male	0.0521	0.0046
Age difference (age of male - age of female)		
greater than 9	−2.9876	< 0.001
from 7 *to* 9	−1.4633	< 0.001
from 4 *to* 6	−0.4862	0.0108
from −3 *to* 3	0	
from −6 *to* −4	−1.4360	< 0.001
from −10 *to* −7	−2.8137	< 0.001
smaller than −10	−3.0582	< 0.001
Difference in educational level		
male is higher or equally educated	0.6424	< 0.001
Marriage history of female		
female was married before	−0.2811	0.1833
Number of potential pairs: 1078		

Model 2

Variable	**Coefficient**	**p-value**
Age of male	0.0550	0.0013
Age difference (age of male - age of female)		
greater than 10	−3.5428	< 0.001
from 4 *to* 10	−1.0867	0.0021
from −3 *to* 3	0	
from −4 *to* −10	−1.0105	0.0943
smaller than −10	−3.1277	0.0196
Difference in educational level		
male is higher or equally educated	0.7825	0.0148
Children with former partner		
female has children	1.6754	< 0.001
Number of potential pairs: 394		

Table 12.1.: Regression results of Model 1 (entering first cohabitation) and 2 (entering higher order cohabitation).

12.3. Results

12.3.1. Evidence from the compatibility measure

The coefficients of the estimated logit models are summarized in Table 12.1 and 12.2. In all models we find a positive effect if the male is higher/equally educated than/as the female. We could not generally detect any significant effect of the marriage history on the mating propensity of the male. The results indicate, however, that when they marry or cohabitate for the first time men are more prone to mate women who were not married before. Women and men who mate are more likely to be of the same age. The effect is stronger in the case of first order marriages and cohabitations. In all four models the direct effect of the age of the male is very small. Modelling compatibility by partnership order might be a reason for this phenomenon. First partnerships are usually started at younger ages, higher order partnerships follow later in life. In the analysis, therefore, a man's age at partnership onset is already indirectly described by the partnership type. We find that men have a greater probability to undergo a higher order cohabitation with a female who already has children with former partners. Surprisingly, for the other partnership types we cannot detect any significant effects of the presence of children with former partners on a male's mating probability.

12.3.2. Re-estimating empirical transition rates

To assess the quality of the proposed mate-matching strategy, validating the simulation output is good and useful practice. Besides basic validation of the simulation output, important hints for model improvement can be gained from careful analysis of the results. We made several simplifying assumptions during the mate-matching process, e.g., shifting event times, and these may have an undue impact. A very basic validation step is the re-estimation of the empirical transition rates which were used as input, by occurrence-exposure rates (Keiding, 1990). To smooth these rates, we employed a two-dimensional P-splines technique (Currie et al., 2006) that has been implemented in an R package named `MortalitySmooth` (Camarda, 2009). We compared the output of simulation runs with and without mate-matching. The re-estimation of rates shows that a simulation with mate-matching causes consistent output. Some results are plotted in the Figures 12.1 and 12.2. Both figures are level plots[2]. Empirical rates along with re-estimated rates are presented. Rates are given along calendar time and age. Their values are depicted on a "rainbow" color scale: red areas correspond to very low rates and violet-pink areas correspond to high rates.

Figure 12.1 shows transition rates of childless women with a lower secondary (medium) education who experience a transition from "being single after leaving parental home" to "first cohabitation". Compared to a setting without mate-matching, in the first years of simulation a slight postponement of event times is observable at higher ages.

---

[2] A level plot is a type of graph that displays on a colored surface three dimensional data points. Different colors map different valued data points. Commonly, abscissa and ordinate correspond to time or geographical dimensions. A level plot is an alternative to a contour plot.

Model 3

Variable	**Coefficient**	**p-value**
Age of male	0.0646	0.0650
Age difference (age of male - age of female)		
greater than 10	−3.3997	< 0.001
from 7 *to* 10	−1.4934	0.0110
from 3 *to* 6	−0.8026	0.0692
from −2 *to* 2	0	
from −5 *to* −3	−1.5026	0.0263
smaller than −5	−4.3357	< 0.001
Difference in educational level		
male is higher or equally educated	0.8493	0.0525
Marriage history of female		
female was married before	−0.4314	0.4873

Number of potential pairs: 198

Model 4

Variable	**Coefficient**	**p-value**
Age of male	−0.0120	0.6618
Age difference (age of male - age of female)		
greater than 8	−2.9174	< 0.001
from 4 *to* 8	−1.6287	0.0547
from −3 *to* 3	0	
smaller than −4	−3.2270	< 0.001
Difference in educational level		
male is higher or equally educated	1.2949	0.0743

Number of potential pairs: 82

Table 12.2.: Regression results of Model 3 (entering first marriage) and 4 (entering higher order marriage).

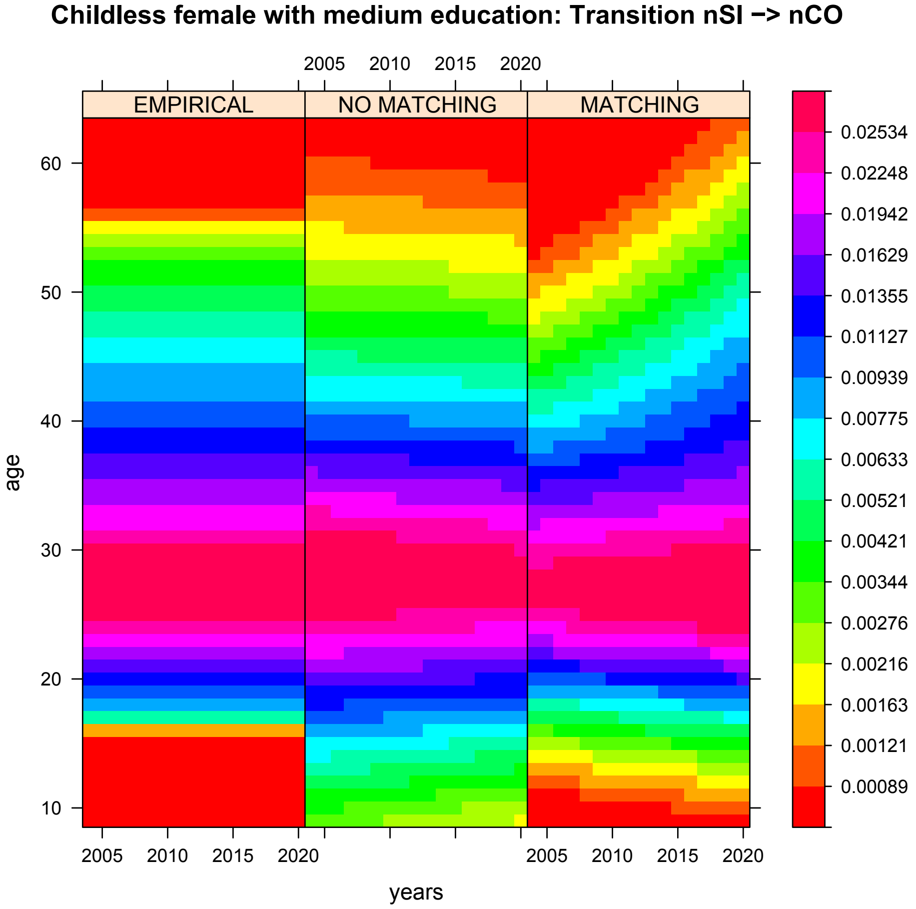


Figure 12.1.: Re-estimation of transition rates of childless females with a lower secondary (medium) education who experience a transition from “being single after leaving parental home (nSI)” to “first cohabitation (nCO)”.

Figure 12.2 depicts transition rates of highly educated males who experience a transition from “being single after leaving the parental home” to “first marriage”. In this case, we also observe a slight postponement of partnership onsets in the first simulation period. However, mate-matching does not significantly affect the occurrence and timing of simulated life-course events. Comparing the results of simulation runs with and without mate-matching, the re-estimation confirms the general suitability of the proposed mate-matching procedure, with respect to the stability of transition rates, despite the fact that some events were moved.

12.3.3. Analyzing the pool of unsuccessful seekers

To figure out whether the simulated partnership market resembles human mating outcomes, we analyze its composition and development over time. In particular, we study unsuccessful individuals. During the simulation, 40,585 partnership formations have been performed, 19,050 due to marriage and 21,535 due to cohabitation. Thus, 81,170 individuals have successfully entered and left the partnership market. Because the algorithm could not find a

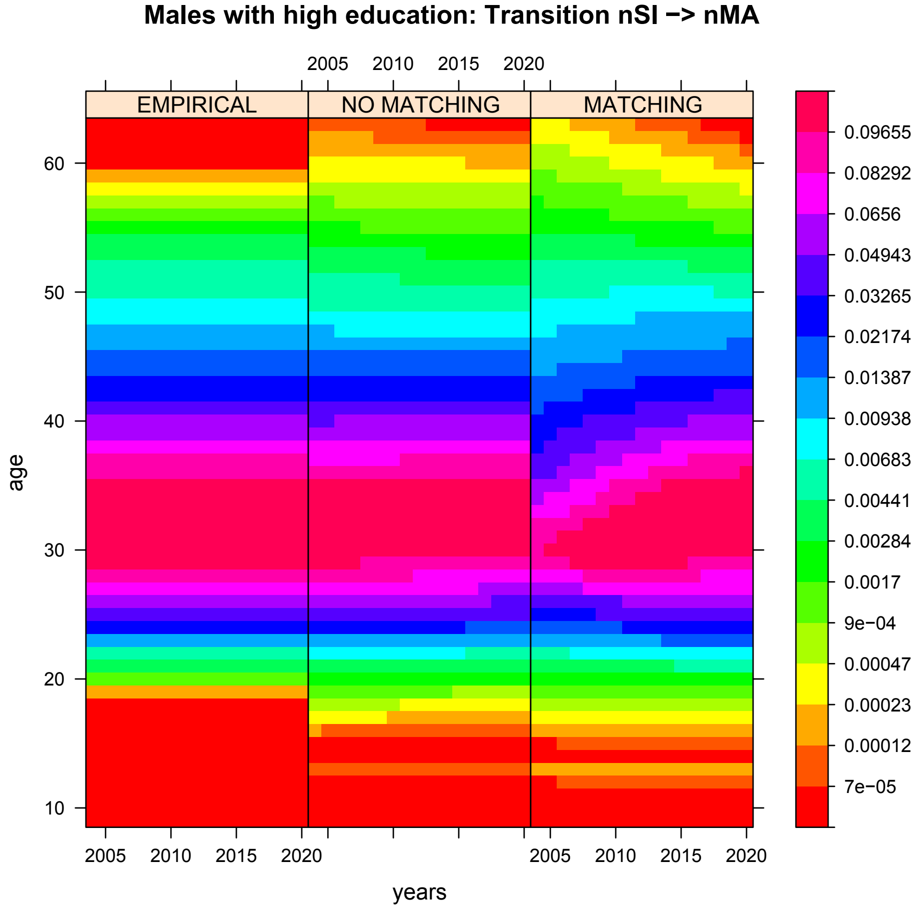


Figure 12.2.: Re-estimation of transition rates of highly educated males who experience a transition from "being single after leaving the parental home (nSI)" to "married (nMA)."

proper spouse in time, the searching period of 9, 779 seekers had to be extended. Over 17 years of simulation, 5, 397 individuals had entered the market without finding a spouse. Figure 12.3 displays the age distribution of unsuccessful seekers according to the year when they had entered the partnership market. We find that individuals who initialize a partner search at older ages remain more often unpaired than their younger counterparts. This phenomenon occurs because during simulation the synthetic mating pool for individuals at older ages is not as rich as it is for younger persons. Consequently, for older individuals the chance to find an appropriate spouse is relatively small. The figure also shows that the age distribution of unsuccessful seekers does not significantly change over the simulation horizon. We contrast these findings with the number of unsuccessful seekers along the year, see Figure 12.4. A decline along calendar time is obvious. In summary, we find that along calendar time the number of unsuccessful seekers decreases, but the age distribution of unsuccessful seekers remains stable. This phenomenon is caused by the age structure of the model population. Generally, at older ages only a small proportion of individuals enters the partnership market.

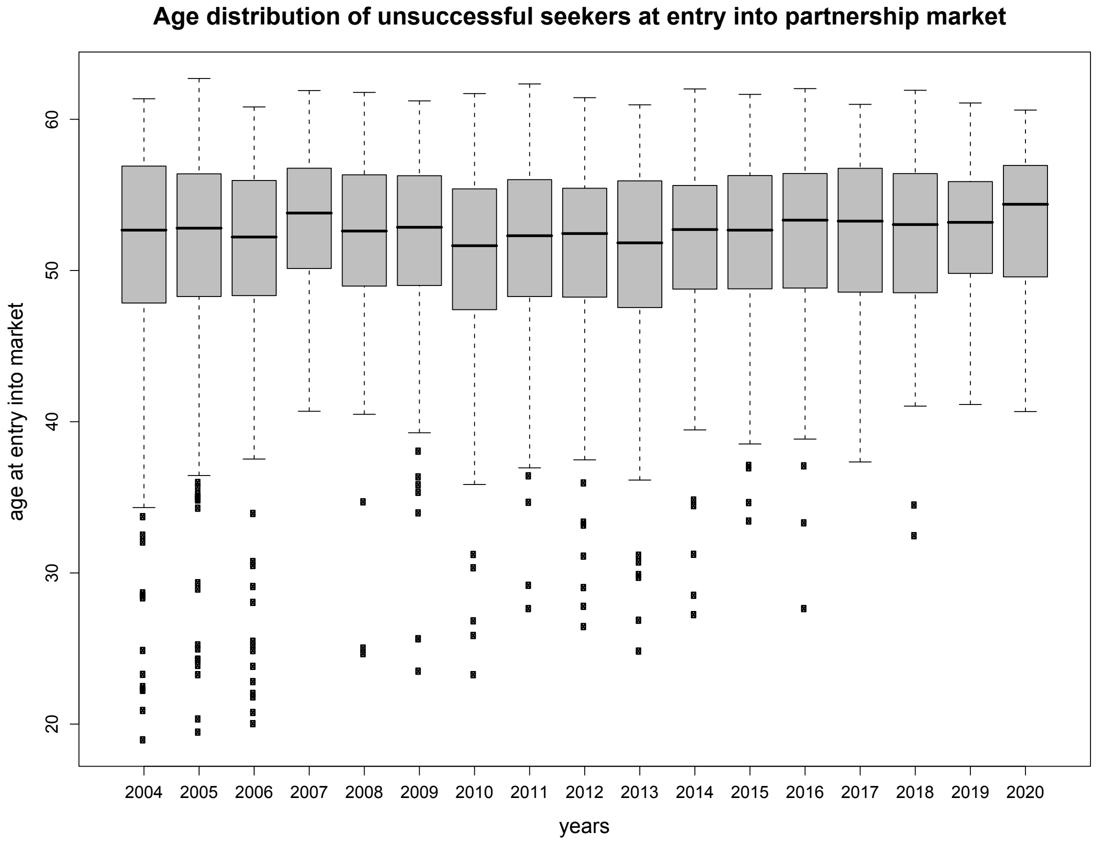


Figure 12.3.: Age distribution of unsuccessful seekers at the time when they enter the partnership market.

As the model population ages during simulation, over time the number of mature adults who enter the market, however, goes up. Therefore, for these adults, the chance to find an assortative mate increases, and in the synthetic pool the number of aged individuals who remain unpaired shrinks. Compared to younger seekers, the mating pool of older adults is still reduced; e.g., due to mortality or a high proportion of married persons at the same age.

Comparison of attributes of actual and simulated couples

It is essential for the usefulness of the proposed mate-matching strategy that it resembles actual characteristics of partners in couples. Therefore, as a further validation step, we analyze the distribution of joint characteristics, with a special emphasis on differences in educational attainment and age. We compare the differences in the educational level of synthetic couples to those of couples given in the NKPS (in the range from 1990 to 2002). Figure 12.5 contrasts simulated and actual data concerning the educational level within cohabiting (graph on left hand side) and married couples (graph on right hand side). The simulation produces on average 20 (80) percent cohabitations in which the female is higher than (equally/less educated as/than) the male. This is contrasted by 15 (85) percent of comparable cohabitations in

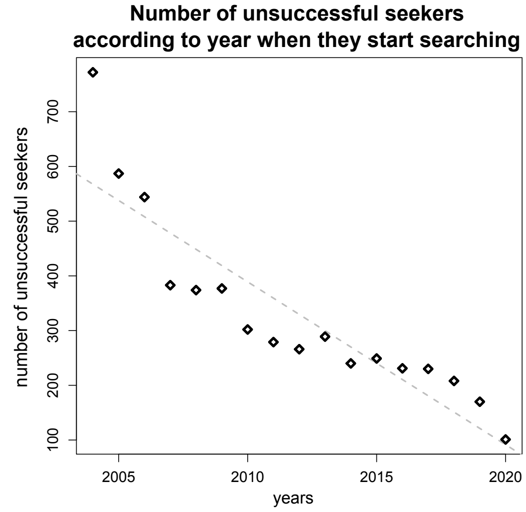


Figure 12.4.: Number of unsuccessful seekers according to the year when they initialize a partner search.

the data. In 17 (83) percent of the synthetic marriages, the female spouse is higher than (equally/less educated as/than) the male. By contrast, the NKPS comprises 14 (86) percent of such couples. Consequently, compared to the observed numbers, in the simulation too many partnerships are constructed in which the female holds a higher education than her spouse. This discrepancy might be caused by the fact that the individual aspiration level of females was decreased with waiting time: as a consequence, the longer a female has to wait in the partnership market, the higher is the probability of a match with a man with a level of education that is lower than in the data. In general, the simulation satisfactorily captures the overall pattern of differences in the educational level of spouses. Figure 12.6 depicts the

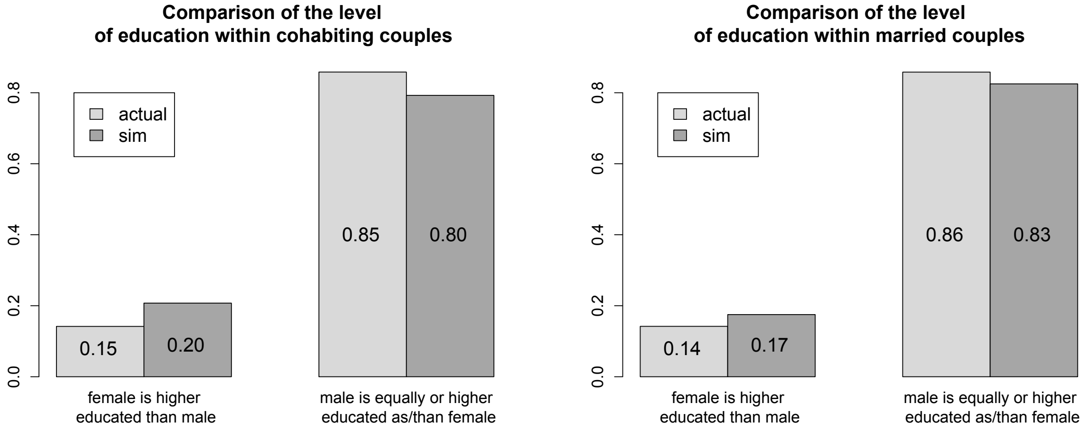


Figure 12.5.: Differences in the educational level of spouses in observed and simulated couples. Each bar shows the percentage of females in the corresponding category.

distribution of age differences of cohabitating and married spouses (age of male minus age of female). The shapes of the simulated and actual frequency distributions are very similar. Stochastic mate-matching algorithms that have so far been employed generally produce age difference distributions that are too flat (Leblanc et al., 2009). They are not capable of reproducing the observed peak at differences of [-1,1]. The proposed mate-matching algorithm is able to overcome this problem. It only produces slightly too many couples for which the male is much older than the female. A reason for this discrepancy could be the small number of corresponding cases in both the actual and the simulated data.

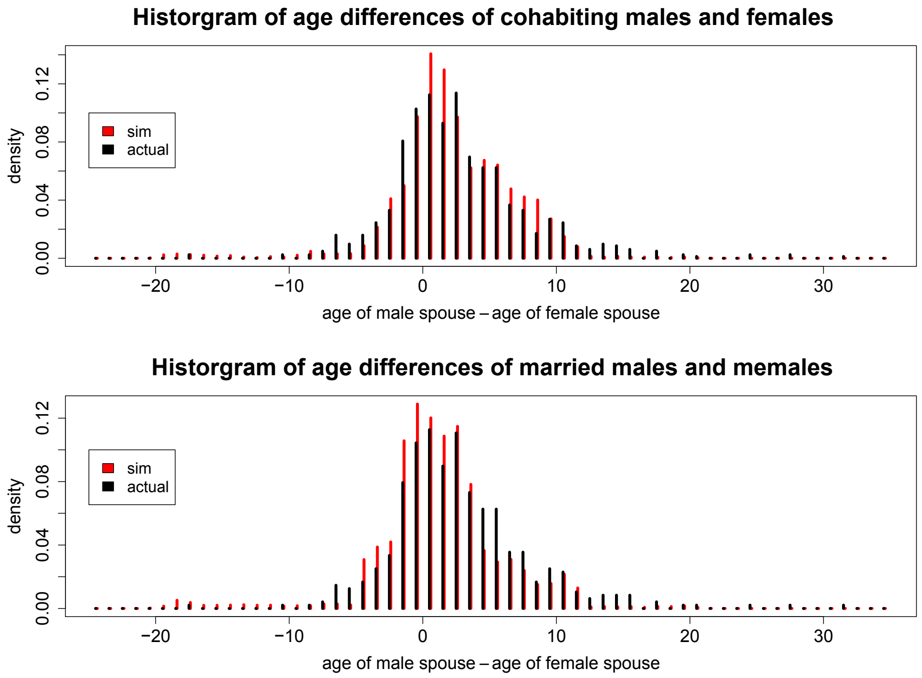


Figure 12.6.: Age differences of spouses in observed and simulated couples.

Handling situations of (relative) competition

A fundamental feature that demographers demand from a realistic mating model is the handling of (relative) competition (McFarland, 1972): an extra supply of single men of a certain age should decrease the number of paired men at all other ages (competition). The effect should be most pronounced at ages neighboring the age group with the surplus of single men (relative competition). At the same time, the number of partnered women should increase over all ages. To check whether the present mate-matching procedure can handle such a situation, in the initial population the number of single men aged 27 was doubled. A comparison of the partnerships built under the initial and the modified setting shows that the proposed mate-matching procedures copes well when strong competition is present. In the cohort of men who turned 28 in 2004, an increased number of partnered men could be observed, but the number did not double as compared to the initial setting. Because of the surplus of younger rivals, the number of partnered men older than 28 declined. The effect is strongest in the neighborhood of age 28 - which attests that the matching algorithm is able to handle situations of relative competition. For ages lower than 26, the number of partnered men is nearly identical to the initial setting. In this model the conditions, whether a female enters the partnership market or not, depends on her personal propensity and not on the availability of men. Therefore, even with a surplus of men in the mating pool, the number of partnered females remains stable.

13. Matching of Individuals: Discussion and Outlook

In contrast to conventional population projection models, dynamic microsimulations are suitable to directly handle interactions between individuals; e.g., married couples (van Imhoff and Post, 1998). Modelling relations between spouses, however, requires establishing a partnership (marriage or cohabitation), and for this purpose an effective mate-matching procedure is needed.

After an extensive literature review, we have proposed a two-sided stochastic mate-matching algorithm for continuous-time microsimulations. The algorithm tries to mimic human partner choice. Concepts were adopted from behaviorial models with implicit and explicit mating rules. In order to construct synthetic pairs, a stochastic marriage algorithm was developed, and embedded into a closed mating model. Preferences concerning attributes of potential spouses are mapped by a compatibility measure. This measure describes the similarity/dissimilarity between potential spouses based on certain important attributes, like age and educational level. Each simulated individual exhibits a certain level of aspiration regarding his/her desired compatibility with another person. If an individual experiences a partnership onset, he/she enters a partnership market. Here, the individual meets a limited number of potential partners at random. A couple is formed if the joint compatibility value of two individuals is at least as high as their aspiration levels. In this way, we encourage assortative mating. Individuals that were inspected, but rejected, accordingly lower their level of aspiration. A marriage queue was used to implement the partnership market as basis for the matching procedure.

In our microsimulation, we model and simulate individual life-courses as sequences of waiting times to next events. As a consequence, we know in advance when individuals will experience the onset of a marriage or a cohabitation. This circumstance is fundamental for the way we construct synthetic couples: we use the waiting time until a marriage/cohabitation onset for scanning potential partners. That is, alike in the Coale-McNeil model, the waiting time to a partnership event can be interpreted as being composed by three latent phases: the phase until the partner search starts, the phase of dating, and the phase between engagement and partnership onset.

We have incorporated our mate-matching algorithm into the microsimulation implementation described in Chapter 5. To demonstrate its capability we have used data on marriage behavior from the Netherlands. Population forecasts were conducted over 17 years, from 2004 to 2020. We found that the proposed algorithm produces acceptable results. It reproduces the observed peak of the frequency distribution of age differences of spouses.

Nevertheless, the presented mate-matching algorithm shows difficulties. Slight changes to event times are necessary to form partnership onsets. Another, more severe problem arises in the case of unsuccessful searches. If an individual is not able to find an appropriate partner during his/her searching period, we currently extend his/her searching period for half a year. A seeker might generally suffer from an exploited partner market, and thus, his/her searching period is extended over and over again. Such shifts of event time mean to ignore already generated events and is in conflict with the stochastic model that is used for simulating life-course events. To assess the extent of this conflict, we have re-estimated the input rates of two different simulation settings with identical stochastic models: one incorporated mate-matching and the other did not. The results of the latter prove that the implementation of the stochastic model is reliable. Hence, in the setting with mate-matching, any discrepancies between empirical and re-estimated rates are an implication of the mate-matching procedure (and not of an implementation error). In a corresponding comparison we find that empirical and re-estimated rates are not remarkably different. Consequently, we deem the proposed mate-matching algorithm suitable to be successfully incorporated into a continuous-time microsimulation.

The presented matching strategy shows several extension points. Preferences concerning the characteristics of a partner can be subject to secular trends, and such changes over time could be included in the mate-matching process. Currently, we only use compatibility measures that are based on actual preference patterns. Pairing individuals generally raises problems concerning the modelling and simulation of linked lives. An approach to specify the behavior of couples over time and a DEVS (Discrete Event Specification Language)-based model to establish and simulate couples in continuous-time are described in the next part of this thesis.

Part III.

A COMPLEX MODEL WITH INTERACTION BETWEEN TWO INDIVIDUALS

To realistically describe individual behavior, the effect of inter-individual interaction has to be considered. For example, in the majority of cases the decision to have children depends not only on the woman, but also on the potential father, and presumably additionally on the social network of both (Bernardi, 2003). Individual interaction patterns are usually ignored within demographic microsimulations. This is also true for the microsimulation that we have introduced within this thesis: up to now we have assumed that synthetic life-courses evolve independently of each other.

The only type of inter-individual linkage that has been so far considered in microsimulation models is demographic kinship (Smith, 1987; van Imhoff and Post, 1998; Wachter et al., 1998). When modelling kin at this level, however, mostly very simplifying and therefore distorting assumptions are made (Ruggles, 1993). For example, in nearly all cases characteristics of members of kin groups are stated to be uncorrelated, which is an assumption that is simply wrong. We know of only one case when a correlation between relatives has been included into a microsimulation model: Murphy (2006) describes the usage of an intergenerational transmission factor for capturing the correlation between a daughter's and her mother's fertility. However, apart from this study, correlation between demographic events of related individuals has so far been neglected. No general modelling or simulation technique exists that accounts for correlation between linked lives in a microsimulation. To give an example: many microsimulation models incorporate a marriage market to perform mate-matching. (So do we, see part II of this thesis.) However, after a couple has been matched, the subsequent modelling and simulation of the individuals forming the couple is not clear: in the literature, neither a stochastic model of the joint behavior of paired individuals is described, nor related simulation algorithms are discussed.

In the third part of this thesis we propose a way to extend the considered microsimulation model to tackle this shortcoming. We suggest a microsimulation model and related simulation techniques to account for interdependencies between the life-courses of spouses of (heterosexual) partnerships. That is, we restrict our consideration to binary linkages. The construction of interaction networks comprising more than two individuals is straightforward from a pure technical point of view: binary links between all interacting individuals make up a complete graph mapping an interaction network. Parameterizing a model describing the behavior of linked individuals can, however, become very difficult. This is mainly due to the fact that factors driving interactions and causal relationships are widely unobservable (Argyle, 2009; de Vos and Palloni, 1989). Partnership dynamics is a field that has been studied extensively. Various mechanisms driving the behavior of spouses have been explored and analyzed; see e.g., Lichter and Carmalt (2009), Boyle et al. (2008), Kalmijn et al. (2007), and Smith and Zick (1994). In summary, we find a considerable number of studies dealing with interdependencies within marital and cohabiting couples. These can be used as a sound basis for parameterizing a model of the behavior of spouses.

Within this thesis, we propose to model couple behavior by considering both spouses of a pair jointly. We detail the related processing in Section 14. In Section 15, we propose a

ML-DEVS model that formalizes the described model of individual and couple behavior. To efficiently execute the ML-DEVS model, we have extended the so-called sequential hierarchical simulator for PDEVS. The capability of the novel microsimulation is illustrated in Section 16 using a simplified example that aims at forecasting the contemporary Dutch population. Here we analyze partnership stability and the individual propensity to quit smoking, considering dual- and single-smoker couples. We conclude the thesis in Section 17 by summarizing the key lines of our work, by discussion the suitability of the developed simulation approaches, and by presenting problems that remain for future work.

14. Model Description

To derive a microsimulation model that captures, besides individual behavior, also interdependencies between life-courses of married and cohabiting individuals, we extend the microsimulation model described in Chapter 2 accordingly. In this chapter, we present a model of couple dynamics that can be used in combination with our model of individual behavior. The general idea is to specify couple dynamics by considering couples as units. That is, as soon as a female and a male mate, the 'couple unit' guides the life-course dynamics of the partnered individuals. If a couple experiences an emigration, dissolution, or widowhood event, the history of the couple ends; i.e., its 'life-course' terminates. Like individual life-courses, we describe couple behavior by using continuous-time multi-state models (cf. Section 2.3). That is, we specify the 'life-courses' of couples as trajectories of continuous-time Markov chains or semi-Markov processes. In doing so, we rely on a method to describe household dynamics suggested by Galler (1988) and Courgeau (1997). They model households as superior entities guiding the dynamics of their individual members.

Using continuous-time multi-state models to describe couple dynamics means that for all possible events a couple can experience transition rates are needed. Often, data sets that are substantive enough to estimate such rates might, however, not be available. An idea to obtain transition rates for couples anyhow, is to use transition rates of individuals and to combine them in a way that they reflect couple dynamics. We detail corresponding methods in Section 14.2.

