The Zurich Case Study of UrbanSim

Author(s):
Axhausen, Kay W.; Schirmer, Patrick; Zöllig, Christof; Müller, Kirill; Bodenmann, Balz Reto

Publication Date:
2011

Permanent Link:

https://doi.org/10.3929/ethz-a-006620683 →

The Zurich Case Study of UrbanSim

Patrick Schirmer

Christof Zöllig

Kirill Müller

Balz R. Bodenmann

Kay W. Axhausen

ERSA 2011 June 2011

Institut für Verkehrsplanung und Transportsysteme
Institute for Transport Planning and Systems

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Contents

ERSA 2011

The Zurich Case Study of UrbanSim

Patrick Schirmer, Christof Zöllig, Kirill Müller, Balz R. Bodenmann, Kay W. Axhausen
IVT
ETH Zürich
8093 Zürich
phone: +41-44-633 62 58
fax: +41-44-633 10 57
{schirmer,zoellig,mueller,bodenmann,axhausen}@ivt.baug.ethz.ch

June 2011

Abstract

This paper reports on implementing the land use model *UrbanSim* in the canton of Zurich, as one of three case studies of the SustainCity project. SustainCity aims at advancing the modelling of interactions between land use and transport and adapting the software *UrbanSim* to the European context. The result will be *UrbanSimE*, a tool intended to be used by European governmental institutions and planning agencies. The paper documents the data acquisition, processing and definitions of the Zurich case study and the technical approach to create a base year for running a "first run" of *UrbanSimE*, *i.e.*, a very basic initial operationalisation.

Keywords

UrbanSim; Urban Simulation; SustainCity; Zurich Case Study

1 Introduction

UrbanSim (Waddell, 2002) is an open-source software, simulating land use development in cities based on the choices of households, businesses, land owners and developers, interacting in urban real estate markets. UrbanSim can be coupled with a multimodal transport model. Previous implementations of UrbanSim used grid cells as spatial resolution for its choice models; however, the latest version of the UrbanSim software also allows simulating at parcel resolution.

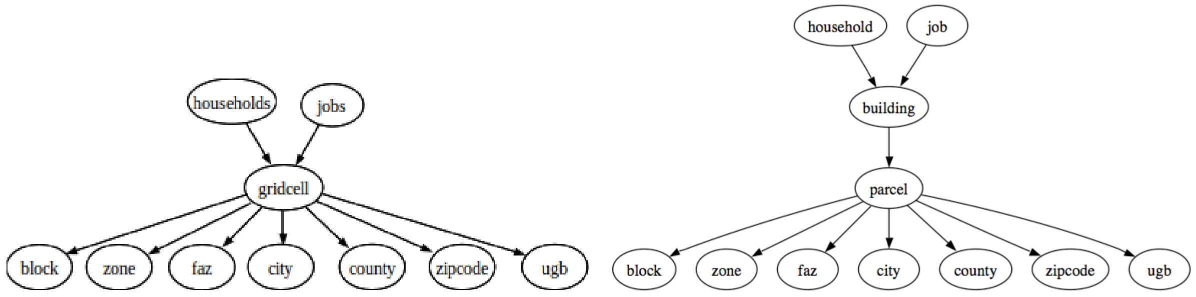
SustainCity (2011) is an EU-funded project with twelve European research institutions. The aim of SustainCity is to extend the integrated land use model UrbanSim to the so-called version UrbanSimE, being adapted to the European context of cities: new choice models will be calibrated and additional models for households, demographics and firmographics will be created. UrbanSim will be used in three case studies: Brussels, Paris and Zurich. The work necessary to set up an UrbanSim environment in these urban areas, as well as the results of the simulation process will be compared within the project. In addition the integration of various traffic models and the integration of new models on demographics, within-household decisions and developers' behaviour will be expertised. All three case studies have a running UrbanSim environment of previous studies at their disposal. For Zurich this has been the Project "Zukunft urbaner Kulturlandschaften" (Löchl *et al.*, 2007a), which was based on the grid cell version of UrbanSim. A main goal of the case study is to migrate the grid cell version to a parcel version, which implies a fundamental change in the model structure shown in Figure 1. Further targets of the case study are:

- Implement a more detailed data structure,
- Improve the existing models and extend them through new models,
- Implement a workflow based on open-source software only,

Figure 1: Basic model structures of UrbanSim

(a) Basic model structure of grid cell version

(b) Basic model structure of parcel version



Source: Waddell (2010)

- Create a scripted processing framework that allows for reproduction.

2 Basis data

2.1 Simulation area and time period

With the change of the spatial reference from grid cell to parcel, `UrbanSim` demands additional and more detailed input data. In the ZUK project, relevant information of the basis data has been grouped in form of aggregated information per grid cell of 100x100m, *e.g.*, possible constructions per use. In the parcel approach this kind of aggregation is not possible and is replaced through linking various main tables.

Therefore, to fit into the new structure, the migration of the ZUK project to the new parcel based approach of SustainCity can only be performed through acquisition of new data and new processing. This processing has been created in form of a flexible work frame that allows for reproduction and is described in Section 3.

We choose the canton of Zurich as simulation area. Including the neighbouring cantons would allow to observe the interactions to those, as well as reduce boundary effects. Unfortunately, it was unrealisable to gather the necessary data and joining them on this high level of detail, as the data structure of the cantons differ too much.

For reasons of better data disposability we decided to shift the base year of the simulation from 1997, as used in the ZUK project, to the year 2000. The simulation period is set to be from year 2000 to year 2030 and will use the latest register information of 2010 for validation.

2.2 Data used in the Zurich case study

Gathering data on this high level of detail is a difficult and time consuming process: access is restricted for security reasons, sometimes it is just not available and additionally it is hosted by a large number of different data owners. For details on these issues and related costs we refer to previous works (Schirmer, 2010). The data we obtained and that now has been used for the Zurich case study can be grouped into geometries, cross sectional and longitudinal information, which will be discussed here briefly. Table 1 presents an overview.

Geometries are used as spatial entities in the simulation process of `UrbanSim`. Within the data processing they often form the only option to join different data through spatial joins. They are represented through the cadastral data of the municipalities in the canton of Zurich, but are not

Table 1: Overview of the data used in the Zurich case study

Entity described	Number of objects	Data set	Data owner	Spatial resolution	Date	Data type
Buildings	255801 (2010)	GWR-BFS	Federal Statistic Department	Addresses and coordinates [m]	2004-2010	Table
	212497 (2010)	GWR-ARV	Canton Zürich	Coordinates [m]	2010	Table
	286468 (2000)	GVZ	Cantonal Building Assurances	Addresses	2010	Table
	534594 (2010)	Vector 25	Swisstopo	Polygon in projected coordinate system	2010	GIS shape
Appartments	671169 (2010)	GWR-BFS	Federal Statistic Department	Building (via EGID)	2004-2010	Table
	478378 (2010)	GWR-ARV	Canton Zürich	Building (via EGID)	2010	Table
Entrances	261650 (2010)	GWR-BFS	Federal Statistic Department	Addresses and coordinates [m]	2004-2010	Table
	237481 (2010)	GWR-ARV	Canton Zürich	Coordinates [m] (and addresses)	2010	Table
New buildings	40849 (2000-2010)	GVZ	Cantonal Building Assurances	Addresses	2001-2010	Table
Development projects	57734 (2010)	GWR-ARV	Canton Zürich	Building (via EPROID)	1967-2015	Table
	60056 (1998-2012)	Documedia	DOCUMEDIA (Baublatt)	Addresses	1998-2012	Table
Parcels	367314 (2010)	Cadastral plans	Canton Zürich	Polygon in projected coordinate system	2005, 2007-2010	GIS shape
Soil coverage	837732 (2010)	Cadastral plans	Canton Zürich	Polygon in projected coordinate system	2005, 2007-2010	GIS shape
Households and persons	1267478 (2000)	Population census	Federal Statistic Department	Coordinates [100 m]	2000 (2010 expected soon)	Table
	5837 (2005)	Mobility micro census	Federal Statistic Department	Coordinates [100 m]	2000, 2005	Table
Enterprises (locations)	75107 (2010)	Enterprise census	Federal Statistic Department	Coordinates [m]	2001, 2005, 2008	Table

consistent for the whole simulation area. The digitisation within the 171 municipalities is an ongoing process and some smaller municipalities are not included in the latest data. Creating a data set for the base year (2000) will require some backward editing of the data: Taking the data from 2010, which is the most complete information, and deleting the buildings constructed in the period form 2000 to 2010. Other changes as the network or merging and splitting of parcels will not be accounted for, as reliable information is missing. The first setup described here will ignored these issue and use the current data on geometries as data of the base year.

