Innovative
Policy Modelling and Governance Tools
for Sustainable Post-Crisis Urban Development

D5.1 Enhanced version of Albatross simulation model

Project acronym :	INSIGHT
Project title :	Innovative Policy Modelling and Governance Tools for Sustainable Post-Crisis Urban Development
Grant Agreement number :	611307
Funding Scheme :	Collaborative project
Project start date / Duration:	01 Oct 2013 / 36 months
Call Topic :	FP7.ICT.2013-10
Project web-site :	http://www.insight-fp7.eu/
Deliverable :	D5.1 Enhanced version of Albatross simulation model
Issue	1
Date	20/4/2016
Status	Approved
Dissemination Level :	Public

Authoring and Approval

Prepared by		
Name & Affiliation	**Position**	**Date**
Harry Timmermans	Professor	Feb 2016
Soora Rasouli	Assistant Professor	Feb 2016

Reviewed by		
Name & Affiliation	**Position**	**Date**
Ricardo Herranz (Nommon)	Scientific/Technical Coordinator	April 2016

Approved for submission to the European Commission by		
Name & Affiliation	**Position**	**Date**
Ricardo Herranz (Nommon)	Scientific/Technical Coordinator	April 2016
Iris Galloso (UPM)	Management Coordinator	April 2016

Record of Revisions

Edition	Date	Description/Changes
Issue 1 Draft 1	13/01/2016	Initial version
Issue 1 Draft 2	20/04/2016	Typo correction and elaboration of the conclusion
Issue 1	20/04/2016	Approval for submission to the EC

Table of Contents

Executive Summary

As part of the INSIGHT program, to improve the integration of different models, the Albatross model system was systematically linked with a residential choice model and dynamic population synthesizer. The relevance of this extension is based on the fact that the house is a key variable, which acts as one of the pegs influencing the action space of individuals. Residential location choice itself may be influenced by several factors such as changes in social demographic of the household. The Albatross model system on the other hand simulates how individuals and households organise their activities in time and space, subject to a set of constraints. Space-time constraints are most critical in this regard. They are dictated by their social demographic, economic situation, location of a households' house relative to work and other activity locations and the chosen/available transportation mode.

Although the current version of Albatross has a synthetic population component, the fact that it exogenously generates a synthetic population on an annual basis jeopardizes the integrity of the data. A dynamic synthetic population ensures the integrity of individual and household data that are subjected to demographic and residential mobility processes.

This report documents the code written for this purpose and illustrates its application using a hypothetical example, related to the City of Rotterdam, The Netherlands. The report re-iterates the model specifications and then continues with a discussion of the implementation and the example. A real world application pertaining to an urban planning policy will be reported in deliverable D7.3 Rotterdam Case Study Report.

1. Introduction

As the term suggests, integrated land use transportation models combine in some systematic manner two or more models, each concerned with a different choice domain. More specifically, integrated land use transportation models predict traffic patterns as a function of accessibility of land use, which in turn is modelled as a function of traffic flows and constitutes one of the key variables of model land use decisions. Thus, accessibility is viewed as the most important location factor to explain the growth in land use, which in turn attracts traffic, resulting in a certain degree of accessibility.

The development of integrated land use transportation models has always showed a gap with the development of its constituent parts (e.g., Wegener, 2004; Timmermans, 2006). For example, over the last 15 years, traditional four-step and trip/tour based models of travel demand have been replaced by considerable more detailed activity-based models (Rasouli & Timmermans, 2014). However, models of that level of detail and complexity have not yet been fully integrated with land use models. Operational integrated land use transportation models are typically based partially or fully on previous generations of models.

This characterisation also applies to the Albatross model system (Arentze & Timmermans, 2004). This computational process model predicts for each adult in households in a coordinated fashion which activities are conducted where, when, for how long, with whom and the transportation mode involved, subject to a set of constraints. It is the only operational and validated activity-based model that has all this functionality. Thus, it is a powerful model to predict travel demand. If needed, predicted high-resolution activity-travel patterns can be processed to generate time-dependent origin-destination matrices, which can be used as input to (dynamic) traffic assignment models. In turn, the results of such traffic assignment models can be used in integrated land use transportation models. The difference would be that a trip or tour-based model is replaced with an activity-based model of travel demand.

Along these lines, the Albatross model could thus replace more traditional transportation models in integrated land use transportation model systems. However, this approach would not avoid, even amplify traditional problems of integrity. Therefore, as a first step into the development of integrated models that are based on a coherent set of decision principles and information, we showed in deliverable D4.2 Housing Location Model how a cross-sectional population forecasting model could be turned into a dynamic population synthesiser to maintain the consistency at the level of individuals and households. A model of residential choice behaviour then includes the dynamic synthesizer. As a result, dynamics in the housing markets, resulting in residential moves, will be affected systematically by changing properties of the synthetic population. The updated household/individual status in terms of socio-demographic characteristics as well as the updated housing location will in turn drive the way they organize their activity travel patterns.

The residential choice model in this study was strongly influenced by the (lack of) publicly available data. In particular, individual level data on housing careers at the level of PCA or zone resolution are not available for privacy reasons. It can be easily replaced with a more advanced residential choice model, however, without affecting the underlying modelling principles. Such more advanced models would include more variables if such data would be collected and/or become publicly available. Other realistic extensions would involve a simulation of transitions in housing supply, which in the current version is assumed a planning variable, which could be considered endogenously, including a simulation of deterioration and renovation of the housing stock. Likewise, property values could be modelled as an extension of the formulated approach.

This report documents the implementation of a model of residential choice behaviour in which a dynamic population synthesizer is embedded. The city of Rotterdam is used as the case study to illustrate the principles of the approach. As said, the level of detail was strongly influenced by the lack of free, micro-level data. The approach can, however, be elaborated in a straightforward manner once such data will become available. The report therefore focuses on the principles of the implementation of the newly developed approach. A full application of the approach based on the Albatross model system will be illustrated later in the project under the policy scenario studies.

The report is organised as follows. First, we will reiterate the formulation of the residential choice model with the embedded dynamic population synthesizer and the data used. Next, we discuss the implementation and the integration of those in Albatross model system. This report is finalized with discussions.

2. The dynamic population synthesizer and residential choice model

2.1. Introduction

Because Albatross is an agent-based micro-simulation model, it simulates activity-travel patterns of all adult members of a household. To reflect the allocation of tasks and resources within households, activity travel patterns of adults in a single household are coordinated. They are also synchronised to reflect joint activity-travel participation.

The choice of using adults in households as decision-making units implies that Albatross needs as input a profile of socio-economic characteristics to do justice to the fact that the nature of activity-travel behaviour differs between socio-demographic segments. Unfortunately, such data do not exist, mainly due to privacy reasons. The best solution therefore is to impute the individual profiles from publicly available aggregate distributions. This is known as a synthetic population. A synthetic population thus is a population, described in terms of individual socio-demographic profiles that have been imputed from publicly available information such that the summation of socio-demographic variables across the synthetic population is consistent with the known aggregate marginal distributions of the variables and covariances that are either assumed or derived from a sample.

In principle, different approaches can be adopted for the imputation process. The synthetic population underlying Albatross is based on the iteratively proportional fitting algorithm, which is equivalent to a loglinear model. This approach adjusts the cell frequencies proportionally across the different dimensions of the multiway frequency tables until the discrepancy from the marginal distributions are smaller than some threshold. In addition, covariance conditions should be met, which in the loglinear representation are captured by the first and higher order interactions of the model.

