Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

TOWARDS AN AUTOMATED FRAMEWORK FOR AGENT-BASED SIMULATION OF REFUGEE MOVEMENTS

Diana Suleimenova
David Bell
Derek Groen

Brunel University London
Department of Computer Science
London, UB8 3PH, UNITED KINGDOM

ABSTRACT

Forced migration is a growing global problem, and the world now has a record amount of 22.5 million refugees. Models that predict refugee movements are few and far between, and constructing these models requires a substantial amount of manual effort while erupting refugee crises require a very rapid response. Here we present a vision towards establishing an automated framework, aimed to enable researchers to construct simulations of refugee movements more quickly and systematically. Our approach incorporates a diverse range of data sources, and uses the FabSim toolkit in conjunction with the Flee simulation code to quickly generate simulation workflows. In addition, we highlight a few key steps that we have already taken towards realizing this vision and discuss opportunities for wider applicability.

1 INTRODUCTION

Human migration is a global phenomenon with a long history. To understand human migration today, several categorization forms, such as a size of the movement, time, lawful status, cause and territory, are required to explain the nature of migration. The most examined category in literature is migration cause that drives people to migrate voluntarily and involuntarily (or forcibly) from their origin country (Faist 2000). Moreover, a combination of these categories brings defining labels of migrant types. Specifically, forced internal and international migration have several types, namely oustees, internally displaced persons and refugees.

The migration type of oustees refers to people who are forcibly internally displaced due to natural preservations, actions undertaken by governments and organizations. The internally displaced persons (hereafter IDPs) migrate internally within a country of residence caused by scenarios of armed conflict and violence. Moreover, the same causes also force refugees to migrate but internationally meaning that people displaced across borders of their home country. According to the 1951 Refugee Convention, a refugee is ”someone who is unable or unwilling to return to their country of origin owing to a well-founded fear of being persecuted for reasons of race, religion, nationality, membership of a particular social group, or political opinion” (UNHCR 2010, p. 3).

UNHCR (2017) highlights that there are 65.6 million people who are forcibly displaced worldwide and 22.5 million of which are refugees. These displacements occur due to internal or external conditions that trigger movement of people (Davenport, Moore, and Poe 2003) listed in Table 1. As a result, countries experience disaster, war or development (Shami 1996). There are various and often complex reasons behind the decision of people to move, but motivation, desire and pressure can derive most of the cases (Boano et al. 2003).

978-1-5386-3428-8/17/$31.00 ©2017 IEEE

Table 1: Internal and external conditions explaining forced migration (Davenport, Moore, and Poe 2003).

Internal conditions	**External conditions**
Violence	Colonialism
Worsening of socio-economic conditions	Unfair trade regulations
Famine	Global Inequality
Natural disasters	Impact of corporations on local economies
	Deterioration of the environment

There are various methods and models forecasting migration but very few appropriate for involuntary movements (Disney et al. 2015). For instance, early warning models can predict forced migration, but ignore refugee forecasts as they can be inaccurate and not flexible to represent complex changes (Lopez-Lucia 2015). Hence, there is a decisive gap in the research area of forced migration. The use of computational approach and improvements in data sources can be possible solutions in forecasting forced migration. Groen (2016) developed an agent-based simulation model and the Flee code, which predicts the distribution of fleeing refugees across target camps.

This paper concentrates on the approach of automating the Flee simulation construction from data gathering to construction of network maps, as well as visualization of refugee movements. We aim to generate an automated agent-based simulation tool from start to end that will be rapid, efficient and consistent, as well as with reduced efforts in developing refugee models for conflict situations. As conflicts forcing people to migrate can erupt at any time and create urgent situations for which organizations or policy-makers require solutions or tools to predict and assist refugees in need. The next sections will discuss an agent-based modeling paradigm and its application to human systems, as well as a progress in developing an automated approach to ABM of refugee movements.