Cross sectional data is microscopic information needed for the creation of agents and objects (persons, jobs, buildings, *cf.* (Schirmer, 2010) for a differentiation of agents and objects). For persons and jobs these could be based on the population census (2000), the enterprise census (2003, 2005, 2008) and the micro census of mobility behaviour (2000, 2005). As described in Section 3.7, the use of populations synthesis might avoid the work of acquisition and linking those data in future.

Data about buildings is taken from the federal register of buildings (GWR), the cantonal building assurance (GVZ) and the cadaster (footprints). The GWR is a highly detailed database including information on buildings and apartments therein. The information on apartments will help to extend the `UrbanSimE` data structure and link the households to apartments instead of buildings in the future. The GWR was originally meant to include all buildings with residential use, but is currently being extended to all buildings. We have data sets from several points in time at our disposal, but – similar to the geometries – completion is an ongoing process, so that 2010 represents the most complete data. Thus, 2010 will be used as starting point to create the year 2000 through backward editing as well.

One data set of the GVZ is a snapshot of the year 2000 including most of the buildings in the canton Zurich. In addition we possess a data set that includes all buildings newly registered from 2000-2010. Unfortunately, we do not have data on the destruction of buildings. While it is possible to create various points in time, these will include doubled buildings which have to be filtered out. In conclusion it can be said, that also the creation of a reference table for the buildings for the use in `UrbanSim` will need some further synthesis algorithms, matching the different data sets, the different years of reference and the geometries between the different data sources.

The longitudinal information is data from the statistic departments (cantonal and national). The data contains aggregated information on developments and is consistent for the years of simulation (2000-2010). Examples are the increase of the population or the amount of new construction. Some specific statistics, as probabilities of moving or changing the job, already have been collected in the previous ZUK project and can be used again (Beige and Axhausen, 2005). Various other surveys and studies have been undertaken at the IVT in the past. Although not implemented for the very first setup described here, the data of those and the resulting models will be essential for defining future models in `UrbanSimE` .

2.3 Needs of UrbanSim

In the following we describe the data needs of `UrbanSim` for the "super simplified" setup of the first run described here. To get an overview it might be helpful to categorise the data needs according to data content:

1. Simulation tables
   a) Main tables with simulated entities, *e.g.*, parcels, buildings, households, persons, jobs
   b) Travel data (zone to zone skims)
   c) Geographies for aggregation, analysis and visualization, *e.g.*, cities, zones, counties, fazes, large areas
2. Model estimation tables, *e.g.*, households and jobs
3. Definition tables
   a) Assumptions / boundary conditions
      - Annual control totals for households and employment
      - Annual relocation rates for households and employment
      - Target vacancies
      - Plan types
      - Development constraints
      - Building square feet per job
   b) Classifications
      - Building types
      - Employment sectors
      - Land use types
      - Household characteristics for household transition

The main tables contain the simulated entities. These are the largest tables. Simulated entities are parcels, buildings, households, persons and jobs. The parcels are used as spatial reference and thus as aggregation unit in respect of built space characteristics and inhabitants, and serve as atomic units for application of development constraints and ownership. The tables in Appendix A give an overview of how the main tables have been filled so far.

Definition tables define assumptions about the scenario as well as classifications. The entities in the main tables have to fit these. The `household_characteristics_for_transition` table defines population classes for which control totals can be specified. Accordingly, the classifications have to be present in the control totals table using attributes used in the `households` or `persons` table. The `development_constraints` table defines land use shares permitted on a parcel. Development constraints are assigned to a plan type (see Subsection 3.6 for further details). Land use type and constraints have to correspond. Appendix B shows where the data filled into the definition tables is derived from.

The estimations of location choice models require observations on those choices. As the census contains information on whether a household recently moved, we can create a subset of the population representing recent movers. The current home of these households is then interpreted as their location choice.

UrbanSim considers travel times in the form of a zone to zone matrix. For the first setup these travel times are imported from the four step model of the canton Zurich and will not change during the simulation. Later we will implement the multi agent simulation approach of MATSim and update the travel time data.

Besides the importance of geographies when linking the data, as described in Section 3.3, geographies are used to visualize and summarise results of the simulation through aggregation. They are included as shape files and stored in a separate folder of the base year cache. The geographies thus should be chosen by the user according to the research question. For our study we have various kinds of geographies that can be used, starting with the parcel as smallest and ending with the canton as biggest entity.

3 Data processing

3.1 Introducing a generic work frame

Creating the base year out of the original data demands for combining various tools . Although direct import to *UrbanSim* supports most common formats, *UrbanSim* does not support further post-processing in *UrbanSim* can only be done through Python within the software. Additionally all information on geometry and location will be lost as not stored in the databases of *UrbanSim*.

The previous ZUK project showed that it is useful to script all the processings to have them documented and eventually to repeat them in form of generic processes. For the SustainCity project we extended this approach to a generic framework that will allow for combination of various tools into a general processing script and to repeat any step of processing undertaken. To be consistent with the open source approach of *UrbanSim*, we integrate only open source tools into this framework.

We choose PostgreSQL with the spatial extension PostGIS as database to host all the data. The add-on PostGIS allows to implement most spatial queries and transformations into the

PostgreSQL environment. For specific GIS applications we connected GRASS-GIS to the PostgreSQL database and for quick visualisation of the data additionally Q-GIS. Statistical summaries, analyses of distributions and transformations of data could be tested and performed through linking the database to the statistic software R.

Through running the script provided on the `UrbanSim` homepage (UrbanSim, 2011), we created an empty database in PostgreSQL that includes all relevant main tables for setting up an initial `UrbanSim` environment. The advantage is, that the main tables are linked through primary keys and controlled when being filled up.

By combining the tools described above, we could fill up those tables and import them to `UrbanSim`. Using these tables as interface in combination with the generic framework described above, allows redefine and enhance the processings that have been undertaken and still begin working with a data set of reduced detail in `UrbanSim` at the same time.

3.2 Data import

We use SVN as master storage for our source data, and use the PostgreSQL environment only for intermediate storage. The data in the PostgreSQL environment can be recreated at any time directly from the master storage.

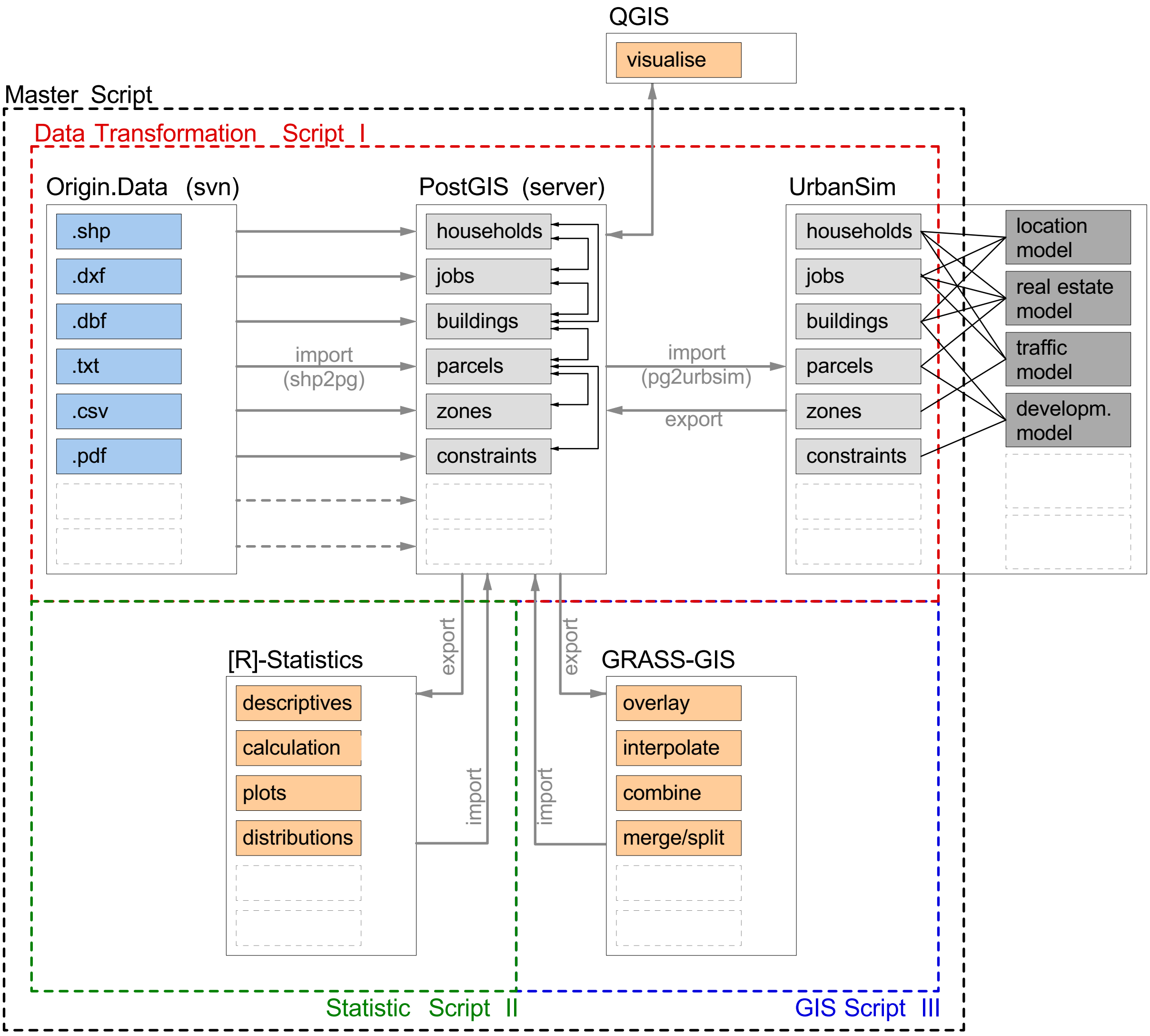
The source data comes in three different formats: tab-separated text files, ESRI shape files, and INTERLIS files, the latter being an exchange format for spatial data used mostly in Switzerland. All data is imported into a PostgreSQL/PostGIS database as a first step. The import is performed by custom configurable shell scripts called `shp2pg`, configured by (a) a master file that lists all data sources and their location in our SVN tree, and (b) several configuration files per data source that specify, *e.g.*, column names and formats (for tab-separated text files) or SQL statements specifying the post-processing. The data is imported "as is", with the only exception that a column specifying the municipality is added to those data sets that do not have such a column.