2.2 Notation

Consider a study area, consisting of *I* zones. Let $i \in I$ be an index for the zones, and let h be an index for households, who live in one of the set of zones *i* of the study area $h \in \mathcal{H}_i$. At the start of the considered time period $t = 0$, each zone has a certain number of households $\mathcal{H}_{icr}^{t=0}$ living in residential units of type *c* and ownership status *r*. Each member q =1, 2, .. Q of household *h*=-1,2, …, $\mathcal{H}$ can be characterized in terms of a profile or set z_{kqhi} of $k = 1, 2, \dots K$ socio-demographic and possible other characteristics. The concept of a synthetic population means that the summation of an attribute across individuals in the synthetic population is consistent with corresponding known statistics Z_{ki} for zones or larger spatial entities, while the correlation between attributes of the synthesized population is consistent with the observed correlation in the sample used for the synthesis. Thus,

$$\sum_{h=1}^{H_i} \sum_{q=1}^{Q_i} z_{kqhi} = Z_{ki}\,; \quad \forall\, k\,, i \qquad (1)$$

$$\rho(z_{Q_{ik}} z_{Q_{ik'}}) = \rho(z_{Q'_{ik}} z_{Q'_{ik'}}) \qquad k, k' \in K \qquad Q' \subset Q \qquad (2)$$

2.3 Dynamic synthetic population

If we wish to keep the individual and household level integrity of the synthetic population over the years, we need to simulate the processes that operate on the population that are relevant to the activity-based model of travel demand. Because households are situated agents, both demographic and residential choice processes need to be simulated. Some demographic processes like aging apply to every individual of the synthetic population. Other demographic processes occur with some probability that may be age and gender-specific. Because some of these processes (like giving birth) is not independent of time, conditional probability distributions reflecting event interval times were used. Thus, inter events temporal probability distributions were used to reflect the temporal dependencies of the demographic processes. Finally, there are some logical constraints in the transitions. For example, one can only divorce if one was married. Under the Dutch law, one can only be married with one person at the same time; implying one has to be single to get married

2.3.1 Demographic processes

The dynamic population synthesizer includes the following set of socio-demographic variables: gender, age, civil status and number of children. As the accounting system is based on a single year, the age of each individual of the synthetic population is increased with one unit every year. A shift in civil status may be caused by marriage and divorce. Changes in civil status due to marriage and divorce were simulated using Monte Carlo draws from age (*A*) and gender (*G*)-specific marriage and divorce rates available from the Central Bureau of Statistics. Thus,

$$p_q^{s,m}(t+1,t) = f((A,G)_q) \quad (5)$$

$$p_q^{m,s}(t+1,t) = f((A,G)_q) \quad (6)$$

where,

$p_q^{s,m}(t+1,t)$ is the probability that the civil status of individual *q* will change from married to single between *t* and *t*+1.

$p_q^{m,s}(t+1,t)$ is the probability that the civil status of individual *q* will change from single to married between *t* and *t*+1.

Marriage and divorce may also change the number of households in each of the zone, depending on the residential choice processes, which happens later in time. In case of a marriage without residential moves beyond the previous zone of residence, two single member households are replaced by one multi-person household, and hence the number of households in zone *i* will be decreased by one. If one of the members has moved from another zone to *i* where the other partner already resided, the change in the number of households in zone *i* will be zero. If both members have moved from another zone to i, the number of households in zone *i* is increased with one. In case of divorce, if both members continue to live in zone *i*, the number of households in that zone will increase by one. If one continues to live in *i* and the other member leaves that zone, there will be no change in the number of households in *i*. In case both move outside of zone *i*, the number of households in zone *i* will be reduced by one.

Marriage means that two partners need to be matched. To that effect, gender-dependent age difference distributions, derived from the MON travel survey data were used. Thus, we derived from the MON data the probability that a man in a particular age group will marry a woman in the various age groups. First, the model identifies using Monte Carlo draws all individuals in the synthetic population who get married and then males and females are matched taking the age difference probabilities into account.

Next, births and deaths also need to be simulated. Births will affect the number of individuals in the population and household size. Death also affects the number of individuals, civil status and household size. While the simulation of death involves Monte Carlo draws from gender and age-specific mortality rates, the birth simulation takes into account age specific fertility rates considering having already a certain number of children. Applying these simulated processes to a baseline synthetic population will simulate change in individual and household socio-demographic profiles at the end of each time period *t* or start of *t+1* due to the simulated demographic processes.

Demographic change is a strong stimulus for people to move house. Giving birth may induce parents to move to a more youth-friendly area with a safer environment or simply require them to think about moving to a bigger house in the same area. Divorce or the situation when the children leave house, on the other hand, can potentially motivate them to move to a smaller dwelling. These processes define the dynamics in the number of households, occupying particular dwelling types with a particular ownership by zone.

$$\mathcal{H}_{icr}^{t+1} = \mathcal{H}_{icr}^{t} \pm \mathcal{M}_{icr}^{t} \pm \mathcal{S}_{icr}^{t} \pm B_{icr}^{t} - \mathcal{D}_{icr}^{t}$$

where,

$\mathcal{H}_{icr}^{t}$ is the number of households living in residential units of type *c* and ownership status *r* in zone *I in time t*

$\mathcal{M}_{icr}^{t}$ is the number of households moving in/out residential units of type *c* and ownership status *r* in zone *i* due to marriage.

$\mathcal{S}_{icr}^{t}$ is the number of households moving in/out residential units of type *c* and ownership status *r* in zone *i* due to divorce/separation

B_{icr}^{t} is the number of households moving in/out residential units of type *c* and ownership status *r* in zone *i* due to giving birth

$\mathcal{D}_{icr}^{t}$ is the number of households moving out residential units of type *c* and ownership status *r* in zone *i* due to death.

Not to forget that the above process is embedded in the integration of the dynamic synthetic population and housing choice model. The synthetic population dictates the changes in social demographics, which in turn affect the willingness to move via the social demographic attributes of the housing model reflected in equation 7.

2.4 Residential choice processes

2.4.1 Imputing residential characteristics

In order to integrate a residential choice model and the activity-based model of travel demand, the current residential profile of each individual and household should be known. However, the synthesized population lacks the residential characteristics. Therefore, data fusion is required, involving in the present case merging the synthetic population with the WoON data (H'). WoON is a national sample on residential preferences and movement. The latter data set describes for a sample of the households with known socio-demographics characteristics, their type of dwelling c in which they live and their ownership status r. Hence, this data set can be used to establish the relationship between socio-demographic profiles and residential profiles, described in terms of dwelling type and ownership status. The fusion process can be described in terms of the following pseudo-code:

For h' = 1: H'

For h=1:H

If $z_h = z_{h'}$

Then $c_h=c_{h'}$ & $r_h=r_{h'}$ (3)

$H=H-\{h\}$

end

end

H' is the total number of households in WoON and h' is an index of these households, while H and h correspond to the synthesized population. The algorithm thus finds matches in socio-demographic characteristics of households in the WoON data and households in the synthetic population and fuses the housing profile of the housing data into the created synthetic population. If more than one housing type and ownership status can be matched with a specific social demographic profile, the assignment is based on proportionality. Matching was based on “household composition”, “income” and “age of children”. It should be noted that, disregarding vacancy, because the proportion of each housing type is known for each zone i, the matching procedure satisfies the constraint that the simulated number of each housing type in each zone is proportional to the existing number.