2 AGENT-BASED SIMULATION

Computational science is a promptly growing multidisciplinary field in which researchers build and modify models until they start to simulate the reality (Alder 2001). One of the classical simulation paradigms is an agent-based modeling (ABM) that consider complex systems from minimalistic to large scale, and captures emergent phenomena. ABM comprises of agents, agent relationships and their environments (Allan 2010, Macal and North 2010). The agents or a group of agents have characteristics, such as interacting, autonomous, self-directed and self-contained, and attributes (e.g. resources and memory). Moreover, the behavior of agents allows to make decisions, interact with agents and function in their environment (Castle and Crooks 2006, Macal and North 2009). The application of ABM present in various practical areas of organization, market, diffusion and flow simulations (Bonabeau 2002).

To understand the use of ABM in the context of forced migration, we reviewed literature focusing on simulation modeling of social conflicts, civil violence, military and armed conflicts. There is a clear trend on the use of ABM and simulation within conflict and peace studies (Yilmaz, Oren, and Ghasem-Aghaee 2006, O'Brien 2010). Moreover, the study conducted by Amblard et al. (2010) provide an overview of theoretical approaches to social conflict situations, such as structural, interactional and Schellenberg's classification. Yet they stress the importance of ABM dealing with the complexities and uncertainties of social conflicts or changes. Particularly, ABM of conflicts is able to store, access and interpret extracted event datasets from varied platforms, as well as add value by explaining directions of conflict and peace (Yilmaz 2007).

Importantly, we examined the application of ABM on human systems, particularly migration and/or refugee movement scenarios. Schmeidl (1997) suggested a model outlining root and proximate causes, and intervening factors of forced migration that can be used as variables in running simulation. However, a limitation of this model is its inability to determine the importance or impact of a specific parameter. To achieve more effective predictions, Sokolowski and Banks (2014) developed a method for forced migration using ABM. Precisely, they establish their simulation using Schmeidls early warning model of forced migration, match the factors with the United Nations Human Rights Council (UNHRC), develop an ABM

Environment Matrix and applied to Syrian refugee crisis. Their simulation can be adopted for a specific situation or study accurately represent forced migration movements using simulation. However, further investigation and analysis are required to determine the key causes and intervening factors in these ABM simulations (Sokolowski, Banks, and Hayes 2014).

In addition, Junjie, Carley, and Cheng (2016) explore the link between population shifts and networks of alliances, linguistic similarities, economic influence and age distribution that might influence agents' decisions to migrate. However, their model does not take into consideration war or conflicts that can be a major reason to force people to flee. In contrast, Collins and Frydenlund (2016) propose ABM approach to investigate the interaction of refugees with groups over long distances and their strategic decisions on speed towards safer locations. They found that when refugees travel with lower speeds then they are more likely to form groups, while refugees with higher speeds result on a pair or single traveling. The main limitation of this study is that authors have not conducted empirical validations on their proposed model, mainly due to data unavailability.

The simulation model by Groen (2016) predicts the distribution of refugees across camps and uses several datasets from publicly available sources. First, the United Nations High Commission for Refugees (UNHCR) provides the total number of refugees in the conflict area and identifies potential destination camps in neighboring countries. Second, the Armed Conflict Location and Event Data Project (ACLED) (Raleigh et al. 2010) allows to obtain data on battle occurrences and conflict locations from where refugees flee. Third, the Bing Maps and Google Maps assist simulation construction by estimating distances and interconnecting links between conflict zones and camps. Forth, population counts for identified location points are manually extracted from population databases to distribute refugees from conflict locations to camp destinations. Moreover, the refugees' choice of destination determined by a weighted probability function, meaning that weights (equivalent to 1) divided to the distance traveled (Groen 2016). Last but not least, the simulation code Flee is written in Python language with additional libraries for analysis such as pandas and matplotlib. To validate, the Flee compares obtained results from constructed simulation with UNHCR dataset for each camp considered within the model.

To sum, refugee flow prediction using the Flee, which is a simulation code and one step in the chain, requires extracted data from publicly available databases, construction of network maps and graphical visualization of refugee movements from conflict zones to camps. Currently, these processes are proven to be time-consuming and inefficient. It is because conflict eruptions and refugee flow creating urgent situations require simulation construction and prediction to be as quick as possible. When there is a prediction urgency of refugee crises or when simulations are adopted for multiple conflict scenarios, we do not want to spend months or years constructing these simulation models. Conflicts are not always anticipated and when they occur, they can occur on short time scales. Hence, an automated approach can increase the chance of reducing completion time of simulation.