14.1. Modelling and simulating couples as 'units'

14.1.1. State space and events

A multi-state model specifying life-courses of couples has a state space that comprises all states that both partners can occupy. These states indicate, for example, the partnership type (i.e., cohabitation or marriage), characteristics of the couple (e.g., the presence of children), and personal attributes of the spouses (e.g., smoking status). Table 14.1 depicts an illustrative example.

The state space for couples comprises states for all possible combinations of female and male attributes. Therefore, depending on the state variables considered and their possible values, the state space can become very large. The small example given in Table 14.1 already yields a pretty extensive state space, see Table 14.2. Multi-state models specify life-courses by sequences of state occupancies and state transitions/events. Table 14.3 shows possible events

State Variable	Possible Values
marital status	married; divorced; widowed
fertility status	no child; child(ren)
smoking status	non-smoker couple; dual smoker couple; she smokes, he is non-smoker; he smokes, she is non-smoker
mortality	both dead; she is dead, he is alive; he is dead, she is alive; both alive

Table 14.1.: Example of state variables that couples might occupy, inclusively possible values, separated by semicolons.

of couples associated to the state space given in Table 14.2. Subsequently, we denote the state space of a couple model by Ψ^P.

couple gets divorced
female starts smoking
female quit smoking
both quit smoking
female dies, he gets widowed
both die

childbirth
male starts smoking
male quit smoking
both start smoking
male dies, she gets widowed

Table 14.3.: Example of possible events that couples can experience.

14.1.2. Stochastic model of couple dynamics

To determine life-courses of couples, we have to provide transitions rates describing their propensity to experience events. For a realistic picture of demographic incidence, these rates should depend on the age of the female spouse and the age of the male spouse, and preferably also on calendar time. Some applications require to consider, besides age and calendar time, the time that a couple has already spent in its current state. For example, it is well known that the divorce risk strongly depends on the duration of a marriage (Becker et al., 1977; Diekmann and Schmidheiny, 2008).

Here we describe the life-course of a couple using a stochastic process $V(t)$, $t \in \mathbb{R}_0^+$, from the family of Markovian processes (Kijima, 1997). The process time t maps the time span over which we 'observe' a couple's life-course. The time t is set to zero when a couple is created, and evolves along the 'life time' of a couple. The process $V(t)$ is fully defined by the two-dimension process $(J_n, T_n)_{n\in\mathbb{N}_0}$ where $(J_n)_{n\in\mathbb{N}_0}$ is a Markov Chain that maps all states that a couple occupies, $J_n \in \Psi$, and $(T_n)_{n\in\mathbb{N}_0}$ is the sequence of the corresponding transition times along process time t, $T_n \in \mathbb{R}_0^+$, $T_0 = 0$. The random waiting time, for which a couple remains in state J_n, is determined by $W_n = T_{n+1} - T_n$. Generally, the process time can be mapped into age and calendar time: The function $A^f(T_n)$ maps the age of the female spouse at T_n, the function $A^m(T_n)$ maps the age of the male spouse at T_n, and the function $C(T_n)$ maps the calendar time at T_n. All three functions are translations. At the birth time of the female spouse, $A^f(\cdot)$ takes the value zero, at the birth time of the male spouse $A^m(\cdot)$ takes

(married, childless, non-smoker couple)
(married, childless, she smokes, he is non-smoker)
(married, childless, he smokes, she is non-smoker)
(married, childless, dual smoker couple)
(divorced, childless, non-smoker couple)
(divorced, childless, she smokes, he is non-smoker)
(divorced, childless, he smokes, she is non-smoker)
(divorced, childless, dual smoker couple)
(she is widowed, childless, she is non-smoker, he is dead)
(he is widowed, childless, he is non-smoker, she is dead)
(she is widowed, childless, she is smoker, he is dead)
(he is widowed, childless, he is smoker, she is dead)
(married, child(ren), non-smoker couple)
(married, child(ren), she smokes, he is non-smoker)
(married, child(ren), he smokes, she is non-smoker)
(married, child(ren), dual smoker couple)
(divorced, child(ren), non-smoker couple)
(divorced, childless, child(ren), he is non-smoker)
(divorced, child(ren), he smokes, she is non-smoker)
(divorced, child(ren), dual smoker couple)
(she is widowed, child(ren), she is non-smoker, he is dead)
(he is widowed, child(ren), he is non-smoker, she is dead)
(she is widowed, child(ren), she is smoker, he is dead)
(he is widowed, child(ren), he is smoker, she is dead)
(both dead)

Table 14.2.: Example state space of a couple model.

the value zero, and $C(\cdot)$ is zero at 01-01-1970 00:00:00.[1]

If we assume that the propensity of experiencing an event depends only on age and calendar time (and not on the time already spent in a state), $V(t)$ can be described as a non-homogeneous continuous-time Markov chain. The transition rates of a non-homogeneous continuous-time Markov chain describing couple behavior can be written as:

$$\lambda_{s_j,s_k}(t) = \lim_{h \downarrow 0} \frac{1}{h} P[J_{n+1} = s_k, T_{n+1} \in (t, t+h] \mid J_n = s_j, T_{n+1} \geq t], \qquad (14.1)$$

where $s_j, s_k \in \Psi^P$, and t, $t \in \mathbb{R}_0^+$, maps the process time. That is, the transition intensity $\lambda_{s_j,s_k}(t)$ describes the propensity of a couple to undergo a transition from state s_j to s_k at

---

[1] In computer science, 01-01-1970 00:00:00 commonly serves as a reference point from which time is measured.

process time t. Accounting for the age and calendar time dependence of the process, we yield

$$\begin{aligned}\lambda_{s_j,s_k}(c,a^f,a^m) = \lim_{h\downarrow 0}\frac{1}{h}P[J_{n+1} = s_k, C(T_{n+1}) \in (c,c+h],\\ A^f(T_{n+1}) \in (a^f,a^f+h], A^m(T_{n+1}) \in (a^m,a^m+h] \mid \\ J_n = s_j, A^f(T_{n+1}) \geq a^f, A^m(T_{n+1}) \geq a^m, C(T_{n+1}) \geq c].\end{aligned} \tag{14.2}$$

Some applications necessitate an additional duration dependency of the stochastic process used to describe couple behavior. In this case, instead of non-homogeneous continuous-time Markov chains, we use non-homogeneous semi-Markov processes (Monteiro et al., 2006) to describe couple behavior. The corresponding transition rates are defined as follows:

$$\lambda_{s_j,s_k}(w,t) = \lim_{h\downarrow 0}\frac{1}{h}P[J_{n+1} = s_k, W_n \in (w,w+h] \mid J_n = s_j, T_n = t, W_n \geq w],$$

i.e., $\lambda_{s_j,s_k}(w,t)$ describes the propensity of a couple to experience a transition to state s_k at process time $t+w$, after a waiting time of w in s_j. Here, unlike in equation (14.1), the parameter t does not indicate the process time, but the time at entry into a state. Accounting for the age and calendar dependencies of the process, we get:

$$\begin{aligned}\lambda_{s_j,s_k}(w,c,a^f,a^m) = \lim_{h\downarrow 0}\frac{1}{h}P[J_{n+1} = s_k, W_n \in (w,w+h] \mid J_n = s_j, C(T_n) = c, A^f(T_n) = a^f,\\ A^m(T_n) = a^m, W_n \geq w].\end{aligned} \tag{14.3}$$

Once the transition rates of a Markovian process are known, the distribution function of the waiting times in the distinct states of the state space can be derived. The distribution function of leaving state s_j for moving on to state s_k is:

$$F(w_{s_j,s_k},c,a^f,a^m) = 1 - \exp\{-\Lambda_{s_j,s_k}(w_{s_j,s_k},c,a^f,a^m)\}, \tag{14.4}$$

where $s_j, s_k \in \Psi^P, s_j \neq s_k$, a^f is the age of the female spouse at the last transition, a^m the age of the male spouse at the last transition, c the corresponding calendar time, and w_{s_j,s_k} the waiting time in s_j until moving on to s_k. The function $\Lambda_{s_j,s_k}(w_{s_j,s_k},c,a^f,a^m)$ describes the integrated hazard of the Markovian process. Assuming only age and calendar time dependency we yield:

$$\Lambda_{s_j,s_k}(w_{s_j,s_k},c,a^f,a^m) = \int_0^{w_{s_j,s_k}} \lambda_{s_j,s_k}\left(c+\nu, a^f+\nu, a^m+\nu\right)\, d\nu,$$

i.e., we obtain the integrated hazard of a non-homogeneous continuous-time Markov chain by integrating the transition rate (14.2) over calendar time, the age of the female, and the age of the male.

Taking a semi-Markov process to describe couple dynamics results in:

$$\Lambda_{s_i,s_j}(w_{s_i,s_j}, c, a^f, a^m) = \int_0^{w_{s_i,s_j}} \lambda_{s_i,s_j}\left(\nu, c, a^f, a^m\right)\, d\nu.$$

That is, we obtain the integrated hazard by computing the integral of the transition rate (14.3) over the waiting time in state s_i until moving to state s_j.

Alternative to the specification as a sequence of events and the waiting times between these events, a life-course can always be described as a sequence of waiting times to next events. That is, once we can derive the distribution functions of the waiting times, we can construct life-courses of couples by generating sequences of random waiting times. The processing used to construct synthetic life-courses is equivalent to the one elaborated in Section 2.3.1 and Section 2.3.2:

As already noted, the transition rates (the intensity function) of a Markovian process are its key quantities. Once they are known, one can compute all kind of process implications – inclusively the distribution functions of the waiting times. If longitudinal data for couples is available (stratified by the age of both spouses, and optionally also by duration), we can theoretically estimate transition rates for couples using the same methods we use to estimate transition rates for individuals (cf. Section 3.2) - although some more statistical modelling may be needed because many more transitions have to be considered.

The current data situation might, however, hamper the estimation of transition rates for all types of events to which couples might be exposed (Huinink and Feldhaus, 2009; Wolf, 2001). An idea to describe couple behavior anyhow is to combine models of individual behavior. For this purpose we have to set assumptions of how individual behavior has to be interlinked to yield couple behavior. This means, we have to model how (and for which transitions) the transition rates of the couple need to be modified relative to the rates of the individual spouses. An example: We may have age-specific rates to quit smoking for men and women. If in a couple one partner quits smoking, this rate for the other partner suddenly will be much higher than the rate for the respective sex by age.

In principle, interlinking individual behavior this way is a task in statistical modelling and many general approaches are available. We will have to base the analysis on mostly external knowledge of the phenomenon studied and the data available to model it. Approaches that deal in a general way with such problems are shared frailty concepts and copula models. In the following they will be summarized briefly.

Within this thesis, we have presented two process classes that can be used to specify individual behavior and couple dynamics: non-homogeneous continuous-time Markov chains and non-homogenous semi-Markov processes (see also Section 2.3). For simplicity reasons, in the next section we concentrate on non-homogeneous continuous-time Markov chains.

14.2. Shared frailty models and copula models

Both shared frailty models and copula models define correlation between two individuals by means of the joint sojourn time function. The joint sojourn time function of a couple indicates the probability that a female and a male spouse have a waiting time longer than w time units in their current states. If we assume that the female's propensity to experience a transition from s_i to s_j does not depend on the male's propensity to experience a transition from v_i to v_j and vice versa, the joint sojourn time function of a couple is:

$$\begin{aligned} S^d_{[s_i,v_i],[s_j,v_j]}(t,w) &= S^f_{s_i,s_j}(t,w) S^m_{v_i,v_j}(t,w) \\ &= \exp\{-(\Lambda^f_{s_i,s_j}(t,w) + \Lambda^m_{v_i,v_j}(t,w))\}, \end{aligned}$$

where $(s_i, s_j) \in E_f, (v_i, v_j) \in E_m, s_i, s_j, v_i, v_j \in \Psi$, and

$$\begin{aligned} S^f_{s_i,s_j}(t,w) &= \prod_{s_j} \exp\{-\int_0^w \lambda^f_{s_i,s_j}(t+u)\ du\}, \\ S^m_{v_i,v_j}(t,w) &= \prod_{v_j} \exp\{-\int_0^w \lambda^m_{v_i,v_j}(t+u)\ du\} \end{aligned}$$

are the marginal probability distributions of the female and the male spouse having a waiting time longer than w time units in their current states. Here, $\lambda^f_{s_i,s_j}(t)$ and $\lambda^m_{v_i,v_j}(t)$ are the (marginal) transition intensities of the individual spouses.[2] They indicate the individual propensity to change the current state, independently of the behavior of the spouse.

Otherwise, we can use parametric models to express dependencies between the behavior of the female and the male spouse. For example, if we assume that spouses are exposed to common, but unobservable, risk factors, then a shared frailty hazard model is suitable to describe $S^d_{[s_i,v_i],[s_j,v_j]}(t,w)$ (Hougaard, 2000, Chapter 7):

$$\begin{aligned} S^d_{[s_i,v_i],[s_j,v_j]}(t,w) &= \exp\{-Y(\Lambda^f_{s_i,s_j}(t,w) + \Lambda^m_{v_i,v_j}(t,w))\} \\ &= L(\Lambda^f_{s_i,s_j}(t,w) + \Lambda^m_{v_i,v_j}(t,w)), \end{aligned} \tag{14.5}$$

where $L(s) = \int_0^\infty e^{-st} f_{Y(t)}\ dt$ is the Laplace transformation, and $f_{Y(t)}$ the density function of the assumed frailty distribution. Copula models also suit well to the definition of bivariate common risk models (Glidden, 2007). Adapted to our purposes, we yield the following copula representation of $S^d_{[s_i,v_i],[s_j,v_j]}(t,w)$:

$$S^d_{[s_i,v_i],[s_j,v_j]}(t,w) = C_\gamma(S^f_{s_i,s_j}(t,w), S^m_{v_i,v_j}(t,w)), \tag{14.6}$$

where C_γ is a parametric family of paired distributions with uniform margins. Both model classes (shared frailty models and copula models) specify a parametric bivariate dependence structure. They are closely related. In fact, the shared frailty model 14.5 determines a copula

---

[2] These rates correspond to non-homogeneous continuous-time Markov chains.

function C_γ for a frailty model with density $f_{Y(t)}$. Originally, shared frailty models and copula models have been designed to describe bivariate survival times. An extension to our setting is, however, straightforward. More details about both model classes can be found in Hougaard (1995), Glidden (2007), and Wienke (2011, Chapter 4 & 6).

15. Microsimulation Model as Dynamic System

Naturally, a microsimulation works on the micro level of a population. Here, the attributes and dynamics of individual actors are described. Including partnership dynamics and global mate-matching rules into a microsimulation model increases its specificity, and thus suggests a more accurate description. To comply with this, we propose to extend the classical microsimulation design by adding a macro level comprising rules that guide mate-matching, partnership onset, and dissolution. Furthermore, the macro level is designed to handle changes in the population composition: it sees to integrate newborns and immigrants into the population, and to eliminate deaths and emigrants from the population.

In the first part of this thesis we describe a microsimulation model that neglects inter-individual interaction. For its specification we suggest the DEVS formalism, see Chapter 4.[1] In particular, we propose the usage of DYNPDEVS which supports variable structure models, i.e., models that change their own structure (Uhrmacher, 2001). Extending the developed DYNPDEVS network (cf. Section 4.2) to allow for partnership creation and dynamics is possible, but certainly not convenient. Global rules, like mating rules, could be handled by a 'workaround' solution, namely, by adding to the existing DYNPDEVS network a sub-model that holds them. This sub-model then mimics a macro model. However, such processing poses several problems (Uhrmacher et al., 2007), e.g., to realize up- and downward communication the ports of the 'macro model' have to be coupled to the ports of all micro models, which significantly burdens modelling and due to event broadcasting model execution.

Within this chapter, we propose to use the ML-DEVS formalism (Uhrmacher et al., 2007) to specify the extended microsimulation model that regards partnership dynamics. This formalism is a variant of the classical DEVS model language, extending it by explicitly supporting multi-level modelling. We structure this chapter as follows: first, we describe a ML-DEVS model for a demographic microsimulation that allows to consider partnership dynamics. Then, we describe the execution semantics that have originally been suggested for the execution of ML-DEVS models. With respect to our application, however, we find that these semantics hamper meaningful population projections. To tackle this problem, we develop a novel ML-DEVS simulator. This simulator is an extension of the so called sequential hierarchical simulator for PDEVS, and allows to efficiently run the extended microsimulation model.

---

[1] DEVS is a modelling language that directly allows to formulate discrete event systems.

15.1. A ml-DEVS population model

In ML-DEVS, a macro level model is described by a coupled DEVS model, equipped with a state and behavioral rules by their own, and micro models are described by ordinary atomic DEVS models.

Communication between micro and macro models is handled by exchanging messages. The propagation of information from the micro level to the macro level is facilitated by equipping micro models with the ability to change their ports. By this means, the macro model can access the information given in the exhibited ports of the micro models. That way the micro models can influence macro level dynamics. The macro model can concurrently activate (several) micro models by signalizing messages via value coupling. Value coupling means that at the macro level information is mapped to specific port names, and every micro model can access the information by forming input ports with corresponding port names.

Similar to DYNDEVS, the ML-DEVS formalism supports variable structure models. However, in contrast to DYNDEVS where structural changes result from models successively generating themselves, ML-DEVS operates structural changes top-down, directed by the macro model and executed on the micro level.

We describe the extended microsimulation model using a ML-DEVS model consisting of a macro model that encloses two types of micro components: individuals and pairs. (See Figure 15.1.)

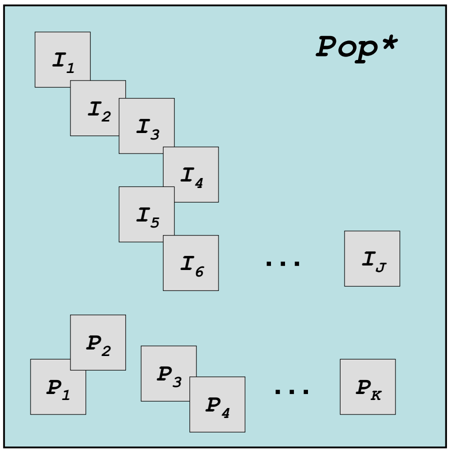


Figure 15.1.: The ML-DEVS population network **Pop*** comprises a finite number J of individual models I_j and a finite number K of pair models P_k.

The special structure of the considered microsimulation model implies that not the entire functionality of the original ML-DEVS formalism is engrossed, e.g., we do not employ any couplings between micro models, and on the macro level the population does not undergo any internal events. Therefore, for reasons of clarity, we adapt the original ML-DEVS approach such that in the subsequent description we let out functionality that we do not demand.

15.1.1. The population: a macro model

We formulate the population macro model $\mathbf{Pop}^\star$ as structure:

$$\mathbf{Pop}^\star = \langle X, Y, S, s_{\text{init}}, \mathbf{I}, \mathbf{P}, \delta, \lambda_{\text{down}}, \lambda, sc\rangle$$

where

$$X = \begin{cases} \text{IM} = [{}^I\psi_1^0, \cdots, {}^I\psi_{im_1}^0, {}^P\psi_1^0, \cdots, {}^P\psi_{im_2}^0] & \text{if } im_1 \text{ single immigrants} \\ & \qquad \text{and/or } im_2 \text{ couples enter,} \\ \varnothing & \text{otherwise,} \end{cases}$$

${}^I\psi_i^0$ $(i = 1, \cdots, im_1, im_1 \in \mathbb{N})$ describes the state of an immigrant, and

${}^P\psi_j^0$ $(j = 1, \cdots, im_2, im_2 \in \mathbb{N})$ describes the state of an immigrating couple,

Y is an output port for single emigrants or pairs to leave the population,

S is the set of possible states of $\mathbf{Pop}^\star$, a state $s = [s_1, \cdots, s_5]$, $s \in S$, indicates

s_1: whether the transition of an individual model leads to a structural model change; indicators of structural model changes invoked by individual models are $actionOfInd = \{immigrating, emigrating, dying, childbirth\}$,

s_2: whether the transition of a pair model leads to a structural model change; indicators of structural model changes invoked by pair models are $actionOfPair = \{immigrating, emigrating, dying, childbirth, dissolution, widowhood\}$,

s_3: (if any) the two individuals that have last been found to form a proper match, and their mating time, otherwise $s_3 = \varnothing$,

s_4: (if any) all searching individuals that were concerned by the latest mate matching round and could not properly be matched, otherwise $s_4 = \varnothing$,

s_5: (if any) two individual models that are due to form a couple, otherwise $s_5 = \varnothing$,

that is

$$S = \underbrace{\Big(\big(actionsOfInd \times \mathcal{J}\big) \cup \varnothing\Big)}_{\text{struct. changes due to ind. model}} \times \underbrace{\Big(\big(actionsOfPair \times \mathcal{P}\big) \cup \varnothing\Big)}_{\text{struct. changes due to pair model}} \times \underbrace{\Big(\big(\mathcal{J} \times \mathcal{J} \times \mathbb{R}_0^+\big) \cup \varnothing\Big)}_{\text{last couple found}} \times$$
$$\underbrace{\Big(\bigtimes_{i \in \mathcal{J}} i \cup \varnothing\Big)}_{\text{reduce aspir. level}} \times \underbrace{\Big(\big(\mathcal{J} \times \mathcal{J}\big) \cup \varnothing\Big)}_{\text{couple due for mating}},$$

where $\mathcal{J}$ is the index set of all individual models $\mathbf{I}$ (see Section 15.1.2) and $\mathcal{P}$ is the index set of all pair models $\mathbf{P}$ (see Section 15.1.3),

$s_{\text{init}} = [\varnothing, \cdots, \varnothing]$ is the initial state of $\mathbf{Pop}^{\star}$,

$\delta : X \times S \times \mathcal{J} \times \mathcal{P} \rightarrow S$ is the state transition function of $\mathbf{Pop}^{\star}$; δ is composed by four component functions:

δ_1: if individuals are immigrating, or an individual signalizes an emigration, a death, or a childbirth event, δ_1 updates the first component s_1 of the state of $\mathbf{Pop}^{\star}$ accordingly, otherwise $s_1 = \varnothing$,

δ_2: if pairs are immigrating, or a pair signalizes an emigration, a death, a childbirth, a dissolution, or a widowhood event, δ_2 updates the first component s_2 of the state of $\mathbf{Pop}^{\star}$ accordingly, otherwise $s_2 = \varnothing$,

δ_3: if an individual model signalizes the onset of a partner search, then δ_3 executes a mate-matching algorithm

- if two individuals can be identified to form a proper match, then δ_3 reports these two individuals and their corresponding mating time in s_3,
- if no individuals can be identified to form a proper match, δ_3 sets s_3 to $\varnothing$;
- besides this, δ_3 reports in s_4 all searching individuals who during the mate matching process were inspected and could not properly be matched;

otherwise $s_3 = s_4 = \varnothing$,

δ_4: if two individuals signalize that their mating time is due, δ_4 reports this in s_5, otherwise $s_5 = \varnothing$.

$\lambda_{\text{down}} : S \rightarrow 2^{\bigcup_{i \in \mathcal{J}} X_i}$ is the downward output function:

- it informs individual models about upcoming mating times (reported by s_3), and
- instructs searching individuals, who were inspected during the latest mate matching round and for whom no proper match could be found (reported by s_4), to lower their aspiration level;

$\lambda : S \rightarrow Y$ is the output function; it forwards the states and the birth dates of emigrating pairs and individuals to Y.

$sc : S \rightarrow \mathbf{I} \times \mathbf{P}$ is the structural change function; sc is composed by seven component functions:

sc_1: *creates individual models*:

if s_1 indicates immigrating individuals, sc_1 creates im_1 new individual models for immigrants,

if s_1 or s_2 indicate childbirth, sc_1 creates for each newborn an individual model,

sc_2: *deletes an individual model*:

if s_1 indicates that an individual is dying or emigrating, sc_2 deletes the corresponding individual model,

sc_3: *creates pair models*:

if s_2 indicates immigrating pairs, sc_3 creates im_2 new pair models for immigrants,

sc_4: *deletes a pair model model*:

if s_2 indicates that a pair is emigrating and both partners of a pair die, sc_4 deletes the corresponding pair model,

sc_5: *creates a pair model and deletes two individual models*:

if s_5 indicates a mating event, sc_5 creates a new pair model for the mating individuals and deletes the corresponding individual models,

sc_6: *deletes a pair model*:

if s_2 indicates the dissolution of a pair: sc_6 deletes the concerned pair model and creates for the separating partners two individual models,

sc_7: *creates an individual model and deletes a pair model*:

if s_2 indicates a widowhood event: sc_7 deletes the corresponding pair model and creates for the surviving partner an individual model;

otherwise sc is not defined.

Concerning most aspects the specification of $\mathbf{Pop}^\star$ equals the one of the network model **Pop**, described in Section 4.2 on page 36:

- Immigrants will be generated by a model $OUTSIDE$, to which the population model is coupled. $OUTSIDE$ maps all those populations that are not described $\mathbf{Pop}^\star$. The $OUTSIDE$'s outputs serve as an input to our population model. Immigrants enter the model via X.
- Via $\mathbf{Pop}^\star$'s output port Y information about emigrants are sent to the $OUTSIDE$ model.
- Death, emigration, immigration, and childbirth events cause structural model changes.

The main difference between $\mathbf{Pop}^\star$ and **Pop** is the consideration of partnerships between women and men. Besides individual models, $\mathbf{Pop}^\star$ also comprises pair models for marital couples and non-marital couples. Like individual models, pair models feature an intrinsic behavior, see Section 15.1.3. The macro model $\mathbf{Pop}^\star$ guides the onset of partnerships, i.e., the creation of pair models. To this end, $\mathbf{Pop}^\star$ performs a mate matching procedure that involves all individuals who signalize their disposition to mate, i.e., exhibit their *searching* port. The procedure that $\mathbf{Pop}^\star$ applies for this purpose is described in Chapter 11. If two individuals are found to form a proper couple, $\mathbf{Pop}^\star$ records this in its state and informs the related individuals via λ_{down} about the upcoming mating time. Individual models receive this information on their $foundMate$ input port. Additionally, $\mathbf{Pop}^{\star}$ instructs individuals who were unsuccessfully inspected during a mate-matching round to lower their level of aspiration (via their input port *redAspLevel*). When two individual models forward the due date of their partnership onset (via their *mating* port), $\mathbf{Pop}^{\star}$ replaces them by a pair model. Accordingly, if a pair model informs $\mathbf{Pop}^{\star}$ about a dissolution event, $\mathbf{Pop}^{\star}$ replaces this pair model by two individual models describing the separated partners. Likewise, if a pair model signalizes a widowhood event, $\mathbf{Pop}^{\star}$ replaces this pair model by one individual model that describes the surviving partner. Apart from immigration, the Figures 15.2 and 15.3 illustrate the different types of structural model changes that $\mathbf{Pop}^{\star}$ carries out.

15.1.2. Individuals: micro components

$\mathbf{I}$ is the set of all individuals models I. We formulate I as structure

$$\langle X, Y, {}^{I}\Psi, \psi_0, p, \delta, \lambda, ta \rangle$$

where

X is the input port of I; it is $X = \{foundMate, redAspLevel\}$;

1. $\mathbf{Pop}^{\star}$ informs I via $foundMate$ about an upcoming partnership onset, and
2. $\mathbf{Pop}^{\star}$ instructs I via *redAspLevel* to lower the aspiration level.

Y is the set of output ports of I; $Y = \{searching, childbirth, emigration, death, mating\}$; we differ between two types of output ports:

1. the port *searching* that is permanently exhibited when I is searching for a mate, and
2. the output ports that indicate structural model changes:
   (i) if a childbirth event is due, I signalizes this on the port *childbirth*,
   (ii) if a death event is due, I signalizes this event on the port *death*,
   (iii) if an emigration event is due, I shows this on the port *emigration*, and
   (iv) if a mating event is due, I signalizes this on the port *mating*.

${}^{I}\Psi$ is the set of possible states that I can occupy; ${}^{I}\Psi$ comprises

(i) I's current state $s_c \in \Psi$ (Ψ is the state space of the stochastic process that maps the individual life-course),

(ii) I's upcoming state $s_u \in \Psi$,

(iii) I's birth date b, $b \in \mathbb{R}_0^+$,

(iv) I's age a, $a \in \mathbb{R}_0^+$,

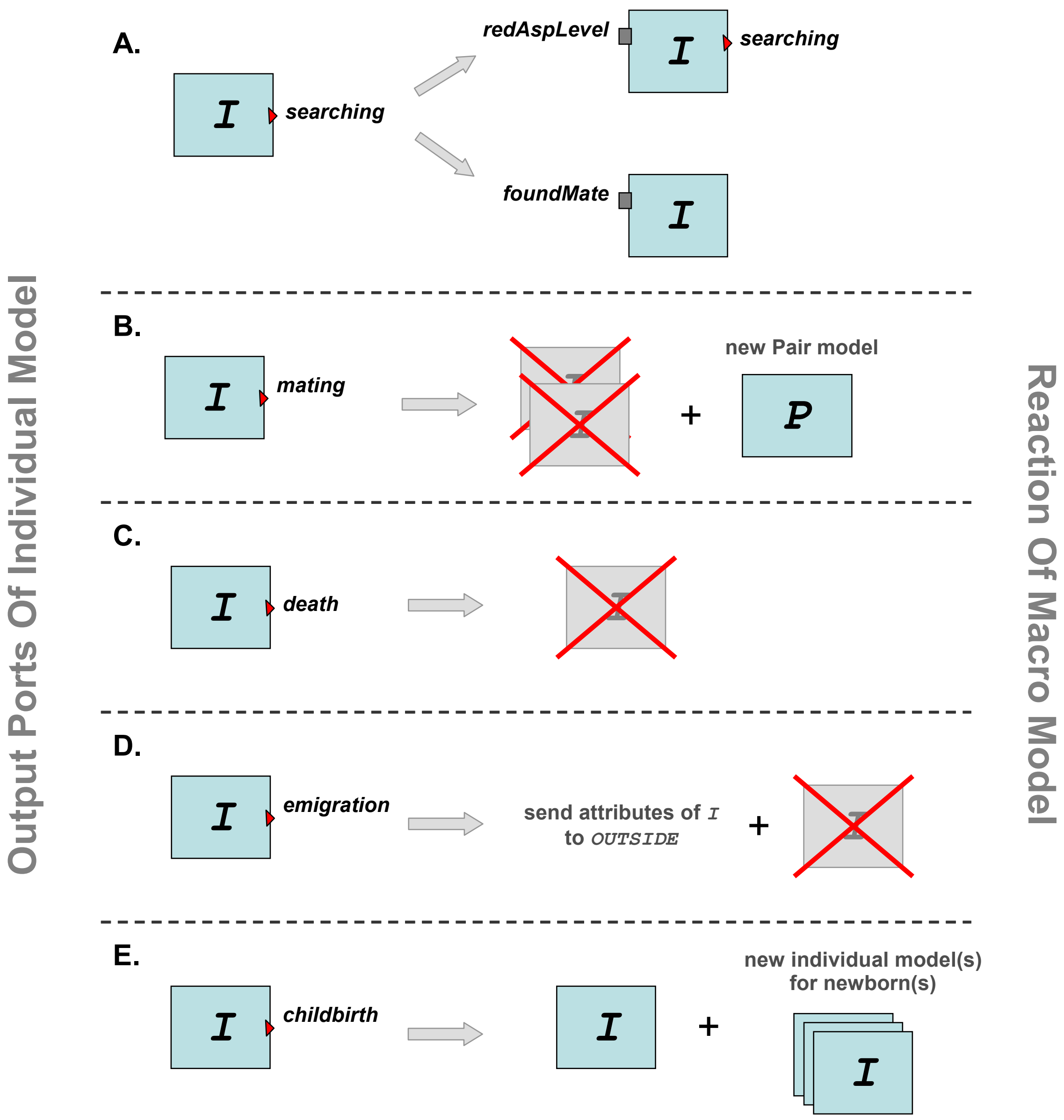


Figure 15.2.: Possible output messages of the individual models and reactions of the population model.