In addition to housing type and ownerships, the residential choice model that will be discussed later in detail also includes years living in the dwelling, size of the dwelling and value of the dwelling and the distance between the house and job, as explanatory variables. The first three attributes were also imputed in the process of fusion explained before from the WoON dataset. Once a household in WoOn is concatenated to the corresponding one generated by synthetic population, all characteristics of their dwelling will be merged. The levels for the above attributes are classified into the following classes: Value of property (Euros): € 74,999 or less, € 75,000 to € 149,999, € 150,000 to € 249,999, € 250,000 or more; Years of living: 5 years or less, 6 to 10 years, More than 10 years. The distance between the housing location and job was simulated by activating the work location choice module of Albatross, which consists of a sequence of inter dependent decision trees.

It should be noted that the outcome of the residential choice location, in turn, can affect the next round of synthetic population as follows; the outcome of the housing location choice model can be of three kinds: a household may decide not to move; it may decide to move outside of the study area, and it may decide to move inside the study area. In the first case, the residential profile of the household will not change. In the second case, the household and corresponding individuals should be deleted from the synthetic population, while in the third case their residential choice process should be simulated. Because housing moves often involve a change in either ownership and/or housing type, moves are simulated in terms of transition probabilities between combinations of ownership and housing type. In addition, households may move between zones in the study area and hence also change in location is simulated by a similar transition probability between various zones in Rotterdam.

2.4.2 Residential choice model

The simulation of residential choice processes is based on a residential choice model. It differentiates between the intention or willingness to move and the actual move. The intention to move is a function of the current housing satisfaction.

$$W_h^{t+1,t}(c,r) = \frac{1}{1+\exp(-(\beta_0 + \mathbf{Z}_h\boldsymbol{\theta}' + \mathbf{X}_{\boldsymbol{h}}\boldsymbol{\eta}' + \boldsymbol{A}_{\boldsymbol{q}\in\boldsymbol{h}}\boldsymbol{\delta}'))} * \phi \tag{7}$$

where,

$W_h^{t+1,t}(c,r)$ is intention to move of household *h* to move house from type *c* and ownership *r*

$\mathbf{Z}_h$ is a row vector of socio-demographic variables of household *h*.

$\mathbf{X}_{\boldsymbol{h}}$ is a row vector of housing attributes of the current house of household *h*

$\boldsymbol{\theta}'$ is a row vector of the effects of socio-demographic variables on the intention to move

$\boldsymbol{\eta}'$ is a row vector of the effects of housing attributes on the intention to move

$\boldsymbol{A}_{\boldsymbol{q}\in\boldsymbol{h}}$ is a row vector of accessibility measures of household member *q* belonging to household *h*.

$\boldsymbol{\delta}'$ is a row vector of the effects of accessibility on the intention to move

ϕ is a correction factor

The correction was introduced to accommodate findings of empirical research that on average only 50% of the households expressing the intention to move will actually move. The detailed attributes levels of the model and the estimated coefficients are listed in Table 1.

Table 1. Estimation results of intention to move in two years

Variable	Level	Coefficients
Constant		0.609***
Household Composition	Single non-worker	0.00062
	Single worker	-0.61234***
	Double one-worker	0.15504***
	Double workers	0.56857***
	Double non-workers	-0.11189*
Income	Below average	-0.28544***
	Average	-0.07207*
	More than average less than 2 times average	0.11654***
	Two times average or more	0.24097***
Age of children	No children	0.2191***
	Younger 6	-0.31794***
	6 to 11	0.3422***
	Older 12	-0.24336***
Age of respondent	34 or less	1.25007***
	35 to 54	0.15767***
	55 to 64	-0.19729***
	65 to 74	-0.36603***
	>75	-0.84442***
Number of cars	No car	0.15506***
	1 car	-0.09372
	2 or more cars	-0.06134
Gender	Male	0.00873
	Female	-0.00873
Working hours respondent	no work	0.29759***
	Less than 32 hours week	-0.62411***

	32 hours or more week	0.32652***
Working hours partner	no work	0.04611
	Less than 32 hours week	0.00026
	32 hours or more week	-0.04637
Distance to work respondent	0-5km	-0.02732
	5-10km	0.02709
	More than 10km	0.00023
Distance to work partner	0-5km	-0.0066
	5-10km	0.00682
	More than 10km	-0.00022
Type of dwelling	House	-0.99433***
	Apartment	-0.60782***
	Other	1.60215***
Years living in the dwelling	5 years or less	-0.27234***
	6 to 10	0.00324
	More than 10	0.2691***
Ownership	Owner	-0.19952***
	Renter	0.19952***
Size of the dwelling	50 m² or less	0.31483***
	50 to 99 m²	0.00332
	100 to 129 m²	-0.09323**
	130 m² or more	-0.22492**
Value of the dwelling (Euros)	74,999 or less	1.58196***
	75,000 to 149,999	-0.35713***
	150,000 to 249,999	-0.61283***
	250,000 or more	-0.612

Note: ***, **, *: Significance at 1%, 5%, 10% level

The number of residential moves from zone *i* in any given year, from housing type *c* and ownership status *r* thus equals:

$$\widetilde{M}^{t+1,t}(c,r,i) = \sum_{h\in\mathcal{H}_i} M_h^{t+1,t} \quad (c,r) \tag{8}$$

It should be noted here that in the actual simulation, the simulated latent number of moves is subject to simulation error due to the use of Monte Carlo draws.

Next, it has to be decided where the people move. Which zone, which type of house and the new ownership status need to be known. Since the actual data at the aggregated level was available about the percentage of move between zones as well as transition between the housing type and ownership status, this information was used to calibrate the residential choice model outcome in reality. More specifically, let $p(i'|i)$ and $p(c',r'|c,r)$ denote the transition probability of moving from zone i to zone i' and housing type *c* and ownership status *r* to from housing type *c'* and ownership status *r'*, then the latent number of residential moves from zone *i* in any given year, from housing type *c* and ownership status *r* to from housing type *c'* and ownership status *r'* equals:

$$\widetilde{M}^{t+1,t}(i'|i) = p(i'|i) \sum_{h\in\mathcal{H}_i} M^{t+1,t}\,(i) \quad \forall i \tag{9}$$

$$\widetilde{M}^{t+1,t}(c',r'|c,r,i) = p(c',r'|c,r) \sum_{h\in\mathcal{H}_i} M^{t+1,t}\,(c,r) \quad \forall c,r,i \tag{10}$$

These transition probabilities derived for the study area are listed in Appendix 2 and 3.

3. Integration of Albatross, housing model and dynamic synthetic population module

The simulation process starts with a baseline population generated through IPF (iterative proportional fitting) for the Rotterdam area at time period t. The cell frequency of the multiway table is obtained from the MON data, which is a travel survey data at the household level. The concept of a relation matrix was used to convert individual count data to household count data for age groups and work status. More specifically, an age-group relation matrix specifies the relation between our age groups and three household status positions for females and males. The frequencies of a similar work-status relation matrix were obtained in a similar vein. This matrix distinguishes no work, part-time work and full-time work, which are linked to the household status categories. It results in a distribution of 15 household types comprising 3×3 double work status groups, 3 single female work status groups and 3 single male work status groups (Arentze et al., 2007). Albatross simulates the activity travel pattern of this base population.

The second step is preparing the index matrix for four demographic process; marriage, divorce, death and birth. The index matrix specifies whether any of the demographic processes apply to each individual. The index matrices are generated based on the age/ gender-specific probability of each of those events. Marriage and divorce are simulated using Monte Carlo draws from empirical data published by the Central Bureau of Statistics relating the probability of marriage and divorce to gender and age, making sure that trivial but essential conditions hold; i.e. a person can only get married if he/ she is single and can divorce if (s)he has been married. The probability of giving birth, which is naturally related to females is a function of age and the number of previous kids. Finally, death is simulated based on the age and gender specific probabilities.

