4th General Conference of the International Microsimulation Association
Canberra, Wednesday 11th to Friday 13th December 2013

The Danish Microsimulation Model SMILE – An overview

Peter Stephensen
DREAM

Abstract

The SMILE model is a Danish, dynamic, data-driven microsimulation model. The current version forecasts demography, education level, socioeconomic characteristics and housing demand for the period 2010-2050. The basic idea with SMILE is to unite the pre-models that the Danish institution DREAM already uses in a full dynamic microsimulation model. The new elements of the model are described and the development strategy is outlined. The model is based on a new *Event Pump architecture.* This is a Lego-block-like object oriented technique where the model is built as an *Agent Tree* consisting of *Agent* objects. The model take extensive use of a method called CTREE, which is a decision tree technique that has not previously been used for microsimulation modelling. Finally, a matching algorithm called SBAM (Sparse Biproportionate Adjustment Matching) has been developed.

Keywords: population projections, education, household projections, housing demand, microsimulation

The Danish Microsimulation SMILE – An overview

1. Introduction

The SMILE[1] model is a Danish, dynamic, data-driven microsimulation model. The current version forecasts demography, education level, socioeconomic characteristics and housing demand for the period 2010-2050.

The model has been developed by the semi-governmental institution DREAM[2]. Many of the expertises needed for the development of a microsimulation model is present in DREAM. DREAM has existed since 1997, and has long run structural economics and fiscal sustainability as its main focuses. The DREAM model system consists of three so called pre-models and the macroeconomic model DREAM (an overlapping generations model). The three pre-models are a population projection, an education projection and a socioeconomic projection. The population projection is the official projection of the Danish population, developed in cooperation with Statistics Denmark. The population projection and the socioeconomic projection are classical cell based models, whereas the education projection is a small microsimulation model. The basic idea with SMILE is to unite these models in a full dynamic microsimulation model.

The SMILE model is *dynamic* in the sense that an initial population (the entire Danish population of approximately 5.5 million persons) is forecasted into the future. Demographic events such as death, birth, immigration, emigration etc. are modelled. Many of the demographic assumptions are similar to the assumptions done in DREAM's national population projection. An important example of this is the projections of death probabilities that use the Lee-Carter econometric method (Lee & Carter, 1992). Other features of SMILE are new compared to DREAM's national population projection: Parity is included in fertilities, the model is subdivided in 11 regions and a matching algorithm called SBAM (Stephensen & Markeprand, 2013) is added to the model.

The modelling of *education* decisions is based on a regionally subdivided version of DREAM's education projection. This projection describes the development in current participation in education and highest completed level of education. The model is based on transition probabilities calculated from Danish register data and it thus forecasts education levels by employing historical educational behavior.

The modelling of labor market affiliation is currently based on a relatively simple transition probability approach. A more advanced approach is under way (Bækgaard, 2013).

A central feature in the current version of SMILE is the modelling of household residential choice. This is a fairly complex issue. To deal with this complexity we break down the individual process into a succession of steps, each step representing one elementary decision. First, it is simulated whether each household moves from its current home, since movements are the outcome of a binary choice: Households can

[1] Simulation Model for Individual Lifecycle Evaluation.

[2] The institution is named after its macroeconomic model DREAM: Danish Rational Economic Agents Model.

either choose to move or stay where they already live. If so, the household's choice of a new dwelling is simulated with probabilities indicating how likely it is that the household moves to a dwelling with certain characteristics. The choice of dwelling is the outcome of a series of discrete choices: The households choose the location of the dwelling (province and town size), owner and rental status (housing type), use (physical use), area (the size of dwelling) and year of construction (the age of dwelling).

The moving probability is divided by background characteristics of the household and by characteristics of the household's current dwelling. Together this results in a lot of explaining variables why the moving probability is calculated as a CTREE (defined in 2.1).

The SMILE model is a *data-driven model*, based on rich Danish register data. The data cover the entire Danish population on annual basis in the period between 1986 and 2011. On each individual our dataset contains information about the person himself (gender, age, educational background, labor market participation etc.), the person's family situation (single/couple, number of children living at home etc.) and information about the dwelling that the person's household lives in (location, owner/rental status, dwelling type and size etc.). We derive data from seven different sources made available through Statistics Denmark. The main data sources are the Danish Civil Registration System (*CPR-registret*), the Housing Register (*Bygnings- og Boligregistret, BBR*), the education register (*Uddannelsesregistret*) and the labor force statistics (*Registerbaseret Arbejdsstyrkestatistik, RAS*).