An option to filter the data by municipality has been implemented in the script. This allows to easily explore small subsets of the data, and to reduce processing times. For the first run, we stick to the city of Zurich as a subset that allows us to evaluate our approach.

3.3 Linking the data

The linkage of the data has been performed in PostgreSQL. Through **`select`** and **`insert`** queries, different basis data could be combined and filled into the template tables in PostgreSQL. The spatial extension PostGIS allows to perform joins through primary keys and as spatial join. An

Figure 2: Data processing scheme



overview of the different variables in the main tables and the links to the original data for each of those can be found in the annex of this paper. In the following we thus only discuss some methodological issues.

As the data have been provided from different sources and represent various reference years they do not have consistent primary keys. We therefore combined various methods of distinct selections for linking the data sets, in order to create a first test set of data for `UrbanSim`. Mainly to combine GWR and GVZ into the building table we had to use spatial joins. These use the coordinates of a building to be joined to the cadastral footprint of this building (done with **`join`** and `st_contains`). If several objects are placed inside this polygon, the first object found will be used. Matching data per primary keys, showed that these are not distinct for all objects. In that case a distinct selection is used that will ignore all doubled features. The third method of matching data is the match per address (city, street, house number). As no spatial information and no primary key is given, this method is the only possible approach for the GVZ data to be matched to the GWR data and from there to the geometries. This approach only provides a result for about two thirds of the data.

These first approaches allowed to roughly fill up the `UrbanSim` tables, but significantly reduced the number of object in the base year as only one method of join can be included for each variable. The future work therefore concentrates on creating a reference data set of each entity used in `UrbanSim`, *e.g.*, a buildings set combining the geometry, the GWR and the GVZ data. This will be created by combining various methods of linking the data for a variable. It thus will avoid some of the data looses experienced in our first approach and will permit to create a more complete reference data set. This combination of joining methods will demand to include a scoring function on how accurate the join is expected to be.

3.4 Geocoding data sets

For performing spatial joins, data needs to be geocoded first. Most of the data have some latitude-longitude information that could be used to create a spatial reference in PostSQL, but some only have an address.

This is most essential for the buildings, that need to combine various data sets within one table. Our first approach to solve this task, was using the GWR data set to geocode the data of the GVZ. The GWR includes coordinates and adresses, but unfortunately the data set is substantially smaller then the GVZ data. The address match will additionally only match address strings that are completely identical. As a consequence only 60% of the GVZ data could be geocoded with that approach.

Similar to the combined methods of linking the data already mentioned, we will have to combine different methods of geocoding. We expect the most consistent one to be extracting the address out of the cadastral information within the INTERLIS format. Those include addresses related to the entities of cities, street axes and building entrances. Second option could be to use the addresses registered by the Swiss Post, which is available in form of the “postmatch” data.

For both approaches we will need to define a matching algorithm, that looks for similarities and ignores “small mistakes” which might have happened through spelling mistakes or abbreviations.

3.5 Enriching the data in GIS

As mentioned above we connected two GIS systems to the PostgreSQL database for full GIS support. The previous ZUK project used ArcGIS as GIS application for the spatial transformations. To be consistent with the open source approach of `UrbanSim`, we decided to avoid the use of any commercial product.

GIS will mainly be needed to extract some information such as view sheds, slope, noise or to derive certain aspects on urban shapes. It has been effective to calculate and visualise densities of population and certain households in previous works. It is also a big advantage to perform a visual control of the content of data.

Best GIS to connect to PostgreSQL has proven to be QGIS. This one can directly be connected to the database and visualize data out of it. Unfortunately, any transformation of those data will create a shape file outside of the database, which has to be imported into the database again. Additionally, the toolbox of QGIS is relatively small for sophisticated processings, so that it can not be used for those.

One of the alternatives to QGIS would be GRASS-GIS. This GIS is widely used from an active community and has a very large set of tools, including strong raster support and 3D-visualisation. It can directly be connected to PostgreSQL and write to PostgreSQL as well. The problem of GRASS-GIS is that it uses an own topological format for its vectors instead of the simple feature approach of PostgreSQL and QGIS. This demands for time consuming and error-prone import and export of vector files into the GRASS-workspace to use the full power of the GIS-system. Raster files cannot be imported into PostgreSQL anyway, as this one is not supporting raster files currently.

First tests with GRASS-GIS in the Zurich Case Study have succeeded through using the tool `v.`**`external`** that will allow for linking to PostgreSQL and avoid import, but it has to be tested how many of the tools can be used in that mode and if the computational time rises significantly. Additionally, we lately obtained the vector data of the cadastral information as topology in form of the INTERLIS data, so that we need to test if this can be transformed to the topological GRASS format without any errors.

However, any experience that has proofed to have significant effect for the simulation process in `UrbanSim` will need to be included into the simulation environment of `UrbanSim`, which would mean to script those GIS applications in Python.

3.6 Definitions on building types, plan types, constraints and land use types

The structure of `UrbanSim` defines initial main tables and variables that are expected to be complete when exported from PostgreSQL. Although most of the variables are precise in terms of what they represent (*e.g.*, age of head of household), the definition and interpretation of some other variables is more flexible and might depend on the kind of data that is available. The definition for those can be set by the user, but needs to be consistent within a project, as they significantly influence the simulation.

Following we will describe the structure defined for building types, plan types, constraints and land use types, that has been developed for the Zurich Case Study. It should be mentioned, that this structure has not been totally included for our first version running in `UrbanSim`, but will be within the ongoing work. Figure 3 gives an overview on how these variables are interpreted within the Zurich project of SustainCity.

According to that definition building types will not only represent built volumes, but be based on the classes of soil coverage that can be found in the cadastral information. These have a higher detail then the parcel, include all the footprints of buildings (as class "buildings"), but additionally include information on the type of soil coverage, *e.g.*, pedestrian way or grass. Through defining those classes as main classes of buildings, the whole coverage of a parcel can be included into the simulation process. This will allow to implement types of open space or probability of their changes into the discrete choice models. To represent different types of uses of a building we additionally defined five subclasses of building: Residential, Commercial, Industrial, Governmental and Retail.

Summing up all the floorspace per building type (which includes the open space) and comparing their share allows to derive the land use of a parcel. We have defined nine main types of land uses, which are equal to the main classes of plan types found in Table 2.

Comparing the existing floorspace per building type to the maximum floorspace permitted within a plan type, will be a important variable for the development model of `UrbanSim` and defining development options. Table 2 gives an overview of the nine main classes of plan types and the related 20 subclasses of those.