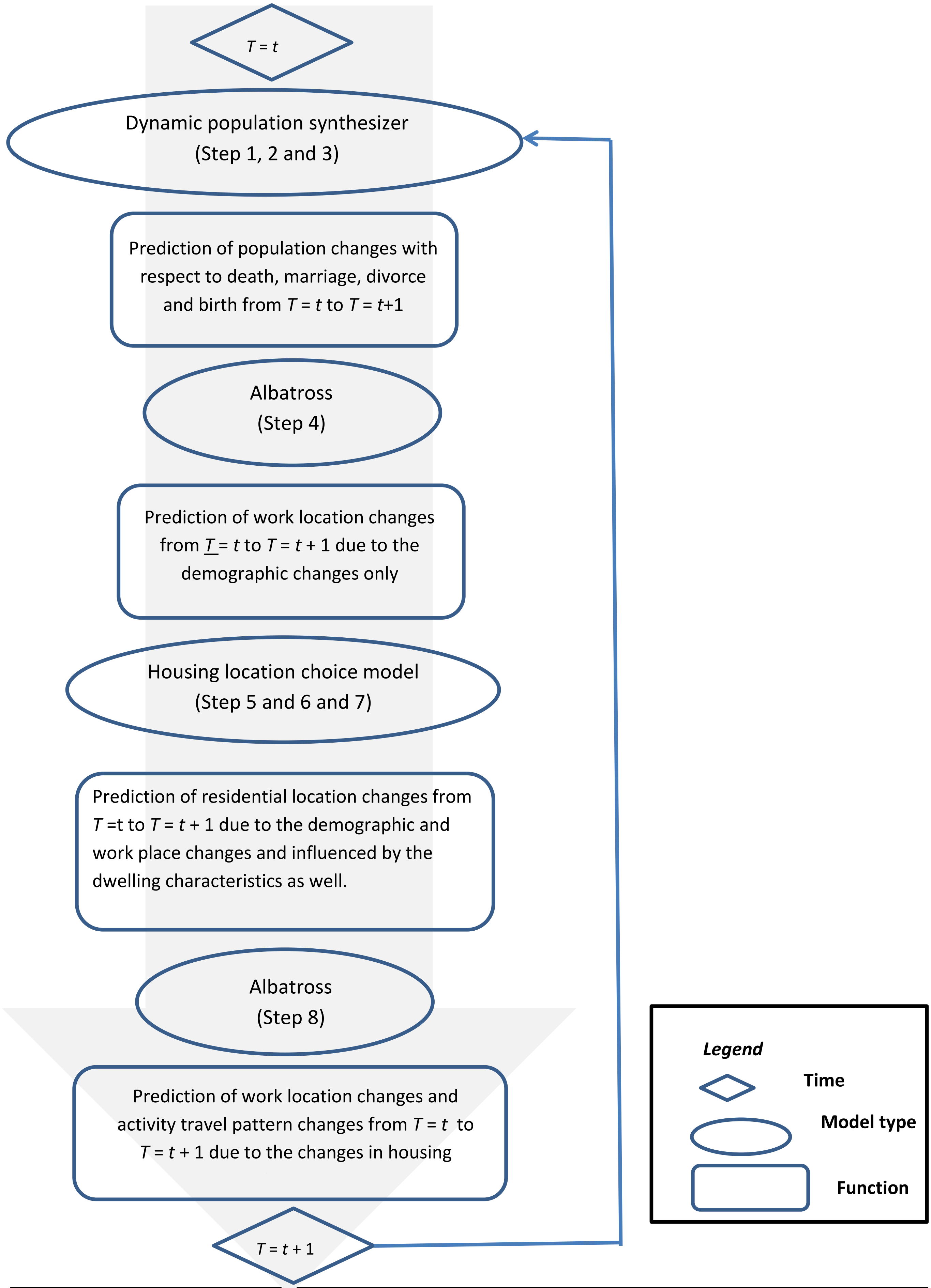
Step 3 uses the index matrix to generate the updated population. For instance, if two persons get married the id of those single household are deleted and a new household with a new composition is created and accommodated in the population. Likewise, if a couple divorces, one adult keeps the previous id of the old household and the other receives a new household id. In this case, the kids will go to mother with 90% probability and to father with 10%. Household composition for each of the separated partners will be updated. The death incident is the most straightforward phenomenon as the records of the dead person will become inactive. Finally, giving birth changes the composition of the household with adding one more person to the number of household members. Age of all household members is increased by one unit for the next year.

Step 3 leads to the population configuration at *t*+1, which is essential to activate the housing choice module. Still one essential attribute required in the housing model is missing and that is the work location as the distance between work and dwelling is one of the influential attributes in the model. To obtain this information, in step 4, the work location choice decision tables of Albatross are activated. It consists of a series of sequential decision trees which ultimately assign a work location to each individual.

Step 5 activates the housing choice model. Attributes in the residential choice model, reflected in equation 7 and in Table 1 consist of 3 kinds; Social demographic attributes which are known as the result of step 3 explained above, accessibility related attributes which are confined to the distance between house and work established in step 4, and the attributes of the dwelling which have to be assessed in step 5. The WoON data base is fused with the synthesized population with the three social demographic variables as the matching criteria. The housing type, ownership status, years of living, size and value of the dwelling are assigned to each synthesized household.

Step 6 uses the housing choice model and predicts whether the household actually moves or not. In step 7, the zone, type and ownership of the new house (if moving happens) are simulated according to the empirical transition probability. Step 8, uses the Albatross framework to predict the activity travel pattern of the new population with the updated demographic characteristics as well as the new configuration of their residential location. The location of work place is also modelled as part of this step. The process continues by going back to step 2. The evolution of activity travel behaviour at individual as well as aggregate level are valuable outcomes of many iterations to predict the changes in travel behaviour for longer periods of time as a function of demographic process and housing location choice.

Figure 1 summarizes this process.


T = t
Dynamic population synthesizer
(Step 1, 2 and 3)
Prediction of population changes with respect to death, marriage, divorce and birth from T = t to T = t+1
Albatross
(Step 4)
Prediction of work location changes from T = t to T = t + 1 due to the demographic changes only
Housing location choice model
(Step 5 and 6 and 7)
Prediction of residential location changes from T =t to T = t + 1 due to the demographic and work place changes and influenced by the dwelling characteristics as well.
Albatross
(Step 8)
Prediction of work location changes and activity travel pattern changes from T = t to T = t + 1 due to the changes in housing
T = t + 1
Legend
Time
Model type
Function

4. Illustration

In this section, we illustrate the kind of output that will be obtained as a result of the modelled demographic and residential choice processes. It should be emphasized, it is a limited illustration in the sense that long distance moves beyond Rotterdam are not incorporated in the simulations.