(v) the time w that I has already spent in s_c, $w \in \mathbb{R}_0^+$,

(vi) the complete waiting time τ that I has to spend in s_c, $\tau \in \mathbb{R}_0^+$, as well as

(vii) I's aspiration level la concerning a potential spouse, $la \in \mathbb{R}_0^+ \cup \varnothing$, $la = \varnothing$ indicates that I is not searching for a mate,

that is

$$
{}^{I}\Psi = \underbrace{\Psi}_{\text{current state}} \times \underbrace{\Psi}_{\text{upcoming state}} \times \underbrace{\mathbb{R}_0^+}_{\text{birth date}} \times \underbrace{\mathbb{R}_0^+}_{\text{age}} \times \underbrace{\mathbb{R}_0^+}_{\text{wait. time already spent}} \times \underbrace{\mathbb{R}_0^+}_{\text{compl. wait. time}} \times \underbrace{\left(\mathbb{R}_0^+ \cup \varnothing\right)}_{\text{aspiration level}},
$$

$\psi_0 = [s_0, s_0, b, a_0, w_0, \infty, \varnothing]$, comprises I's initial state s_0, I's birth date b, I's age a_0 at model initialization, the waiting time w_0 that I has already spent in s_0, the entire waiting time that I has to spent in his/her current state (in the initialization phase we set this time to ∞), and the initial level of aspiration concerning a potential spouse is $\varnothing$.

To ease the subsequent description of the ML-DEVS functions δ, p, λ and ta, we define the following auxiliary functions:

$$
\begin{aligned}
\mathit{search} &: \Psi \times \Psi \to \{\mathit{true}, \mathit{false}\},\\
\mathit{death} &: \Psi \to \{\mathit{true}, \mathit{false}\},\\
\mathit{emigration} &: \Psi \to \{\mathit{true}, \mathit{false}\},\\
\mathit{childbirth} &: \Psi \times \Psi \to \mathbb{N}_0.
\end{aligned}
$$

The function *search* shows if an event implies the onset of a partnership, the functions *death* and *emigration* indicate a death and an emigration event, and the function *childbirth* gives the number of newborns that a state transition implicates.

$\delta : X \times {}^{I}\Psi \times \mathbb{R}_0^+ \to {}^{I}\Psi$ is the state transition function of I (x indicates input messages, $\psi \in {}^{I}\Psi$, and e describes the time elapsed):

$$
\delta(x, \psi, e) = \begin{cases}
[s_u, s_{j^\star}, b, a + \tau, 0, \tau^\star, \varnothing] & \text{if } \mathit{search}(s_u, s_{j^\star}) = \mathit{false}\\
[s_u, s_{j^\star}, b, a, 0, \infty, la_0] & \text{if } \mathit{search}(s_u, s_{j^\star}) = \mathit{true}\\
[s_c, s_u, b, a, 0, \infty, \min(0, la - \delta_A)] & \text{if an input message has been received via}\\
 & \qquad \text{the port } \mathit{redAspLevel}\\
[s_c, s_u, b, a, 0, t^{on} - b - a, \varnothing] & \text{if the actual mating time } t^{on} \text{ has been}\\
 & \qquad \text{received via the port } \mathit{foundMate}\\
\text{not defined} & \text{otherwise,}
\end{cases}
$$

where $s_{j^\star} = \text{argmin}_{\,s_j \in \Psi/\{s_u\}}(w_{s_u,s_j})$, w_{s_c,s_j} is the (random) waiting time between the states s_c and s_j (see the stochastic model of individual behavior described in Section 2.3), $\tau^\star = \min\{w_{s_c,s_j}\}$, la_0 is the aspiration level in the moment when an individual enters the mate searching phase, and δ_A is the decrement to lower the aspiration level in case of an unsuccessful search round, $\delta_A \in \mathbb{R}^+$.

$p : {}^{I}\Psi \to 2^{\mathcal{P}_I}$ ($\mathcal{P}_I = \{\mathit{searching}, \mathit{mating}, \mathit{death}, \mathit{emigration}, \mathit{childbirth}\}$) selects the ports available in a given state of the individual model, i.e.,

$$p(\psi) = \begin{cases} \mathit{searching} & \text{if } \mathit{search}(s_c, s_u) = \mathit{true} \text{ and } la \neq \varnothing, \\ \mathit{mating} & \text{if } \mathit{search}(s_c, s_u) = \mathit{true} \text{ and } la = \varnothing, \\ \mathit{death} & \text{if } \mathit{death}(s_u) = \mathit{true}, \\ \mathit{emigration} & \text{if } \mathit{emigration}(s_u) = \mathit{true}, \\ \mathit{childbirth} & \text{if } \mathit{childbirth}(s_c, s_u) > 0, \\ \varnothing & \text{otherwise,} \end{cases}$$

$\lambda : {}^{I}\Psi \to Y$ is the output function; it is composed by five component functions that fill the outport ports $\mathit{searching}, \mathit{mating}, \mathit{childbirth}, \mathit{death}$ and *emigration*:

λ_1: If $\mathit{search}(s_c, s_u) = \mathit{true}$ and $a \neq \varnothing$, λ_1 forwards the information $(s_u^m, b, t^{\text{des}} = b + a + \tau^\star, la)]$ necessary for mate searching, where $s_u^m \subseteq s_u$ comprises the individual attributes that are relevant for finding a proper spouse (like age and educational attainment), and t^{des} is the desired mating (calendar) time; otherwise $\lambda_1 = \varnothing$.

λ_2: If I is due to mate (i.e., $\mathit{search}(s_c, s_u) = \mathit{true}$ and $la = \varnothing$), λ_2 forwards via the output port *mating* the upcoming state and the birth date of I, otherwise $\lambda_2 = \varnothing$.

λ_3: If I is due to give birth (i.e., $\mathit{childbirth}(s_c, s_u) > 0$), λ_3 forwards via the output port *childbirth* the number of newborns, otherwise $\lambda_3 = \varnothing$.

λ_4: If I is due to die (i.e., $\mathit{death}(s_u) = \mathit{true}$), λ_4 signalizes this to the output port *death*, otherwise $\lambda_4 = \varnothing$.

λ_5: If I is due to emigrate (i.e., $\mathit{emigration}(s_u) = \mathit{true}$), λ_5 forwards via the output port *emigration* the current state and the birth date of I, otherwise $\lambda_5 = \varnothing$.

$ta : {}^{I}\Psi \to \mathbb{R}_0^+ \cup \{\infty\}$ is the time advanced function: $ta(\psi) = \tau$.

The main difference between the definition of the micro model I and the definition of the individual model of the DYNPDEVS network **Pop** (cf. Section 4.2 on page 36) is the consideration of a mate searching phase and the external determination of a mating time. Apart from this, the design of I is very similar to the one of the individual model of **Pop**. In particular, the computation of events and waiting times to events relies on the same method; i.e., the stochastic model of individual behavior described in Section 2.3.

Immediately after it is clear that the next event of I implies the onset of a partnership, he/she enters a phase of mate searching. In this phase, I features a positive aspiration level la concerning the traits of a potential spouse. The higher this level, the more demanding is I. The initial level of aspiration can be chosen as being beta distributed. (In Section 11.3 on page 105 we detail the parameterization of the mate-matching procedure.) The parameters of the distribution might depend on the state of I, e.g., on his/her age and sex. Besides the determination of an aspiration level, I indicates his/her mating willingness by permanently exhibiting his/her *searching* output port. This port holds I's birth date, the desired mating time, and the attributes of I that are relevant for the mate matching procedure, e.g., I's educational level. Such modelling allows to cache private information and only discloses data that is relevant for the external mate matching operation.

The permanent exhibition of the *searching* port ensures that every time $\mathbf{Pop}^{\star}$ is conducting a mate matching round, it can retrieve that data that is relevant for mating. Meanwhile, every time $\mathbf{Pop}^{\star}$ considers (the searching) I in a mate matching round and no proper spouse could be detected for I, $\mathbf{Pop}^{\star}$ instructs I (via its input port *redAspLevel*) to lower his/her aspiration level. During each mate matching round, $\mathbf{Pop}^{\star}$ checks whether I's desired mating time t^{des} is expired.[2] If this is the case, the options listed in Section 11.4 on page 108 offer a way out. At this place, we opt for shifting the desired mating time, and to pair I as soon as possible. The latter might be accomplished by enforcing I to be part in each upcoming mate matching round, until he/she is partnered. If then a proper spouse for I is found, $\mathbf{Pop}^{\star}$ informs her/him about the upcoming mating event. In line with this, I receives an accordant message on its input port $foundMate$.

Besides the *searching* port, I possesses the four output ports: $death, emigration, childbirth$, and *mating*. The port *death* signalizes that I makes a transition to 'death'. Likewise, *emigration* shows an emigration event, and *childbirth* indicates the number of children that a transition implicates. On the *mating* port, I signalizes that he/she enters a partnership. Figure 15.2 displays the output ports of I. It further illustrates the operations that $\mathbf{Pop}^{\star}$ conducts in response.

Modelling messages to the macro model via ports allows us to encapsulate the internal state and the internal functionality of the individual model. The individual model only reveals information which the macro model can handle. This way we ensure that between the micro models and the macro model only required information is processed.

15.1.3. Pairs: micro components

$\mathbf{P}$ is the set of all pair models P. We formulate P as structure

$$\langle Y, {}^{P}\Psi, \psi_0, \delta, \lambda, ta \rangle$$

where

Y is the output port of P; $Y = \{dissolution, widowhood, death, emigration, childbirth\}$; it forwards structural changes like dissolution (port: *dissolution*), widowhood (port: *widowhood*), death of both partners (port: *death*), emigration (port: *emigration*), or childbirth (port:

---

[2] Within this thesis we propose a mate matching procedure that uses so called 'searching periods' during which individuals can date; see Section 11.1 on page 103 for more details.

childbirth),

${}^{P}\Psi$ is the set of possible states that P can occupy; ${}^{P}\Psi$ comprises

(i) P's current state $s_c \in \Psi$ (Ψ is the state space of the stochastic process that maps the dynamics of the couple),

(ii) P's upcoming state $s_u \in \Psi$,

(iii) the birth date b_m, of the male spouse, $b_m \in \mathbb{R}_0^+$,

(iv) the birth date b_f, of the female spouse, $b_m \in \mathbb{R}_0^+$,

(v) the age a_m of the male spouse, $a_m \in \mathbb{R}_0^+$,

(vi) the age a_f of the female spouse, $a_f \in \mathbb{R}_0^+$, as well as

(v) the time w that P has already spent in s_c, $w \in \mathbb{R}_0^+$.

That is

$$
{}^{P}\Psi = \underbrace{\Psi}_{\text{current state of pair}} \times \underbrace{\Psi}_{\text{upcoming state of pair}} \times \underbrace{\mathbb{R}_0^+}_{\text{birth date of male}} \times \underbrace{\mathbb{R}_0^+}_{\text{birth date of female}} \times \\
\underbrace{\mathbb{R}_0^+}_{\text{age of male}} \times \underbrace{\mathbb{R}_0^+}_{\text{age of female}} \times \underbrace{\mathbb{R}_0^+}_{\text{waiting time already spent}},
$$

$\psi_0 = [s_0, s_0, b_m, b_f, a_m, a_f, w_0]$, comprises P's initial state s_0, the birth date b_m of the male spouse, the birth date b_f of the female spouse, the age a_m of the male spouse at model initialization, and the age a_f of the female spouse at model initialization, as well as P's waiting time w_0 in s_0.

To ease the subsequent description of the ML-DEVS functions δ, p, λ and ta, we define the following auxiliary functions:

$$
\begin{aligned}
dissolution &: \Psi \times \Psi \to \{true, false\}, \\
widowhood &: \Psi \times \Psi \to \{true, false\}, \\
deathOfBoth &: \Psi \to \{true, false\}, \\
emigration &: \Psi \to \{true, false\}, \\
childbirth &: \Psi \times \Psi \to \mathbb{N}_0.
\end{aligned}
$$

The function *dissolution* indicates whether a state transition implies the dissolution of the partnership, the function *widowhood* indicates the death either of the male or the female spouse, and $deathOfBoth$ shows the death of both partners; *emigration* indicates an emigration event, and the function *childbirth* gives the number of newborns that a state transition implicates.

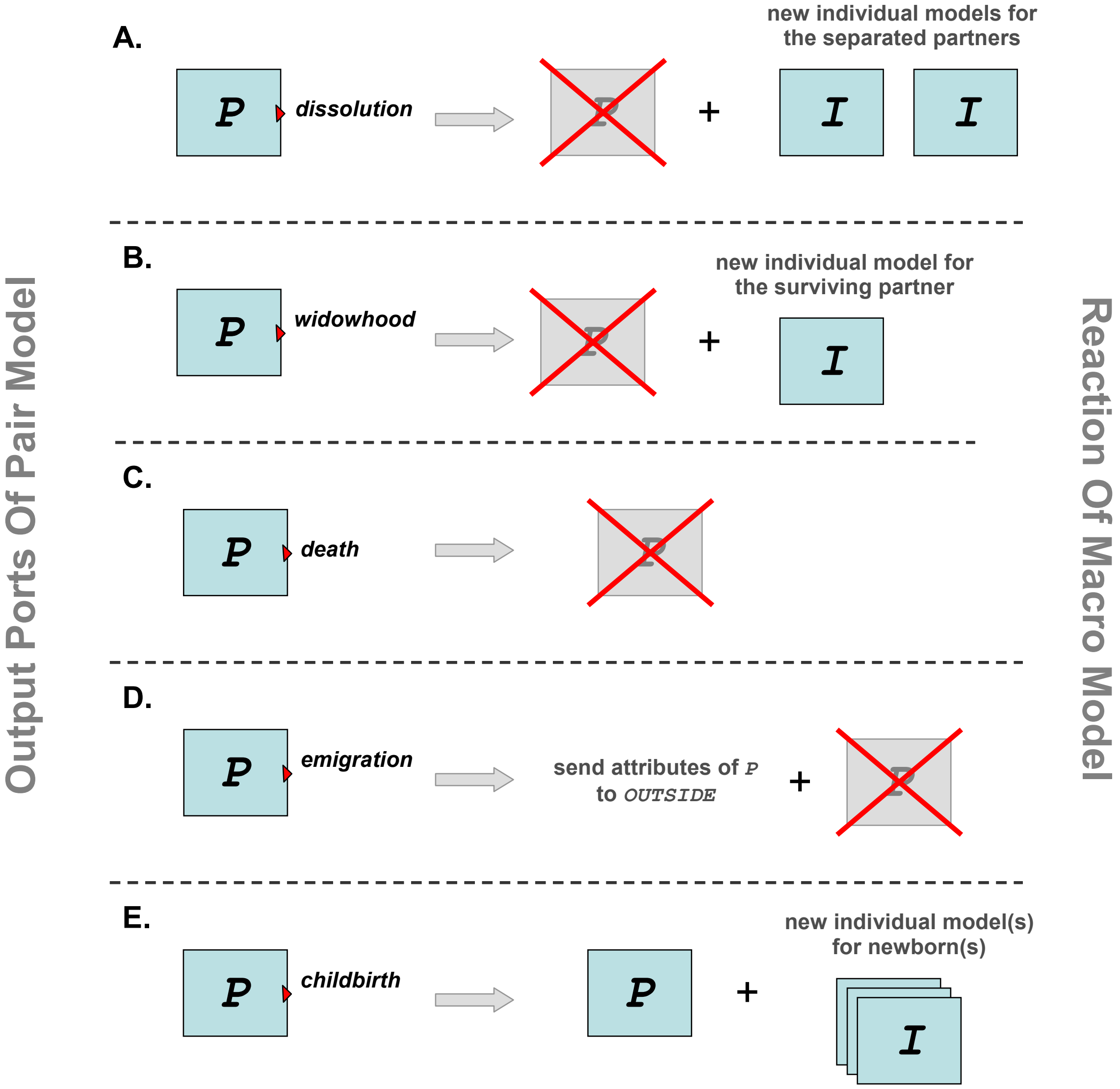


Figure 15.3.: Possible output messages of the pair models and reactions of the population model.

$\delta : {}^P\Psi \to {}^P\Psi$ is the state transition function of P:

$$\delta(\psi)(s_c, s_u, b_m, b_f, a_m, a_f, w) = (s_u, \underset{s_j \in \Psi / \{s_u\}}{\operatorname{argmin}} (w_{s_u,s_j}), b_m, b_f, a_m + \tau, a_f + \tau, 0),$$

where $\psi \in {}^P\Psi$, w_{s_u,s_j} is the (random) waiting time between the states s_u and s_j, and $\tau = \min_{s_j \in \Psi/\{s_u\}} (w_{s_u,s_j})$.

$p : {}^P\Psi \to 2^{\mathcal{P}_P}$ $(\mathcal{P}_P = \{dissolution, widowhood, death, emigration, childbirth\})$ selects the ports available in a given state of the pair model, i.e.,

$$p(\psi) = \begin{cases} dissolution & \text{if } dissolution(s_c, s_u) = true, \\ widowhood & \text{if } widowhood(s_c, s_u) = true, \\ childbirth & \text{if } childbirth(s_c, s_u) > 0, \\ death & \text{if } deathOfBoth(s_u) = true, \\ emigration & \text{if } emigration(s_u) = true, \\ \varnothing & \text{otherwise,} \end{cases}$$

$\lambda : {}^P\Psi \rightarrow Y$ is the output function; it is composed by five component functions that fill the outport ports $dissolution, widowhood, childbirth, death$ and $emigration$:

λ_1: If P is due to dissolve (i.e., $dissolution(s_c, s_u) = true$), λ_1 forwards via the output port *dissolution* the upcoming state of P and the birth dates of both spouses, otherwise $\lambda_1 = \varnothing$.

λ_2: If P is due to experience a widowhood event (i.e., $widowhood(s_c, s_u) = true$), λ_2 forwards via the output port *widowhood* the state and the birth date of the surviving partner, otherwise $\lambda_2 = \varnothing$.

λ_3: If P is due to experience a childbirth event (i.e., $childbirth(s_c, s_u) > 0$), λ_3 forwards via the output port *childbirth* the number of newborns, otherwise $\lambda_3 = \varnothing$.

λ_4: If P is due to experience the death of both partners (i.e., $deathOfBoth(s_u) = true$), λ_4 signalizes this to the output port *death*, otherwise $\lambda_4 = \varnothing$.

λ_5: If P is due to emigrate (i.e., $emigration(s_u) = true$), λ_5 forwards via the output port *emigration* the current state of P and the birth dates of both spouses, otherwise $\lambda_5 = \varnothing$.

$ta : {}^P\Psi \rightarrow \mathbb{R}_0^+ \cup \{\infty\}$ is the time advanced function: $ta(\psi) = \tau$.

The specification of the pair model P is very similar to the specification of the individual model I. It records in its state the attributes, the ages, and the birth dates of the female and the male spouse. Output ports are used to inform the macro model **Pop*** about structural changes. State transitions are specified in the same way as in the case of I. Only the method that is used to compute the waiting times between transitions differs.

To describe partnership dynamics, a stochastic model for individual behavior does not apply per se. In Section 14, we describe an approach that models couple dynamics by considering couples as units. Waiting times to next events are computed here for both spouses jointly (see equation (14.4) on page 132).

To inform **Pop*** about structural changes, the pair model P features five output ports (*dissolution*, *widowhood*, *death*, *emigration* and *childbirth*). If P experiences a dissolution event, it forwards via the port *dissolution* to **Pop*** the birth dates and the upcoming attributes of both spouses. In response, **Pop*** creates for the separated partners two individual models, and deletes P. The processing in case of a widowhood event is similar. The only difference is that solely for the surviving partner the individual model is created. If P experiences an emigration event, it forwards to **Pop*** the current attributes and the birth dates of the female and the male spouse. In response, **Pop*** sends P's data to the $OUTSIDE$ model (cf. description of **Pop*** model on page 141), and deletes P. Equally, the death of both spouse causes the deletion of P. In case of a childbirth event, P forwards the number of newborns to **Pop***. **Pop*** reacts by creating as many individual models as newborns have been reported. Figure 15.1.3 shows the output ports of P. The figure further illustrates the structural model changes that **Pop*** conducts in response to activated ports.

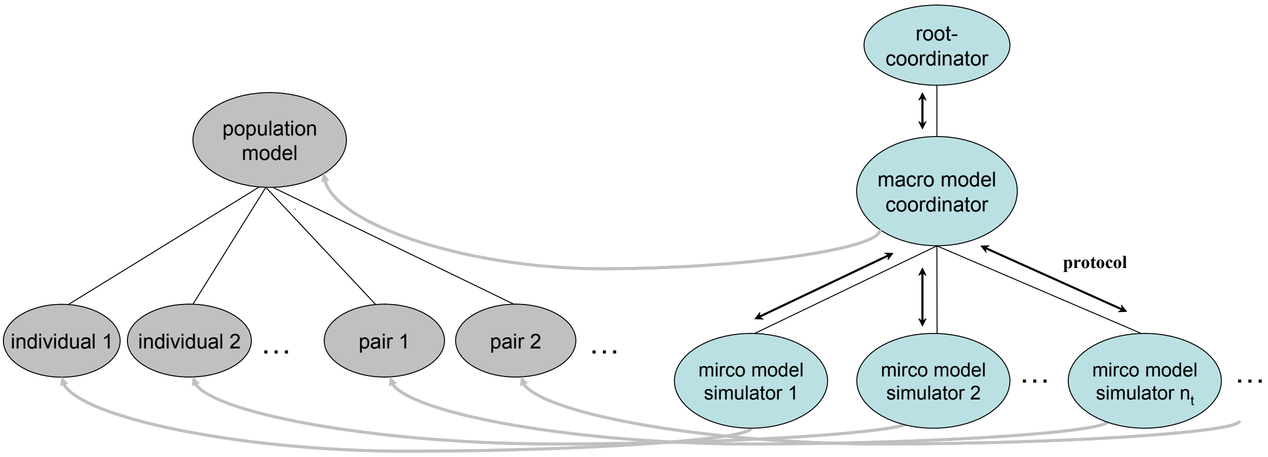


Figure 15.4.: Each ML-DEVS model can be mapped into a processor tree: to each individual model and pair model corresponds a simulator, and to the macro model a coordinator. The root-coordinator monitors the overall execution of the simulation. The communication between the different levels of the tree is conducted based on protocols.

15.2. Execution semantics of the ml-DEVS population model

For the execution of the presented ML-DEVS model, execution semantics must be provided. Traditionally, the execution semantics of a DEVS model is described by the *abstract simulator*, which comprises *simulators* and *coordinators* (Zeigler et al., 2000). In a ML-DEVS model, simulators correspond to micro models and coordinators to macro models. In line therewith, the abstract simulator of a ML-DEVS model comprises three algorithms: the algorithm that describes the processing of the root-coordinator, the algorithm that specifies the processing of macro model coordinators, and the algorithm corresponding to the simulators of micro

Notation	***Message Type***
$\star$-message	activation message, message indicating that an internal event is due
y-message	output message
x-message	input message
done-message	termination message

Table 15.1.: DEVS message notation.

models.

The root-coordinator manages the overall simulation loop. It initializes a new simulation and instructs the model execution until some termination criteria is met. Its processing scheme equals the processing scheme of the root-coordinator of DYNPDEVS described in Section 4.3.1, see Algorithm 1 on page 42.

A ML-DEVS coordinator implements the event queue for the next event times of the macro model and its micro model components. It is responsible for the correct synchronization of its model components and for the handling of external events (in our case: the arrival of immigrants). For this purpose, a coordinator employs communication protocols: it waits for protocols sent by its subordinates and forwards them to its parent. Figure 15.4 displays the processor tree corresponding to the ML-DEVS population model.

To ensure consistency within a simulation step, messages are processed in a well-defined order (Table 15.1 summarizes the different types of messages): if a $\star$-message activates a model, a y-message with output data is forwarded. Afterwards, the simulator waits for a x-message comprising input information. Finally, a *done*-message signalizes the completion of a simulation step. For the considered population network, Figure 15.5 depicts the communication protocol between the micro model simulator and the macro model coordinator, as well as the communication protocol between the macro model coordinator and the root coordinator.

15.2.1. An abstract simulator for ml-DEVS

In this section, we describe the abstract simulator that Uhrmacher et al. (2007) have originally developed for the execution of their ML-DEVS formalism.[3] They present a simulator for micro models that is very similar to the DYNPDEVS simulator of Uhrmacher (2001), see Section 4.3.1 Algorithm 3 on page 44. It mainly differs with respect to only three aspects: first, instead of defining three transition functions for conducting internal, external and confluent events, in the ML-DEVS simulator all kind of events are handled by one single transition function ('delta'). Second, the ML-DEVS simulator does not process any structural model changes.[4] Finally, opposed to DYNPDEVS, ML-DEVS models allow to select a set of available

---

[3] Originally, the ML-DEVS simulator of Uhrmacher et al. (2007) includes a "value coupling downward" method that couples state variables of the macro model to input ports of the micro models. Furthermore, originally the macro model features an "activation constrain" for its execution. In our setting, both functionalities are not applied. Therefore, in the description of the abstract simulator we skip their definition.

[4] Note that this feature is special to the presented ML-DEVS population model. In the original ML-DEVS approach micro models can adapt their own behavior.

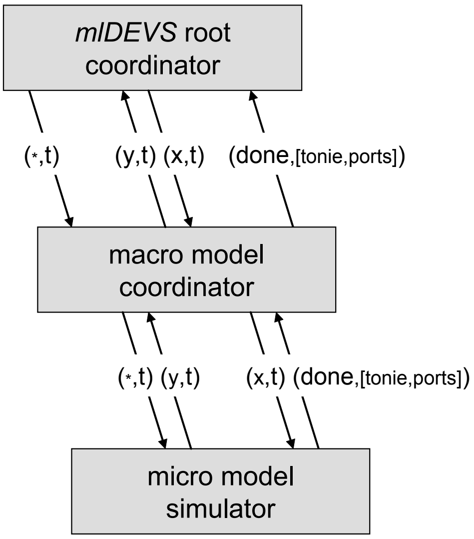


Figure 15.5.: ML-DEVS communication protocol. The ⋆-, y-, and x- comprise beside their regular information also the actual simulation time t, and the *done* message comprises the time *tonie* of the next internal event and all approachable ports *ports*.

ports (depending on the model state). Combined with the time *tonie* of the next event, these *ports* are sent via the *done* message to the coordinator. Algorithm 6 shows the source code of the micro model simulator of Uhrmacher et al. (2007).

Algorithm 6 Pseudocode of the simulator of the micro model.

```
when receive ⋆ or $x$ message
if message is ⋆ message then
  execute model.lambda
  send $y$ message to parent
  wait for receive $x$ message
end if
execute model.delta
execute model.timeAdvance
ports = model.availablePorts
send done message including tonie and ports
end when
```

The ML-DEVS coordinator combines the functionality of DEVS coordinators and simulators. That is, it is designed to coordinate on the one side the execution of the micro model simulators, and to conduct on the other side macro model transitions. The coordinator developed by Uhrmacher et al. (2007) works as follows: initially, the coordinator is activated by a ⋆-message from the root-coordinator. Then, as a first step, it activates its imminent micro model components by sending them a ⋆-message. The coordinator waits for their response, and then records received messages. If the current model time t corresponds to the coordinator's actual *tonie*, it executes the output function of the macro model and its downward output function. Afterwards, it sends its own output message and the output messages of its children to its parent, and waits for response. It then merges the response message from the parent and the value of its downward output function, and sends the resulting x-message to its imminent and influenced children. Now, the coordinator waits for their *done* messages, and stores their next event times *tonie* and currently exhibited ports. If the coordinator has received a message (about an external event) by its parent or its *tonie* is due, it conducts a state transition. For this purpose, it calls the transition function and the time advanced function of the macro model. A transition of the macro model might imply structural model changes of children or the coupling structure. If this is the case, the coordinator performs these changes. Subsequently, it updates the available ports *myports* of its children models, and sends a *done* message, including *myports* and the next event time to its parent. This processing is repeated as long as the root-coordinator sends $\star$-messages and/or x-messages.

To avoid problems when processing structural model changes, Himmelspach and Uhrmacher (2004) recommend to sort them in each step. They suggest to first create models and ports, then to create and delete couplings, to subsequently delete ports, and finally to delete models.

15.2.2. Limitations of a direct implementation of the abstract simulator

The abstract simulation that Uhrmacher et al. (2007) have developed for ML-DEVS is essentially a direct implementation of the ML-DEVS processing scheme. Such a processing might pose problems (Himmelspach and Uhrmacher, 2006). In Section 4.3.2 we have already discussed the potential problems of a direct implementation of the DYNPDEVS processing scheme. As mainly all DEVS variants rely on similar processing schemes, the problems we have described for DYNPDEVS also apply to the processing of ML-DEVS. That is, depending on the depth of the processor tree and the number of models, passing message through the processor tree can result in a remarkable overload. Furthermore, the ML-DEVS simulator of Uhrmacher et al. (2007) demands for each micro model one thread.[5] Usual Java virtual machine implementations, however, do not support more than a few thousand parallel threads. In our case, this implies that meaningful population projections would not be possible. To overcome this limitation, we design a novel simulator for ML-DEVS models.

15.2.3. A sequential abstract simulator for ml-DEVS

In this section, we present a simulation algorithm for ML-DEVS that extends the "sequential abstract simulator" for PDEVS developed by Himmelspach and Uhrmacher (2006). The sequential abstract simulator is conceptualized to sequentially compute (parts of) the processor tree. It maintains the basic concept of the classical DEVS coordinators and simulators, inclusive their communication scheme. However, it modifies the simulation processing by splitting it into subsequent steps. Relying on the setting for PDEVS, we implement these steps using

---

[5] Within a program, a thread is a sequentially executed stream of instructions.