The cantonal zoning strategy consists of categories that define constraints on the use which is permitted on a parcel. Besides, it often includes information on the height, volume or density that is permitted. Based on these definitions we interpolated the average density that it permitted for each class and included those in form of constraints. Constraints refer to construction for the five uses named above. In total we thus have 164 plan types that refer to 820 constraints.

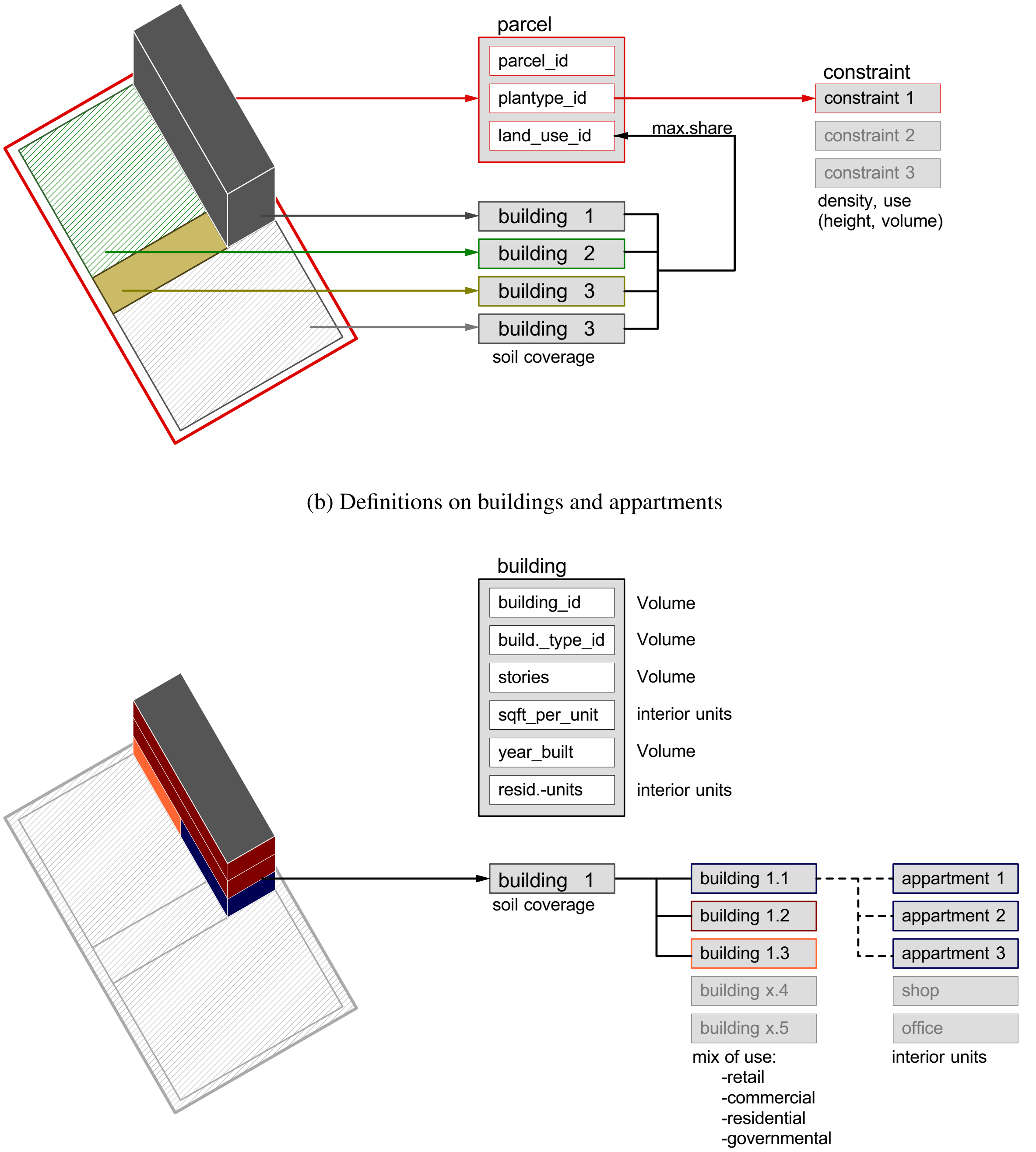
3.7 Population synthesis

The availability of the Swiss census permits us to use a full sample for the population of the study area. However, most other projects will not have this opportunity. In addition, given the confidentiality restrictions, using a population synthesised from freely available data would be

Table 2: Defintions on plan types and constraints

		SC		Developable for UrbanSim		Max Fractions (%)					Min Fractions (%)				
Plan type – main class	**Description/Comment**	**PLAN TYPE (SC)**	**Generic Use (SC)**			**Residential**	**Commercial**	**Retail**	**Industrial**	**Governmental**	**Residential**	**Commercial**	**Retail**	**Industrial**	**Governmental**
Residential	100% Residential	1000	1	1		100					90				
	Residential district (> 90%)	1100	1	1		100	10	10			90				
	Residential district (< 90%)	1200	1	1		90	30	30			70	10		10	
Residential & Commercial	Residential with (loud) commercial and industrial	2000	2	1		100	10		10		90				
	Residential (< 90%) with (loud) commercial and industrial	2100	2	1		90	30		30		70	10		10	
Center	Central area; Mix of residential, commercial, retail	3000	3	1		100	100	100		100					
	Urban center: Mix of residential, commercial, retail	3100	3	1		100	100	100		100					
Governmental	Public/Governmental	4000	4	0						100					
Industry & Commercial	Industry & Commercial/Industrial	5000	5	1			80		80			20		20	
Industry	Industry	6000	6	1			20		100					80	
Open Space	Recreation area	7000	7	0											
	Conservation area	7100	7	0											
	Reserve	7200	7	0											
	Agricultural	7300	7	0											
	Forest	7400	7	0											
	Water	7500	7	0											
Infrastructure	Infrastructure	8000	8	0											
	Airport	8100	8	0											
Undefined	Unzoned	9000	9	0											
	Undefined	9100	9	0											

Figure 3: Definitions for Zurich Case Study



(a) Definitions on parcels, constraints and land-use

(b) Definitions on buildings and appartments

preferable even for the SustainCity project. We are currently evaluating the generation of a synthetic population from the Swiss Public-Use Sample. Müller and Axhausen (2011) present an overview of the techniques available to date.

Unfortunately, no information on income and car ownership has been gathered for the Swiss census. We will impute the missing information from the mobility and transport microcensus, enhancing the approach presented by Bürgle (2007) by matching both data sets not only by household size, but also by the age of the household persons and by the availability of a second residence.

3.8 Load data into UrbanSim

UrbanSim requires tabular data without spatial attribute in a specific structure as input. This applies to both table names (*cf.* Figure 1(b)) and column names. Thus, after preparing the data, the final step is to fit it into the *UrbanSim* schema. An empty database schema for the parcel version of *UrbanSim* is created by the script create_parcel_database.py, which is then filled using *pg2urbansim*, a set of custom data transformation scripts.

The order in which the scripts must be executed is determined by data dependencies. For example, the *households* table cannot be filled before the *buildings* table. Currently, each of these data transformation scripts has a numerical prefix that explicitly defines the position within the execution sequence. We will replace this by a more flexible approach using GNU *make*: The dependencies between the scripts will be specified in a Makefile, allowing *make* to automatically determine a feasible order of execution.

There are three kinds of tables in the *UrbanSim* schema: Simulation, definition and estimation tables (*cf.* Section 2.3). Only the first two types of tables are usually imported; tables of the last type are the result of model estimation and specification which is done with UrbanSim (see Section C.2).

4 First UrbanSim run

This section informs about the approach taken and lessons learned when working with *UrbanSim*. A more detailed documentation of the working steps can be found in Appendix C.

There are two basic approaches distinguishable. The first is to take a sample project from the repository and to replaced the data and models with own specifications and data. This approach is recommended in (Patterson and Bierlaire, 2008). The second approach is to start from an *UrbanSim* model template. For our work we started with this second approach using the urbansim_parcel template.

Reasons to start from the urbansim_parcel template are the slim model specification, making it easier to be supported with data. Additionally a lot of specialities in the sample projects can be omitted. Through the divers datastructure across European countries in our understanding starting from a template will be the most common use case, that we want to test.

Following the recommendation from a previous project we tried to start working with `UrbanSim` as fast as possible. Therefore, we chose the strategy to set up a first dummy run which can be extended stepwise. Thus the first target is to have an oversimplified simulation running transition and locations choice models of households and employment only.