Tables 4.1 and 4.2 describe the distribution by ownership and type of dwelling by zones for two consecutive years. Combining these tables thus illustrates the dynamic in the population addressing the demographic and residential choice processes

Table 4.1 Aggregated results for the type and ownership before applying the residential moving model

	Own			Rent			
Zone	**House**	**Apartment**	**Other**	**House**	**Apartment**	**Other**	Total
1	962	2012	4660	951	8270	953	17808
2	2164	6112	4989	2578	22592	1061	39496
3	1296	635	1973	1081	2283	358	7626
4	1215	4846	3767	1775	17854	717	30174
5	4440	1460	2597	3178	6090	540	18305
6	2351	4278	4134	2201	16429	809	30202
7	10655	3318	3693	6904	13534	751	38855
8	5898	4295	3392	4104	15956	642	34287
9	6537	2612	3355	4491	9985	705	27685
10	4095	4833	4566	3188	17585	873	35140

Table 4.2 Aggregated results for the type and ownership after applying the residential moving model

	Own			Rent			
Zone	**House**	**Apartment**	**Other**	**House**	**Apartment**	**Other**	Total
1	1285	3735	3389	1420	7588	708	18125
2	2563	9100	3715	3249	19685	874	39186
3	1314	1429	1517	1139	2303	285	7987
4	1595	6609	2847	2138	15361	595	29145
5	4306	3087	1998	3255	5780	438	18864
6	2596	6544	3081	2588	14386	653	29848
7	10121	6259	2908	6686	12574	645	39193
8	5758	7008	2633	4303	14264	571	34537
9	6285	4683	2615	4378	9249	580	27790
10	4259	7423	3408	3527	15546	740	34903

Table 4.3 Comparison of individual characteristics before and after applying dynamic population synthesizer

	Iteration 0	Iteration 1
	Population	Population
<34	138890	138716
34<= <54	131746	131567
54<= <64	80619	80258
64<= <74	35570	34819
>74	74284	67077

Table 4.3 illustrates change in the age distribution .

Table 4.4 Comparison of household characteristics before and after applying population synthesizer

	Iteration 0	Iteration 1
Single, 0 worker	78838	82407
Single, 1 worker	64243	64344
Double, 1 worker	34288	34312
Double, 2 worker	51971	51593
Double, 0 worker	50238	45177

The final table for illustration is Table 4.4. It captures the change in household composition, based on the number of adult workers in the households.

5. Conclusions

In this report, we document work related to the extension of the Albatross model with a dynamic population synthesis that includes a model of residential moves and housing choice. The residential choice has been modelled as a binary choice between willingness to move or not. To obtain more realistic values for the actual move, the willingness to move was corrected by a correction factor based on empirical analysis.

The dynamic synthetic population was developed from scratch by considering the probability of a couple of life events as a function of social demographic characteristics. Those probabilities were extracted from the census.

In order to connect the synthesized population and the residential choice model, the WoON database was used. For linking the outcome of the residential choice and the movement between different house types and zones, BAG and the Rotterdam database were used. These two newly developed components were linked to Albatross in order to evaluate the long term evolution of activity travel patterns as a result of the aforementioned life cycle events and residential movement.

These additions offer better opportunities for the application of the model. First, residential choice is a more integral part of the simulation of activity travel patterns. Previous research has indicated residential and job locations are the pegs around which people organise their daily life and travel. This integration acknowledges this important link. Secondly, the dynamic synthesizer ensures that forecasts are consistent.

Future research might expand this development. In addition, other lifecycle events could be simulated in a similar manner. Budget permitting, individual level data would improve the estimation of the model.

Annex I. References

Arentze, T.A. and H.J.P. Timmermans (2004), A learning-based transportation oriented simulation system, *Transportation Research B*, 38, 613-633.

Rasouli, S. and H.J.P. Timmermans (2014), Activity-based models of travel demand: Promises, progress and prospects, *International Journal of Urban Sciences*, 18, 31-60.

Timmermans, H.J.P. (2006), The saga of integrated land use and transport modelling: How many more dreams before we wake up? In: K. Axhausen (ed.), *Moving Through Nets: The Physical and Social Dimensions of Travel*, Elsevier, Oxford, pp. 219-239.

Verhoeven, M., T.A. Arentze, H.J.P. Timmermans and P.J.H.J. van der Waerden (2005), Modeling impact of key events on long-term transport mode choice decisions: Decision network approach using event history data, *Transportation Research Record*, 1926, 106-114.

Wegener, M. (2004), Overview of land use transport models. In: D.A. Hensher and K. Button (eds.), *Transport Geography and Spatial Systems. Handbook in Transport*. Pergamon Press, Kidlington, UK., pp. 127-146.

Appendix 1: Distribution dwelling types per zone

Zone	House	Apartment	Other
(1) Rotterdam Centrum	7.3%	65.2%	27.4%
(2) Delfshaven	7.1%	79.5%	13.4%
(3) Overschie	25.4%	44.8%	29.8%
(4) Noord	4.5%	83.5%	12.0%
(5) Hillegersberg-Schiebroek	36.0%	49.1%	14.9%
(6) Kralingen-Crooswijk	9.9%	76.4%	13.7%
(7) Prins Alexander	39.1%	50.4%	10.4%
(8) Feijenoord	23.5%	66.4%	10.2%
(9) IJsselmonde	32.5%	54.5%	13.0%
(10) Charlois	15.3%	71.6%	13.0%

Source: BAG 2012

Appendix 2: Movements between zones in Rotterdam

Zone	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
(1) Rotterdam Centrum	**37%**	14%	2%	10%	6%	11%	6%	7%	2%	4%
(2) Delfshaven	7%	**48%**	3%	9%	3%	7%	6%	7%	4%	8%
(3) Overschie	3%	14%	**43%**	7%	6%	5%	6%	6%	5%	4%
(4) Noord	7%	13%	4%	**36%**	9%	12%	7%	5%	2%	4%
(5) Hillegersberg-Schiebroek	5%	6%	4%	9%	**51%**	7%	10%	4%	2%	3%
(6) Kralingen-Crooswijk	12%	10%	2%	11%	4%	**44%**	8%	4%	2%	3%
(7) Prins Alexander	4%	6%	2%	5%	5%	7%	**61%**	4%	3%	4%
(8) Feijenoord	4%	8%	1%	4%	2%	4%	4%	**46%**	12%	16%
(9) IJsselmonde	2%	5%	1%	2%	2%	3%	6%	15%	**50%**	13%
(10) Charlois	2%	8%	2%	3%	2%	3%	4%	17%	11%	**50%**

Source: Rotterdamcijfers.nl

Appendix 3: Transition probabilities: Ownership status and dwelling type

		Owner future dwelling			Renter future dwelling			Sum
		House	Apartment	House	Apartment	House	Apartment	
Owner current dwelling	House	8.9%	39.4%	3.4%	27.1%	20.7%	0.5%	100%
	Apartment	6.0%	18.5%	1.1%	39.7%	33.2%	1.6%	100%
	Other	10.5%	21.1%	5.3%	21.1%	36.8%	5.3%	100%
Renter current dwelling	House	16.7%	56.8%	7.3%	8.5%	10.6%	0.1%	100%
	Apartment	8.4%	63.4%	2.3%	8.6%	17.1%	0.2%	100%
	Other	9.8%	61.2%	17.3%	1.9%	9.3%	0.5%	100%