Algorithm 7 Pseudocode of the coordinator of the macro model.

```
//Create during initialization
myports = all currently accessible ports of children

when receive $\star$ or $x$
if message is $\star$ message then
  send $\star$ message to imminent children
  wait for receive $y_{\text{micro}}$ messages from the imminent children
  if $t$ is getTonie(macroModel) then
    $y_{\text{macro}}$ = macroModel.lambda
    $x_{\text{down}}$ = macroModel.lambdaDown
  end if
  $y$ = union $(y_{\text{macro}}, y_{\text{micro}})$
  send $y$ message to parent
  wait for receive $x_{\text{parent}}$ message from parent
end if
from $x_{\text{parent}}$ create $x$ messages for all influenced or imminent children
if $t$ is getTonie(macroModel) then
  $x$=union $(x, x_{\text{down}})$
end if
send $x$ messages to all influenced or imminent children
wait for receive done messages from all influenced or imminent children
store done.tonie and done.ports of all influenced and imminent children
$x_{\text{mac}}$ is message for macroModel included in $x_{\text{parent}}$
if $x_{\text{mac}}$ is not empty or $t$ is getTonie(macroModel) then
  execute macroModel.delta($x_{\text{mac}}$,ports)
  execute macroModel.timeAdvance
  macroModel.processStructuralChanges
  myports = macroModel.availablePorts
end if
  tonie = min(childTonies,getTonie(macroModel))
send done-message including tonie and myports
end when
```

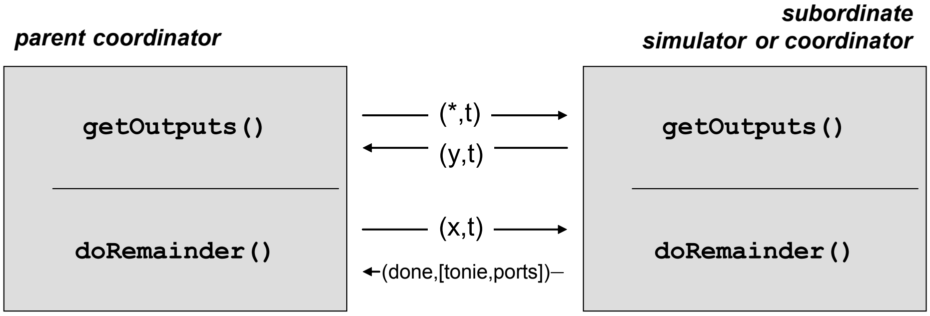


Figure 15.6.: Processing scheme of the sequential abstract simulator of ML-DEVS.

the two methods `getOutputs` and `doRemainder`, cf. Figure 15.6. During simulation processing, these methods are successively called. The simulator and coordinator of a ML-DEVS model realize the methods `getOutputs` and `doRemainder` differently, cf. Algorithm 9 and Algorithm 8.

If its `getOutputs` method is called, a macro model coordinator activates its imminent children, and reads their output messages. If the current model time t corresponds to the coordinator's actual *tonie*, the coordinator executes subsequently the output function of the macro model and its downward output function. Finally, `getOutputs` forwards to the parent of the coordinator the children's output messages and its own output message.

Afterwards, the `doRemainder` method of a macro model coordinator is called. As a first step, it identifies all individuals who are influenced by external events. Then, if the current model time t equals the coordinator's actual *tonie*, it combines the input message from the parent and the value of its downward output function, and sends the resulting message to its imminent and influenced children. Otherwise (if $t \neq tonie(\text{macroModel})$), the coordinator solely forwards the message from its parent to its imminent and influenced children. The coordinator executes the children's `doRemainder` method and requests their next event times (*tonies*) and currently exhibited ports, and stores them. Subsequently, the coordinator checks whether it has received a message (about an external event) by its parent. In case it has received a message or its *tonie* is due, it executes the transition function and the time advanced function of the macro model. Eventual caused structural model changes are subsequently executed. The `doRemainder` method exits by updating the available ports *myports* of all children models, and by determining and forwarding the system's next event time.

The micro model simulator employs the method `getOutputs` to call the model's output function and to forward the respective output information. By `doRemainder`, a micro model simulator performs a model's state transition and determines the next internal event. Finally, it forwards to its superordinate coordinator the ports available in the current state of the model and its next event time. The functionality of the micro model simulator basically equals the one that Himmelspach and Uhrmacher (2006) have described for PDEVS. Additionally, it

Algorithm 8 Pseudocode of the coordinator of the 'sequential abstract simulator for ML-DEVS'.

```
//Create during initialization
myports = all currently accessible ports of children

function Messages getOutputs()
  for each child in imminents do
    allMsgs = union (allMsgs,child.getOutputs())
  if $t$ is getTonie(macroModel)
    macro.msgs = macroModel.lambda
    macro.downMsgs = macroModel.lambdaDown
    msgs = union (allMsgs,macro.msgs)
  end if
  return msgs
end function

function (double,ports) doRemainder(msgs)
  influencees = union (influencees,propagateInputOutputCouplingMessages(msgs))
  if $t$ is getTonie(macroModel)
    msgs = union (msgs,macro.downMsgs)
  end if
  for each child in union (influencees, imminents) do
    (childTonies,ports) = union(msgs,child.doRemainder())
  macroMsg = getMsgForMacro(msgs)
  if (macroMsg is not empty) or $t$ is getTonie(macroModel)
    execute macroModel.stateTransition(macroMsg,ports)
    execute macroModel.timeAdvance
    macroModel.processStructuralChanges
    myports = macroModel.availablePorts
  end if
  tonie = min(childTonies,getTonie(macroModel))
  return union (tonie,myports)
end function
```

Algorithm 9 Pseudocode of the simulator of 'the sequential abstract simulator for ML-DEVS.'

```
function Messages getOutputs()
  execute model.lambda
  return getMsgs(model)
end function

function (double,ports) doRemainder(msgs)
  execute model.stateTransition(msgs)
  execute model.timeAdvance
  ports = model.availablePorts
  return union (tonie,ports)
end function
```

features a function to determine the ports that the micro model exhibits in a given state.

Using the sequential abstract simulator for ML-DEVS to execute $\mathbf{Pop}^\star$ means that independently of the population size we need only two threads (one for the macro model coordinator and one for the root-coordinator). That is, opposed to a direct implementation of the ML-DEVS processing scheme, the sequential abstract simulator for ML-DEVS allows to define a model's size independently of the number of threads supported by an Java virtual machine.

16. Application: A Synthetic Population Resembling the Netherlands

We conduct a small case study to illustrate the capabilities of the novel microsimulation model that accounts for partnership dynamics. For this purpose, we look at a synthetic population that (roughly) resembles the Dutch population. The setting of this case study is similar to the one of the MicMac microsimulation case study (cf. Section 6). We consider fertility and partnership behavior, and smoking behavior. In particular, we study how partners influence each other's smoking behavior. Note that the presented application is mainly based on synthetic data, and should not be used to draw conclusions about actual behavior. It only serves as a means to demonstrate the potential of a microsimulation that accounts for interdependencies between life-courses of married or cohabiting individuals.

In the following, we assume that life-course transitions of individuals and couples only depend on age and calendar time; i.e., we neglect possible 'duration dependencies' of demographic processes.[1] Furthermore, we specify couple dynamics by considering couples as units. We describe the behavior of unlinked individuals employing the model described in Section 2.3.1, and we specify couple behavior using the model presented in Section 14.1.2. This chapter is structured as follows: First, we introduce the data and the scenario that we use for our illustration. Then we show some results.

16.1. The synthetic population

Starting on January 1, 2008, we generate life-courses of 1% of a synthetic population that resembles the Netherlands. The simulation horizon ranges from January 1, 2008 to December 31, 2020. During simulation, we only consider individuals aged between 0 and 63. The initial population comprises 70,296 males and 68,269 females.

The state space that we employ for the individual model (cf. Section 2.1) consists of the following elements (variables with corresponding values given after the colons, separated by semi-colons):

- gender: female; male

---

[1] Disregarding duration dependencies means e.g., to neglect the fact that evidentially the divorce risk depends on the time that has elapsed since wedding and that the propensity of giving birth to a second child depends on the time elapsed since the delivery of the first child.

- marital status and living arrangement: living in the parental home and never married; married; being single; cohabitating;
- fertility: childless; at least one child
- smoking status: non-smoker; smoker
- mortality: dead; alive

The elements of the state space for the couple model (cf. Section 14.1.1) are:

- marital status: married; dissolved; cohabiting
- fertility status: childless; parents
- smoking status: non-smoker couple; dual smoker couple; female smoker, male non-smoker; male smoker, female non-smoker
- mortality: both dead; female dead, male alive; male dead, female alive; both alive

If the value 'living alone' is assigned to an individual, he/she lives either alone and never lived in a union before, or he/she lives alone but cohabitated before, or he/she lives alone and was married before. The value 'dissolved' indicates the separation of a married or cohabiting couple. Spouses that are dissolved or widowed enter the 'living alone' state of the individual model. During simulation, individuals and couples can experience the following events:

Possible events of individuals	leaving parental home, getting divorced or separated, giving birth (for females), launching a cohabitation, marrying, quitting to smoke, starting to smoke, dying
Possible event of couples	getting divorced or separated, childbirth, female starts smoking, male starts smoking, female quit smoking, male quit smoking, both quit smoking, both start smoking, female dies and male gets widowed, both die male dies and female gets widowed

To quantify the propensity of individuals and couples to experience events, we employ either empirical or synthetic transition rates. We have estimated (non-parity specific) fertility rates and transition rates to change the marital status and/or the living arrangement for unlinked individuals using the FFS_NL 2003 (the corresponding rates are given in the appendix, see page 201ff). For this purpose, we have applied a slightly modified version of the MAPLES estimation procedure[2] (Impicciatore and Billari, 2007). We use hypothetical death rates and

---

[2] MAPLES estimates age profiles from longitudinal survey data using a generalized additive model and piecewise cubic splines, cf. Section 3.2.1.

transition rates of changing the smoking behavior. It is well known that smokers have a higher mortality risk than non-smokers (Doll et al., 2004; Jacobs et al., 1999). We account for this fact by adapting the mortality rates presented in Section 6 (see Figure 7.2 on page 69) accordingly: mortality rates for smokers are obtained by increasing these rates by 10% and mortality rates for non-smokers are obtained by reducing them by 10%. The resulting mortality schedule – and the transition rates that we use to describe the propensity of individuals to change the smoking behavior – are given in the appendix on page 201ff.

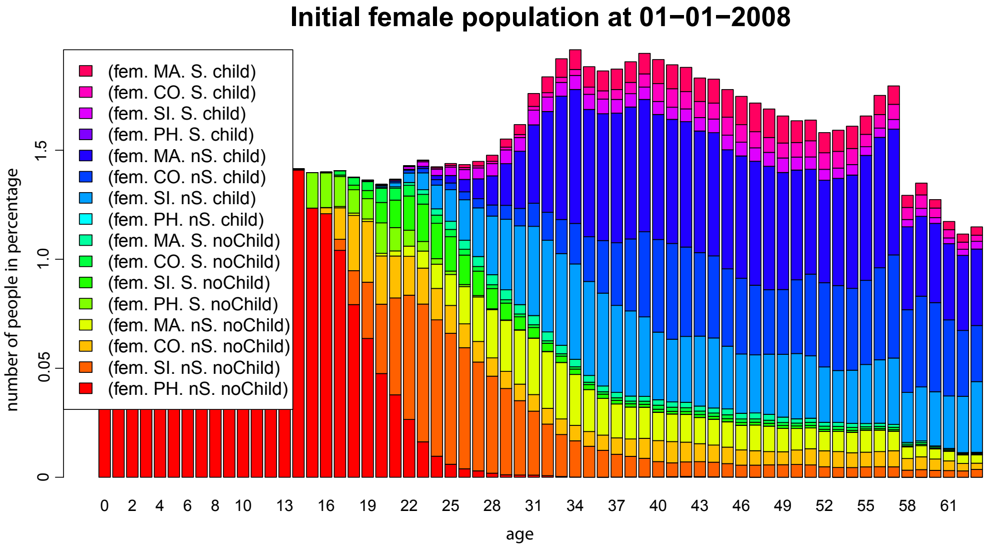


Figure 16.1.: The female initial population refers to January 1, 2008. The figure shows the female population, categorized according to age, fertility status, marital status, and smoking behavior; coding: noChild (childless), child (being mother), PH (living at parental home), SI (living alone), CO (cohabiting), MA (being married), S (smoker), nS (non-Smoker).

To describe couple behavior, for simplicity we assume that

- the same fertility rates apply to linked and unlinked women,
- the risk to quit smoking is higher for a female/male who is married/cohabiting with a non-smoker than it is for a female/male who is married/cohabiting with a smoker, but does not depend on the presence of children,
- the risk to start smoking depends on whether the partner smokes or not, but not on the presence of children,
- the influence that the smoking behavior of the spouse has on the own smoking behavior does not vary with age,
- the divorce risk and the risk to break up depends on the presence of children, but not on the smoking behavior of the spouses,

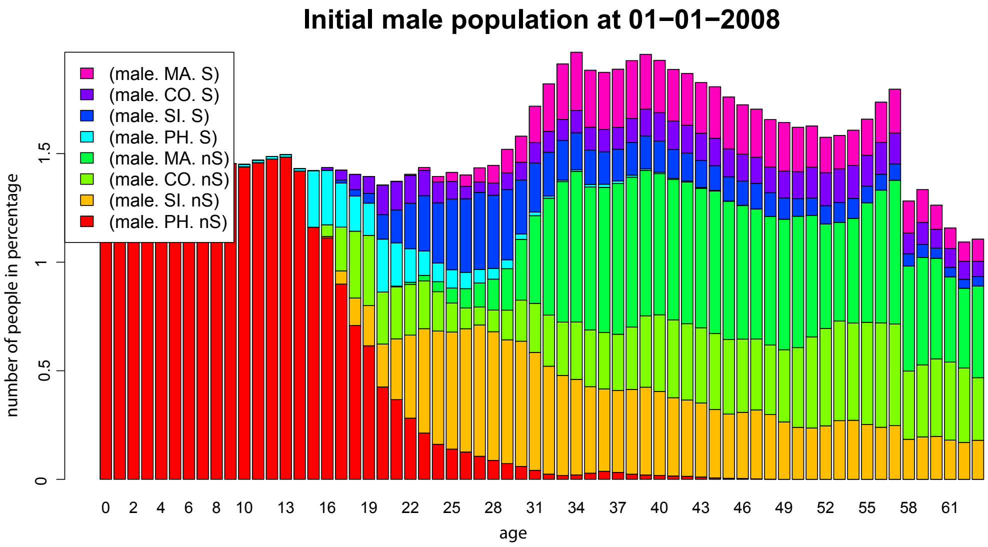


Figure 16.2.: The initial population refers to January 1, 2008. The figure displays the male population, categorized according to age, marital status, and smoking behavior; coding: PH (living at parental home), SI (living alone), CO (cohabiting), MA (being married), S (smoker), nS (non-Smoker).

- to linked and unlinked individuals the same mortality risk applies, and
- the risk that both spouses die at the same time is very low (we set it to 10^{-5}).

The transition rates of couples to change their marital status or their smoking behavior are given in the appendix, see page 201ff. We assume that the mortality rates vary with age, sex and calendar year, and the other rates are age- and sex-specific, but are held constant over calendar time.

In Section 3.3, we have presented data and methods that are suitable to construct an initial population of unlinked individuals. Adding pair information to an initial population does theoretically not pose any difficulties. Problems arise due to the fact that micro information on couples is often hard to access. In contrast to single individuals, we need "double information" to determine a couple: we need the birth dates and the attributes of both spouses. Here, we settle for an initial population that comprises only individuals. For our illustration, we construct an initial population using the method of iterative proportional fitting (cf. Section 3.3.1). The marginal frequency distributions needed to estimate these numbers have been taken from EuroStat (EuroStat, 2010) and the FFS_NL 2003 survey. In the Figures 16.1 and 16.2, the resulting population numbers are given. The first figure depicts the initial female population cross-classified according to age, sex, marital status, and fertility status, and the second figure shows the initial male population according cross-classified according to age, sex, and marital status. Generally, our simulation model allows the consideration of migration, however, for reasons of simplicity we neglect migration for this example.

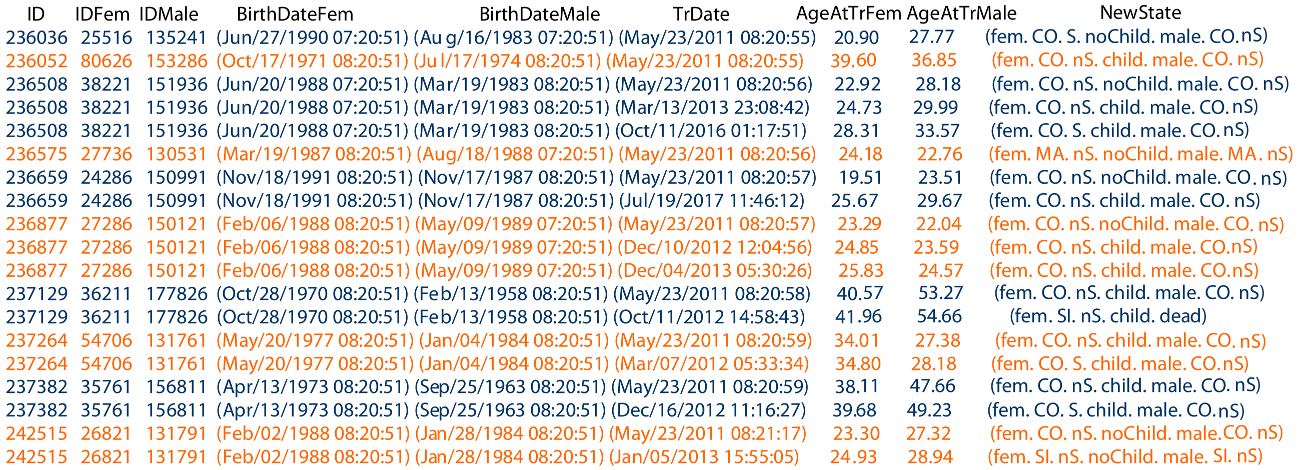
During simulation, we conduct mate-matching. For this purpose, we employ the algorithm presented in Section 11. Compared to the illustration of the mate-matching algorithm given in Section 12, however, we present here a simplified version: We do not consider the educational attainment of individuals and we do not differentiate between first and higher order partnerships. Nonetheless, our setting is sufficient to illustrate the capability of the micro-macro population model. We use two logit models to quantify the compatibility between potential spouses: The first model describes the probability to enter a cohabitation and the second model describes the probability to enter a marriage. For estimating the models, we employ the first wave of the NKPS (conducted in the period from 2002 to 2004) (Dykstra et al., 2005). We only consider partnerships that started in the years from 1990 to 2002. Our data set contains a record for each observed couple, which consists of the age of the male spouse, the age difference between the female and the male spouse (in integer years), and the number of children that the spouses have with former partners. The coefficients of the estimated logit models are given in the appendix, see page 205.

16.2. Results

For each simulation run the micro-macro population simulation provides information on the simulated life-courses in three files:

1. an ASCII file containing the birth dates of all simulated individuals,
2. an ASCII file containing the dates of transitions and the corresponding destination states for all simulated individuals, and
3. an ASCII file containing the dates of transitions and the corresponding destination states for all simulated couples.

These files have a well-defined format, which can be accessed and managed further by arbitrary tools. We use R to summarize and visualize the simulation output. Subsequently, we show a few descriptive statistics for simulated couple data. For the evaluation of data of unlinked individuals we refer to the description given in Section 5.3 and in Section 6.2. The results shown here are not comprehensive and only aim at indicating the potential of a demographic microsimulation with linked lives. As a first step we converts the simulation output into a format resembling event history data. This format eases further computation. In Figure 16.3 typical life-courses of ten simulated couples are given.



ID	IDFem	IDMale	BirthDateFem	BirthDateMale	TrDate	AgeAtTrFem	AgeAtTrMale	NewState
236036	25516	135241	(Jun/27/1990 07:20:51)	(Au g/16/1983 07:20:51)	(May/23/2011 08:20:55)	20.90	27.77	(fem. CO. S. noChild. male. CO.nS)
236052	80626	153286	(Oct/17/1971 08:20:51)	(Ju l/17/1974 08:20:51)	(May/23/2011 08:20:55)	39.60	36.85	(fem. CO. nS. child. male. CO. nS)
236508	38221	151936	(Jun/20/1988 07:20:51)	(Mar/19/1983 08:20:51)	(May/23/2011 08:20:56)	22.92	28.18	(fem. CO. nS. noChild. male. CO. nS)
236508	38221	151936	(Jun/20/1988 07:20:51)	(Mar/19/1983 08:20:51)	(Mar/13/2013 23:08:42)	24.73	29.99	(fem. CO. nS. child. male. CO. nS)
236508	38221	151936	(Jun/20/1988 07:20:51)	(Mar/19/1983 08:20:51)	(Oct/11/2016 01:17:51)	28.31	33.57	(fem. CO. S. child. male. CO. nS)
236575	27736	130531	(Mar/19/1987 08:20:51)	(Aug/18/1988 07:20:51)	(May/23/2011 08:20:56)	24.18	22.76	(fem. MA. nS. noChild. male. MA. nS)
236659	24286	150991	(Nov/18/1991 08:20:51)	(Nov/17/1987 08:20:51)	(May/23/2011 08:20:57)	19.51	23.51	(fem. CO. nS. noChild. male. CO. nS)
236659	24286	150991	(Nov/18/1991 08:20:51)	(Nov/17/1987 08:20:51)	(Jul/19/2017 11:46:12)	25.67	29.67	(fem. CO. nS. child. male. CO. nS)
236877	27286	150121	(Feb/06/1988 08:20:51)	(May/09/1989 07:20:51)	(May/23/2011 08:20:57)	23.29	22.04	(fem. CO. nS. noChild. male. CO. nS)
236877	27286	150121	(Feb/06/1988 08:20:51)	(May/09/1989 07:20:51)	(Dec/10/2012 12:04:56)	24.85	23.59	(fem. CO. nS. child. male. CO.nS)
236877	27286	150121	(Feb/06/1988 08:20:51)	(May/09/1989 07:20:51)	(Dec/04/2013 05:30:26)	25.83	24.57	(fem. CO. nS. child. male. CO.nS)
237129	36211	177826	(Oct/28/1970 08:20:51)	(Feb/13/1958 08:20:51)	(May/23/2011 08:20:58)	40.57	53.27	(fem. CO. nS. child. male. CO. nS)
237129	36211	177826	(Oct/28/1970 08:20:51)	(Feb/13/1958 08:20:51)	(Oct/11/2012 14:58:43)	41.96	54.66	(fem. SI. nS. child. dead)
237264	54706	131761	(May/20/1977 08:20:51)	(Jan/04/1984 08:20:51)	(May/23/2011 08:20:59)	34.01	27.38	(fem. CO. nS. child. male. CO. nS)
237264	54706	131761	(May/20/1977 08:20:51)	(Jan/04/1984 08:20:51)	(Mar/07/2012 05:33:34)	34.80	28.18	(fem. CO. S. child. male. CO.nS)
237382	35761	156811	(Apr/13/1973 08:20:51)	(Sep/25/1963 08:20:51)	(May/23/2011 08:20:59)	38.11	47.66	(fem. CO. nS. child. male. CO. nS)
237382	35761	156811	(Apr/13/1973 08:20:51)	(Sep/25/1963 08:20:51)	(Dec/16/2012 11:16:27)	39.68	49.23	(fem. CO. S. child. male. CO.nS)
242515	26821	131791	(Feb/02/1988 08:20:51)	(Jan/28/1984 08:20:51)	(May/23/2011 08:21:17)	23.30	27.32	(fem. CO. nS. noChild. male.CO. nS)
242515	26821	131791	(Feb/02/1988 08:20:51)	(Jan/28/1984 08:20:51)	(Jan/05/2013 15:55:05)	24.93	28.94	(fem. SI. nS. noChild. male. SI. nS)

Figure 16.3.: The simulated life-courses of ten couples. The column ‘BirthDateFem’ gives the birth dates of the female spouses, the column ‘BirthDateMale’ comprises the birth dates of the male spouses, ‘TrDate’ contains the transition dates, ‘AgeAtTrFem’ the corresponding transition ages of the female spouses and ‘AgeAtTrMale’ the corresponding transition ages of the male spouses. ‘NewState’ gives the states that couples enter when they undergo an event.

Each record describes an event that a couple has experienced during simulation. It gives the ID of the couple and the birth times of both spouses (‘BirthDateFem’ and ‘BirthDateMale’). Furthermore, it contains the transition date (‘TrDate’), the transition age of the female spouse (‘AgeAtTrFem’), the transition age of the male spouse (‘AgeAtTrMale’), and the state that the couple has entered (‘NewState’). The first transition of a couple corresponds to the onset of the marriage/cohabitation. If a couple experiences a dissolution event, the spouses return to the ‘unlinked population’. Likewise, in case of a widowhood event, the surviving spouse becomes part of the unlinked population. During simulation, 7, 858 couples have been created, of these 2.6% experience a dissolution event, and 0.9% undergo a widowhood event, i.e., in our setting most relationships persist.

Having a partner who smokes can influence the spouse’s initiation of smoking. That is, a smoking spouse might incite his/her non-smoking partner to start smoking, or prevent his/her smoker partner from quitting smoking. It is also possible that a non-smoking partner urges his/her spouse to stop smoking. We study whether in our simulation output we can find accordant behavior. For this purpose, we determine how many partnered smokers quit smoking, conditional on the smoking status of the partner. The subsequent table shows the results:

Partner smokes? *Gender*	yes	no
male	19.05%	43.62%
female	17.50%	26.31%

That is, during simulation 19.05% of the male smokers who are married/cohabiting with a smoking woman quit smoking. This is contrasted by 43.62% smokers partnered with a non-smoking woman. Furthermore, we find that 26.31% of the smoking women who are partnered with a non-smoking man quit smoking, opposed by 17.50% who are partnered with a smoker. Overall, being partnered with a non-smoking partner has a positive effect on an individual’s propensity to quit smoking. In the simulation output the effect is more pronounced for males than for females. Among the unlinked individuals, 28.9% of the women and 17.10% of the men quit smoking during simulation time. We obtain the following percentages of spouses who start smoking, conditioned on the partner’s smoking behavior:

Partner smokes? *Gender*	yes	no
male	8.50%	4.6%
female	3.63%	1.34%

Here we see that a smoking spouse strongly influences the smoking behavior of his/her partner. Again, it seems that women have a stronger influence on their partners than men. Among the unlinked individuals we find 5.44% male non-smokers and 2.43% female non-smokers who start smoking. All these results are in accordance with the input transition rates for smoking behavior; see Figure A-21 and Figure A-20 in the appendix on page 201ff.

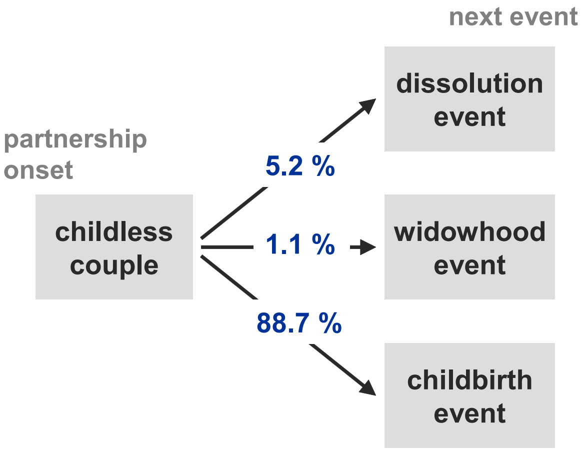


Figure 16.4.: First event after partnership onset.

As a further aspect, we study the behavior of childless couples once they have married or launched a cohabitation. We study dissolution and childbirth events, and restrict our consideration to the first event that happens after partnership onset. Figure 16.4 shows the respective results. After having entered marriage or cohabitation, 88.7% of the childless couples experience a childbirth event, 5.2% undergo a dissolution event, and 1.1% suffer from a widowhood event. Hence, in our virtual population, childless couples have a very high propensity of becoming parents.

As mentioned before, numerous summaries and insights can be derived from the simulation output of the micro-macro population model, but we restrict ourselves here to the few examples given above. Again, please note that the results shown are only illustrative and do not reflect observed behavior – because they are based on input rates and model assumptions that are not solely derived from empirical data.

17. Conclusions

Microsimulation is a methodology that is well suited for population projections as it very closely mimics life-course dynamics.

To face the challenges of aging societies, policy makers need reliable information about future population composition, and, in particular, answers to the questions of how different policies might affect individual behavior. Therefore, an adequate monitoring and forecasting of the lifestyle and life courses of individuals is required. What particularly is needed a methodology that complements the demographic projections by age and sex with projections of the way people live their lives concerning marriage, fertility behavior, health related behavior, educational and professional careers, etc. Microsimulation lends itself naturally to this objective.

In this thesis, we described the development of the demographic microsimulation that we have realized in the context of the project 'MicMac - Bridging the micro-macro gap in population forecasting'. Furthermore, we detailed extensions that we have added to overcome some limitations of the initial version of the MicMac microsimulation. In its initial version, the MicMac microsimulation does not account for linked lives, but considers individuals as being independent of each other. Such a setting certainly oversimplifies reality. To tackle this shortcoming, we first developed a mate-matching algorithm for the MicMac microsimulation, and, in a second step, we designed a micro model to describe the life-course dynamics of married and cohabiting couples.

17.1. Summary of findings

A tool for population projection: the MicMac microsimulation

Microsimulation is a technique that models and simulates the behavior of micro units (like individuals or firms) over time. The centerpiece of a demographic microsimulation is the virtual population that (commonly) resembles a real population. The virtual population comprises individuals characterized by demographic attributes like sex, current marital status, educational attainment, children ever born, etc. It evolves over time because, based on a stochastic model, individuals experience certain demographic events over their life time. The length of spells between events gives, together with the individual's birth date, the ages at which events are experienced.

In demographic microsimulations, time can be modelled to advance either in discrete steps (usually in years) or continuously. In discrete-time models, individual attributes and behavior are updated at each step, while in continuous-time microsimulation models events can happen at any instant in time along a continuous time scale. A continuous-time microsimulation defines individual life-courses as sequences of discrete events; hence a continuous-time microsimulation model is a discrete event model.

To describe individual behavior, we use a continuous-time multi-state model; i.e., we use a continuous-time microsimulation model. Multi-state models are commonly determined by stochastic processes that at any point in time occupy one out of a set of discrete states (Hougaard, 1999). In this thesis, Markovian processes are employed to specify individual life-courses (cf. Chapter 2). The propensity for experiencing certain demographic events is usually age-dependent but also varies with calendar time (e.g., decreasing mortality). In some cases, also the time that has been elapsed since the last event affects an individual's propensity to experience a transition, e.g., the divorce risk depends on the time that has elapsed since wedding. According to the application, if we assume that propensities to experience life-course events depend only on age and calendar time, non-homogeneous continuous-time Markov chains are suitable to describe life-course dynamics. Otherwise, if we assume that the propensities to experience events also depend on the time elapsed in a state, we suggest to use non-homogeneous semi-Markov processes. Current age-, time-, and (optionally) duration-specific incidence rates can be estimated from vital statistics and social surveys; e.g., by using occurrence-exposure rates (cf. Sections 3.1 and 3.2). Assumptions about future rates then define the projection scenarios (cf. Section 3.2.2).