There is no default specification for the models. In this respect it might be helpful to have default model specifications that are applicable in most urban regions. To do so a literature review is currently been done, that will report on location choice models in European projects, trying to come up with a default specification with most frequently used significant variables.

Working with `UrbanSim` requires good programming skills preferable with Python programming language. The program cannot yet be mastered via the graphical user interface only.

The script `create_parcel_database.py` defines the input data as an empty template. This is a good approach for the data preparation, as the use of a database enforces the data to be consitstant regarding their linkage before the import to `UrbanSim`.

When getting error messages it might help to compare your project to one of the example projects. Examples give usable hints on how configurations or data structures should look like.

Debugging `UrbanSim` with Eclipse is not straight forward because `UrbanSim` runs multiple threads. However, `UrbanSim` can be configured to be run in a single thread to allow debugging.

The models in `UrbanSim` depend on variables, which in turn might depend on other variables. In the *Variable Library*, there are two tools that help satisfying these dependencies: (a) a dependency graph shown upon choosing *View dependencies* from a variable's context menu, and (b) a *Validate Selected* button that checks whether the currently selected variables can be computed and outputs the reason if they cannot. To check whether a model will run without error, one needs to manually validate all variables the model is using. A dependency scanner that is run automatically before the actual simulation would simplify this a lot.

5 Modelling

5.1 Overview

After conditioning data and implementing a first base model in `UrbanSim` including only dummy variables, the next step concerns the implementation of the basic model structure (see for more technical details also the Econometric Guidance of the SustainCity project by Picard and Antoniou (2011)). The focus lies on the estimation of the following models. Partially, parameters of these models have already been estimated, corresponding references are added in brackets:

- Household location, transition and relocation choice (Belart, 2011),
- Employment location, transition and relocation choice (Bodenmann and Axhausen, ????),
- Land development: Real estate price model (Löchl, 2010), development proposal choice, building construction, process pipeline events.
- Urban shape options, parcel transition model,

As a considerably large part of apartments in the agglomerations of Zurich and Winterthur are provided by dwelling cooperatives, the Zurich case study will additionally focus to model the according market and alternatives for residential (re-)location choice.

The model structure will be implemented in `UrbanSim` and is developed in two phases (see Figures 4 and 5). However, changes in the second phase concerns basically only the models regarding companies and their employees. There, the aim is to distinguish between decisions concerning companies and employees - analogous to the distinction household - resident.

5.2 Household location, transition and relocation choice

The survey of Löchl *et al.* (2005) provided the basis for the modelling of residential location choice as used in the ZUK project (Bürgle, 2007).

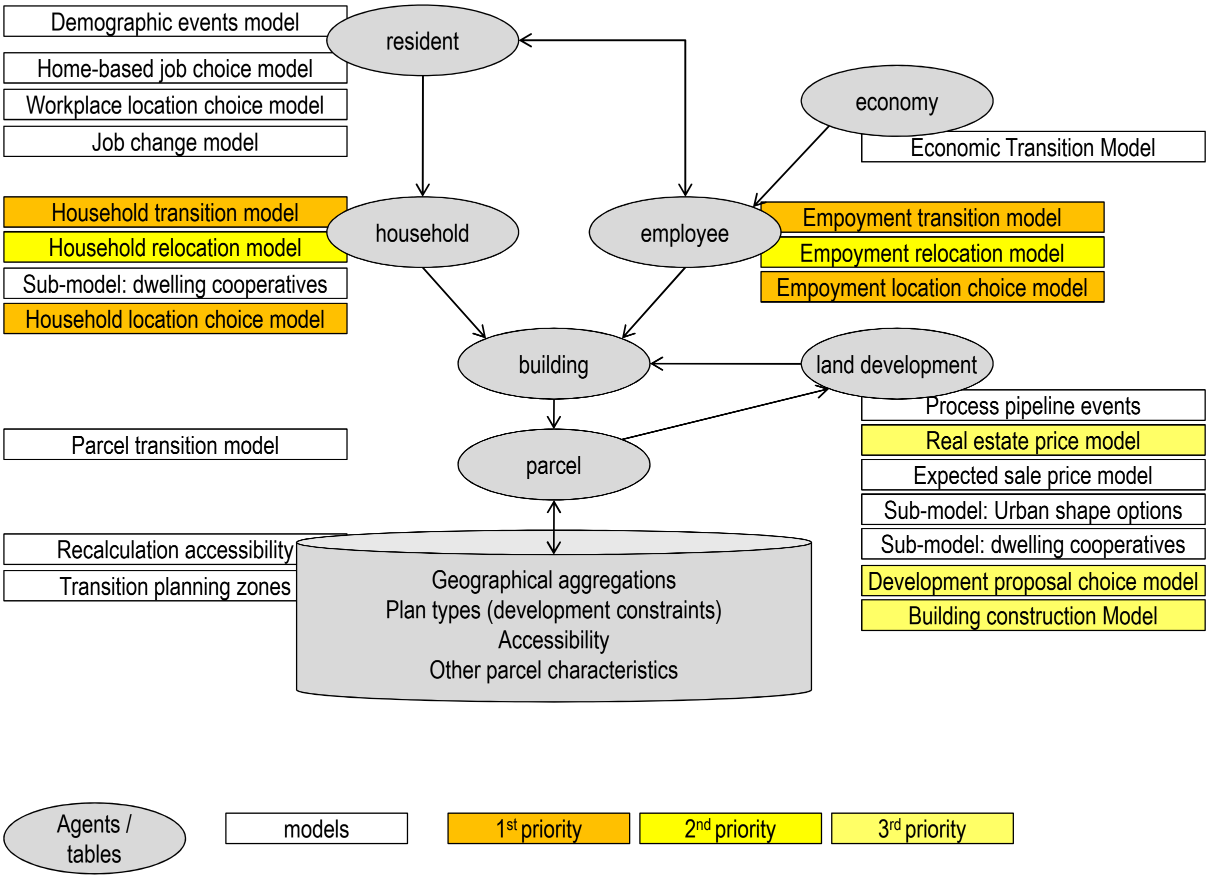
A new survey was undertaken end of 2010, that has been asking 5,000 households having recently moved within the Canton of Zurich. This survey asked for attributes of the location and the building of the current and the previous residential location, as well as for social networks and attitudes of lifestyle and has been examined within the masterthesis of Belart (2011).

The models of those survey can be used within the project of SustainCity and are currently being implemented into the generic work frame mentioned earlier.

Both modelling approaches select 49 non-chosen alternatives with the help of a choiceset-sampling out of a sample of available residences queried from a website in 2005 and 2010. The choice-set used in 2005 would select alternatives located around the chosen alternative, in 2011 a random sampling was used for this purpose. Future work will concentrate on other sampling strategies and evaluate their effects (Schirmer *et al.*, 2011).

All models created in 2005 and 2010 are MNL-models with explaining variables of domains: life style, dwelling type, location, but several models have been created on subsets of the