3 AUTOMATION TOOLS

Constructing a valid automated simulation model is time-consuming and challenging. Despite this, simulation modeling requires a generation of automation as it improves simulation process and provides better integration of applications (Huang, Seck, and Verbraeck 2011). Therefore, there is a requirement for tools which can assist the process of developing simulation models (Swaminathan, Smith, and Sadeh 1998). In other words, automation of simulation with available computational advancements can solve complex problems.

There are a wide range of languages, open-source software and automation tools which assist scientists and programmers in developing computational research. To demonstrate, domain-specific languages (DSLs) are small and specialized for a specific aspect of a software application. Hence, the use of DSLs, such as CSS or APIs, helps developers to build programs written in general-purpose languages. Mernik, Heering, and Sloane (2005) provide a detailed guidance for design, analysis, and implementation processes of DSL development for developers. Alternatively, there is a package management software named Homebrew, developed for Macintosh operating systems, integrated with the command-line and favored for its ease of use. Homebrew helps to instantly install required software.

Another example of existing computational advancement is the swift parallel scripting engine reducing complexities of structuring file systems (Wilde et al. 2011). It provides a basis for writing and execution of codes or programs across dispersed computing resources from computer clusters to supercomputers. There are also automation tools such as Longbow to run on HPC resources, which automatically stages input datasets and executes large simulation results. According to Gebbie-Rayet et al. (2016), Longbow provides an opportunity for users to simply and instantly run tasks on HPC using their own desktop environment.

In terms of workflow-specific automation engines, Kepler provides a platform to operate with various data formats, merge software components and execute locally or online to generate results. In addition, Kepler and other workflow engines, such as Taverna, simplify data access, flow of control and ensemble distribution of databases with remote machines. Curcin and Ghanem (2008) discuss several workflow systems and distinguish them in terms of their flow of control and data flow characteristics.

Importantly, Groen et al. (2016) released a highly modifiable automation toolkit named FabSim under a BSD 3-clause license, which has a purpose of time management by automating and simplifying a range of computational activities. For example, researchers save their time as FabSim helps them construct, manage and organize input and output files, user and machine configurations, as well as application executions using remote resources, all using one-line commands.

FabSim has already been applied successfully across three different disciplines, including computational fluid dynamics, materials science, and biomolecular simulations. It is able to automate a range of complex computational tasks, including iterative optimization procedures that involve simulations on remote supercomputers and local data processing (Suter, Groen, and Coveney 2015), remote code compilation and deployment across heterogeneous platforms (Groen et al. 2016), and the construction and execution of simulations involving multiple models run on different machines (Itani et al. 2015) (so-called multiscale or hybrid simulations).

FabSim consists of a software toolkit that is easy to use, navigate, explain and adjust. Hence, the main strength of this automation toolkit is its ease of customization in structuring models helping to maintain computational tasks. To provide easy remote access, FabSim relies on low-level secure shell (SSH) software that is used on most Unix-based machines and supercomputer, and is written in Python 2. It does not require any administrative or additional heavyweight installations, and therefore can be easily applied to perform activities on remote resources. FabSim relies on the Fabric library to access and manage these remote machines conveniently while using the YaML library as a platform to provide a dense, instinctive and human-readable data structure. Figure 1 demonstrates the overall construction of FabSim automation toolkit.

4 AUTOMATED SIMULATION CONSTRUCTION

Computational researchers and organizations can predict refugee movements using a wide range of information sources, codes and tools. Specifically, we refer to the construction of agent-based simulation model demonstrated in Figure 2 that involves five main processes of extraction, construction, instance definition, execution and visualization. Manual extraction of data sources, construction of network maps, exploration of different parameters, and which aspects to include in a simplified model are time-consuming and error-prone procedures that require careful scrutiny. Simulation models are also limited in reusability and thus, it is essential to construct simulation with reusable components (Swaminathan, Smith, and Sadeh 1998).