The execution of a continuous-time microsimulation can be very efficient compared to a discrete-time microsimulation. This is because in a continuous-time microsimulation, individual attributes are only updated when an event occurs. Only very few continuous-time microsimulations exist. Many modelers consider them as being too cumbersome to deal with (Galler, 1997). For instance, Scott et al. (2003, p. 5) argue that "continuous-time simulation has certain theoretical advantages, but implementation is more difficult, and model operation less transparent to the user, than for discrete-time". However, several formalisms exist that support discrete event based models and ease their realization, e.g., DEVS (Zeigler et al., 2000), stochastic Petri Nets (Murata, 1989), and stochastic pi-calculus (Priami, 1995). Due to its modular and state-based concept, the DEVS formalism is well-suited to specify a continuous-time microsimulation. There are a couple of tools around that support the implementation of DEVS models (like the DEVS variants in JAMES II). Therefore, within this thesis, we deem DEVS as being appropriate to realize the MicMac microsimulation. In Chapter 4 we proposed two DEVS model specifications of the MicMac microsimulation. We described an atomic DEVS model, and we revealed strengths but also limitations of this solution. The limitations were addressed by exploiting a DEVS variant that supports variable structures, i.e., DYNPDEVS. We described a DYNPDEVS model that specifies individuals as atomic models and the population as a network model. We detailed the semantics that have originally been developed for the execution of a DYNPDEVS model. We found that a simulation based on this semantics hinders meaningful population projections because it demands at least one process/thread per model scheme. To tackle this problem, we worked out execution semantics that allows to efficiently run meaningful population projections with the developed DYNPDEVS model (cf. Section 4.3.3). The designed semantics is an extension of one of the sequential simulation algorithms originally developed for PDEVS (Himmelspach and Uhrmacher, 2006).

It is also possible to translate a continuous-time microsimulation model directly into source code, without specifying it beforehand as discrete event system. In line with this, we directly realized the MicMac microsimulation model using the modelling and simulation framework JAMES II as a library (cf. Chapter 5). This procedure can entail some unpleasant problems; for example, extensions of the microsimulation (such as the inclusion of linked lives) might be costly and laborious. For the MicMac microsimulation (without any extension), however, such an immediate implementation is entirely convenient. We used this implementation to illustrate the capabilities of the MicMac microsimulation (see Chapter 6). For this purpose, we conducted population projections for a synthetic population resembling the Dutch population. The evaluation of the simulation output showed mainly reasonable results. Profiting from the experimentation layer of JAMES II we could evaluate the performance and suitability of the microsimulation implementation. We carried out performance experiments to find an efficient data structure to maintain event lists, and to study the behavior of different pseudo random number generators (see Chapter 7). Our experiments showed that for a continuous-time microsimulation the 'mlist' event queue is an appropriate choice. We tested the three random number generators RANDU, java random number generator, and Mersenne Twister for their suitability. To summarize, we could not find any significant differences between the results produced by the three different PRNGs. To asses the suitability of our simulation results, we re-estimated empirical transition rates, and cross-model validated the microsimulation using the MicMac macroprojection. We found that the MicMac microsimulation performs well.

A mate-matching algorithm

Despite of all its capabilities, the MicMac microsimulation features some limitations. It assumes individual life-courses to be independent, i.e., the concept of linked lives is not considered. Possible changes in the behavior of one person due to, for example, characteristics of the partner after union formation or marriage are not modelled. A related problem is the inclusion of a proper marriage market. The MicMac microsimulation does not match partners if a union formation or marriage are simulated. We tackled both problems, and extended the MicMac microsimulation accordingly: first, we proposed and implemented a technique to create synthetic couples, i.e., to match individuals. Then, we designed a microsimulation model, inclusive simulation algorithm, that involves both independent individuals and married/cohabiting pairs.

Modelling relations between spouses necessitates creating synthetic (married or cohabiting) couples, and therefore we need an effective mate-matching procedure. Two different approaches exist to handle the matching of spouses (cf. Chapter 10). In open models, an appropriate partner is created when needed. By contrast, in closed models an appropriate partner has to be found among the individuals of the model population. In terms of modelling and simulation, open models are straightforward. However, their interpretation is difficult: How can we explain the appearance of a spouse that was not present before? Closed models have a more plausible interpretation as marital matches are formed between individuals that already exist. Corresponding modelling and simulation approaches have to incorporate sophisticated matching algorithms that ensure the creation of synthetic marriages that follow a realistic pattern. In this context, two problems have to be solved: (1) when do individuals marry and (2) from a set of candidates, whom do they marry. Several suitable matching algorithms exist, however all of them apply to discrete-time models. No matching algorithm has so far been proposed for closed microsimulation models with a continuous time scale. In discrete-time models, annual marriage markets are applied to create couples. This is a natural choice as in discrete-time models events are determined to occur in a given year. In continuous-time models by contrast, where events can occur at any point in time, annual marriage markets do not fit (cf. Section 11.1).

In our microsimulation, individual life-courses are specified as sequences of waiting times to a next event. That is, we can determine in advance when individuals will experience the onset of a marriage or a cohabitation. We take advantage of this circumstance by using the waiting time until marriage or cohabitation onset for scanning potential partners.

In this thesis, we suggested to implement one marriage market that individuals can enter and leave over the complete simulation time range (cf. Chapter 11). Each “mating-minded” individual remains in the market for a specific period of mate searching and matching. In order to build up synthetic couples in the market a two-fold stochastic approach is used. First, we assign to each individual a random value that captures his/ her aspiration level regarding a partner. An empirical likelihood equation reveals the probability that a given woman and a given man would mate. (Relying on the theory of assortative mating, pairs with similar attributes, like similar ages and educational levels, have a higher probability of mating.) Subsequently, we simulate a decision making process whether two individuals date each other applying individuals’ aspiration levels and their mating probability. A couple is formed if a positive decision has been made and the timing of the couple’s marriage events is consistent regarding their individual searching periods. Individuals that were inspected (but rejected) lower their level of aspiration accordingly.

In order to illustrate the capability of the presented algorithm, we extended the MicMac microsimulation tool. We run simulations to forecast a synthetic population based on the population of the Netherlands using microdata from Statistics Netherlands and the Netherlands Kinship Panel Study. We found that the proposed algorithm produces acceptable result; e.g., it is capable to reproduce the observed peak of the frequency distribution of age differences of spouses.

Microsimulation and interaction between two individuals

To realistically describe couple dynamics, matching individuals solely does not suffice. Also, the effect of inter-individual interaction has to be considered. In a microsimulation, life-course events are commonly determined based on empirical distribution functions which rely on data reported for single persons. Individual interaction patterns are usually ignored in this context. No general modelling or simulation technique exists that accounts for correlation between linked lives in a microsimulation. To give an example: many microsimulation models incorporate a marriage market to perform mate-matching. After a couple has been matched, however, the subsequent modelling and simulation of the individuals forming the couple is not clear: in the literature neither a stochastic model of the joint behavior of paired individuals is described, nor related simulation algorithms are discussed. In this thesis we developed a model that, when describing two linked life-courses, maintains the concept of modelling individual life courses using stochastic processes. Couple dynamics are described by considering couples as units. Like individual life-courses, couple behavior are specified by using continuous-time multi-state models. That is, we use continuous-time Markov chains or semi-Markov processes to describe the 'life-courses' of couples (cf. Section 14.1).

Using a continuous-time multi-state approach for modelling couple dynamics may imply very large state spaces that are associated with a corresponding need for transition rates. Data sets that are rich enough to estimate such rates often do not exist. To anyhow describe couple behavior models of individual behavior can be interlinked such that a realistic model of couple behavior results. With shared frailty model and copula models we presented two very general approaches that facilitate such modelling (cf. Section 14.2).

In the first part of this thesis, we presented a microsimulation model that neglects inter-individual interaction. For its specification, we proposed using the DEVS formalism (see Chapter 4). In order to circumvent shortcomings (discussed in Chapter 15) we developed a ML-DEVS model to specify the extended microsimulation model that regards partnership dynamics (cf. Section 15.1). The ML-DEVS formalism extends the classical DEVS model language by explicitly supporting a multi-level setup. In ML-DEVS, a macro level model is specified by a coupled DEVS model, comprising an own state and behavioral rules, and micro models are specified by ordinary atomic DEVS models. Like DYNDEVS, the ML-DEVS formalism supports variable structure models, and therefore it suits well to specify population dynamics. To describe a microsimulation model that considers couple dynamics, we set up a ML-DEVS model that consists of a macro-DEVS model comprising two types of micro components: individuals and pairs. These micro components handle the life-course dynamics of unlinked individuals and the dynamics of married or cohabiting couples. The macro-DEVS model guides the onset of partnerships (marriages or cohabitations), i.e., it performs mate matching and instructs the creation of pair models. If a pair model signalizes an emigration, a dissolution, or a widowhood event, the macro-DEVS model instructs its extinction or dissolution.

We presented the simulation semantics that have originally been developed for the execution of ML-DEVS models (cf. Section 15.2). To execute the original simulation algorithm we would, however, need one process or one thread per micro-DEVS model and one process or one thread per macro-DEVS model. That is, like in the case of DYNPDEVS, a simulation based on the original semantics does not allow meaningful population projections (cf. Section 15.2.2). To overcome this limitation, we designed simulation semantics that facilitates an efficient execution of the population ML-DEVS model. The developed simulation semantics extends one of the sequential simulation algorithms initially worked out for PDEVS.

We conducted a small case study to illustrate the capabilities of the novel microsimulation model that accounts for partnership dynamics (cf. Chapter 16). For this purpose, we studied a synthetic population that (roughly) resembles the Dutch population. We looked at fertility, partnership, and smoking behavior; in particular we analyzed how partners influence each other's smoking behavior. The conducted application indicates that the ML-DEVS population model works fine.

17.2. Critical assessment

We have designed a demographic microsimulation to efficiently conduct realistic population projections on a detailed level. However, the methodology still shows limitations. In this section, we detail ideas for improvement and point to directions of research leading to more realistic population projection models.

Data requirements of microsimulations

A main impediment for the usage microsimulations is their demand for data. To run meaningful microsimulation, applications a lot of data have to be provided. In social science, a lack of suitable data often hinders the usage of micro models (Huinink and Feldhaus, 2009). Nonetheless, at present researchers observe improvements concerning data accessibility. Considerable advancements concerning data availability in the field of life-course sociology were noted during the last years by Mayer (2009). He emphasizes the buildup of national individual-level longitudinal databases. Furthermore, Mayer finds a rising amount of data sources that are concerned with the effects of the social environment on life-courses. This points to an improvement of the data situation and therefore to an increased applicability of micro models.

Drawbacks and opportunities of our implementation approach

Still many simulation applications are developed from scratch, which causes various problems (Edmonds and Hales, 2003; Galan and Izquierdo, 2005; Hannay et al., 2009; Merali, 2010; Minar et al., 1996), among others:

- missing transparency and documentation,

- no repeatability of results,
- high development costs because of redevelopments,
- no product tests or validation for lack of time at the end of a project.

In this thesis, we tried to avoid these problems by (1) using an established formalism for discrete event simulation to specify our microsimulation model and (2) by using the M&S framework JAMES II to realize it. This way, we could profit from up-to-date M&S methodology, make use of already existing M&S tools, and validate our simulation product.

Using the DEVS metaphor to specify a microsimulation system is bears some problems. Although DEVS suits well to specify population dynamics in the intended way, it introduces some modelling overhead. The DEVS formalism is very rigid. This results in bulky model specifications; see e.g., the longish definition of the ML-DEVS population model in Section 15.1. An alternative to DEVS are process algebras like the stochastic pi to specify population models. The stochastic pi is an individual-based approach that supports dynamic composition and interaction pattern (John et al., 2010). It is originated in systems biology and has been developed for spatial modelling of cell-biological processes. Its basic approach is to let individual entities move according to stochastic processes and to let them interact by communicating over ‘channels’. Stochastic pi is a neat and very intuitive approach, and in theory, we deem it appropriate to readily model life-course dynamics. However, currently the stochastic pi formalism does only support homogeneous Markov chains to describe system behavior. Although an extension to other classes of stochastic processes is work in progress.

Improving mate-matching

The matching strategy we presented can be improved: At the moment, we match individuals based on actual preference patterns. However, preferences concerning the characteristics of a partner might change over calendar time, and such changes should be considered in the mate-matching process.

Furthermore, in the current model, individuals enter the partnership market based on empirical marriage rates or based on rates indicating cohabitation propensities. To correctly determine partnership events though, instead of marriage or cohabitation rates, we would have to use rates indicating the willingness to mate. However, we know of no data source that allows estimating such rates. A way to anyhow obtain those rates would be to hypothesize them based on external knowledge of the phenomenon.

Networks of linked individuals

So far, the novel microsimulation with linked lives only devises married couples and consensual unions. An extension to other kinds of bidirectional relationships seems to be straightforward. For example, in the presented way we can also model mother-child relationships. The situation differs when more than two individuals are involved, and non-hierarchical interaction patterns should be specified. The only rough idea that, to our knowledge, has been mentioned in the literature in this context is to consider linked individuals as being part of a "larger simulation unit that contains all individual units that are mutually dependent" (Galler, 1997, p. 14). However, how this should be realized has not been elaborated yet.

To describe household formation and dissolution, Galler (1988) and Courgeau (1997) realize microsimulation models where individuals are linked to superior household entities. At a first glance, these models seem to provide the methodology that is necessary to formally specify interaction between individuals and groups of individuals. Unfortunately, both models have been designed only for specifical data structures, and as individuals are per definition part of exactly one household, also a hierarchical structure is posed on the model population. As a consequence, a generalization of the approaches of Galler (1988) and Courgeau (1997) to account for various types of inter-individual interaction is not apparent.

Micro, meso, macro - multi-level simulations

The problem of accounting for interaction among individuals in a micro model is closely related to the problem of describing the dynamics of societal groupings. Naturally, a microsimulation works on the micro level of a population. However, as soon as a micro model accounts for interaction and linkage patterns between individuals also a group level becomes emergent, the meso level. Here interacting individuals are grouped according to the type of interaction, e.g, into households, families, firms, or even party membership. Generally, complex dynamic systems, like populations, can be specified at three different structural levels: micro, meso, and macro.[1] The latter gives the conditions under which the target system is considered: If a human population is studied, macro conditions are, e.g., the political and the economic system, distinct policies, the taxation and benefit system, the environmental conditions etc.

Demographic microsimulation currently mostly models the effects of *stable* macro conditions on individual life course decisions. The *dynamics* of macro structures, like a changing political system, are not modelled - usually, the macro-level is measured as aggregates of individual attributes (Gilbert and Troitzsch, 2008). A feedback from the individual behavior on the macro condition is hence not foreseen. Many real world examples, however, show that individual behavior has a strong influence on the macro level, e.g., the growing environmental awareness of the people in the Western World world has a strong effect on political decisions. Modelling population dynamics but neglecting upward causation from the individual level to the macro level is hence very unsatisfactory.

In conjunction with microsimulations, the concept of meso levels has de facto never been described as such. Two versions of microsimulation models exist that support the formation of groups of individuals - which might be interpreted as a kind of meso level realization. First, as already mentioned, the consideration of demographic kinship leads to the formation

---

[1] In the context of population projections the notions 'micro' and 'macro' are not only used to describe the structural levels at which a system is specified. They are also used to specify at which aggregation level population forecasts are produced.

of families. Then, in spatial microsimulations the target population is not only specified on the individual level, but also according to the regions where individuals live. These regions are usually not modelled as self-contained entities, but as attributes of individuals, and even if they are modelled as self-contained entities, their dynamics is restricted to the in- and out-migration of individuals (Holm et al., 2002). In summary, creating family ties and discriminating individuals according to regions of residence can at the utmost be seen as realizing a slimmed down version of a meso level in a multi-level system.

In order to make further step towards a multi-level approach, a set of challenges has to be met to specify and implement a demographic microsimulation that allows interaction among individuals, and between individuals and their environment. This includes the development of a stochastic model to accurately capture population dynamics, an effective implementation (preferably using an already existing simulation framework), and finally the provision of a software that, without too much effort, can be used by others, particularly by demographers.

Appendices

Derivation of distribution function of waiting times

If the transition rates of a Markovian process are known, the distribution function of the waiting times in the distinct states of the state space can be derived. Subsequently, we detail the derivation of the distribution function of the waiting times of a non-homogeneous continuous-time Markov Chain. We explain the related processing by considering for simplicity reasons in a first step only a homogeneous setting and the transition between two states. (Subscripts related to states are omitted because of clarity.) We denote the (probability) density of experiencing a transition after a duration w by $f(w)$, the corresponding waiting time in the current state is $S(w)$, and $\lambda(w)$ the associated transition rate. It is

$$\begin{aligned}
\lambda(w) &= \frac{f(w)}{S(w)} \;\Rightarrow\; f(w) = \lambda(w)S(w) \\
\lambda(w) &= [-lnS(w)]' \\
S(w) &= \exp\{-\int_0^w \lambda(\nu)d\nu\} = \exp\{-\Lambda(w)\},
\end{aligned}$$

where the integral $\int_0^w \lambda(\nu)d\nu$ is called cumulative (integrated) hazard rate $\Lambda(w)$. Consequently, the (cumulative) distribution function of leaving a current state can be written as

$$F(w) = 1 - \exp\{-\Lambda(w)\}.$$

From this derivation the distribution function of the waiting times of a non-homogeneous continuous-time Markov Chain can be deduced in a straightforward manner. We denote by $F(w,t)$ the corresponding distribution function that depends on the time t of the last transition and, via the time s of the next transition, on the waiting time w spent in the current state ($w = s - t$). It is

$$F(w,t) = 1 - \exp\{-\Lambda(w,t)\},$$

where

$$\Lambda(w,t) = \int_0^w \lambda\,(t + \nu)\;\; d\nu.$$

Equivalently, if we account for the age and calendar time dependence of the process, we get

$$F(w,c,a) = 1 - \exp\{-\Lambda(w,c,a)\},$$

where

$$\Lambda(w, c, a) = \int_0^w \lambda\left(c + \nu, a + \nu\right)\ d\nu,$$

and $\lambda(\cdot)$ describes the corresponding set of transition rates.

Simulation study: Two alternative algorithms to construct life-courses

In Section 2.3.1 we state that, starting from an initial state, life-course events can be determined by generating random waiting times for all possible destination states and then choosing the shortest among them to determine the actual destination state. An alternative method to simulate life-course events, which is laid out in Wolf (1986), is to first determine the waiting time in the current state, and then to specify the destination state. To illustrate the equivalence of both methods we conduct a simple simulation study using the so called disability model with reactivation (Hougaard, 2000, Section 5.3.2). The state space of this model comprises three states: (1) healthy, (2) disabled, and (3) dead. Individual can undergo transitions from healthy to disabled, from healthy to dead, from disabled to healthy, and from disabled to dead. The transition intensities that describe the propensities of individuals to undergo transitions are assumed to be independent of calendar time, but to depend on age. That is, we use a non-homogeneous continuous-time Markov chain to describe individual life-courses. Figure A-1 depicts the transition intensities are employed in this simulation study. We assume that they are piecewise constant.[2]

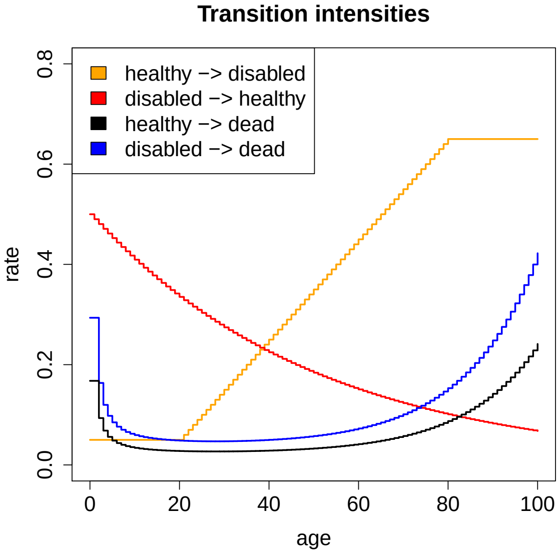


Figure A-1.: Transition intensities employed in this simulation study.

---

[2] The transition intensities we use in this example are synthetic, however, resemble observed transition patterns. Thanks to Ekaterina Ogurtsova who has prepared these data.

To ensure that the results obtained by the two distinct simulation algorithms (the one that we present in Section 2.3.1 and the method suggested by Wolf (1986)) are comparable, we integrate both methods into the same framework. For this purpose, we still use JAMES II and the microsimulation implementation described in Chapter 5. Here, we only replace simulation algorithms.

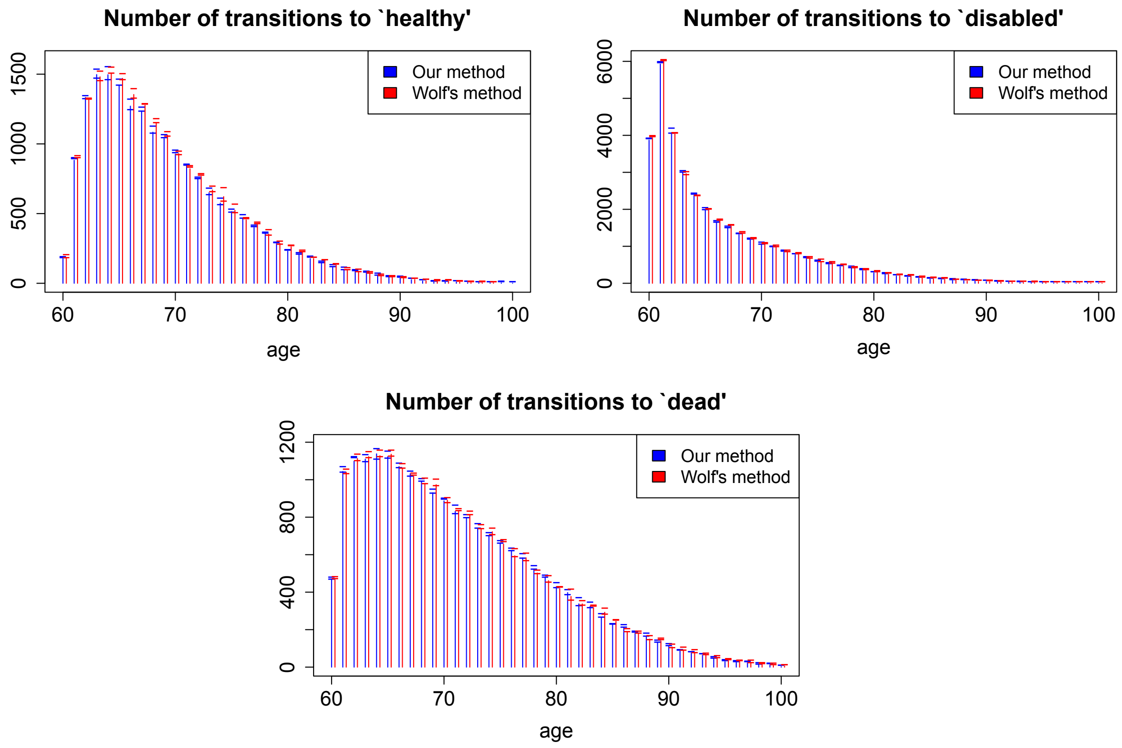


Figure A-2.: Number of transitions according to destination state and age.

The simulation starts at time $c = 0$ and ends after 40 years. We determine the life-courses of 20,000 individuals who are in state (1) at model initialization, and are of age $a = 60$. To account for sampling variation, we conduct five simulation runs per method. For the evaluation of the simulation results we use R. Figure A-2 shows a comparison of the number of transitions generated by the two different methods. Each vertical line maps the mean number of transitions (out of five runs) for each age. The horizontal lines at the top of each vertical line mark the second and the third quartile of the empirical distribution, i.e., they indicate the variability of the studied distribution. Numbers are given according to destination state and age. Figure A-3 depicts a comparison of the mean distributions of the simulated event times. Also here the little horizontal lines show the variability between the different runs. For both methods, we find almost identical results. In conclusion, our evaluation demonstrates that the method that we present to simulate life-course events and the method that Wolf (1986) suggests are equivalent. A comparison of run times does not reveal any significant differences: running the simulation five times, we find run times ranging between 12.99 and 15.65 seconds employing Wolf's method and run times ranging between 13.66 and 17.36 seconds employing our method.

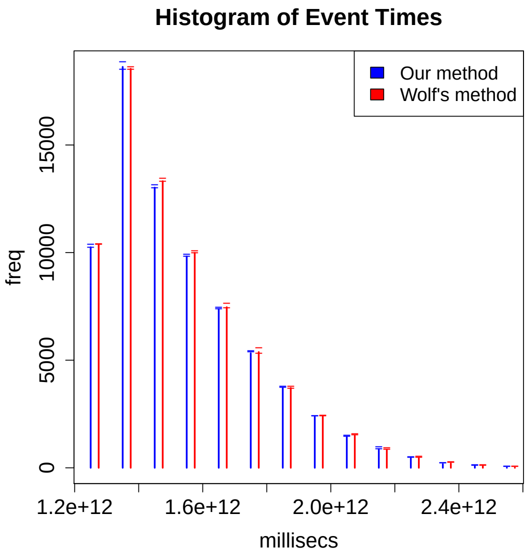


Figure A-3.: Comparison of simulated event time distributions.

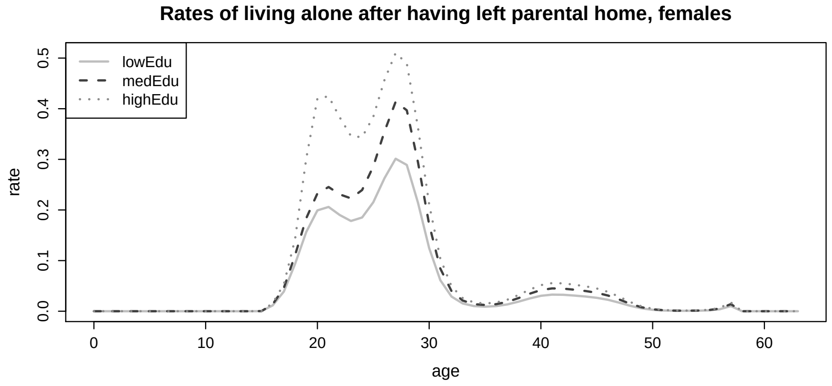
Transition rates for Chapter 6: Application to the Netherlands

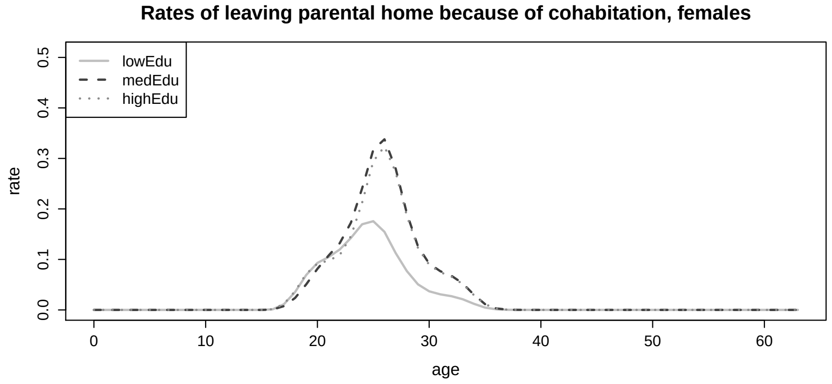
In Chapter 6 (see page 55ff.) we describe an application for a population which closely resembles the contemporary Netherlands. In this appendix we present the age-profiles of the transition rates used in this application.

Transition rates related to changes in living arrangement and marital status in Figures A-4, A-5, A-6, A-7, A-8 (for females), and also in Figures A-11, A-12, and A-13 (for males). Figure A-9 and Figure A-10 top show the birth rates of females. while we assume that during simulation transition rates for changing the marital and parity vary with age, but are constant over calendar time. Figure A-10 bottom shows the transition rates of females to a higher level of educational attainment, while Figure A-13 bottom shows the corresponding rates for males. Furthermore, the following assumptions were made to simplify the model and to reduce the number of required transition rates.

- For both sexes the rates to leave the parental home do not depend on the presence of children.
- Additionally, for males the transition rates to leave the parental home because of marriage are independent of their educational attainment.
- Divorce rates are independent of the marriage order.
- Females move into a cohabitation after living alone, once they have left parental home, independent of the presence of children.
- For both sexes the transition rates from cohabitation to marriage do not depend on the partnership order.
- The propensity to move into cohabitation after a previous partnership is independent of the type of the former partnership (marriage or cohabitation).
- For both sexes the risk to split up is independent of the partnership order and the presence of children,
- Rates that describe a transition to parity three or more do not depend on the marital status or the living arrangement of the woman.

- The probability to change the educational attainment is independent of marital status and living arrangement, and also of the presence of children.