Figure 4: Model structure Zurich case Study: phase 1

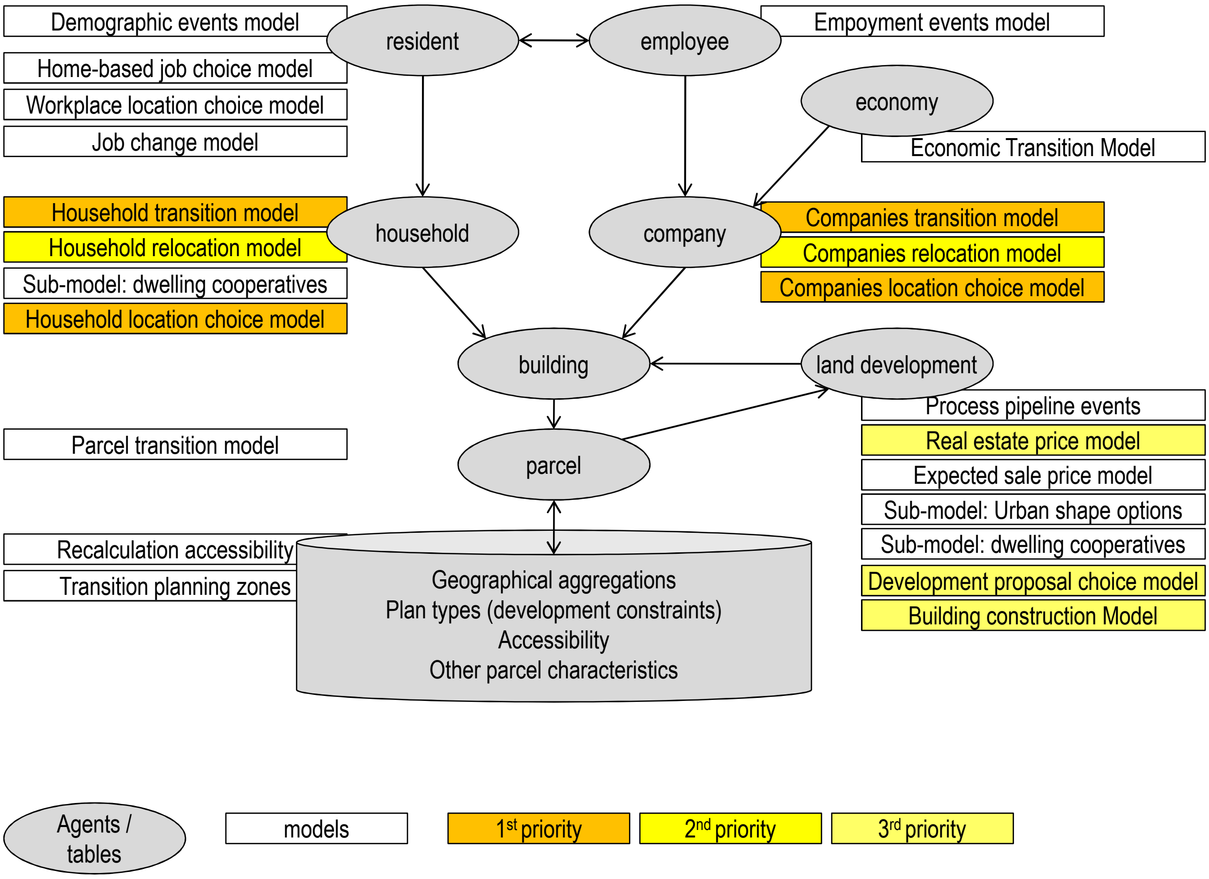

Demographic events model
Home-based job choice model
Workplace location choice model
Job change model
resident
economy
Economic Transition Model
Household transition model
Household relocation model
Sub-model: dwelling cooperatives
Household location choice model
household
employee
Empoyment transition model
Empoyment relocation model
Empoyment location choice model
building
land development
Process pipeline events
Real estate price model
Expected sale price model
Sub-model: Urban shape options
Sub-model: dwelling cooperatives
Development proposal choice model
Building construction Model
Parcel transition model
parcel
Recalculation accessibility
Transition planning zones
Geographical aggregations
Plan types (development constraints)
Accessibility
Other parcel characteristics
Agents / tables
models
1st priority
2nd priority
3rd priority

survey. Multinomial logit was deemed an appropriate approach to the estimation task at hand in accordance with McFadden (1978) who showed that unbiased parameters can be produced in the face of a large number of alternatives by using a random sample of the universe of the available choice set for alternatives.

In 2005 the behaviour of various socio-demographic groups were estimated separately, in 2010 the behaviour of different tenure types and lifestyles were estimated. The estimation of these models in form of NL-models has not been done yet, but will become content of future work. Both surveys show that there are various attributes that are significant for the residential location choice, but are not included in *UrbanSim* yet. A reduced model will therefore have to be included in the beginning, eventually *UrbanSim* can be extended later on. The selection of explanatory variables was geared to the following working hypothesis and to the data availability:

- Factors influencing residential location choice depend on the type of household making the location decision
- Households prefer to spend as little as possible of their income on housing
- Households with employed persons prefer housing locations close to their place of em-

Figure 5: Model structure Zurich case Study: phase 2


Demographic events model
Home-based job choice model
Workplace location choice model
Job change model
resident
employee
Empoyment events model
economy
Economic Transition Model
Household transition model
Household relocation model
Sub-model: dwelling cooperatives
Household location choice model
household
company
Companies transition model
Companies relocation model
Companies location choice model
building
land development
Process pipeline events
Real estate price model
Expected sale price model
Sub-model: Urban shape options
Sub-model: dwelling cooperatives
Development proposal choice model
Building construction Model
Parcel transition model
parcel
Recalculation accessibility
Transition planning zones
Geographical aggregations
Plan types (development constraints)
Accessibility
Other parcel characteristics
Agents / tables
models
1st priority
2nd priority
3rd priority

ployment

- Households with children prefer to live in areas with many children
- Young households without children prefer locations with high population density
- Older and retired households prefer locations with a high proportion of open spaces
- Municipality characteristics like the tax index or the rate of vacant housing units influence residential location choice
- Households tend to avoid locations with heavy noise emissions
- Environmental site characteristics like proximity to bodies of water or sunshine exposure may increase the utility of a residential location
- Households generally value a good local supply of retail trade
- The accessibility by individual or public transport in the Greater Zurich Area does not show differences big enough to significantly influence residential location choice but good accessibility by public transport is important for households without a car

5.3 Employment location, transition and relocation choice

Zurich Case Study bases employs firmography results from the adjacent St. Gallen region. These models address location behaviour of companies (on the level of plants). Therefore, the intended model structure will model transition, relocation and location choice of firms. The jobs provided are subsidiary modelled based on the behaviour of the companies (Figure 5).

It is proposed to model firmography in a three-step model which decomposes the plants and subsidiary jobs evolution:

1. Firmographics events like birth / closures / relocation / growth of plants
2. Location choice of new established plants
3. Location choice of relocating plants

Using data from the three cantons of St. Gallen and both Appenzell, most of these models have been estimated and calibrated. This data set provides information on more than 50,000 companies during a period from 1991 to 2006. The first model will consist of different sub-models and draw from a macro-econometric transition model. Due to data restrictions, firmographics in Switzerland generally distinguish between ten sectors (Bürgle, 2006; Bodenmann and Axhausen, ????). If possible, the sector of service and finance is additionally divided in smaller sections: i) finance, ii) business services, and iii) public and personal services (see Bodenmann and Axhausen, ????).

In general, firmographic events model is based on the results of Bodenmann and Axhausen (2008). Using business demographic data of St. Gallen region, four basic variables show up in the migration behaviour of companies: age, size, branch and location (community type) of the business. Using a logit-loglinear model, the relevant effects on the behaviour of the companies have been quantified. However, The present analysis shows that several more factors play a part in the decisions on choice of location: among others, the availability of building land and the price of space. This shows up particularly in the modelled effects between the branches. These themes as well as that of infrastructure (accessibility for customers and employees) and the behaviour of communities and cantons (*i.e.*, taxes) will be explored in further studies.

The location choice model for newly established plants has to be re-estimated with the data set of St. Gallen region. Based on the results of Bodenmann and Axhausen (2010, ????), Bürgle (2006) and Bodenmann (2007), a Nested Logit (NL) model will be estimated. Generally, the same variables will be used as for the location choice of relocating plants discussed below.

The model regarding location choice of relocating plants is generally based on the results of Bodenmann and Axhausen (????) for different business sectors. In general, the most important factors on location decisions of companies are cities, cantonal business development and tax burden. The fact that a potential site is in a city summarizes different advantages and disadvantages of these locations. This result corresponds to the innovative milieu approach and other agglomeration effects (Hutchinson, 2003; Florida, 2005; Bodenmann and Axhausen, 2010). The differences regarding the valuation of a site in a city suggest that companies in the sector of manufacturing tend to leave cities. In contrast, companies in the sectors of retail trade as well as public and personal services significantly tend to chose new locations in cities. The parameters regarding tax burden and cantonal business development indicate the very positive effect of governmental business friendliness. But also the various accessibility indicators play a strong role in most of the models. In general, railway stations have a larger impact than motorway connections. Interestingly, the various parameters for land prices have all only minor effects. The strongest effect has land price for residential use, this indicates a crowding-out effect between residential and business use.

5.4 Developers

The main underlying hypothesis is that the outcome of a development event depends not only on the site characteristics and demand but also on the characteristics of the developer. Aspects of the developer that may play an important role are purpose, strategy, portfolio structure, size (human and financial resources) and specialisation. Therefore the envisaged developer model tries to consider attributes of developers to explain development events.

In this framework the the development process is driven by developers modeled as autonomous entities. Consequently, we model the developers as agents. This allows us to consider resources of developers and property structures.

In respect of modelling techniques we will first estimate multinomial logit models for exogenously determined markets segments. In a first approach these segments remain constant in the sense that a developer will not change type. This assumption my be relaxed later on because in reality developers may also switch market segments or are present in various of them. In this case we would also have to estimate probabilities for developers to switch the market segment.