Therefore, we propose an automated simulation construction toolkit combined with simulation model - Flee - for predicting the distribution of incoming refugees across destination camps. It is a unique and useful simulation addition since automation toolkit can create the whole environment for researchers and organizations to curate data, add data processing components, construct models and modify simulations,

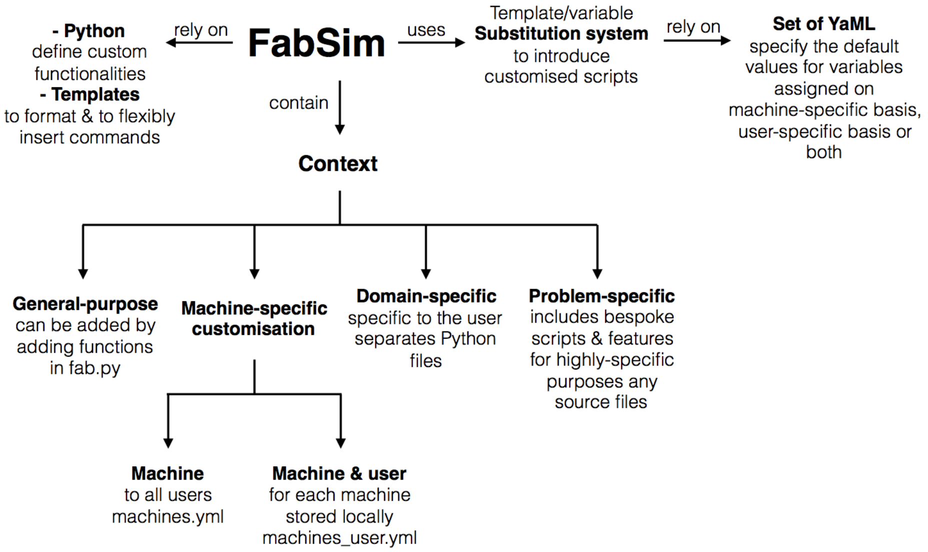


Figure 1: FabSim structure.

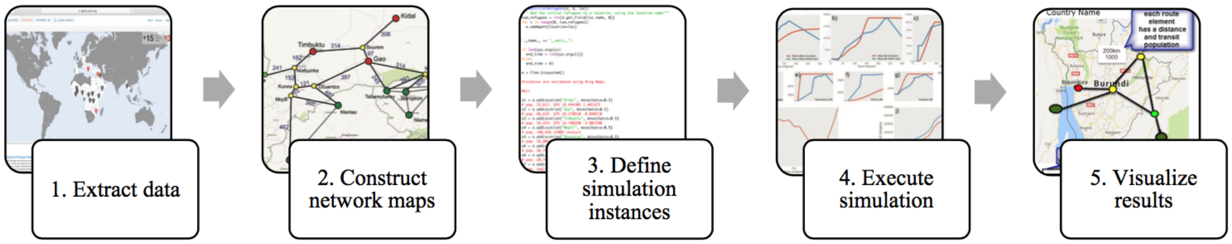


Figure 2: Overview of the steps required to construct, run and visualize a basic refugee simulation.

but mainly to predict refugee movements. We discuss and focus on the current and future thought-outs or intentions of each stage undertaken for full automation of simulation modeling.

4.1 Data Extraction

The first stage to consider is the extraction of data from multiple sources. To understand, we adopt the model building approach suggested by Chandra and Grabis (2003), which demonstrates several raw data sources creating unified input data and dividing databases from the model. Within our data extraction, the model building approach, which is a preliminary version, follows from raw data sources to a model generator as illustrated in Figure 3.

We extract data from three main sources, namely United Nations High Commission for Refugees (UNHCR), Armed Conflict Location and Event Data Project (ACLED) and population databases. We model the migration of refugees in a conflict crises and each of these sources provide essential information for simulation construction. In particular, UNHCR database identifies countries with the former or current refugee crisis, lists camps located in neighboring countries of a refugee situation and keeps a record of refugee numbers for each camp in JavaScript Object Notation (JSON) formats. In the case of armed conflict data, there is the World Bank providing data on fragile and conflict affected areas, but lacking information on armed conflict event dates, their initiation and targets. Another dataset supported by the World Bank is Uppsala Conflict Data Program (UCDP) with Peace Research Institute Oslo (PRIO) that have developed armed conflict dataset. It contains the information required for developing an agent-based