Currently the model does not contain any information on income or wealth. Income- and labor marked dynamics will be added to the model in 2014 (Bækgaard, 2013). As follows from the name of the model (Simulation Model for Individual Lifecycle Evaluation), it is a core ambition to be able to describe the lifecycle of each individual. This includes the ability to calculate future labor market pensions. During the 90's labor marked pensions was implemented for almost all Danes. DREAM already has considerable experience in calculating these forms of pensions and it is the plan to add labor marked pensions to SMILE.

2. What is New?

From a Danish point of view there are several novelties in SMILE. First and foremost it is new to have a full-fledged dynamic microsimulation model. There exists a large static microsimulation model called Lovmodellen ("The Law Model") in Denmark. The model is built around a large and very impressive data set containing thousands of variables at the individual level. The model is used by the government to analyze taxation and distribution. The model does not contain any behavioral assumptions (except for a labor supply effect).

A second new feature is the ability to forecast household formation and family structure. Previously there have only been simple forecasts of the development in the single/couple-composition and in the number of persons per household in Denmark. These numbers are important for many purposes: housing demand, city structure, fertility etc. SMILE can forecast these figures conditional on education and region.

Furthermore, the model can analyze the effect on the family structure of changes in demography, education level etc.

Finally, a long run forecast of the housing demand is new in Denmark. From a planning perspective, it is important to be able to project the future structure of housing demand. Like in many other countries we have a core-periphery issue in Denmark. People are moving to a few major cities, resulting in thinly populated areas. To sharpen SMILE on this issue, DREAM is in the process of subdividing the model further to 98 municipalities instead of 11 regions.

2.1 New methods

When it comes to methodology we have three new methods/techniques and one old method that we think is new in connection with microsimulation.

The CTREE (Horthorn et al., 2006 and Rasmussen, 2013) is a well-known decision tree technique implemented in R. As far as we know it has not been used for microsimulation analysis. It is used in SMILE for two purposes. First, it is used to solve what we like to call *the second wave of 'Curse of Dimensionality'*. Cell based models (like for example standard population projections) suffers under *the first wave of 'Curse of Dimensionality'*: If you add a new categorical variable with *n* levels to a cell based model, the size of the model is multiplied with *n*. A cell based model is relatively easy to make, but its size explodes when you add new features. This problem is solved by microsimulation models. It is hard (and expensive) to make a microsimulation model, but you can add new features to the model without uncontrolled increases in its size. But precisely the fact that it is relatively costless to add new features to the microsimulation model, gives rise to a new kind of curse: *the second wave*. Many features in the model imply that behavior is dependent on many variables. As a result of this, transition probabilities have to be calculated on the basis of 'thin' data. This introduces an extra source of noise into the model. We solve this with the CTREE. The CTREE basically cluster categorical data, using chi-square tests as criteria (for examples of use, see Rasmussen, 2013). This implies a considerable compression of data. As an example, the probability of moving depends on so many variables (age, education, family size, region etc.) that the number of 'raw' transition probabilities[3] is approximately half a million. The CTREE defined on this data set has only approximately 2,000 end points.

The other reason to use the CTREE is that it can be considered an alternative to discrete choice models (logit/probit). Where raw transition probabilities can be accused of *over-fitting*, a simple discrete choice model can easily 'under-fit': it is too simple. A good discrete choice model typically has a lot of interactions between the covariates. It takes skill and time to derive such interactions. The CTREE is a modern descendent of the method called CHAID (CHi-squared Automatic Interaction Detection). It can therefore be seen as a compromise between the too complicated raw transition probabilities and the too simple logit/probit-models. It will typically be better than both, although it probably will have a hard time beating a well done discrete choice model with interactions. But with the very large number of decisions taken in a large dynamic

[3] A 'raw' transition probability is calculated directly from data. Example: We have 20 individuals in data. 5 moves. Therefore the 'raw' probability of moving is 5/20 = 0.25.

microsimulation model, it is not viable to depend on thorough econometric analysis on every detail.

The first methodological novelty of the model is the use of a so called *Agent object*. SMILE is written is C# and is an object oriented model. Most features of the model are therefore defined as objects (Household, Person, Dwelling, Simulation etc.). It is assumed that all these agents are inherited from the same basic object. As a result of this, the model can be described as an *Agent Tree*. We will elaborate further on this in the next section.

The next novelty, and in connection with the above, is the *Event-Pump Architecture*. As in most other dynamic microsimulation models a basic element of the model is *events*. Events are what people do: birth, death, moving etc. The core driver of the model is a loop called the *event pump*. This loop controls when and where things happen in the model. The feature is inspired by the *message pump*, which is a central element in how Windows is programmed (Petzold, 1998).