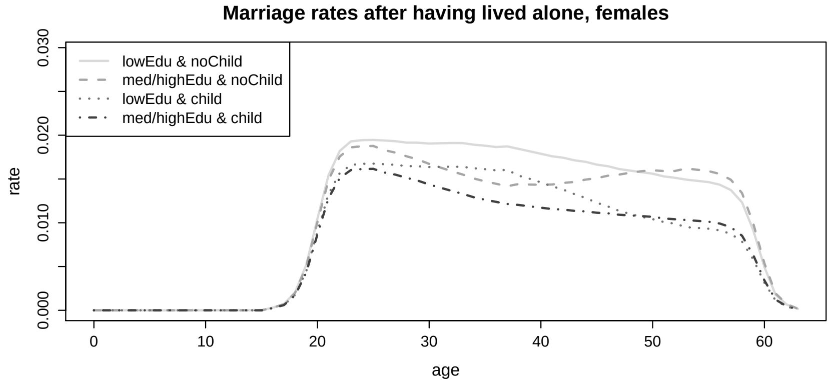
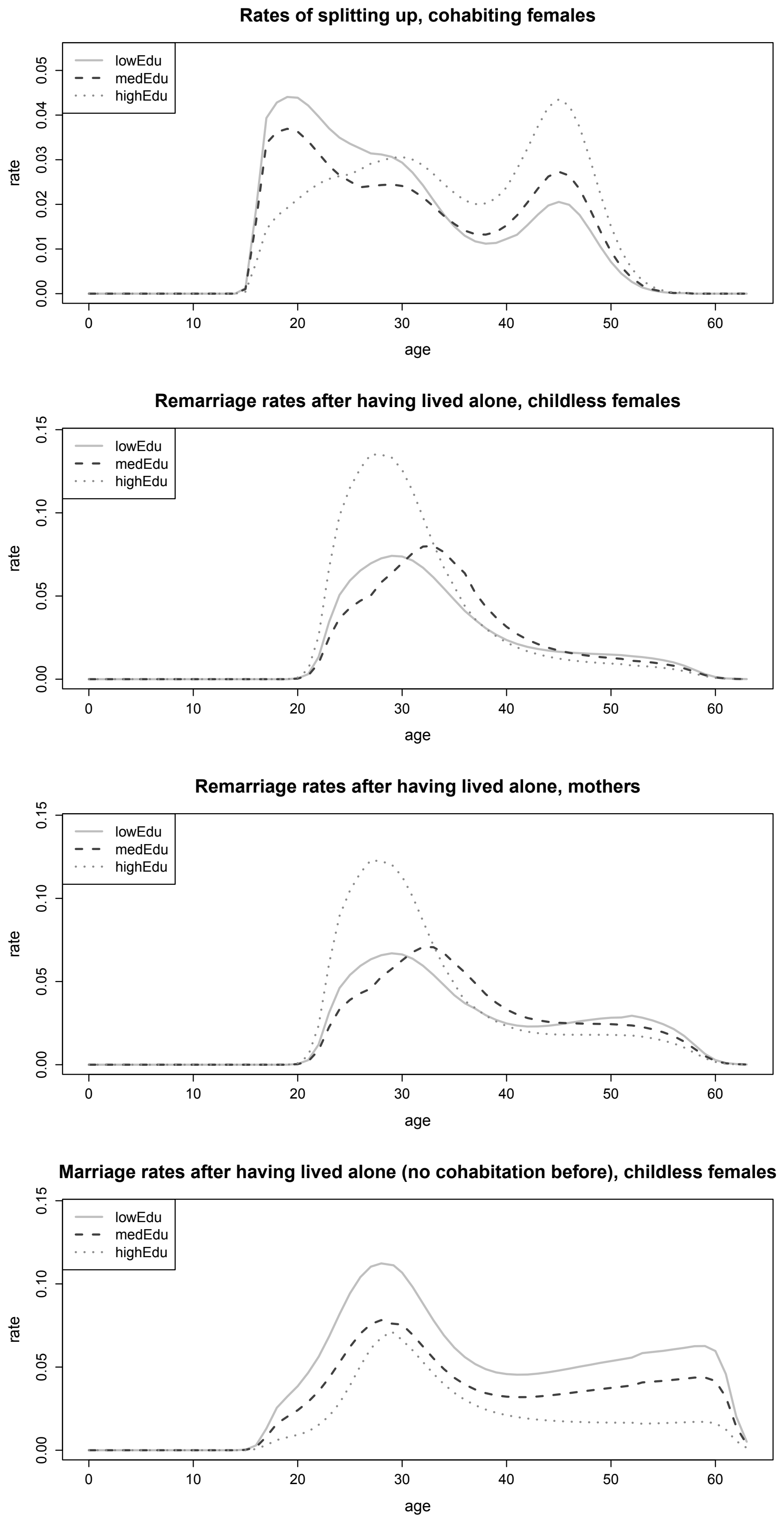


Figure A-4.: Transition rates of females who leave parental home and marriage rates of females by educational attainment; lowEdu: only primary school, medEdu: lower secondary school, highEdu: upper secondary or tertiary education.



Rates of splitting up, cohabiting females

Remarriage rates after having lived alone, childless females

Remarriage rates after having lived alone, mothers

Marriage rates after having lived alone (no cohabitation before), childless females

Figure A-5.: Rates of splitting up, marriage and remarriage rates of females; lowEdu: only primary school, medEdu: lower secondary school, highEdu: upper secondary or tertiary education.

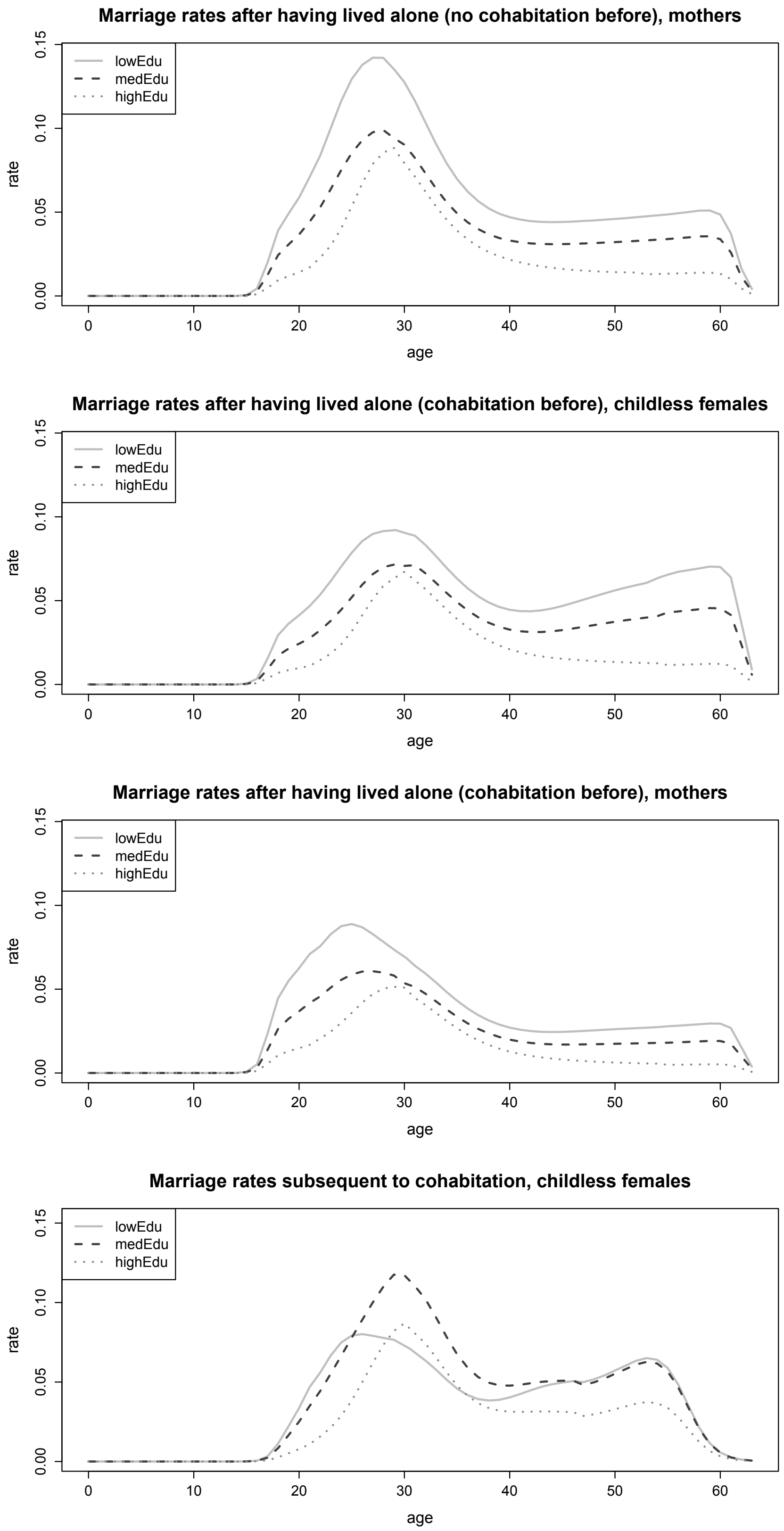
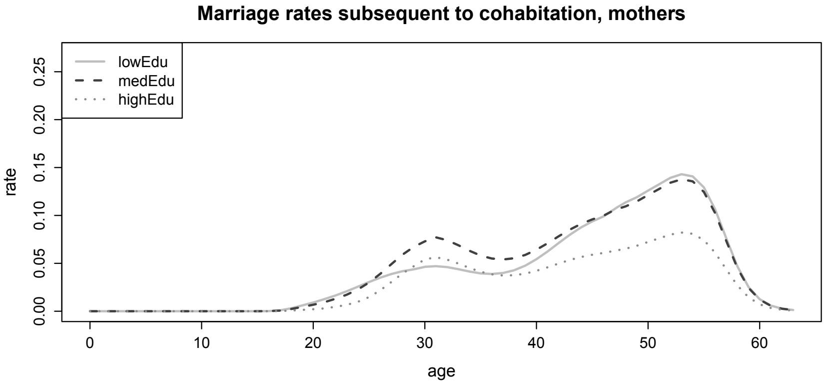
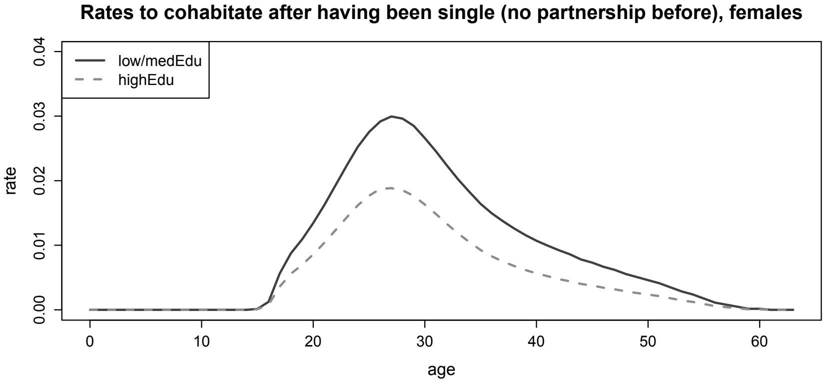
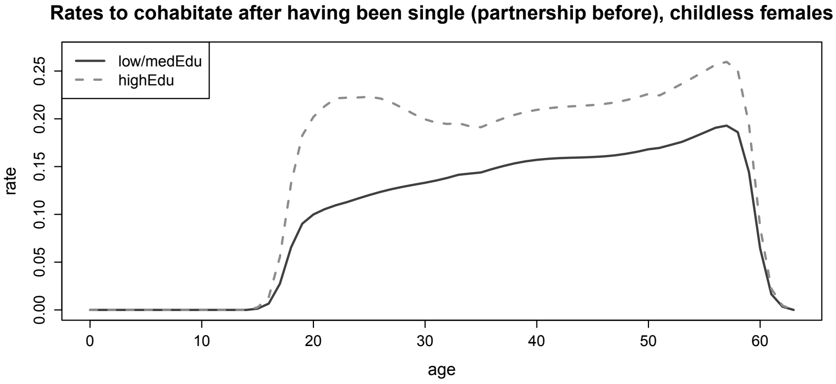


Figure A-6.: Marriage rates of females by educational attainment; lowEdu: only primary school, medEdu: lower secondary school, highEdu: upper secondary or tertiary education.


Marriage rates subsequent to cohabitation, mothers


Rates to cohabitate after having been single (no partnership before), females


Rates to cohabitate after having been single (partnership before), childless females

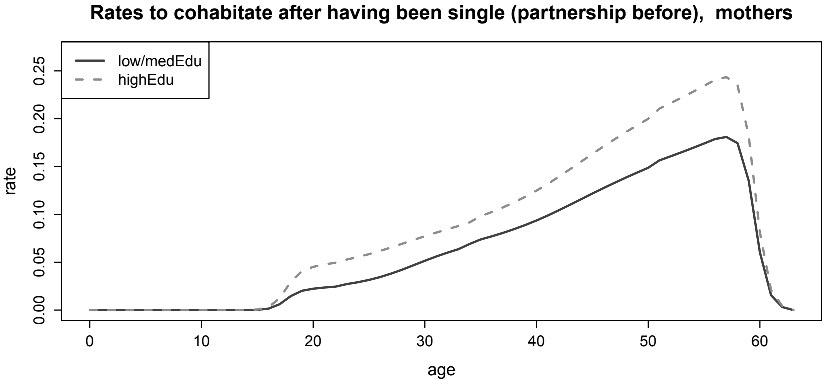

Rates to cohabitate after having been single (partnership before), mothers

Figure A-7.: Marriage rates of females by educational attainment (part 2); lowEdu: only primary school, medEdu: lower secondary school, highEdu: upper secondary or tertiary education.

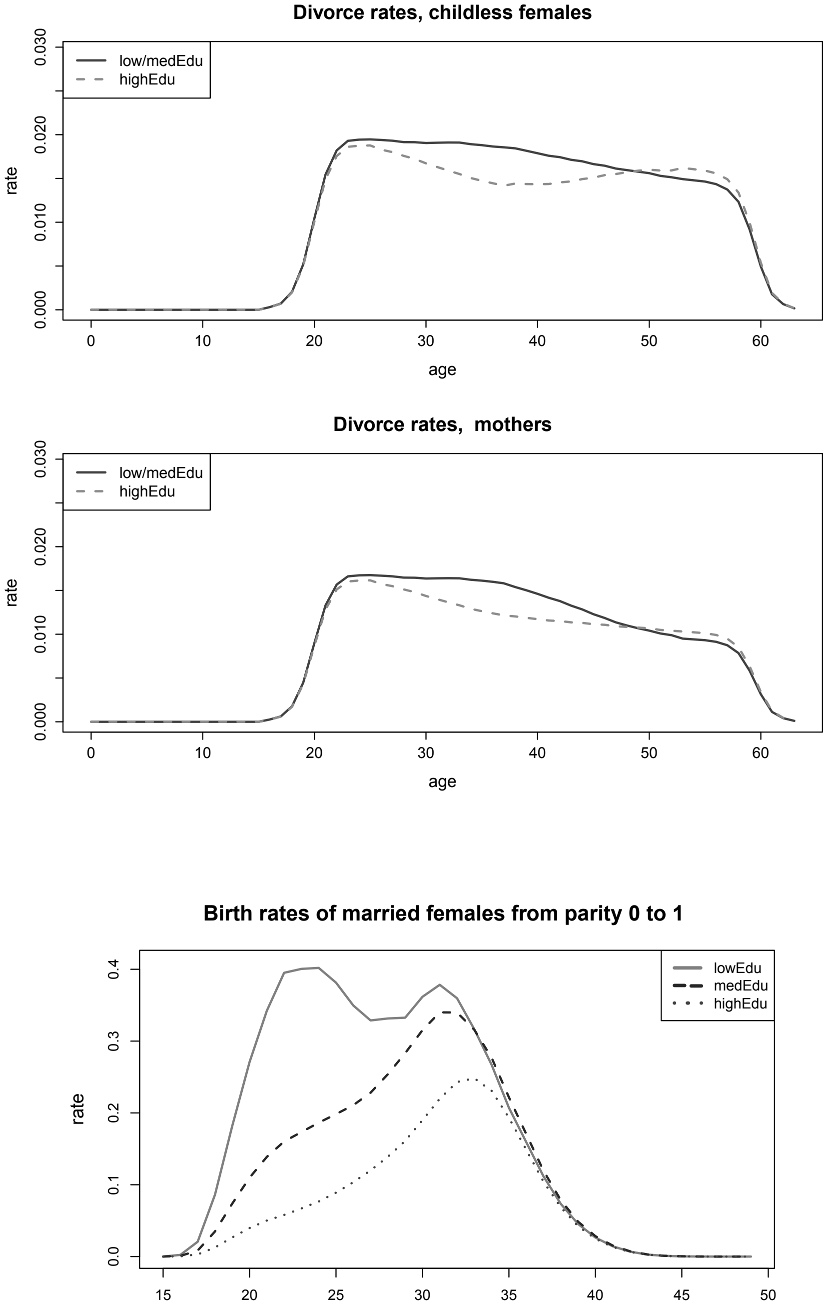
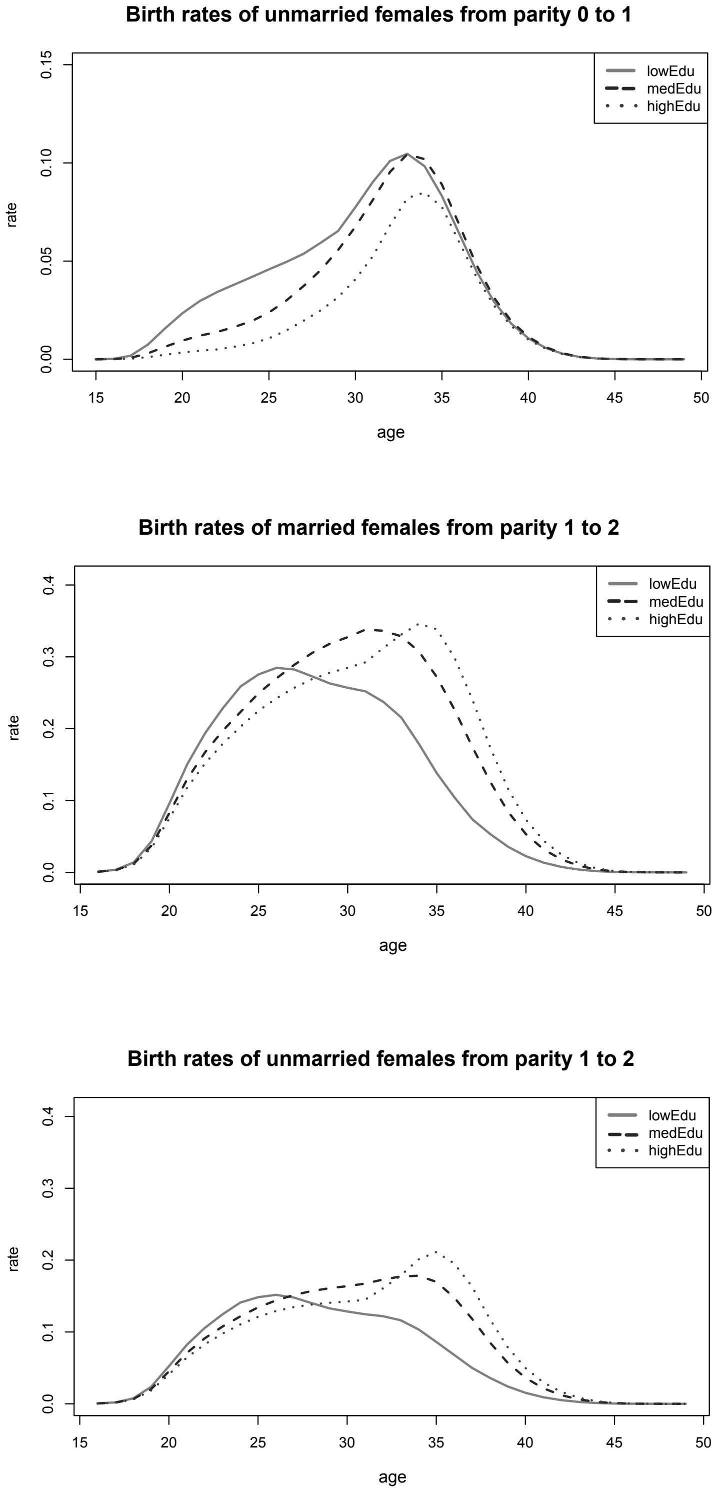


Figure A-8.: Divorce rates of females, and birth rates; coding: lowEdu (only primary school), medEdu (lower secondary school), highEdu (upper secondary or tertiary education).



Birth rates of unmarried females from parity 0 to 1

Birth rates of married females from parity 1 to 2

Birth rates of unmarried females from parity 1 to 2

Figure A-9.: Birth rates by educational attainment; lowEdu: only primary school, medEdu: lower secondary school, highEdu: upper secondary or tertiary education.

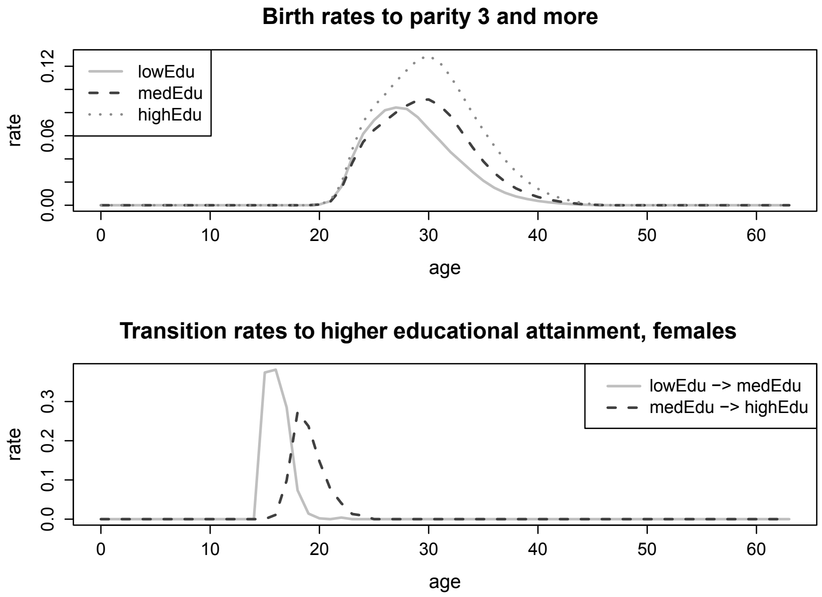


Figure A-10.: Birth rates and transition rates to higher educational attainment, by educational attainment; lowEdu: only primary school, medEdu: lower secondary school, highEdu: upper secondary or tertiary education.

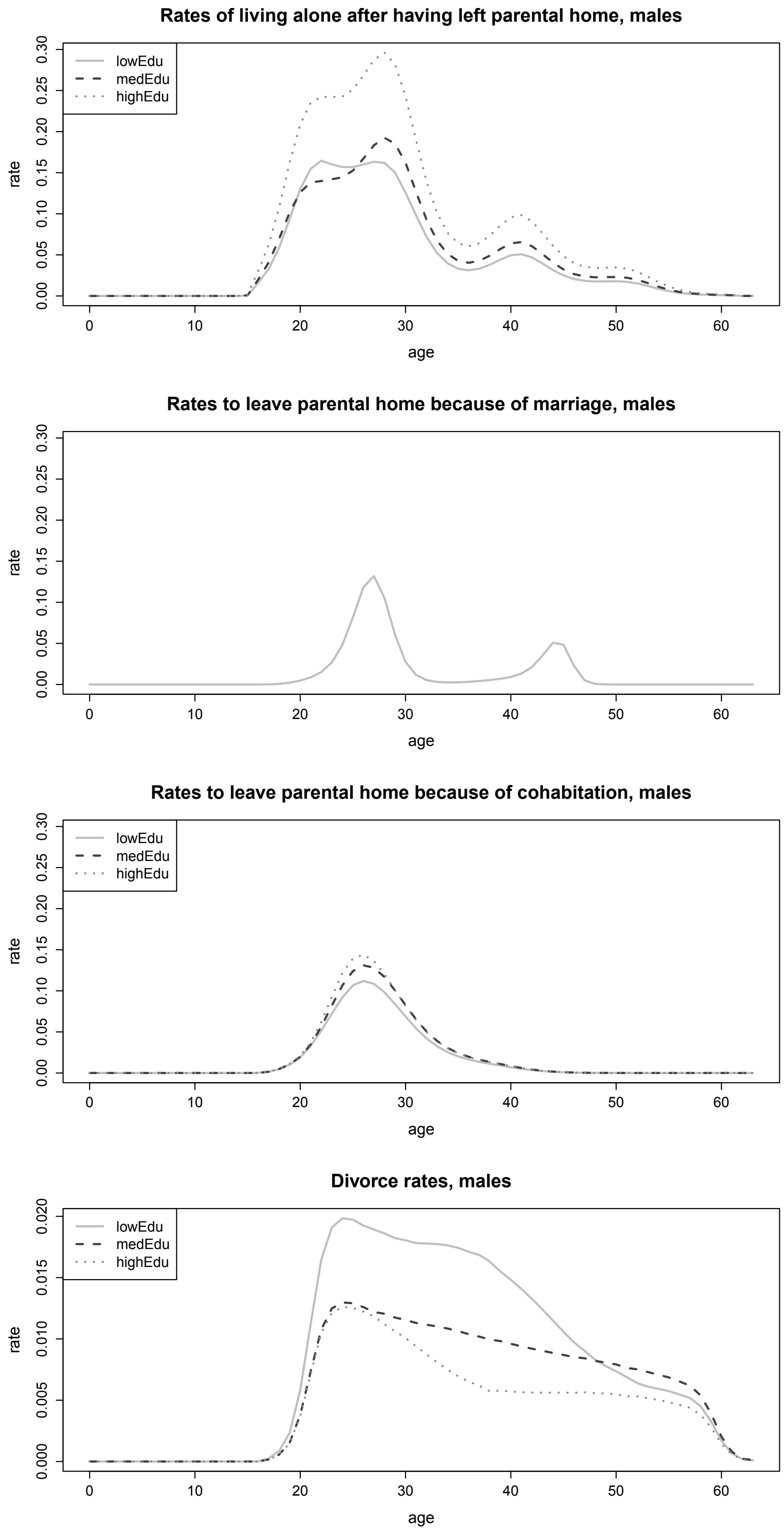


Figure A-11.: Rates to leave parental home and divorce rates of males, by educational attainment; lowEdu: only primary school, medEdu: lower secondary school, highEdu: upper secondary or tertiary education.

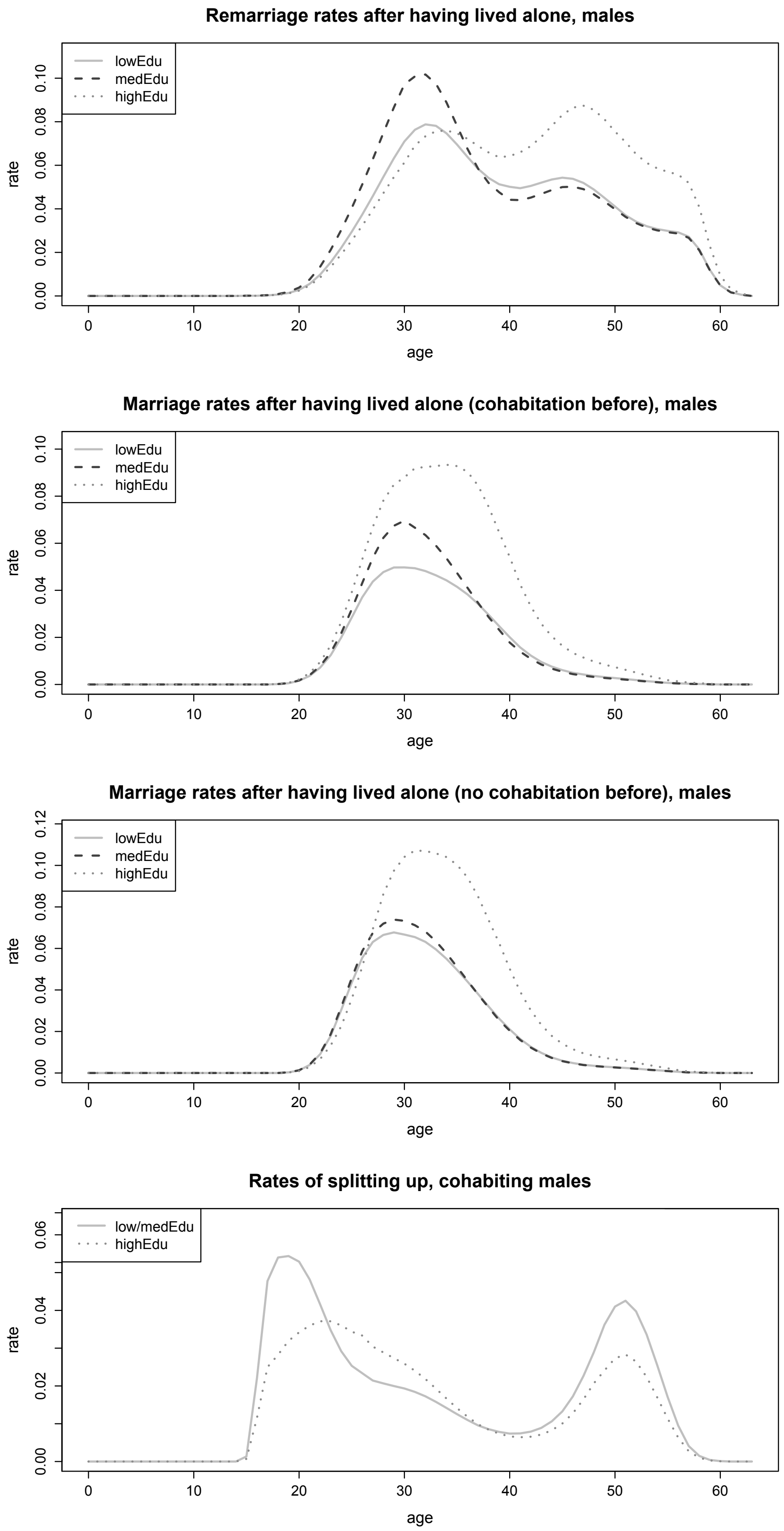


Figure A-12.: Marriage and remarriage rates of males, and rates of splitting up; lowEdu: only primary school, medEdu: lower secondary school, highEdu: upper secondary or tertiary education.