Source WoON 2012

Appendix 4.1: Dynamic synthetic population code

```
load Dynamic.mat

    % Cpop: Current population (individual)
    %%% 1: HHID (#)
    %%% 2: ID (#)
    %%% 3: Age (#)
    %%% 4: Gender (1:male/2:female)
    %%% 5: Status (1:married/2:single)
    %%% 6: Position (1:Adult/2:child)
    %%% 7: Live (1:Alive/2:Death)
    %%% 8: Work Status(1:Worker/2:Non-worker)

    % HHH: Current household
    %%% 1: HHID (#)
    %%% 2: Nrhhold (#)
    %%% 3: nrchild (#)
    %%% 4: Composition (0:Single, no worker/1:Single, one worker/2:Double,
    %%% one worker/3:Double, two workers/4:Double, no worker)

    % Index
    %%% 1: Index1 = married (1/0)
    %%% 2: Index2 = divorce (1/0)
    %%% 3: Index3 = birth (1/0)
    %%% 4: Index4 = death (1/0)

    %%Make Dead people Population Matrix
    Dpop=Cpop;
    Dpop=sortrows(Dpop,+7);
    Dpop(Dpop(:,7)==1,:)=[];

    %%Remove Dead People from Current Population
    Cpop=Cpop;
    Cpop=sortrows(Cpop,+7);
    Cpop(Cpop(:,7)==2,:)=[];

    %% Numbers
    Nc=size(Cpop,1); % Number of current population
    Nh=size(HHH,1); % Number of current household
    Nb=sum(Index(:,3),1); % Number of birth
    Nd=sum(Index(:,4),1); % Number of death
    %Nm=sum(Index(:,1),1); % Number of marriage
    %%% Index5: combination of age and death
    Index(:,5)=times(Cpop(:,6), Index(:,4));

    %%%%Household Index
    Index_H=zeros(Nh,5);
    Index_H(1,:)=Index(1,:);
    m=1;
    for n=2:1:Nc
        if Cpop(n,1)==Cpop(n-1,1)
            Index_H(m,:)=Index_H(m,:)+Index(n,:);
        else
            m=m+1;
            Index_H(m,:)=Index_H(m,:)+Index(n,:);
```

```
        end
    end

    %% Individual
    %%% birth
    Npop_b=[Cpop Index(:,3)]; % matrix combining Npop and birth index
    Npop_b=sortrows(Npop_b,-9); % sort rows for 9th column from larger to smaller
number
    Npop_b(Nb+1:end,:)=[]; % remove the rows of 0 baby
    Npop_b(:,end)=[]; % remove the 8th column (index) from matrix
    T=max(Cpop(:,2)); % generate Sds for new babies
    for n=1:1:Nb
        Npop_b(n,2)=T+1; % ID
        Npop_b(n,3)=0; % Age
        Npop_b(n,4)=round(rand()+1); % Gender
        Npop_b(n,5)=2; % Status
        Npop_b(n,6)=2; % Position
        Npop_b(n,7)=1; % Live
        Npop_b(n,8)=2; %Non-worker
        T=T+1;
    end

    %%%Marriage
    %{
    %%%% finding single hholds who got married
    Pop_m=[Cpop(:,1) Cpop(:,2) Cpop(:,4) Index(:,1)]; % merge Cpop hhid, id, gender
with Index marriage
    Npm=size(Pop_m,1); % Number of merged population

    Pop_m2=zeros(Npm,5);

    for n=1:1:Npm
        Pop_m2(n,1)=Pop_m(n,1);
        Pop_m2(n,2)=Pop_m(n,2);
        Pop_m2(n,3)=Pop_m(n,3);
        Pop_m2(n,4)=Pop_m(n,4);

        if Pop_m2(n,3)==1 && Pop_m2(n,4)==1; %male got married
            Pop_m2(n,5)=1;
        end
        if Pop_m2(n,3)==2 && Pop_m2(n,4)==1; %female got married
            Pop_m2(n,5)=2;
        end
    end

        %%%% pick Id of males getting married
    Pop_m2_male=Pop_m2;
    Pop_m2_male=sortrows(Pop_m2_male,+5);
    Pop_m2_male(~Pop_m2_male(:,5),:)=[];
    Pop_m2_male(Pop_m2_male(:,5)==2,:)=[];
    Pop_m2_male(:,3:end)=[];
    Pop_m2_male(:,3)=rand(size(Pop_m2_male,1),1); %randomize the ids of males
    Pop_m2_male_rand = sortrows(Pop_m2_male,3);
    Pop_m2_male_rand(:,3)=[];


        %%%% pick Id of females getting married
```

```
    Pop_m2_female=Pop_m2;
    Pop_m2_female=sortrows(Pop_m2_female,+5);
    Pop_m2_female(Pop_m2_female(:,5)<=1,:)=[];
    Pop_m2_female(:,3:end)=[];
    Pop_m2_female(:,3)=rand(size(Pop_m2_female,1),1); %randomize the ids of females
    Pop_m2_female_rand = sortrows(Pop_m2_female,3);
    Pop_m2_female_rand(:,3)=[];

        %%%%match making males females
    N_m2_male=size(Pop_m2_male_rand,1);
    N_m2_female=size(Pop_m2_female_rand,1);
    if N_m2_male>N_m2_female
        Pop_m2_couple=[Pop_m2_male_rand(1:N_m2_female,:) Pop_m2_female_rand];
    else
        Pop_m2_couple=[Pop_m2_male_rand Pop_m2_female_rand(1:N_m2_male,:)];
    end
    %}
    Pop_m=[Cpop(:,1) Cpop(:,2) Index(:,1)]; % merge Cpop hhid, id, with Index
partner
    for m=1:1:Nc %turn row no to id
        if Index(m,1)~=0
            Pop_m(m,3)=Cpop(Index(m,1),2);
        end
    end
    Pop_m=sortrows(Pop_m,3); % sort rows for 3th column from larger to smaller
number
    Pop_m(1:max(find(Pop_m(:,3)==0)),:)=[]; % remove the rows
    Pop_m=sortrows(Pop_m,2);

    Pop_couple0=[Pop_m(:,2) Pop_m(:,3)];
    Pop_couple=Pop_couple0;
    for n=1:1:size(Pop_couple,1)
        if Pop_couple(n,1)>Pop_couple(n,2)
            Pop_couple(n,1)=Pop_couple0(n,2);
            Pop_couple(n,2)=Pop_couple0(n,1);
        end
    end
    Pop_couple=unique(Pop_couple,'rows');

    TH=max(Cpop(:,1)); % generate household ID
    for n=1:1:size(Pop_couple,1)
        TH=TH+1;
        Pop_couple(n,3)=TH;
    end
    Pop_mn=[Pop_couple(:,1) Pop_couple(:,3);Pop_couple(:,2) Pop_couple(:,3)];
    Pop_mn=sortrows(Pop_mn,1);

    Npop_marry=[Pop_m(:,1) Pop_mn]; %married population

    %%%Divorce
    HHH_div =[HHH Index_H(:,2)];
    HHH_div(Index_H(:,2)==0,:)=[]; %hholds divorced
    Ndiv=size(HHH_div,1);
    HHH_div_sep_ID=zeros(Ndiv,2); %hhold member who separates to another hhhold id
    HHH_div_stay_ID=zeros(Ndiv,2); %hhold member staying at same hhold id with
other members of hhhold (if there is)
```

```
for n=1:1:Ndiv
    Atemp=ismember(Cpop(:,1),HHH_div(n,1),'rows');
    Atemp2=find(Atemp==1);
    A=zeros(size(Atemp2,1),8);
    for m=1:1:size(Atemp2,1)
        A(m,:)=Cpop(Atemp2(m,1),:);
    end
    A=sortrows(A,-5);
    if rand()<0.9 % ratio (probability) of female taking baby: Female take baby
        if A(end-1,4)==2 % if respondent one before end is female
            HHH_div_stay_ID(n,1)=A(end-1,1); %staying female hhid
            HHH_div_stay_ID(n,2)=A(end-1,2); %staying female id
            HHH_div_sep_ID(n,2)=A(end,2); %seperating male id
            TH=TH+1;
            HHH_div_sep_ID(n,1)=TH; %seperating male getting new hhold id
        else % if respondent one before end is male = end is female
            HHH_div_stay_ID(n,1)=A(end,1); %staying female hhid
            HHH_div_stay_ID(n,2)=A(end,2); %staying female id
            HHH_div_sep_ID(n,2)=A(end-1,2); %seperating male id
            TH=TH+1;
            HHH_div_sep_ID(n,1)=TH; %seperating male getting new hhold id
        end
    else % Male takes baby
        if A(end-1,4)==1 % if respondent one before end is male
            HHH_div_stay_ID(n,1)=A(end-1,1); %staying male hhid
            HHH_div_stay_ID(n,2)=A(end-1,2); %staying male id
            HHH_div_sep_ID(n,2)=A(end,2); %seperating female id
            TH=TH+1;
            HHH_div_sep_ID(n,1)=TH; %seperating female getting new hhold id
        else % if respondent one before end is female = end is male
            HHH_div_stay_ID(n,1)=A(end,1); %staying male hhid
            HHH_div_stay_ID(n,2)=A(end,2); %staying male id
            HHH_div_sep_ID(n,2)=A(end-1,2); %seperating female id
            TH=TH+1;
            HHH_div_sep_ID(n,1)=TH; %seperating female getting new hhold id
        end
    end
    clear Atemp Atemp2 A
end

%%% dead people's hhold
HHH_death =[HHH Index_H(:,4)];
HHH_death(Index_H(:,4)==0,:)=[]; %hholds without death event are erased
Ndeath=size(HHH_death,1);
HHH_death_stay_ID=zeros(Ndeath,2); %any hhold member who stays
Popdeath=[Cpop Index(:,4)];
for n=1:1:Ndeath
    Atemp=ismember(Popdeath(:,1),HHH_death(n,1),'rows');
    Atemp2=find(Atemp==1);
    A=zeros(size(Atemp2,1),9);