Finally we have developed a matching algorithm called SBAM (Stephensen & Markeprand, 2013). SBAM is an acronym for Sparse Biproportionate Adjustment Matching. The method is a data-driven top-down technique used to model the formation of couples. It is fast and the sparsity implies that the method can cope with many covariates.

3. Development strategy

The basic concepts in the development strategy have been *object orientation* and *agent based modelling*. The model is programmed in an object orientated programming language (C#). The advantages of object orientation are often described by the words *encapsulation* and *separation*. By encapsulation you mean that a lot of functionality can be hidden inside an object. That makes it easier for others to use it. This is best demonstrated by an example. Figure 1 shows the *Agent Tree*. This is basically the model. The Simulation object at the top level controls the simulation. The left branch contains the actual agents of the system: the households and the persons. The right branch contains different kinds of functionality. The Demographics object contains information on the behavior of the households and persons, and the Statistics object gathers information (output data). The Demographics object contains all the transition probabilities and (importantly) how to use them. All this functionality is *encapsulated* in the object. When programming the left branch you can use the Demographics object without actually knowing/understanding what's inside it. You could now replace the Demographics object with a Swedish version (ie. with Swedish transition probabilities) and you would have a Swedish instead of a Danish model. This example also demonstrates the related concept *separation*. It is considered as good programming practice to separate various tasks. This makes it easier to maintain code and to find errors. The structure of the model naturally separates these tasks.

In the last 10 to 15 years Agent Based Modelling has been more and more common. The basic idea is that modelling should be done at the individual level, and that

interaction is important. Microsimulation and agent based modelling is related traditions but never the less different. Both traditions work at the individual level. But when it comes to interaction among the agents there are differences. The most important example of interaction in microsimulation models is the formation of couples. In many microsimulation models (also in SMILE) this problem is solved with a top-down approach. In an agent based model the approach should always be bottom-up. Agent based models are more principled, use more theory and less data. Microsimulation models are data-driven and more pragmatic.

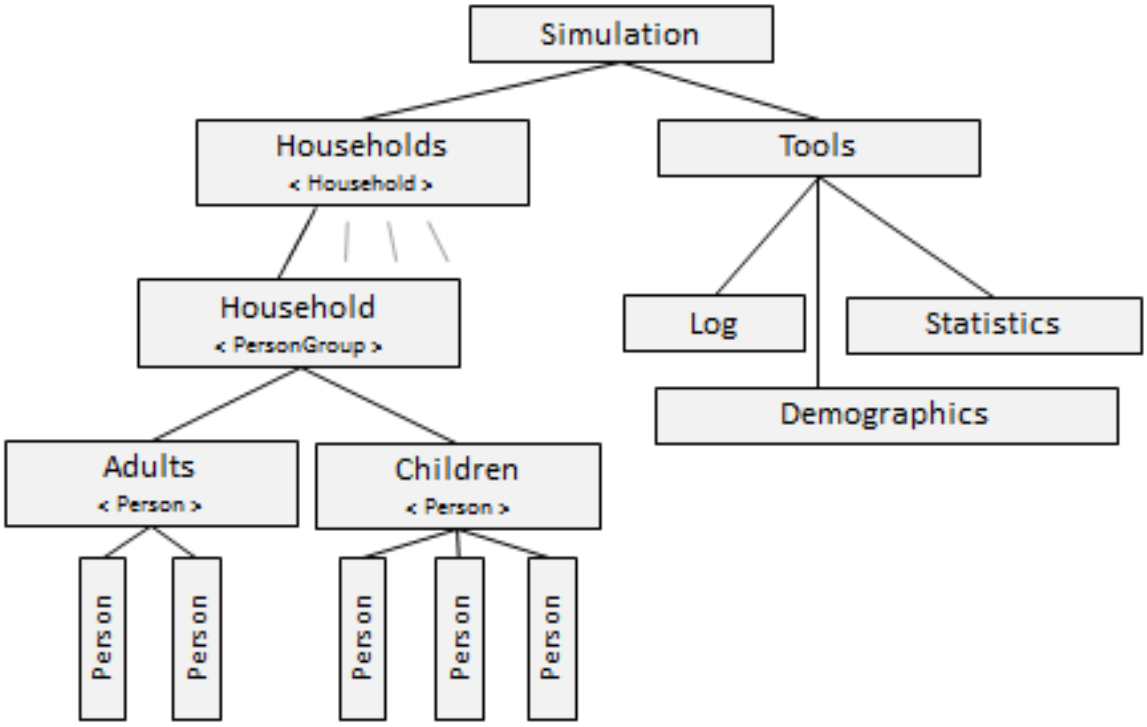

Figure 1 The Agent Tree

Despite these differences SMILE is basically build as an agent based model. It is probably fair to say that microsimulation models is a pragmatic subset of agent based models. The basic argument for using an agent based modelling strategy is therefore generality. Using this approach opens up for a lot of future possibilities of interaction between the agents.