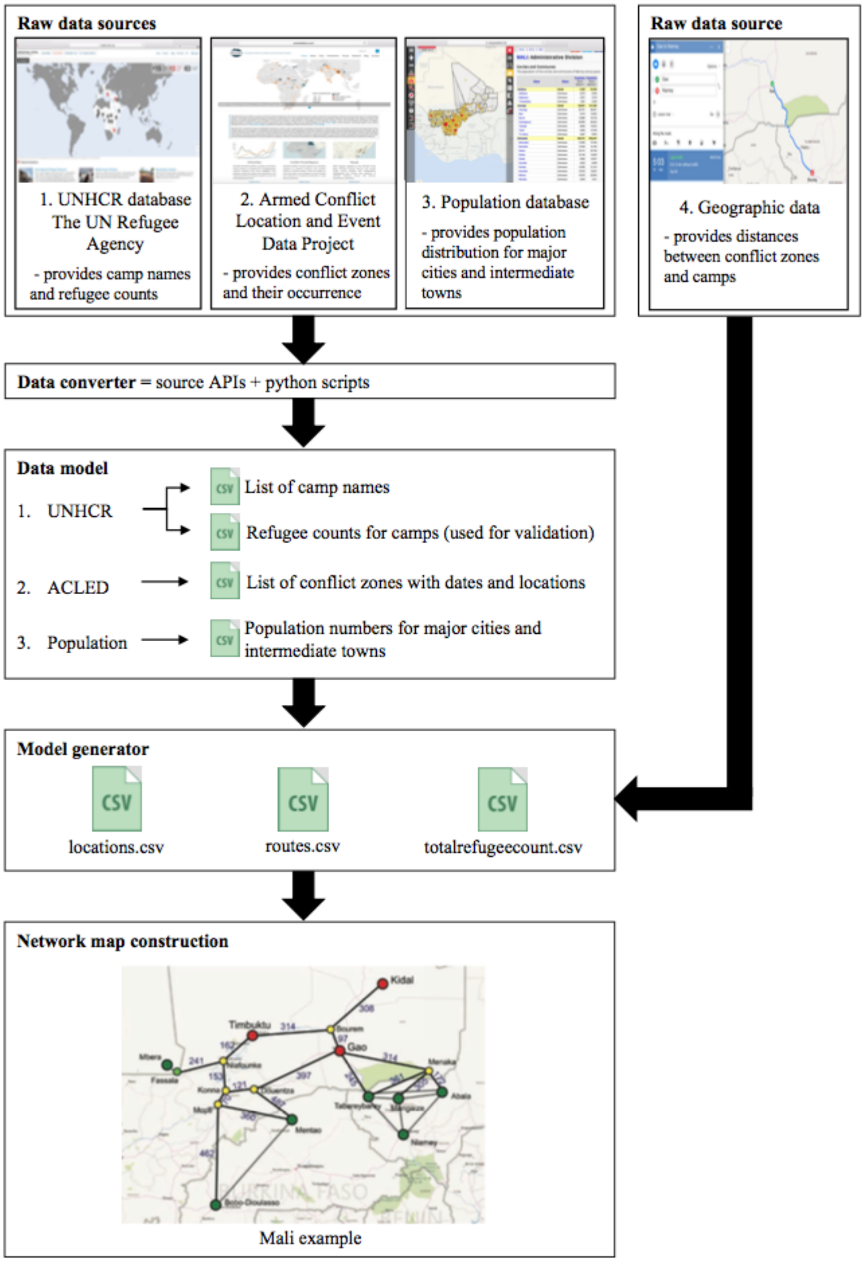

Raw data sources
1. UNHCR database The UN Refugee Agency
- provides camp names and refugee counts
2. Armed Conflict Location and Event Data Project
- provides conflict zones and their occurrence
3. Population database
- provides population distribution for major cities and intermediate towns
Raw data source
4. Geographic data
- provides distances between conflict zones and camps
Data converter = source APIs + python scripts
Data model
1. UNHCR
List of camp names
Refugee counts for camps (used for validation)
2. ACLED
List of conflict zones with dates and locations
3. Population
Population numbers for major cities and intermediate towns
CSV
Model generator
locations.csv
routes.csv
totalrefugeecount.csv
Network map construction
Kidal
Timbuktu
Bourem
Gao
Mbera
Fassala
Niafunke
Konna
Mopti
Menaka
Abala
Tabareybarey
Mangaize
Niamey
Mentao
Bobo-Dioulasso
Mali example