    for m=1:1:size(Atemp2,1)
        A(m,:)=Popdeath(Atemp2(m,1),:);
    end
    A_couple=ismember([A(:,5) A(:,9)],[1 1],'rows');

    if sum(A_couple,1)==1
        for m=1:1:size(Atemp2,1)
            if A(m,9)==0 && A(m,5)==1
```

```
                    HHH_death_stay_ID(n,1)=A(m,1);
                    HHH_death_stay_ID(n,2)=A(m,2);
                end
            end
        end
    end

    %%%change civil status of individuals
    Npop=zeros(Nc,7);
    for n=1:1:Nc % n increases 1 by 1 from 1 to Nc (number of current pop)
        Npop(n,1)=Cpop(n,1);
        Npop(n,2)=Cpop(n,2);
        Npop(n,3)=Cpop(n,3)+1;
        Npop(n,4)=Cpop(n,4);
        Npop(n,5)=Cpop(n,5);
        Npop(n,6)=Cpop(n,6);
        Npop(n,7)=Cpop(n,7);
        Npop(n,8)=Cpop(n,8);
        if Index(n,4)==1 %died, 7th column becomes death
            Npop(n,7)=2;
        end

        for m=1:1:size(Npop_marry) %single to married
            if Npop(n,2)==Npop_marry(m,2)
                Npop(n,1)=Npop_marry(m,3);
                Npop(n,5)=1;
            end
        end

        for m=1:1:Ndiv
            if Npop(n,2)==HHH_div_stay_ID(m,2) %married to single
                Npop(n,5)=2;
            end
        end

        for m=1:1:Ndiv
            if Npop(n,2)==HHH_div_sep_ID(m,2) %married to single
                Npop(n,1)=HHH_div_sep_ID(m,1);
                Npop(n,5)=2;
            end
        end

        for m=1:1:Ndeath
            if Npop(n,2)==HHH_death_stay_ID(m,2)
                Npop(n,5)=2;
            end
        end

    end

%    Npop=[Npop Index(:,4)]; % death matrix combining Npop and death index
%    for n=1:1:Nc
%        if Npop(n,8)==1
%            Npop(n,7)=2;
%        else
%            Npop(n,7)=1;
%        end
```

```
 %   end
 %   Npop(:,end)=[]; % remove the 8th column (index) from matrix
    %%Npop=sortrows(Npop,-7); %sort rows for 7th column from larger to smaller
    %%Npop(1:Nd,:)=[]; % remove the rows with value 1 for death people

    %%% Making new population
    Npop=[Npop;Npop_b;Dpop;]; % combine New pop, baby pop and dead pop
    Npop=sortrows(Npop,1); %sort rows for hhold id

    %% Household
    %%%% predicting household characteritics in terms of current households
    NHHH_C=HHH;
    for n=1:1:Nh
        if Index_H(n,5)==2; %if a person dying is child, decrease nr of child
%(Index_H(n,5)<=18) && (Index_H(n,5) > 0) ; %if a person =<18 dies, decrease
nrchild
            NHHH_C(n,3)=HHH(n,3)-1;
        else
            NHHH_C(n,3)=HHH(n,3);
        end
        if Index_H(n,4)==1; % if there is a death in hhold, then decrease nr hhold
            NHHH_C(n,2)=HHH(n,2)-1;
        else
            NHHH_C(n,2)=HHH(n,2);
        end
        if Index_H(n,3)==1; %if there is a birth, then increase nr hhold
            NHHH_C(n,2)=HHH(n,2)+1;
            NHHH_C(n,3)=HHH(n,3)+1;
        end
        for m=1:1:size(Npop_marry,1)%if single marries, decrease nr hhold
            if HHH(n,1)==Npop_marry(m,1)
                NHHH_C(n,2)=NHHH_C(n,2)-1;
            end
        end
        for m=1:1:size(HHH_div,1)%if divorce, decrease nr hhold
            if HHH(n,1)==HHH_div(m,1)
                NHHH_C(n,2)=NHHH_C(n,2)-1;
            end
        end
    end

    %%%% predicting household characteritics in terms of new households
    NHHH_M=Npop_marry(:,1); % from marriage
    NHHH_M(:,2)=2;
    NHHH_M(:,3)=0;

    NHHH_D=HHH_div_sep_ID(:,1); % from divorce
    NHHH_D(:,2)=1;
    NHHH_D(:,3)=0;

    %%% Merge household
    NHHH=[NHHH_C;NHHH_M;NHHH_D];

    %%% Delete household
    NHHH(NHHH(:,2)==0,:)=[];
    %%% Sort
    NHHH=sortrows(NHHH,1);
```

```
function result = true_or_false(pop,sourceStat)
    %type:  1: dead or alive
    %       2: give birth or not
    %       3: marry or not
    %       4: divorce or not
    %       5: Marry or not ids
    pIDPos          = 1;
    agePos          = 3;
    genderPos       = 4;
    ageClassPos     = 5;        %for marriage
    numOfBabyPos    = 6;        %for woman
    ageClassDivPos  = 7;        %for divorce
    statusPos       = 8;        %single or others...
    ageClassMarrMatchPos   = 9;    %marriage match


    age             = pop(:,agePos);
    gender          = pop(:,genderPos);
    ageClass        = pop(:,ageClassPos);
    numOfBaby       = pop(:,numOfBabyPos);
    ageClassDiv     = pop(:,ageClassDivPos);
    status          = pop(:,statusPos);
    ageClassMarrMatch   = pop(:,ageClassMarrMatchPos);

    result          = zeros(size(pop,1),4);
    checkArr        = zeros(size(pop,1),1);
    %dead: age, male_prob,female_prob
    %birth:age, prob
    %marry:ageclass, male_prob,female_prob
    %divorce:ageclass, male_prob,female_prob
   for i=1:size(pop,1)
       %to 1
       chanceOfDead      = sourceStat.dead(age(i)+1,gender(i)+1);
       tmpR = rand;
       if tmpR < chanceOfDead
            result(i,1) = 1;
       end

       %to 2
       if gender(i) == 2 && age(i) >= 15 %married?! %must be a woman and age over
15
           ageP          = min(age(i)-15+1,36);
           numOfBabyI    = numOfBaby(i);
           babyP         = (ageP-1)*4 + numOfBabyI + 1;
           chanceOfBirth     = sourceStat.birth(babyP,1);
           if rand < chanceOfBirth
               result(i,2)   = 1;
           end
       end

       %to 3
       if age(i) >= 18 && status(i) == 2 %must be over 18 %Should be also single!!
           chanceOfMarr      = sourceStat.marr(ageClass(i),gender(i));
           if rand < chanceOfMarr
               if gender(i) == 1
                   result(i,3)   =
newAssign(sourceStat.marrMatch(ageClassMarrMatch(i),:));
```

```
%checkArr(i)=newAssign(sourceStat.marrMatch(ageClassMarrMatch(i),:));
               else
                   result(i,3)   =
newAssign(sourceStat.marrMatch(:,ageClassMarrMatch(i)));

%checkArr(i)=newAssign(sourceStat.marrMatch(:,ageClassMarrMatch(i)));
               end
           end
       end
       %to 4
       if age(i) >= 18 && status(i) == 1 %must be over 18 %should be married!!!
           chanceOfDiv     = sourceStat.div(ageClassDiv(i),gender(i));
           if rand < chanceOfDiv
               result(i,4)  =1;
           end
       end
   end
    marrPosMale     = find(result(:,3) > 0 & gender == 1);      %pos
    marrPosFem      = find(result(:,3) > 0 & gender == 2);      %pos
    maleAgeclass    = ageClassMarrMatch(marrPosMale);
    femaAgeClass    = ageClassMarrMatch(marrPosFem);
    toFemalAgeClass = result(marrPosMale,3);                    %
    toMaleAgeClass  = result(marrPosFem,3);                     %

    femaAgeClassFlag    = zeros(size(femaAgeClass,1),1);
    for m =1 : length(marrPosMale)
        %s1=length(femaAgeClass)
        %s2=length(femaAgeClassFlag)
        %pause
        pos      = find(femaAgeClass == toFemalAgeClass(m) & femaAgeClassFlag == 0);
        if ~isempty(pos)
            randVal = ceil(rand*length(pos));
            result(marrPosMale(m), 5)          = marrPosFem(pos(randVal));
            result(marrPosFem(pos(randVal)),5)  = marrPosMale(m);
            femaAgeClassFlag(pos(randVal))      = 1;

%             s0=marrPosMale(m)
%             s1=result(marrPosMale(m), 5)
%             s2=marrPosFem(pos(randVal))
%             s3=result(marrPosFem(pos(randVal)),5)
           % pause

        end
    end
end
 function c=newAssign(a)
    b   = cumsum(a);
    tmp = rand;
    c   = length(a);
    for i=1:length(a)
        if tmp <= b(i)
            c = i;
            break;
        end
    end
 end
```