The decision to follow an agent based path let to the idea of a basic Agent object. This object is the Lego block from which the model is build. An Agent is a thing that 1) Potentially has children 2) Potentially do stuff. By ‘having children’ we mean pointing to other Agents. It is this property that makes it possible to build an Agent Tree like the one shown in Figure 1. By ‘do stuff’ we mean having a generic method/function that defines what to do under different circumstances. This method (a ‘function’ is called a ‘method’

in C#) is called `EventProc(idEvent)`. The input `idEvent` is an event ID that defines the current event. The generic behavior of `EventProc` is to do nothing and just send `idEvent` to all its children Agents (if any). If you therefore sends an `idEvent` to the `EventProc` of the top Simulation object in Figure 1, the event will automatically be send down the tree an eventually reach all Agents. If you want your `EventProc` to do something (a Person to die or a Household to move), you just program this behavior before you send the event to the children Agents. Or if you know the event is of no interest for the Agents further down the tree, you can chose to stop the event. This can be a source of speed gains.

Having constructed the Agent tree from our Agent Lego blocks, we just need to send sequences of events down the tree to make the model run. This is done by the *Event Pump*. The Event Pump is a loop located in the top Simulation objects `EventProc`. It sends events down the tree in the correct order. Typically it repeats the same sequence of events every period.

The Event Pump got its name from the so called Windows Message Pump that is a fundamental concept in the original programming of Windows (the Win32 API, see Petzold 1998). In Windows every window is controlled by a callback function called `WinProc`. The input to this function is (among other) a parameter `iMsg`. This is a message ID that defines which message the window receives (pressing a button, moving the mouse over the window etc.). The Message Pump controls the order of these `iMsg`'s. The parallel to our approach is obvious.

The Event Pump Architecture has proven itself very useful and flexible. It is easy to make changes to the model and we have yet to face microsimulation/Agent-based issues that do not have an Event Pump implementation.

4. Concluding remarks

A full-fledged dynamic microsimulation for Denmark has been build. The current version forecasts demography, education level, socioeconomic characteristics and housing demand for the period 2010-2050.

The model is based on a new Event Pump architecture and takes advantage of a method called CTREE, which is a decision tree technique that have not previously been used for microsimulation modelling. The model is still under development, as income- and labor market dynamics is under way as well as a disaggregation to 98 municipalities instead of the current 11 regions.

5. References

Bækgaard, H. (2013): *A Bayesian approach to labour market modeling in dynamic microsimulation*, DREAM Conference Paper, December 2013. The paper can be downloaded from www.dreammodel.dk/SMILE

Hansen, J. Z., Stephensen, P. & Kristensen, J. B. (2013): Household Formation and Housing Demand Forecasts, DREAM Report, December 2013. The report can be downloaded from www.dreammodel.dk/SMILE

Hansen, J. Z. (2013): *Modeling Household Formation and Housing Demand in Denmark using the Dynamic Microsimulation Model SMILE*, DREAM Conference Paper, December 2013. The paper can be downloaded from www.dreammodel.dk/SMILE

Hothorn, T., K. Hornik & A. Zeileis (2006): *Unbiased Recursive Partitioning: A Conditional Inference Framework*, Journal of Computational and Graphical Statistics, Vol. 15, No. 3, page 651–74.

Lee, Ronald D. & Carter, Lawrence R. (1992): *Modeling and Forecasting U.S. Mortality*, Journal of the American Statistical Association, Vol. 87, No. 419, 659-671.

Rasmussen, N. E. (2013): *Conditional inference trees in dynamic microsimulation - modelling transition probabilities in the SMILE model*, DREAM Conference Paper, December 2013. The paper can be downloaded from www.dreammodel.dk/SMILE

Petzold, Charles (1998): *Programming Windows*. 5th edition. Microsoft Press. 1479 pages.

Stephensen, P. & Markeprand, T. (2013): *SBAM: An Algorithm for Pair Matching*, DREAM Conference Paper, December 2013. The paper can be downloaded from www.dreammodel.dk/SMILE