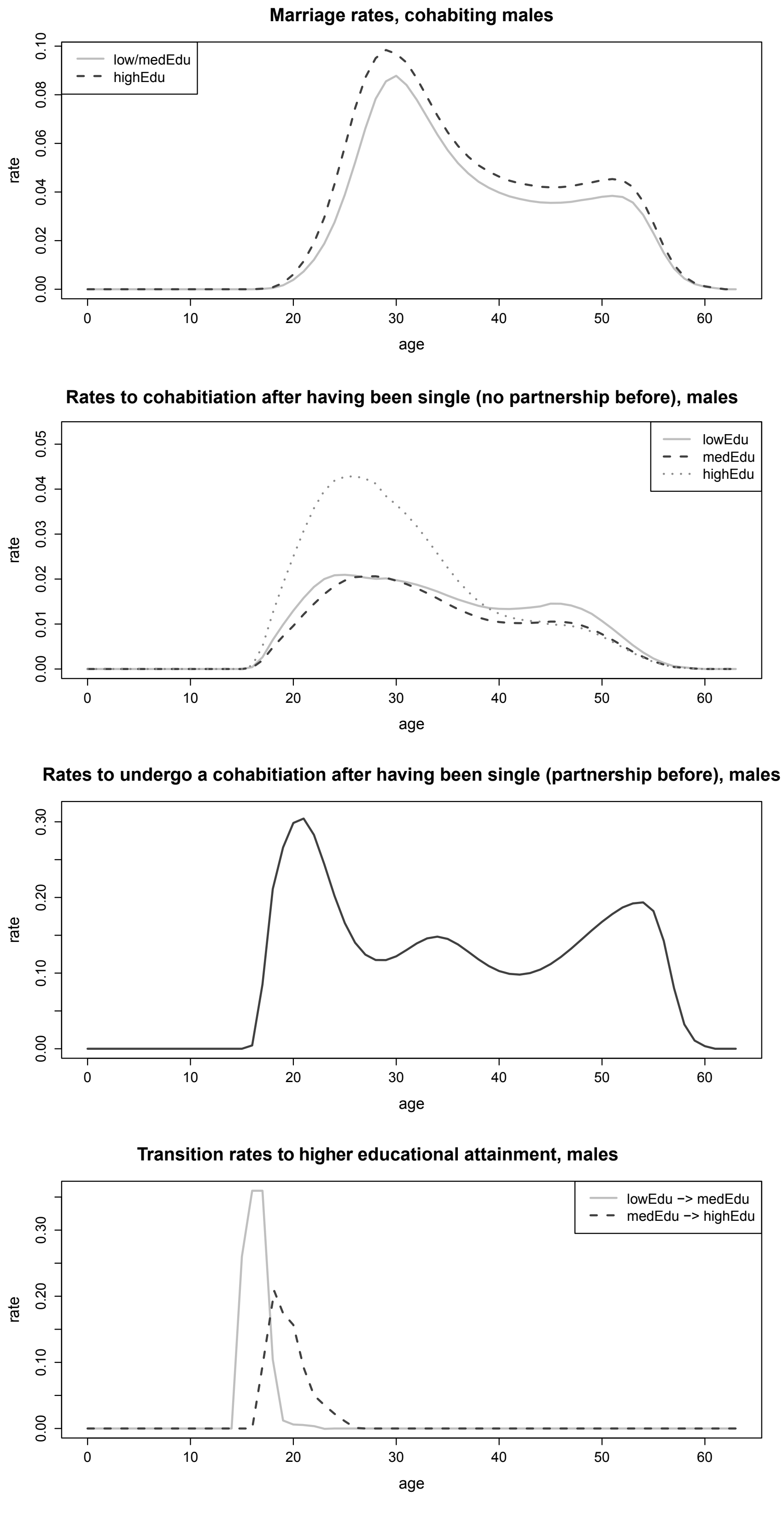


Figure A-13.: Marriage rates of males and rates of males to cohabitation (by educational attainment), as well as transition rates to higher educational attainment; lowEdu: only primary school, medEdu: lower secondary school, highEdu: upper secondary or tertiary education.

Further results of the microsimulation application presented in Chapter 6

In this appendix we show some additional results of the data example elaborated in Chapter (see page 55ff).

Figure A-14 depicts the frequency distribution of the marital status/living arrangement of females and males, at the beginning of each year during the period from 2010 to 2015. As we have assumed that the propensities to change the marital status/living arrangement do not vary over calendar time, the distribution of the studied state occupancies does not remarkably change over time. We further find that the distribution looks quite similar for men and women.

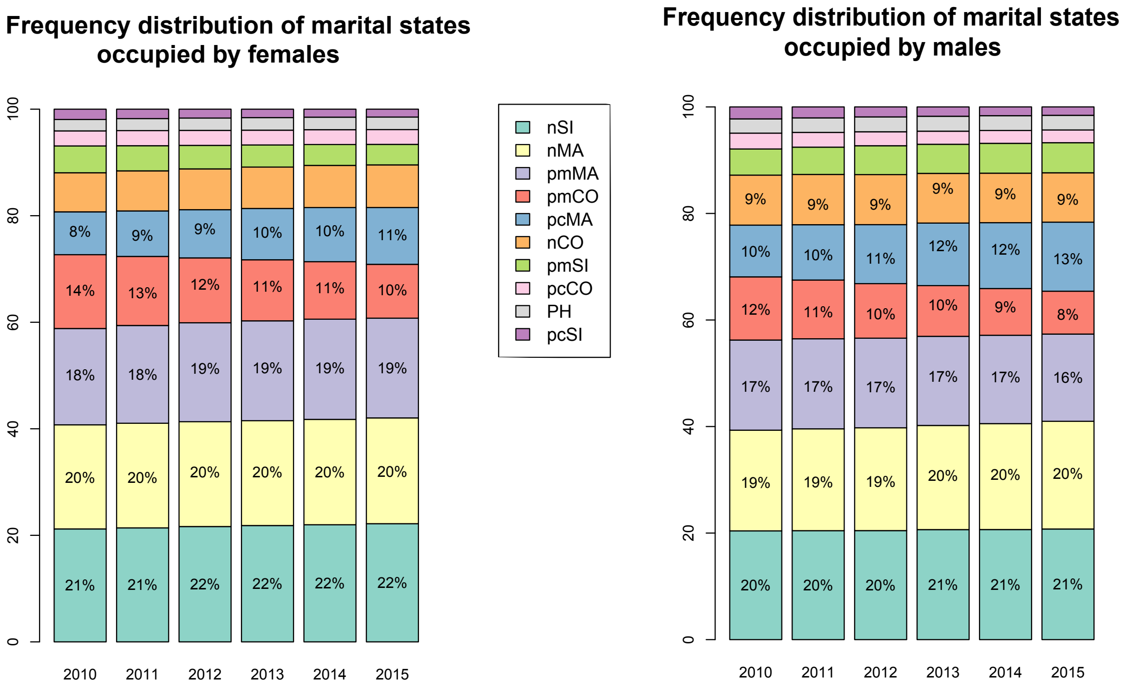


Figure A-14.: Frequency distribution of marital status/living arrangement of females and males, at the beginning of each year from 2010 to 2015.

Analyzing the age distribution of first remarriages over time shows that remarkably more men than women remarry, namely 2108 men as opposed to 1875 women. Figure A-15 shows the frequency distribution of first remarriage events for females and males over age and time.

Comparing the numbers for females and males reveals that the age distribution of women who remarry is wider than the corresponding age distribution of men. As we have assumed time-constant remarriage rates, we do not find pronounced changes in the pattern of first remarriages over time.

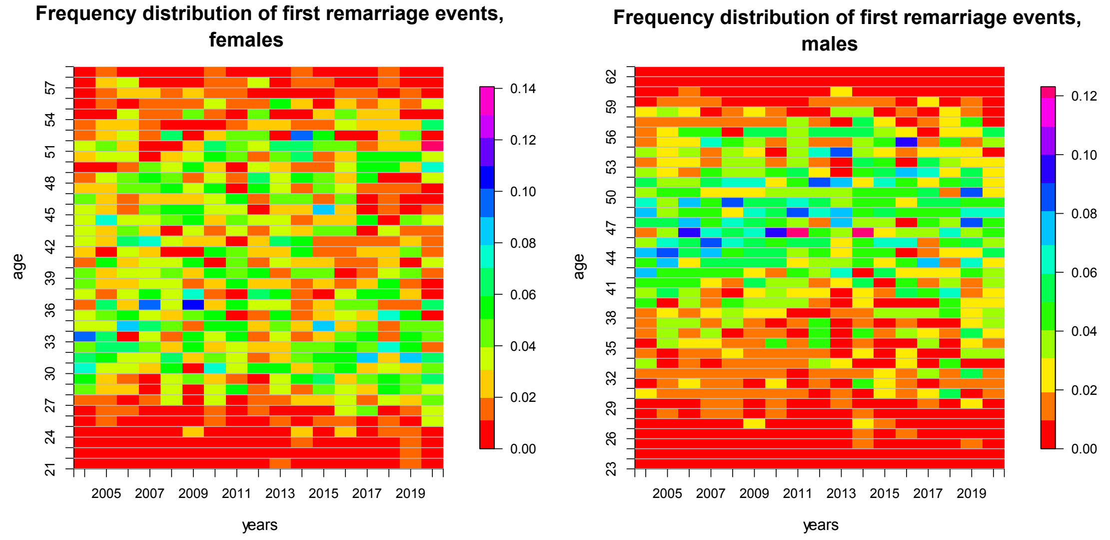


Figure A-15.: Frequency distribution of first remarriage events ('pmSI → pmMA') for females and males.

Figure A-16 shows the waiting time distribution between the first and second birth, by educational attainment of the mother.

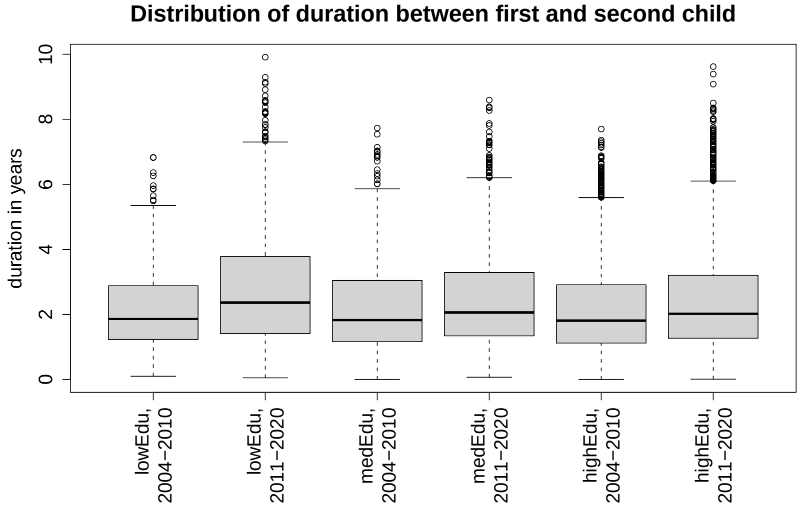


Figure A-16.: The distribution of the waiting time between first and second birth by educational attainment of the mother. We contrast two consecutive periods from 2004 to 2010 and from 2011 to 2020, respectively.

The educational level of the mother has no strong impact on the waiting between first and second birth. Comparing the two periods 2004-2010 and 2011-2020, however, reveals a wider waiting time distribution for the second period. One reason for this effect might be the wider simulation horizon, namely ten instead of seven years.

For the age range from 0 to 63 that was considered the age-at-death distribution remains relatively stable over calendar time, cf. Figure A-17. We see that females generally get slightly older than males. Notwithstanding, mortality differences are more pronounced at older ages, however, and these are not included in the simulation example presented here.

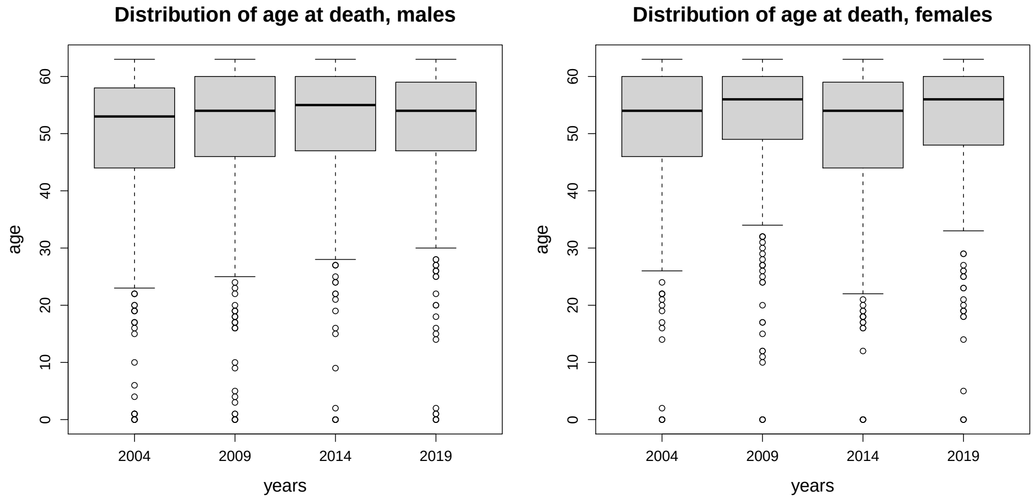


Figure A-17.: Age-at-death distribution of females and males, given in five year intervals.

Transition rates for Chapter 16: Application with linked lives

In Chapter 16 (see page 159ff.) we described an illustration of the novel microsimulation model that accounts for partnership dynamics In this appendix we present the age-profiles of the transition rates used in this application.

The transition rates that describe the propensity of unlinked females and males to change the marital status are depicted in Figures A-18 and A-19. Transition rates of unlinked individuals to change their smoking behavior and (non-parity specific) fertility rates are depicted in Figure A-20. Figure A-21 top shows mortality rates of female and male non-smokers and smokers, Figure A-21 bottom shows the transition rates of spouses to change their smoking behavior. The transition rates of couples to change marital status are given in Figure A-21. Table A-1 contains the coefficients of the logit models which we have estimated to quantify the compatibility between two potential spouses.

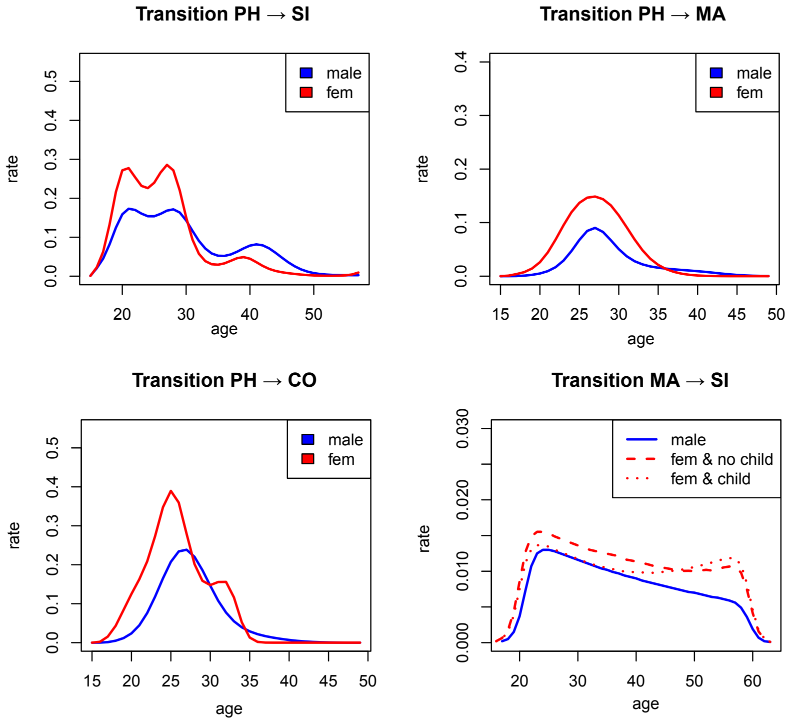


Figure A-18.: Transition rates of of unlinked females and males to change the marital status; PH: living at parental home, SI: being single, MA: married, CO: cohabiting.

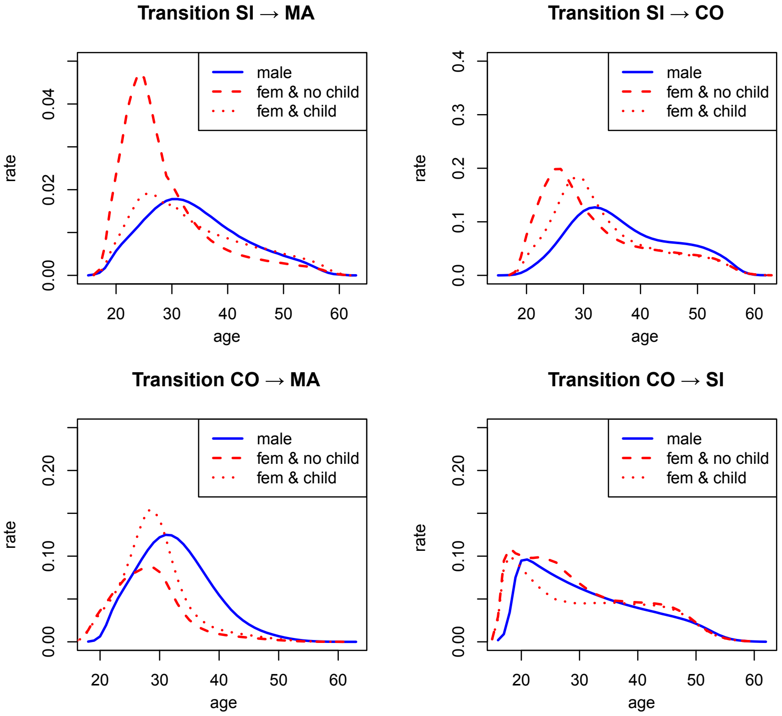


Figure A-19.: Transition rates of of unlinked females and males to change the marital status; S: smoker, nS: non-smoker, CO: cohabiting.

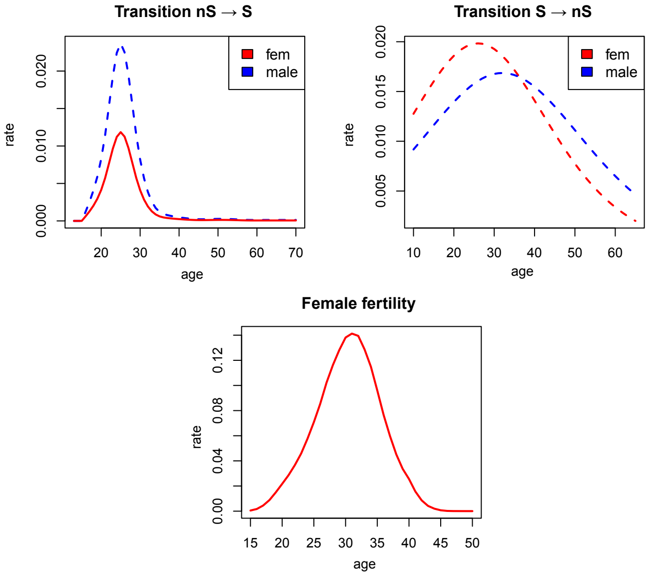


Figure A-20.: Transition rates of unlinked individuals to change their smoking behavior and birth rates; S: smoker, nS: non-smoker, CO: cohabiting.

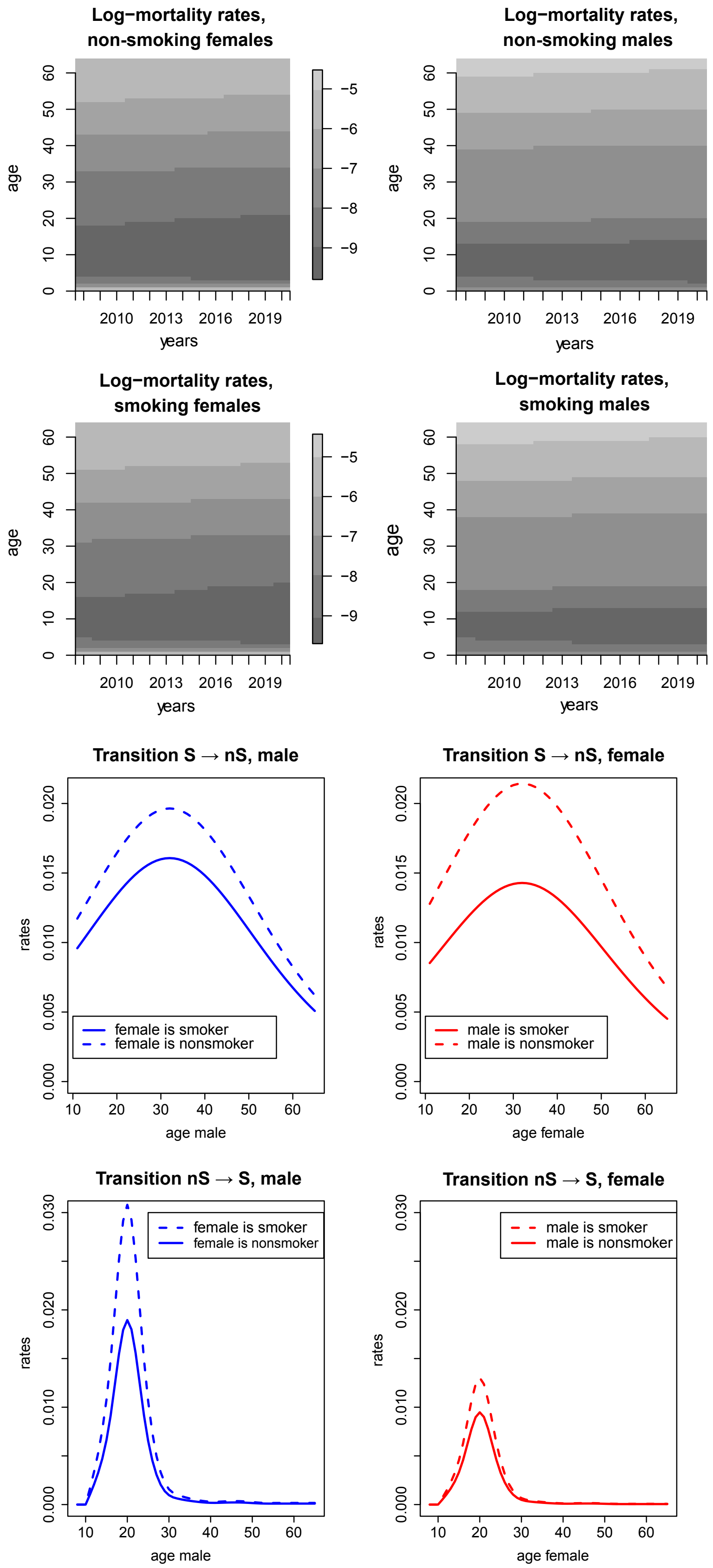
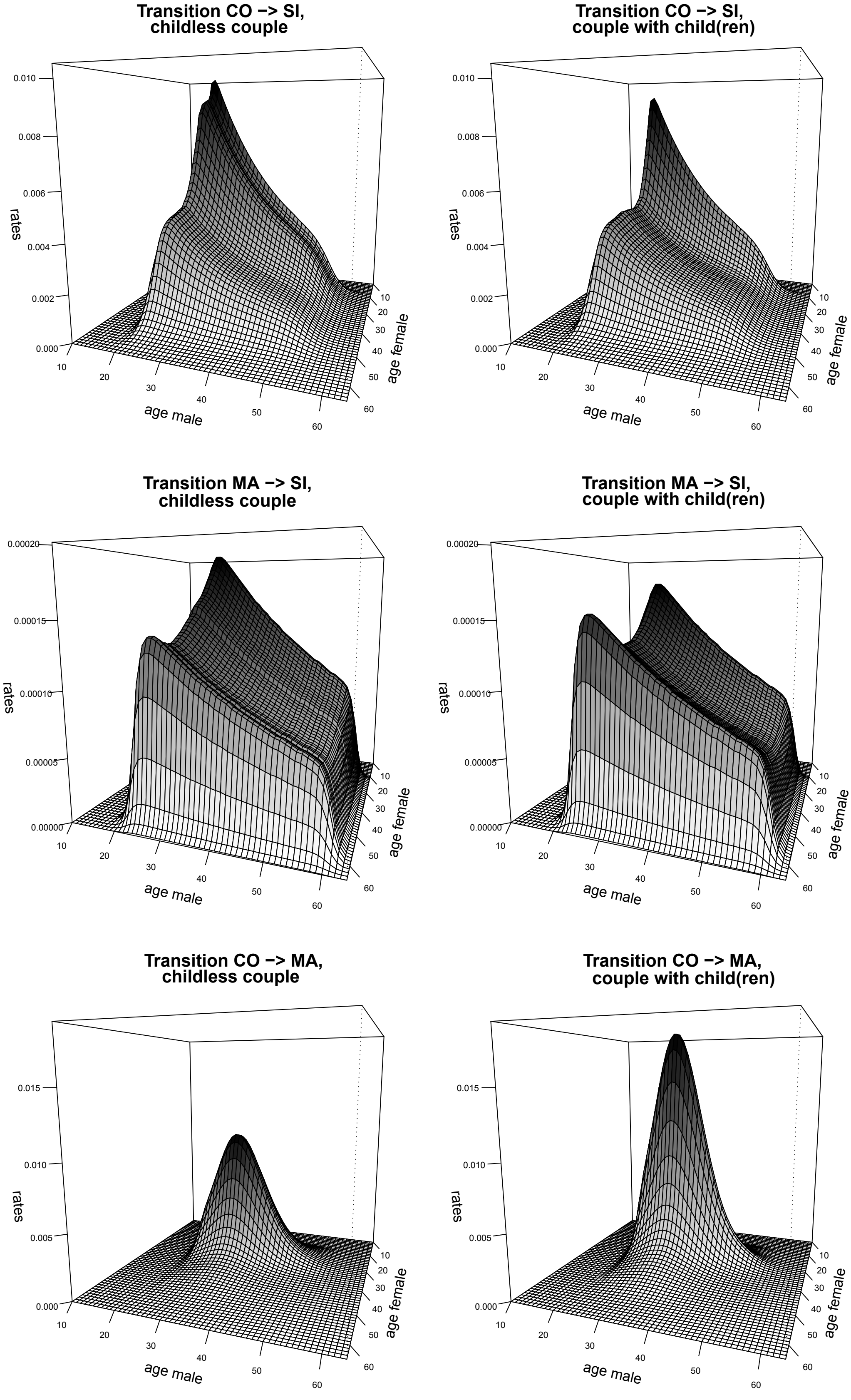


Figure A-21.: Log-mortality rates by age, sex, and smoking status for the period 2004 - 2020 and ages from 0 to 63, and transition rates of spouses to change the smoking status; S: smoker, nS: non-smoker.


Transition CO −> SI, childless couple

Transition CO −> SI, couple with child(ren)

Transition MA −> SI, childless couple

Transition MA −> SI, couple with child(ren)

Transition CO −> MA, childless couple

Transition CO −> MA, couple with child(ren)

Figure A-22.: Transition rates of couples to change the marital status; MA: married, CO: cohabiting, SI: dissolved.

Model 1

Variable	**Coefficient**	**p-value**
Age of male	0.0285	0.0031
Age difference (age of male - age of female)		
greater than 9	−2.7075	< 0.001
from 7 *to* 9	−0.8154	< 0.001
from 4 *to* 6	−0.4665	0.006
from −3 *to* 3	0	
from −6 *to* −4	−1.3531	< 0.001
from −10 *to* −7	−2.6642	< 0.001
smaller than −10	−4.1708	< 0.001
Child(ren) with former partner(s)		
female has child(ren)	0.3926	0.082
Number of potential pairs: 1078		

Model 2

Variable	**Coefficient**	**p-value**
Age of male	0.0181	0.3571
Age difference (age of male - age of female)		
greater than 10	−3.3940	< 0.001
from 7 *to* 10	−1.2144	0.0252
from 3 *to* 6	−0.7533	0.0805
from −2 *to* 2	0	
from −5 *to* −3	−1.6500	0.0061
smaller than −5	−6.2412	< 0.001
Child(ren) with former partner(s)		
female has child(ren)	1.3138	0.0116
Number of potential pairs: 198		

Table A-1.: Results of simple regression models determining compatibility between potential partners. Regression results of Model 1 (entering cohabitation) and of Model 2 (entering marriage).

Bibliography

Ahuja, R., T. Magnanti, J. Orlin, and M. Reddy (1995). Applications of network optimization. In M. Ball, T. Magnanti, C. Monma, and G. Nemhauser (Eds.), *Network Models*, Handbooks in Operations Research and Management Science 7, pp. 1–83. North-Holland, Amsterdam.

Andersen, P. and N. Keiding (2002). Multi-state models for event history analysis. *Statistical Methods in Medical Research 11*(2), 91–115.

Argyle, M. (2009). *Social Interaction* (2nd ed.)., Chapter The development of social human drives, pp. 40–55. Aldine Transactions.

Bacon, B. and S. Pennec (2007, November). APPSIM - modelling family formation and dissolution. Working Paper 4, NATSEM, National Center for Social and Economic Modelling, University of Canberra.

Balci, O. (2003). Verification, validation, and certification of modeling and simulation applications: Verification, validation, and certification of modeling and simulation applications. In S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice (Eds.), *Proceedings of the 2003 Winter Simulation Conference*, pp. 150–158. IEEE Computer Science.

Bartholomew, D. J. (1973). *Stochastic Models for Social Processes* (2nd ed.). John Wiley & Sons.

Becker, G., E. Landes, and R. Michael (1977). An economic analysis of marital instability. *The Journal of Political Economy 85*(6), 1141–1187.

Bernardi, L. (2003). Channels of social influence on reproduction. *Population Research and Policy Review of Economics and Statistics 22*, 527–555.

Bharathy, G. and B. Silverman (2010, September). Challenges of building and validating agent-based social systems models. Submittet to: *Simulation: Transactions of the Society for Modeling and Simulation International.*

Billari, C. (2001). A log-logistic regression model for a transition rate with starting treshold. *Population Studies 55*, 15–24.

Billari, C., A. Prskawetz, B. Diaz, and T. Fent (2007). The "wedding-ring": An agent-based marriage model based on social interaction. *Demographic Research 17*(3), 59–82.

Billari, F. (2000). Searching for mates using 'fast and frugal' heuristics: A demographic perspective. In G. Ballot and G. Weisbuch (Eds.), *Applications of Simulation to Social Sciences*, pp. 53–65. Hermes Science Publications, Oxford.

Blossfeld, H.-P., A. Hamerle, and K. Mayer (1989). *Event History Analysis.* Lawrence Erlbaum Associates.

Blossfeld, H.-P. and G. Rohwer (2002). *Techniques of Event History Modeling: New Approaches to Causal Analysis* (2nd ed.). Lawrence Erlbaum Associates.

Blossfeld, H.-P. and A. Timm (2003). Assortative mating in cross-national comparison: A summary of results and conclusions. In *Who Marries Whom? Educational Systems as Marriage Markets in Modern Societies*, pp. 331–342. Dordrecht, Kluwer.

Bouffard, N., R. Easther, T. Johnson, R. Morrison, and J. Vink (2001). Matchmaker, matchmaker, make me a match. *Brazilian Electronic Journal of Economics 4*(2).

Boyle, P., H. Kulu, T. Cooke, V. Gayle, and C. Mulder (2008). Moving and union dissolution. *Demography 45*(1), 209–222.

Brayton, R., F. Gustavson, and R. Willoughby (1970). Some results on sparse matrices. *Mathematics of Computation 24*(112), 937–954.

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

Burch, T. (1995). Theory of household formation: Progress and challenges. In E. van Imhoff, A. Kuijsten, and L. van Wissen (Eds.), *Household Demography and Household Modeling*, The Plenum Series on Demographic Methods and Population Analysis, pp. 85–95. Plenum Press.

Buss, D. M. (2006). Strategies of human mating. *Psychological Topics 15*(2), 239–260.

Caldwell, S. (1996). CORSIM 3.0 - technical and user documentation. Unpublished manuscript, Cornell University, Department of Sociology.

Caldwell, S., M. Favreault, A. Gantman, J. Gokhale, T. Johnson, and L. Kotlikoff (1999). Social security's treatment of postwar Americans. *Tax Policy and the Economy 13*, 109–148.

Camarda, C. (2009). *MortalitySmooth: Smoothing Poisson counts with P-splines.* R package version 1.0: MPI Rostock. http://CRAN.R-project.org/package=MortalitySmooth.

Cellier, F. (1991). *Continuous System Modeling.* Springer.

Chandola, T., D. Coleman, and R. Horns (1999). Recent European fertility patterns: Fitting curves to 'distorted' distributions. *Population Studies 53*, 317–329.