Appendix 4.2: Residential movement code

```
%seed1 is WOON: (1)concat, (2) type
%zone is CBS: type of dwelling (3 types) per zone(10 zones)
%syn is synthetic population: (1) concat, (2) zone, (3)type, (4)position woon

%1 is concat - input
%2 is zone - input
%3 is type - result
%4 is ID woon - result

function syn=two_d_assign(seed1,zone,syn)
    uniComb     = unique(seed1(:,1));
    seed1Update = get_seed(seed1,uniComb);
    seed1UpLen  = size(seed1Update,1);
    zoneLen     = size(zone,1);
    k=0;
    l=0;
    bigM        = zeros(seed1UpLen,3,zoneLen);

    for i=1:size(zone,1)
        eachZone     = zone(i,:);
        bigM(:,:,i) = each_zone_two_d_assign(seed1Update,eachZone,uniComb);
    end

    syn(:,3:4)  = 0;
    for i=1:size(syn,1)
        %i
        %s0=uniComb(1:10)
        %s1=syn(i,1)
        pos     = find(uniComb==syn(i,1),1) ;
        prob    = bigM(pos,:,syn(i,2));
        prob    = cumsum(prob)./sum(prob);


        tmp     = rand;
        if tmp <= prob(1)
            syn(i,3)    = 0;
        elseif  tmp <= prob(2)
            syn(i,3)    = 1;
        else
            syn(i,3)    = 2;
        end
        pos     = find(seed1(:,1) == syn(i,1) & seed1(:,2) == syn(i,3));
        if ~isempty(pos)
            syn(i,4)    = pos(ceil (rand*length(pos)));
        else
            syn(i,4)    = ceil (rand * size(seed1,1));
        end
        if mod(i,5000)==0
            disp('good')
        end

    end
end
```

```
function reDefinedD = each_zone_two_d_assign(seed1Update,eachZone,uniComb)

    reDefinedD  = seed1Update;
    seedRow     = sum(reDefinedD,2);
    seedCol     = sum(reDefinedD,1);
    targetRow   = seedRow;
    targetCol   = sum(seedCol) .* eachZone;

    while ~converge(reDefinedD, targetRow, targetCol)
        %row adjustment
        for i=1:length(uniComb)
            reDefinedD(i,:)     =  reDefinedD(i,:) .* ( targetRow(i) / sum(
reDefinedD(i,:) ) ) ;
        end
        %column adjustment
        for i=1:3
            reDefinedD(:,i)     =  reDefinedD(:,i) .* ( targetCol(i) / sum(
reDefinedD(:,i) ) ) ;
        end
    end
end

function flag = converge(reDefinedD, targetRow, targetCol)
    seedRow     = sum(reDefinedD,2);
    seedCol     = sum(reDefinedD,1);
    tmpV        = dot ( (seedRow - targetRow), (seedRow - targetRow) ) + ...
                  dot ( (seedCol - targetCol), (seedCol - targetCol) );
    base        = dot (targetRow, targetRow) + dot (targetCol, targetCol);

    if sqrt(tmpV/base)  < 0.005
        flag = 1;
    else
        flag = 0;
    end
    sqrt(tmpV/base);
end

function seed1Update = get_seed(seed1,uniComb)
    seed1Update     = zeros(size(uniComb,1),3); %for three column 0, 1 2

    for i=1:length(uniComb)
        tmpP0       = length( find( seed1(:,1) == uniComb(i) & seed1(:,2) == 0 ) );
        tmpP1       = length( find( seed1(:,1) == uniComb(i) & seed1(:,2) == 1 ) );
        tmpP2       = length( find( seed1(:,1) == uniComb(i) & seed1(:,2) == 2 ) );
        seed1Update(i,:)        = [tmpP0, tmpP1, tmpP2];
    end
end

function syn = mc(syn,refMoveType,refMoveZone)
    syn(:,5:8)=0;
    %1 is prob
    %2 is zone current
    %3 is type
    %4 is own
    %5 is move or not
    %6 is own future
    %7 is type future
```

```
    %8 is zone future
    for i=1:size(syn,1)
        if rand < syn(i,1)
            syn(i,5)    = 1;
            pos1            = 1+syn(i,3)+3*(syn(i,4)+1 - 1);
            newPos1     = newAssign(refMoveType(pos1,:));
            syn(i,6)    = ceil(newPos1/3);
            syn(i,7)    = newPos1 - (syn(i,6)-1)*3;
            %
            pos2        = syn(i,2);
            newPos2     = newAssign(refMoveZone(pos2,:));
            syn(i,8)    = newPos2;
        end
    end
end

function c=newAssign(a)
    b   = cumsum(a);
    tmp = rand;
    c   = length(a);
    for i=1:length(a)
        if tmp <= b(i)
            c = i;
            break;
        end
    end
end
```