So far the segmentation of the developers will follow the following lines:

- purpose (profit oriented, non profit oriented)
- professionalism (professional, occasional) using size, organizational form or frequency as proxy
- Strategy type followed (portfolio, object-oriented)

We also intend to estimate other model types than multinomial logit. With latent class models it might be possible to determine the market segments endogenously. This approach has already been made by others.

When estimating these models on the basis of parcels as alternative we will also have to deal with the choice set definition problem (Lee and Waddell, 2010). The availability of parcels depends on whether the developer has the legal and economical right to develop the site. The choice set further depends on the are the developer is observing. An estimate on the market reach of developers will be necessary therefore.

5.5 Urban shape

We expect that the implementation of attributes on the shape of objects into the discrete choice models of `UrbanSim`, will significantly enhance the simulation results of `UrbanSim`.

As described in previous work (Schirmer, 2010), `UrbanSim` these kind of information, are relevant for the simulation process in two means: (a) it has an effect on the location choice of agents within a simulation, (b) it reduces the amount of development options for an objects within a simulation environment.

These information on the shape can be of multiple kinds. A first (non-terminatory) grouping of those could be: (a) the orientation and form of a building, (b) the relation of shapes to each other and (c) the definition of typologies based on certain combination of shape attributes and the neighboring shape attributes.

Gil *et al.* (2009) show that similarity of shapes allow to define and differ urban typologies through data mining techniques. Also, other works (Meinel *et al.*, 2009) use the shape of buildings and parcels to cluster certain attributes on shapes into typologies or to derive further values, *e.g.*, to define the use and height of a building based on its footprint and various attributes of the shapes (block, buildings) of its environment.

These works show that the shape of objects depend on each other within the build environment and that it is of relevance for the simulation to implement these kind of dependency.

To underline these hypothesis, two example shall be given: (a) a retail facility searches for a new location. For selling its goods, a visual attractive position is preferred, *i.e.*, with a front to the street. On the other side it needs direct access to the storage for the trucks arriving. A location at the corner of a building, *i.e.*, facing two streets, can therefore be expected to be favoured. (b) the definition of plan types for a parcel allow to add 2,000 sqm of floorspace to existing 15,000 sqm on site. The constraints do not allow to ad additional stories to the building and the position of the building on site does not allow for extension as well. The only chance of development will thus be the destruction of the current building to create a new one of 17,000 sqm, which might be to expensive to be of profit.

As the data structure of UrbanSim can not handle geometries, a first approach to implement geometric information into the simulation process, will be to define and categorise attributes of shapes similar to the constraints or plan types. These can then be coded and added as information to PostgreSQL, *e.g.*, through GIS applications in the generic work frame. These attributes can then be implemented into the models of UrbanSim to test there effect on developments.

The transformation of shapes, *i.e.* the implementation of developments as new shapes, can not be handled within the current version of UrbanSim. As described by Schirmer and Kawagishi (2011), defining attributes to code the developments on a parcel could allow to use procedural shape grammars to create these kinds of transformations outside of the current workspace of `UrbanSim`.

6 Conclusions and Outlook

In this paper we report on the work undertaken to set up a case study of `UrbanSim` on the canton of Zurich. We comment on the most important data that has been acquired and give an insight into the issues arising when linking different data due to inconsistency and missing links. As the data structure is already very diverse within this Case Study, it can be expected that the implementation of a European model will need some further observation of the related data used in the other case studies of UrbanSim.

We present the approach through a generic framework based on PostgreSQL. This scripted approach allows to repeat any transformation and processing of the original data and implements different tools of processing. It is thus an essential step to create a first and rough version of a base year in `UrbanSim` ignoring many data problems and still have the chance to correct and enhance the processings in a later step.

At present a first running environment that forms a very basic "super simplified" simulation environment has been realised. This includes very reduced household location choice and employment location choice models. The first phase of the Zurich Case Study within the SustainCity project including data conditioning and implementing the first model in `UrbanSim` is therefore mostly finished. From here the ongoing work will focus on `UrbanSimE` and envision to include more of the variables into the models, as well as build up own models that have previously been estimated outside `UrbanSimE`. Besides it will be essential to enhance the quality of the base year and include further variables in that one.

7 Acknowledgements

This research has been partly funded through the SustainCity project (FP7-244557), co-financed by the EU within the Seventh Framework Programme (FP7). The authors wish to acknowledge the Commission for their support of the project, the efforts of the partners and the contributions of all those involved in SustainCity.

8 References

Beige, S. (2008) Long-term and mid-term mobility decisions during the life course, Ph.D. Thesis, ETH Zurich, Zurich.

Beige, S. and K. W. Axhausen (2005) Feldbericht der Befragung zur langfristigen räumlichen Mobilität, *Technical Report*, **315**, , ETH Zurich, Zurich.

Belart, B. (2011) Wohnstandortwahl im Grossraum Zürich, Master Thesis, , ETH Zurich, Zurich.

Bodenmann, B. R. (2007) Modelle zur Standortwahl von Unternehmen, in T. Bieger, C. Lässer and R. Maggi (eds.) *Jahrbuch 2006/2007 Schweizerische Verkehrswirtschaft*, 5–37, St. Gallen.

Bodenmann, B. R. and K. W. Axhausen (????) Destination choice of relocating firms – a discrete choice model for the region of St. Gallen, Switzerland, *Papers in Regional Science*.

Bodenmann, B. R. and K. W. Axhausen (2008) Schweizer Unternehmen – quo vaditis? Firmendemographische Trends am Beispiel des Wirtschaftsraums St. Gallen, *Raumforschung und Raumordnung*, **66** (4) 318–332.

Bodenmann, B. R. and K. W. Axhausen (2010) Synthesis report on the state of the art on firmographics, *Technical Report*, **2.3**, SustainCity project, , ETH Zurich, Zurich.

Bürgle, M. (2006) Residential location choice model of the Greater Zurich area, paper presented at the *6th Swiss Transport Research Conference*, Ascona, March 2006.

Bürgle, M. (2007) Synthese von Haushaltsdaten für den Kanton Zürich, *Technical Report*, **9**, , , Polyprojekt Zukunft urbane Kulturlandschaften, Zurich.

Florida, R. (2005) *Cities and the creative class*, Routledge, New York.

Gil, J., N. Montenegro, J. N. Beirão and J. P. Duarte (2009) On the discovery of urban typologies, paper presented at the *27th eCAADe Conference*, 269–278, September 2009.

Hutchinson, T. P. (2003) *Produktionsfaktor Wissen: Untersuchung des Zusammenhangs zwischen Wissen und Standort von Unternehmen*, vol. 45 of *Aachener Reihe Mensch und Technik*, Wissenschaftsverlag Mainz, Aachen.

Lee, B. H. Y. and P. A. Waddell (2010) Residential mobility and location choice: A nested logit model with sampling of alternatives, *Transportation*, **37** (4) 587–601.

Löchl, M. (2010) Application of spatial analysis methods for understanding geographic variation of prices, Ph.D. Thesis, ETH Zurich.

Löchl, M., M. Bürgle and K. W. Axhausen (2007a) Implementierung des integrierten Flächennutzungsmodells UrbanSim für den Grossraum Zürich – ein Erfahrungsbericht, *disP – The Planning Review*, **168**, 13–25.

Löchl, M., M. Bürgle and U. Waldner (2007b) Handbuch Simulationsmodell Grossraum Zürich, *Technical Report*, **465**, , ETH Zurich, Zurich.

Löchl, M., U. Waldner and M. Bürgle (2005) Haushaltsbefragung zur Wohnsituation im Grossraum Zürich – Tabellenband, *Technical Report*, **2**, , , Polyprojekt Zukunft urbane Kulturlandschaften, Zurich.

McFadden, D. (1978) Modeling the choice of residential location, in A. Karlqvist (ed.) *Spatial Interaction Theory and Residential Location*, 75–96, North-Holland, Amsterdam.

Meinel, G., R. Hecht and H. Herold (2009) Analyzing building stock using topographic maps and GIS, *Building Research & Information*, **37** (5) 468–482.

Müller, K. and K. W. Axhausen (2011) Population synthesis for microsimulation: State of the art, paper presented at the *90th Annual Meeting of the Transportation Research Board*, Washington, D.C.

Patterson, Z. and M. Bierlaire (2008) Development of prototype UrbanSim models, *Report TRANSP-OR*, **080814**, Ecole Polytechnique Fédérale de Lausanne, Lausanne, August 2008.