Figure 3: The model building approach for refugee, conflict and population data (Chandra and Grabis 2003).

simulation of refugees fleeing due to armed conflicts from 1989-2015 (Gleditsch et al. 2002). However, we use ACLED database as it is more detailed and up to date, and has all required information on conflicts and protests for African and Asian countries that can be downloaded in eXceL Spreadsheet (XLS) formats. In addition, City Population database establishes population distributions of major cities and intermediate towns within conflict areas. WorldPop is also a population distribution database that gives an opportunity to map population distribution with our simulation but is still at the exploration stage.

As raw data sources provide an explanation for data diversity, a data converter examines efficient and effective ways of gathering input files from all databases. To achieve this, firstly, we investigated UNHCR and ACLED application program interfaces (APIs) with some complications. Specifically, we found that the current UNHCR operational portal for refugee situations has API documentations but they appear to refer to an older version of UNHCR platform. Similarly, some of the conflict situations still under old UNHCR API, whereas other situations use new API codes. Henceforth, we present in Tables 2 and 3 the sequence to follow for both versions of UNHCR APIs.

Table 2: An old UNHCR APIs for camp names and refugee counts.

data.unhcr.org/api/	
Variables	Description
1. instances/list.json	provides *id* - string of country code
2. regions/show.json?id=(countryname)	provides list of camps for each country
e.g. regions/show.json?id=mali	*mali* is a country name

Table 3: A new UNHCR APIs for camp names and refugee counts.

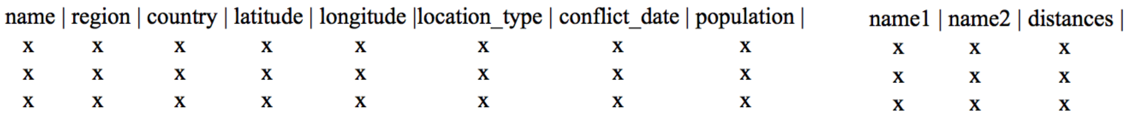
data2.unhcr.org/api/	
Variables	Description
1. en/search	view source page for *geo_id* showing number codes for all cities
2. population/get/timeseries?geo_id=(number)	change *geo_id* to get camps counts
e.g. population/get/timeseries?geo_id=933	*geo_id=933* is Lusenda camp

As for the ACLED data, there are XLS files to download for each conflict situation. However, there is information within these files that do not have value for our simulation construction. Hence, ACLED API documentation provides descriptive guidelines that return the data used for refugee simulation. For instance, an API code for Mali crisis follows with, *gwno=432* is the country code corresponding Mali, *event_type=battle* filters the data and determines the occurrence of conflicts with event type being battle and *fatalities_where=%3E&fatalities=0* modifies the nature of the query providing fatality rate greater than 0. There is still some work required with a little bit more filtering for aimed dataset outcome.

The extraction of population data is currently manual, we are still at the development stage as we are trying to detect the most appropriate population distribution source. Subsequently, we aim to develop Python scripts for all three raw data sources that will download datasets automatically and instantly, as well as generate CSV files described in data model step of Figure 3. As a result, we will produce combined CSV files, namely locations.csv and routes.csv. These input files will contribute towards the construction of network maps discussed in the next section.