Coale, A. (1971). Age patterns marriage. *Population Studies 25*(2), 193–214.

Coale, A. and D. McNeil (1972). The distribution by age at first marriage in a female cohort. *Journal of American Statistical Association 67*(340), 743–749.

Courgeau, D. (1997). Event history analysis of houshold formation and dissolution. In E. van Imhoff, A. Kuijsten, P. Hooimeijer, and L. van Wissen (Eds.), *Household Demography and Household Modeling*, The Plenum Series on Demographic Methods and Population Analysis, pp. 185–202. Plenum Press, New York.

Cumpston, R. (2009). Acceleration, alignment and matching in multi-purpose household microsimulations. Paper presented at the 2nd general conference of the International Microsimulation Association, Ottawa, June 8 - 10, Australian National University, South Melbourne.

Currie, I., M. Durban, and P. Eilers (2006). Generalized linear array models with applications to multidimensional smoothing. *Journal of the Royal Statistical Society B 68*(Part 2), 259–280.

Dabrowska, D., G. Sun, and M. Horowitz (1994). Cox regression in a Markov renewal model: An application to the analysis of bone marrow transplant. *Journal of the American Statistical Association 89*, 867–877.

Dagsvik, J., A. Flaatten, and H. Brunborg (2001). A behavioral two-sex marriage model. *Mathematical Population Studies 9*, 97–121.

de Vos, S. and A. Palloni (1989). Formal models and methods for the analysis of kinship and household organisation. *Population Index 55*(2), 174–198.

Deaton, A. (1985). Panel data from time series of cross-sections. *Journal of Econometrics 30*(1-2), 109–126.

Deming, W. E. and F. F. Stephan (1940). On a least square adjustment of a sampled frequency table when the expected marginal totals are known. *Annals of Mathematical Statistics 11*, 427–444.

Diekmann, A. (1990). Diffusion and survival models for the process of entry into marriage. In K. Mayer and N. Tuma (Eds.), *Event History Analysis in Life Course Research*. The University of Wisconsin Press.

Diekmann, A. and K. Schmidheiny (2008). The intergenerational transmission of divorce: A fifteen-country study with the Fertility and Family Survey. ETH Zurich Sociology Working Papers 4, ETH Zurich, Chair of Sociology.

Doll, R., R. Peto, J. Boreham, and I. Sutherland (2004). Mortality in relation to smoking: 50 years' observations on male British doctors. *BMJ 328*(7455).

Dykstra, P., M. Kalmijn, T. Knijn, A. Komter, A. Liefbroer, and C. Mulder (2005). Codebook of the Netherlands Kinship Panel Study – a multi-actor, multi-method panel study on solidarity in family relationships, wave 1. NKPS Working Paper 4, NIDI, The Hague.

Edmonds, B. and D. Hales (2003). Replication, replication and replication: Some hard lessons from model alignment. *Journal of Artificial Societies and Social Simulation 6*(4).

EuroStat (2010). http://epp.eurostat.ec.europa.eu/portal/page/ portal/statistics (accessed April 2011).

Ferguson, T. S. (1989). Who solved the secretary problem? *Statistical Science 4*(3), 282–289.

Fienberg, S. (1970). The analysis of multidimensional contingency tables. *Ecology 51*(3), 419–433.

French, R. and E. Kus (2008). KAMA: A temperature-driven model of mate-choice using dynamic partner representations. *Adaptive Behaviour 16*(1), 71–95.

Fujimoto, R. (2000). *Parallel and Distributed Simulation Systems.* Wiley Series on Parallel and Distributed Computing. John Wiley & Sons.

Galan, J. and L. Izquierdo (2005). Appearances can be deceiving: Lessons learned re-implementing Axelrod's 'evolutionary approach to norms'. *Journal of Artificial Societies and Social Simulation 8*(3).

Gale, D. and L. Shapley (1962). College admissions and the stability of marriage. *The American Mathematical Monthly 69*(1), 9–15.

Galler, H. (1988). Microsimulation of household formation and dissolution. In N. Keilman, A. Kuijsten, and A. Vossen (Eds.), *Modelling Household Formation and Dissolution*, pp. 139–159. Clarendon Press, Oxford.

Galler, H. (1997). Discrete-time and continuous-time approaches to dynamic microsimulation (reconsidered). Technical report, NATSEM, National Center for Social and Economic Modelling, University of Canberra, University of Canberra.

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

Geyer, J. and V. Steiner (2010, March). Public pensions, changing employment patterns, and the impact of pension reforms across birth cohorts: A microsimulation analysis for Germany. IZA Discussion Paper 4815, German Institute for Economic Research (DIW) and Center for European Economic Research (ZEW), Institute for the Study of Labor (IZA).

Gilbert, N. and K. Troitzsch (2008). *Simulation for the Social Scientist*, Chapter Multilevel simulation models, pp. 100–129. Open University Press.

Gill, R. and N. Keilman (1990). On the estimation of multidimensional demographic models with population registration data. *Mathematical Population Studies 2*(2), 119–143.

Glidden, D. (2007). Pairwise dependence diagnostics for clustered failure-time data. *Biometrika 94*(2), 371–385.

Goh, R. S. M. and I. L.-J. Thng (2003). Mlist: An efficient pending event set structure for discrete event simulation. *International Journal of Simulation - Systems, Science & Technology 4*(5-6), 66–77.

Goujon, A. (2008). Report on projections by the level of education (future human capital: Estimates and projections of education transition probabilities). Technical report, MicMac project, Vienna Institute of Demography, Austrian Academy od Sciences.

Greenfiled, E. A. and N. Marks (2006). Linked lives: Adult children's problems and their parents' psychological and relational well-being. *Journal of Marriage and Family 68*, 442–454.

Gusfield, D. and R. Irving (1989). *The Stable Marriage Problem: Structure and Algorithms.* Foundation of Computing Series. The MIT Press.

Hadwiger, H. (1940). Eine analytische Reproduktionsfunktion für biologische Gesamtheiten. *Skandinavisk Aktuarietidskrif 23*, 101–113.

Hammel, E., D. Hutchinson, K. Wachter, R. Lundy, and R. Deutel (1976). *The SOCSIM demographic-sociological microsimulation program: Operating manual.* Institute of International Studies, University of California, Berkeley. Research Series, No. 27.

Hammel, E., C. Mason, and C. Wachter (1990). *SOCSIM II: A sociodemographic microsimulation program rev. 1.0. Operating manual.* Regents of the University of California.

Hannay, J., H. Langtangen, C. MacLeod, D. Pfahl, J. Singer, and G. Wilson (2009). How do scientists develop and use scientific software? In *Proceedings of the SECSE*, pp. 1–8. IEEE.

Harknett, K. (2008). Mate availability and unmarried parent relationships. *Demography 45*(3), 555–571.

Hasti, T. and R. Tibshirani (1990). *Generalized Additive Models.* Springer.

Hellekalek, P. (1998). Good random number generators are (not so) easy to find. *Mathematics and Computers in Simulation 46*, 485–505.

Hernes, G. (1972). The process of entry into first marriage. *American Sociological Review 37*, 173–182.

Hill, R. A. and R. I. M. Dunbar (2003). Social network size in humans. *Human Nature 14*(1), 53–72.

Himmelspach, J. (2007). *Konzeption, Realisierung und Verwendung eines allgemeinen Modellierungs-, Simulations- und Experimentiersystems, Entwicklung und Evaluation effizienter Simulationsalgorithmen.* Reihe Informatik. Göttingen: Sierke Verlag.

Himmelspach, J., R. Ewald, and A. Uhrmacher (2008). A flexible and scalable experimentation layer for JAMES II. In S. J. Mason, R. R. Hill, L. Mönch, and O. Rose (Eds.), *Proceedings of the 2008 Winter Simulation Conference*, pp. 827–835. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Himmelspach, J. and A. Uhrmacher (2004). Processing dynamic PDEVS models. In D. DeGroot and P. Harrison (Eds.), *Proceedings of the 12th IEEE International Symposium on MASCOTS*, pp. 329–336. IEEE Computer Society.

Himmelspach, J. and A. Uhrmacher (2006). Sequential processing of PDEVS models. In A. Bruzzone, A. Guasch, M. Piera, and J. Rozenblit (Eds.), *Proceedings of the 3rd EMSS*, Barcelona, Spain, pp. 239–244. LogiSim.

Himmelspach, J. and A. Uhrmacher (2007a). The event queue problem and PDEVS. In *Proceedings of the SpringSim '07, DEVS Integrative M&S Symposium*, pp. 257–264. SCS.

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

Hoem, J. (1976). The statistical theory of demographic rates. *Scandinavian Journal of Statistics 3*, 169–185.

Holm, E., K. Holme, K. Mäkilä, M. Mattsson-Kauppi, and G. Mörtvik (2002). The SVERIGE spatial microsimulation model: Content, validation, and example applications. Work report GERUM 2002:4, Kulturgeografiska institutionen/ SMC.

Holmer, M., A. Janney, and B. Cohen (2009). *PENSIM Overview.* Policy Simulation Group and the U.S. Department of Labour.

Hougaard, P. (1995). Frailty models for survival data. *Lifetime Data Analysis 1*, 255–273.

Hougaard, P. (1999). Multi-state models: a review. *Lifetime Data Analysis 5*(3), 239–264.

Hougaard, P. (2000). *Analysis of Multivariate Survival Data.* Statistics for Biology and Health. Springer.

Huinink, J. and M. Feldhaus (2009). Family research from the life course perspective. *International Sociology 24*(299), 299–324.

Impicciatore, R. and F. Billari (2007). Age profiles estimation for family and fertility events based on micro data: The MAPLE (method for age profile longitudinal estimation). In *Joint Eurostat /UNECE Work Sessions on Demographic Projections.*

Jacobs, D., H. Adachi, I. Mulder, D. Kromhout, A. Menotti, A. Nissinen, and H. Blackburn (1999). Cigarette smoking and mortality risk: Twenty-five-year follow-up of the Seven Countries Study. *Archives of Internal Medicine 159*(7), 733–740.

John, M., C. Lhoussaine, J. Niehren, and A. Uhrmacher (2010). The attributed pi-calculus with priorities. *Transactions on Computational Systems Biology XII - Lecture Notes in Computer Science 5945/2010*, 13–76.

Kalbfleisch, J. D. and J. F. Lawless (1985). The analysis of panel data under a Markov assumption. *Journal of the American Statistical Association 80*, 863–871.

Kalbfleisch, J. D., J. F. Lawless, and W. M. Vollmer (1983). Estimation in Markov models from aggregate data. *Biometrics 39*(4), 907–919.

Kalmijn, M. (1998). Intermarriage and homogamy: Causes, patterns, trends. *Annual Review of Sociology 24*, 395–421.

Kalmijn, M., A. Loeve, and D. Manting (2007). Income dynamics in couples and the dissolution of marriage and cohabitation. *Demography 44*(1), 159–179.

Kaneko, R. (2003). Elaboration of the Coale-McNeil nuptiality model as generalized log gamma distribution: A new identity and empirical enhancements. *Demographic Research 9*(10), 223–262.

Karlin, S. and H. Taylor (1975). *A First Course in Stochastic Processes* (2nd ed.). Academic Press.

Kasprzyk, D., G. Duncan, G. Kalton, and M. P. Singh (Eds.) (1989). *Panel Surveys.* Wiley, New York.

Keiding, N. (1990). Statistical inference in the Lexis diagram. *Philosophical Transactions of the Royal Society of London: Physical Sciences and Engineering 332*(1627), 487–509.

Kelly, S. (2003, December). Australia's microsimulation model - DYNAMOD. Technical report, NATSEM, National Center for Social and Economic Modelling, University of Canberra.

Keyfitz, N. and H. Caswell (2005). *Applied Mathematical Demography* (3 ed.)., Chapter The multi-state model, pp. 444–460. Springer.

Kijima, M. (1997). *Markov Processes for Stochastic Modeling*, Chapter Continuous-time Markov chains, pp. 167–241. Stochastic modeling. Chapman & Hall.

King, A., H. Baekgaard, and M. Robinson (1999). Dynamod-2: An overview. Technical report, NATSEM, National Center for Social and Economic Modelling, University of Canberra.

Klevmarken, A. (2002). Statistical inference in micro-simulation models: Incorporating external information. *Mathematics and Computer in Simulation 59*, 255–265.

Kruithof, J. (1937). Calculation of telephone traffic. *De Ingenieur 52*, E15–25.

Labiche, Y. and G. Wainer (2005). Towards the verification and validation of DEVS models. In *Proceedings of 1st Open International Conference on Modeling & Simulation*, pp. 295–305.

Laditka, S. and D. Wolf (1998). New methods for analyzing life expectancy. *Journal of Aging and Health 10*, 214–241.

Law, A. and W. Kelton (2000). *Simulation Modeling and Analysis*, Chapter Generating random variates, pp. 437–495. Industial Engineering Series. Mc Graw Hill Higher Education.

Leblanc, N., R. Morrison, and H. Redway (2009). A match made in silicon: Marriage matching algorithms for dynamic microsimulation. Paper presented at the 2nd general conference of the International Microsimulation Association, Ottawa, June 8 - 10, DYNACAN project, Ottawa.

L'Ecuyer, P. and P. Hellekalek (1998). Random number generators: Selection criteria and testing. In *Random and Quasi-Random Point Sets*, Number 138 in Lectures Notes In Statistics, pp. 223–266. Springer.

Liang, Z. (2000). *The Coale-McNeil Model: Theory, Generalisation and Application.* Rijksuniversiteit Groningen. Dissertation.

Lichter, D. and J. Carmalt (2009). Religion and marital quality among low-income couples. *Social Science Research 38*(1), 168–187.

Lichter, D. and Z. Qian (2008). Serial cohabitation and the marital life course. *Journal of Marriage and Family 70*(4), 861–878.

Liming, C., N. Schenker, and J. Lubnitz (2006). Analysis of functional status transitions by using a semi-Markov process in the presence of left-censored spells. *Applied Statistics 55*(4), 477–491.

Lutz, W. (2009). Toward a systematic, argument-based approach to defining assumptions for population projections. Interim Report IR-09-037, International Institute for Applied Systems Analysis, Austria.

Ma, J. (1995). Stable matchings and rematching – proof equilibria in a two-sided matching market. *Journal of Economic Theory 66*, 352–369.

Marsaglia, G. (1968). Random numbers fall mainly in the planes. *Proceedings of the National Academy of Science 61*(1), 25–28.

Matsuo, H. and F. Willekens (2003). Event histories in the Netherlands Fertility and Family Survey 1998. Research Report 03-1, Population Research Center, University of Groningen.

Mayer, K. (2009). New directions in life course research. *Annual Review of Sociology 35*, 413–433.

Mayer, K. and N. Tuma (Eds.) (1990). *Event History Analysis in Life Course Research.* University of Wisconsin Press.

McClean, S. and E. Montgomery (1999). Estimation for semi-Markov manpower models in a stochastic environment. In *Semi-Markov Models and Applications*, pp. 219–227. Kluwer Academic Publishers. edited by Janssen, J. and Limnios, N.

McClean, S., E. Montgomery, and F. Ugwuowo (1998). Non-homogeneous continuous-time Markov and semi-Markov manpower models. *Applied Stochastic Models and Data Analysis 13*, 191–198.

McFarland, D. (1972). Comparison of alternative marriage models. In T. Greville (Ed.), *Population Dynamics*, pp. 89–106. Academic Press, New York.

Merali, Z. (2010, October). ... why scientific programming does not compute. *Nature 467*, 775–777.

Minar, N., R. Burkhart, C. Langton, and M. Askenazi (1996, Juni). The SWARM simulation system: A toolkit for building multi-agent simulations. Technical report, Santa Fe Institute.

Monteiro, A., G. Smirnov, and A. Lucas (2006). Non-parametric estimation for non-homogeneous semi-Markov processes: An application to credit risk. Discussion Paper TI 2006-024/2, Tinbergen Institute.

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

Müller, P. (Ed.) (1975). *Wahrscheinlichkeitsrechnung and mathematische Statistik, Lexikon der Stochastik* (2nd ed.). Akademie-Verlag Berlin.

Murata, T. (1989). Petri nets: Properties, analysis and applications. In *Proceedings of the IEEE*, Volume 77(4), pp. 541–580.

Murphy, M. (2003). Bringing behavior back into micro-simulation: Feedback meachanisms in demographic models. In F. Billari and A. Prskawetz (Eds.), *Agent-Based Computational Demography - Using Simulation to Improve our Understanding of Demographic Behaviour*, pp. 159–174. Physica-Verlag.

Murphy, M. (2006). The role of assortative mating on population growth in contemporary developed societies. In F. Billari, T. Fent, A. Prskawetz, and J. Scheffran (Eds.), *Agent-Based Computational Modelling: Applications in Demography, Social, Economic and Environmental Sciences*, pp. 61–84. Physica-Verlag.

Nemhauser, G., A. Rinnooy Kan, and M. Todd (1989). *Optimization*, Volume 1 of *Handbooks in Operations Research and Management Science.* North-Holland, Amsterdam.

Neyer, G. and G. Andersson (2008). Consequences of family policies on childbearing behavior: Effects or artifacts? *Population and Development Review 34*(4), 699–724.

NIDI (2006, June). MicMac newsletter. http://www.nidi.knaw.nl/en/micmac/newsletter/newsletter_1/micmac-newsletter-01.pdf.

O'Donoghue, C. (1999). Dynamic microsimulation: A methodological survey. *Brazilian Electronic Journal 4*(2).

Ogurtsova, K., J. Gampe, and S. Zinn (2009). The MicMac Pre-processor. Manual.

O'Harra, J. and J. Sabelhaus (2002, October). Projecting longitudinal marriage patterns for long-run policy analysis. Technical Paper 2002-2, Congressional Budget Office Washington, DC.

Orcutt, G. (1957). A new type of socio-economic system. *Review of Economics and Statistics 39*(2), 116–123.

Pawlowskia, B. and R. I. M. Dunbar (1999). Withholding age as putative deception in mate search tactics. *Evolution and Human Behavior 20*(1), 53–69.

Perese, K. (2002, November). Mate matching for microsimulation models. Technical Paper 3, Long-Term Modeling Group, Congressional Budget Office, Washington, DC.

Perumalla, K. (2005). μsik: A micro-kernel for parallel/distributed simulation systems. In *ACM/IEEE/SCS Workshop on Parallel and Distributed Simulation (PADS)*, Monterey, CA. IEEE Computer Society Press.

Pollard, J. and C. Höhn (1993). The interaction between the sexes. *Zeitschrift für Bevölkerungswisschenschaft 19*, 203–228.

Prentice, R. and R. Pyke (1979). Logistic disease incidence models and case-control studies. *Biometrika 66*(3), 403–411.

Press, W., S. Teukolsky, W. Vetterling, and B. Flannery (2007). *Numerical Recipes - The Art of Scientifc Computing* (3rd ed.). Cambridge University Press.

Pressat, R. (1972). *Analyse Demographique (english).* Aldine-Atherton, Chicago.

Priami, C. (1995). The stochastic pi-calculus. *The Computer Journal 38*, 578–589.

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

Ruggles, S. (1993). Confession of a microsimulator: Problems in modeling the demographic kinship. *Historical Methods 26*(4), 161 – 169.

Satyabudhi, B. and S. Onggo (2008). Parallel discrete-event simulation of population dynamics. In S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler (Eds.), *Proceedings of the 2008 Winter Simulation Conference*, pp. 1047–1054. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Schneider, M. and S. Zenios (1990). A comparative study of algorithms for matrix balancing. *Operations Research 38*(3), 439–455.

Scott, A., M. Evendrou, J. Falkingham, and A. Zaidi (2003). Beyond 2020: Lessons from the development of the SAGE dynamic microsimulation model for Britain. ESRC-SAGE Discussion Paper No. 14, London School of Economics.

Sekhon, J. (2007). Multivariate and propensity score matching software with automated balance optimization: The matching package for R. *Journal of Statistical Software 10*(2).

Siegert, M. (2006). Integrationsmonitoring - State of the Art in internationaler Pespektive, Studie im Autrag des Bundesamtes für Mirgation und Flüchtlinge (Bamf), Nürnberg. Technical report, Europäisches Forum für Migrationsstudien (EFMS).

Simon, H. (1957). *Models of Man.* John Wiley & Sons.

Simon, H. (1990). Invariants of human behavior. *Annual Review of Psychology 41*, 1–19.

Smith, J. (1987). The computer simulation of kin sets and kin counts. In J. Bongaards, T. Burch, and K. Wachter (Eds.), *Family Demography: Methods and their Application*, pp. 249–266. Clarendon Press, Oxford.

Smith, K. and C. Zick (1994). Linked lives, dependent demise? Survival analysis of husbands and wives. *Demography 31*(1), 81–93.

Spielauer, M. (2002). The potential dynamic of microsimulation in family studies: A review and some lessons for FAMSIM+. Working Paper 18, Austrian Institute for Family Studies, Laxenburg, Austria.

Statistics Canada (2011a). The LifePath microsimulation model: An overview. http://www.statcan.gc.ca/spsd/LifePaths.htm (accessed April 2011).

Statistics Canada (2011b). Modgen version 10.1.0: Developer‘s guide. http://www.statcan.gc.ca/microsimulation/modgen/doc-eng.htm.

Teo, C.-P., J. Sethuraman, and W.-P. Tan (2001). Gale-Shapley stable marriage problem revisited: Strategic issues and applications. *Management Science 47*(8), 1252–1267.

Todd, M. (2002). Fast and frugal heuristics for environmentally bounded minds. In G. Gigerenzer and R. Selten (Eds.), *Bounded Rationality, The Adaptive Toolbox*, pp. 51–70. First MIT Press.

Todd, P. (1997). Searching for the next best mate. In R. Conte, R. Hegselmann, and P. Terna (Eds.), *Simulating Social Phenomena*, pp. 419–436. Springer-Verlag, Berlin.

Todd, P., C. Billari, and J. Simão (2005). Aggregate age-at-marriage patterns from individual mate-search heuristics. *Demography 42*, 559–574.

Todd, P. and F. Billari (2003). Population-wide marriage patterns produced by individual mate-search heuristics. In F. Billari and A. Prskawetz (Eds.), *Agent-Based Computational Demography: Using Simulation to Improve our Understanding of Demographic Behaviour*, pp. 118–137. Physica-Verlag.

Trivers, R. (1972). Sexual selection and the descent of man, 1871-1971. In B. Campbell (Ed.), *Parental Investment and Sexual Selection*, pp. 136–179. Aldine, Chicago.

Uhrmacher, A. (2001). Dynamic structures in modeling and simulation: A reflective approach. *ACM Transactions on Modeling and Computer Simulation 11*, 206–232.

Uhrmacher, A., R. Ewald, M. John, C. Maus, M. Jeschke, and S. Biermann (2007). Combining micro and macro-modeling in DEVS for computational biology. In *Proceedings of the 2007 Winter Simulation Conference.*

Uhrmacher, A., J. Himmelspach, and R. Ewald (2010). Effective and efficient modeling and simulation with DEVS variants. In G. Wainer and P. Mosterman (Eds.), *Discrete-Event Modeling and Simulation: Theory and Applications.* CRC Press - Taylor and Francis.

United Nations (Ed.) (2001). *Principles and Recommendations for a Vital Statistics System (Revision 2).* Economic and Social Affairs. United Nations.

van Emde Boas, P., R. Kaas, and E. Zijlstra (1977). Design and implementation of an efficient priority queue. *Mathematic Systems Theory 10*, 99–127.

van Imhoff, E. (1990). The exponential multidimensional demographic projection model. *Mathematical Population Studies 2*(3), 171–181.

van Imhoff, E. and N. Keilman (1991, October). *Lipro 2.0: An application of a dynamic demographic projection model to household structure in Netherlands.* The Hague: NIDI.

van Imhoff, E. and W. Post (1998). Microsimulation methods for population projection. *Population: An English Selection, New Methodological Approaches in Social Sciences 10*(1), 97–138.

van Wissen, L. and P. Ekamper (2009). Estimating a starting population for MicMac. Micmac project report, NIDI, The Hague/ Population Research Center, University Groningen, http://www.nidi.knaw.nl/en/micmac/output/epc-2008-micmac-vanwissen.pdf/epc-2008-micmac-vanwissen.pdf.

Wachter, K. (1995, April). 2030's seniors: Kin and step-kin. Technical report, University of California, Berkeley.

Wachter, K., D. Blackwell, and E. Hammel (1998, February). Testing the validity of kinship microsimulation: An update. Technical report, University of Carlifornia, Berkeley.

Wang, W., P. Scarf, and M. A. J. Smith (2000). On the application of a model of condition-based maintenance. *The Journal of the Operational Research Society 51*, 1218–1227.

Warner, T., W. Manning, P. Giordano, and M. Langmore (2010, August). Relationship formation and stability in emerging adulthood: Do sex ratios matter? Working Paper WP-10-10, National Center for Family & Marriage Research, Bowling Green State University.

Wienke, A. (2011). *Frailty Models in Survival Analysis*. CRC Biostatistics Series. Chapman Hall/CRC Taylor & Francis Group.

Willekens, F. (1980). Entropy, multiproportional adjustment and the analysis of contingency tables. *Systemi Urbani 2*(3), 171–201.

Willekens, F. (1982). Multidimensional population analysis with incomplete data. In K. Land and A. Rogers (Eds.), *Multidimensional Mathematical Demography*, pp. 43–111. Academic Press.

Willekens, F. (1999). Modeling approaches to the indirect estimation of migration flows: From entropy to EM. *Mathematical Population Studies 7*(3), 239–278.

Willekens, F. (2005a). Biographic forecasting: bridging the micro-macro gap in population forecasting. *New Zealand Population Review of Economics and Statistics 3*, 77–124.

Willekens, F. (2005b, May). Bridging the micro-macro gap in population forecasting. MicMac project proposal, NIDI, The Hague.

Willekens, F. (2006). Multistate model for biographic analysis and projection (multistate model for biographic analysis and projection). MicMac project paper Deliverable D1, NIDI, The Hague, The Hague.

Willekens, F. (2007). Continuous-time microsimulation in longitudinal analysis. Technical report, NIDI, The Hague.

Willekens, F. (2009). Continuous-time microsimulation in longitudinal analysis. In A. Zaidi, A. Harding, and P. Williamson (Eds.), *New Frontiers in Microsimulation Modelling*, pp.

353–376. Ashgate, Farnham, Surrey, UK. (Book released during 2nd International Microsimulation Conference, Ottawa, June 2009).

Willekens, F., J. de Beer, and N. van der Gaag (2007). Report on input data requirements of MIC. Technical report, NIDI, The Hague.

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

Wolf, D. (1988). The multistate life table with duration-dependence. *Mathematical Population Studies 1*(3), 217–245.

Wolf, D. (2001). The role of microsimulation in longitudinal data analysis. *Special Issue on Longitudinal Methodology, Canadian Studies in Population 28*(2), 313–339.

Wolf, D. and T. Gill (2009). Modeling transition rates using panel current-status data: How serious is the bias? *Demography 46*(2), 371–386.

Woodbury, S. A. and C. Davidson (Eds.) (2003). *Search Theory and Unemployment*, Volume 76 of *Recent Economic Thought.* Springer.

Zaidi, A. and K. Rake (2002). Dynamic microsimulation models: A review and some lessons for SAGE. Discussion Paper no. 2 SAGEDP/02,, London School of Economics, ESRC: 1-30.

Zeigler, B. (2000). Verification and validation of DEVS models: Applying the theory of modeling and simulation to the needs of simulation based acquisition. In *Proceedings of Summer Computer Simulation Conference.*

Zeigler, B. P., H. Praehofer, and T. Kim (2000). *Theory of Modelling and Simulation* (2nd ed.). Academic Press. (Reissued by Krieger Pub. Co., Malabar, Fla. 1985).

Zinn, S. and J. Gampe (2010). *The MicMac post-processor.* MPIDR, Rostock. Manual.

Zinn, S. and J. Gampe (2011). *The MicMacCore.* MPIDR, Rostock. Manual.

Zinn, S., J. Gampe, J. Himmelspach, and A. Uhrmacher (2010). A DEVS model for demographic microsimulation. In *Spring Simulation Multi-Conference - Symposium on Theory of Modeling & Simulation - DEVS Integrative M&S Symposium (DEVS)*, San Diego, pp. 87–94. SCS Publishing House.

Zinn, S., J. Himmelspach, J. Gampe, and A. Uhrmacher (2009). Mic-core: A tool for microsimulation. In M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls (Eds.), *Proceedings of the 2009 Winter Simulation Conference*, Piscataway, NJ. IEEE.

Thesis Statements

1. For a precise description of population dynamics continuous-time models are an optimal theoretical choice.

2. As opposed to traditional population projection methods (like cohort component models), a microsimulation approach allows to describe intra-cohort variation.

3. A continuous-time microsimulation model is a discrete event model.

4. Due to its modular and state-based concept, the DEVS formalism is well-suited to formulate a continuous-time microsimulation model.

5. The simulation semantics that has originally been developed for the execution of DEVS models hinders population projections of meaningful size.

6. A DYNPDEVS network is more appropriate to describe population dynamics than an atomic DEVS model.

7. Using the M&S framework JAMES II for implementing a microsimulation allows us to profit from already implemented M&S functionality and to validate the final microsimulation product.

8. Within this thesis we propose to determine a next life-course event by computing random waiting times for all possible destination states, and by selecting then the shortest one. An alternative method to compute life-course events is to first determine the waiting time in the current state, and then to specify the destination state. We argue that both methods are equivalent.

9. Defining an infertile period of nine months after delivery might lead to distorted outcomes in the sense that the input fertility rates do not correspond exactly to the simulated output.

10. We analyzed the performance of four event queue data structures: calendar queue, mlist, heap queue, and simple event queue. In the applied continuous-time microsimulation setting the mlist performs best.

11. A way to test the implementation of the microsimulation is to re-estimate the empirical transition rates which were used as input.

12. If we run the MicMac microsimulation and macroprojection consistently parameterized, the aggregated microsimulation output corresponds to the macroprojection results.
13. Neglecting inter-individual relationships in a microsimulation model might lead to less accurate population projections.
14. In discrete-time models annual marriage markets are applied to create couples. This is a natural choice as in discrete-time models events are determined to occur in a given year. In continuous-time models by contrast, where events occur at continuous time points, annual marriage markets are not necessarily the most appropriate choice.
15. In our mate-matching approach, an empirical likelihood equation gives the probability that a given woman and a given man would mate. This way we can account for assortative mating.
16. The developed mate-matching algorithm can handle (relative) competition: an extra supply of single men of a certain age decreases the number of paired men at all other ages (competition). The effect is most pronounced at ages neighboring the age group with the surplus of single men (relative competition).
17. Like individual life-courses, couple dynamics can be specified using continuous-time multi-state models.
18. Extending the developed dynPDEVS population network to allow for partnership creation and dynamics is possible, but not convenient. A ml-DEVS model is more appropriate.