Picard, N. and C. Antoniou (2011) Econometric guidance, *Technical Report*, **5.1**, SustainCity project, University of Cergy-Pontoise, Cergy-Pontoise.

Schirmer, P. (2010) Options and constraints of a parcel based approach in 'UrbanSimE', paper presented at the *10th Swiss Transport Research Conference*, Ascona, September 2010.

Schirmer, P., B. Belart and K. W. Axhausen (2011) Location choice in the greater Zurich area, paper presented at the *11th Swiss Transport Research Conference*, Ascona, May 2011.

Schirmer, P. and N. Kawagishi (2011) Using shape grammars as a rule based approach in urban planning – a report on practice, paper presented at the *29th eCAADe Conference*, September 2011.

SustainCity (2011) The SustainCity project, `http://www.sustaincity.org`.

UrbanSim (2011) Open Platform for Urban Simulation, `http://www.urbansim.org`.

Waddell, P. A. (2002) UrbanSim: Modeling urban development for land use, transportation, and environmental planning, *Journal of the American Planning Association*, **68** (3) 297–314.

Waddell, P. A. (2010) Overview of UrbanSim and the Open Platform for Urban Simulation, presentation, UrbanSim Tutorial, Zurich.

A Data transformation

Table 3: Data transformation for the parcels table

Column name	Type	Required	Status	Final
parcel_id	PrimaryKey	✓	av_liegensch.gid	✓
parcel_id_local	String(20)		Dummy 0	✓
land_use_type_id	Integer	✓	N	
land_value	Integer	✓	Dummy 1	
parcel_sqft	Integer	✓	av_liegensch.flm2	✓
plan_type_id	ForeignKey	✓	Dummy 1001	
centroid_x	Integer	✓	av_liegensch.the_geom	✓
centroid_y	Integer	✓	av_liegensch.the_geom	✓
tax_exempt_flag	Integer		Dummy 0	✓
city_id	ForeignKey		av_liegensch.zgde	✓
county_id	ForeignKey		1	✓
zone_id	ForeignKey	✓	av_liegensch.zonennr1	✓
census_block_id	string		Dummy 0	✓

Table 4: Data transformation for the buildings table

Column name	Type	Required	Status	Final
building_id	PrimaryKey	✓	gwr_gebaude10.egid	✓
building_quality_id	Integer	✓	gwr_gebaeude10.gklas	
building_type_id	ForeignKey	✓	av_bodenbed_f.art_code	
improvement_value	Integer	✓	gvz_geb2000.wert	✓
land_area	Integer	✓	av_bodenbed_f.flaeche_m2	✓
non_residential_sqft	Integer	✓	land_area * stories	
parcel_id	ForeignKey	✓	av_liegensch.gid	✓
residential_units	Integer	✓	gwr_gebaeude10.ganzwhg	✓
sqft_per_unit	Integer	✓	**avg**(gwr_wohnungen10.warea)	✓
stories	Integer		gwr_gebaeude10.gastw	✓
tax_exempt_flag	Integer		Dummy 0	✓
year_built	Integer	✓	gwr_gebaeude10.gbauj	✓

Table 5: Data transformation for the `jobs` table

Column name	Type	Required	Status	Final
job_id	PrimaryKey	✓	`autogenerated`	✓
building_id	ForeignKey	✓	`buildings.building_id`	✓
home_based_status	Integer	✓	**Default** 0	
sector_id	Integer	✓	`sectors ZUK`	✓

Table 6: Data transformation for the `households` table

Column name	Type	Required	Status	Final
household_id	PrimaryKey	✓	`vz.hhnr`	✓
building_id	ForeignKey	✓	`buildings.building_id`	✓
persons	Integer	✓	**COUNT**(vz.person_id)	✓
income	Integer	✓	50000 * **COUNT**(person_id)	
age_of_head	Integer	✓	`vz.valtj`	✓
race_id	Integer	✓	`vz.hmat`	✓
workers	Integer	✓	**COUNT**(person_id)**WHERE** vz.ams **BETWEEN** 11 **AND** 14	✓
children	Integer	✓	zkind > 0	✓
cars	Integer	✓	**COUNT**(person_id)**WHERE** vz.apkwl = 1	

Table 7: Data transformation for the `persons` table

Column name	Type	Required	Status	Final
person_id	PrimaryKey	✓	`vz.person_od`	✓
household_id	ForeignKey	✓	`vz.hhnr`	✓
member_id	Integer			
relate	Integer			
age	Integer			
sex	Integer			
race_id	Integer			
student	Integer			
worker	Integer			
hours	Integer			
work_at_home	Integer			
edu	Integer			
earning	Integer			
job_id	ForeignKey			

B Model definition

Table 8: Tables for model definition

Table name	**Status**
annual_employment_control_totals	linear extrapolation from enterprise census
annual_household_control_totals	Prognosis statistical office canton Zurich
annual_relocation_rates_for_households	Survey data (Beige, 2008)
annual_job_relocation_rates	ZUK definitions (Löchl *et al.*, 2007b, p. 32)
building_types	7, according to soil coverage information (bodenbed_f.art_code)
employment_sectors	ZUK categories (Löchl *et al.*, 2007b, p. 32)
target_vacancies	0.66% (housing), 4.0% (non-housing)
development_constraints	derived from zoning plans
land_use_types	**9 categories**
plan_types	reduced to 146 categories

C Set up UrbanSim

In the following two subsections we describe briefly the installation and the main work steps in *UrbanSim*.

C.1 Install UrbanSim

The installation on Linux is described in detail on the *UrbanSim* wiki.[1] Here we give some general remarks. We use Eclipse as integrated developer environment with the Pydev extension. Python is installed in a virtual environment to be able to have multiple Python versions on the same computer. Some Python packages have to be installed in addition to the default Python 2.6 using the package management system of Linux. For packages not available in the repository we use *easy_install* for installation from a download site. Subversive is used to connect to the *UrbanSim* SVN repository and download the data. Setting a run configuration helps to start the graphical user interface easily within eclipse.

[1] http://www.urbansim.org/Research/SustainCity/UrbanSimInstallationInstructionsForUbuntu

C.2 Work with UrbanSim

This is a very general description of how tho use `UrbanSim`. It follows the main work steps that are also visualised in the graphical user interface with the tabs *General*, *Data*, *Models*, *Scenarios* and *Results* below the main menu.

First you have to create your own project. Note that you create a child project of one of `UrbanSim`'s defaults. Your configurations are stored in an xml file with the project's name.

Importing the data with the `UrbanSim` tool `sql_data_to_opus` is the next step. This will convert the data into the `UrbanSim` format and store it to the `UrbanSim` cache. If you want to import from a database you first have to define your database connections. Further it my be helpful to make a local node and to specify the parameters in the tree of the the *Tools* tab. The parameters will be stored which is convenient if you have to reuse the tool.

There are some other points to note when working with the tool `sql_data_to_opus`.

1. It seems that the tool is not able to connect to a database named differently than the project itself. Instead it will create a new schema in the database with the projects' name. However, it will find tables in the public schema of that database.
2. You cannot enter schema qualified names (like postgres.zurisim).
3. You are not notified if an imported table is empty.
4. If a cell of a table is empty you get the error *Setting an array element with a sequence*.

The default model `urbansim_parcel` does not contain all model specifications. This makes sense because you want to estimate your models on your own data. The basic use case is to use `UrbanSim`'s estimation functionality which creates the needed tables of model specifications and coefficient in the `base_cache_storage` directory (directory in which the base year is stored).

Another use case is to estimate the model with an external software like `biogeme`. Then we would have to provide the specification and coefficients table by importing them like other tables of the base year. In case of taking over existing models the specification and coefficients tables can just be copied. Of course we still have to specify the model in `UrbanSim` for simulation.

`UrbanSim` provides three possibilities for creating a new model:

1. Make a local node (child of parent model)
2. Duplicate a parent model
3. Creation from template

When using the first or second possibilities it is necessary to add manualy a specification tag in the project's xml file. You will find different presettings depending on which possibility you choose. The presentings give hints on how model parameters should be set. The presettings can be changed.

Once a model is specified we can run an estimation. Rightclick the model and choose *Run Estimation* from the context menu. If you want your estimations to be stored, check whether *Estimation Configuration* > *save_estimation_results* is marked in the *Models* tab.

The default scenario configuration is duplicated which inserts an xml tag `Copy of urbansim_scenario_baseline` in your project xml file defining the local node. Within this tag we have to add a tag `datasets_to_preload` manually within which the selectable data sets have to be specified. Before a simulation can run the executable bit has to be checked in the context menu of the run configuration.