4.2 Automated Network Map Construction

We construct network maps with extracted data from sources that provide conflict, camp and intermediate locations. At the second stage of our simulation construction, we use two CSV files with specific information (see Figure 4a and 4b) employing the CARTO map construction platform. At the present, we are able to upload CSV files with location names and geographic coordinates to CARTO, which instantly identifies and marks required locations. Although it reduces the search time for conflicts, camps and intermediate town allocations, linking and determining distances between these locations has yet to be automated.



name	region	country	latitude	longitude	location_type	conflict_date	population
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x

(a)

name1	name2	distances
x	x	x
x	x	x
x	x	x

(b)

Figure 4: The CSV templates for network map construction. a) locations.csv file provides location names, which are determined as conflict, camp or intermediate town, with geographical and population data. It also includes information on the occurrence of conflicts that we identify from ACLED database. b) routes.csv represents distances from conflict zones to camps connected with intermediate towns. If *name1* is location A, then *name2* represents location B and hence, we calculate distances between those to distribute population fleeing conflict areas.

4.3 Define Simulation Instances and Execute Simulations

The Flee simulation uses input files, such as locations.csv and routes.csv, to predict the distribution of incoming refugees across destination camps. In other words, we use conflict locations, camps and intermediate towns, as well as their population and distances between locations obtained from previous steps to construct an agent-based simulation model. Along with the collected data, there is also the choice of agent-specific parameters, such as move speed and awareness level, as well as the simulation duration.

The simulation model involves each of these parameters after a lengthy testing process. Groen (2016) runs multiple simulation models with various travel speeds, refugees' spectrum of locations awareness and simulation periods of conflict crisis. These testing processes are manual and time-consuming. To save time by a simpler, quicker and more systematic performance of the computational activities, we will use FabSim automation toolkit (Groen et al. 2016). Constructing a highly transparent and customized program will allow us to achieve automation of key tasks by simplifying and accelerating activities, including creation, execution and analysis of models. Moreover, FabSim will also provide a platform to ensemble simulation for parameter explorations, predicting alternative conflict scenarios and predict the effects of different refugee camp allocations.

Therefore, FabSim is favorable for our automation construction. Importantly, we renamed FabSim for automated simulation construction as FabFlee. It will have a potential for integrating various modules of data exploration, model construction, execution and analysis of results. In relation to results analysis, Python script named plot-flee-output.py constructs and generates graphical visualization comparing obtained simulation output against UNHCR data. This comparison provides a visual representation for camps within the model for further validation analysis.

Finally, FabSim is helpful for creating multiscale refugee modeling from small to large scales (see Figure 5). To begin with, there is an evacuation modeling, where we focus on refugees within a designated area who are fleeing conflicted town. Specifically, we try to detect where refugee movements happen. Then, there is a refugee modeling constructed using Flee code that predicts refugee movements from a home country towards camps in neighboring countries. The last but not the least, migration modeling focuses on movements of people across countries. Although these three model types are quite different fields of study, there is the discrepancy in time scales between them. However, FabSim is still at exploration process step as simulation construction is not automated yet and neither is framework analysis.

4.4 Visualization of Results

Visualizing the network map assists researchers in constructing simulations for refugee movements. To reveal potential mistakes, bugs in the network graph and to picture the distribution of refugee population for simulation periods, we established a visualization project - Visualflee. Its ultimate aim is to produce a tightly coupled application of the type that would allow the migration of refugees to be modeled in real time.

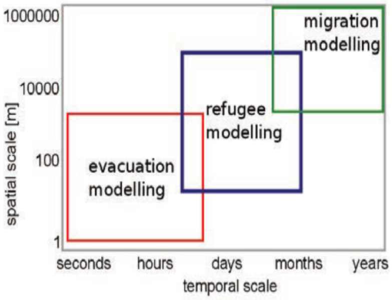


Figure 5: Spatial and temporal scales of evacuation, refugee and migration modeling (Groen 2016).

Python code combines the csv data, describing geographical locations and how their population changed through time, and produces data in the standard GeoJSON format. The visualization itself is created in HTML and JavaScript using the popular Leafletjs library to display maps and the Leaflet.timeline plugin to animate them. We set out to visualize the movement of people during a conflict on a map, with circles which grow and shrink as the population in each location changes as simulation period progresses. The colors of the circles distinguish cities, conflict zones and refugee camps as illustrated in Figure 6. This is a simple and additional application to earlier discussed automation tool. We also intent to add connection links between conflict zones, intermediate towns and camps with distance labeling and hence require further advancements at a later stage.

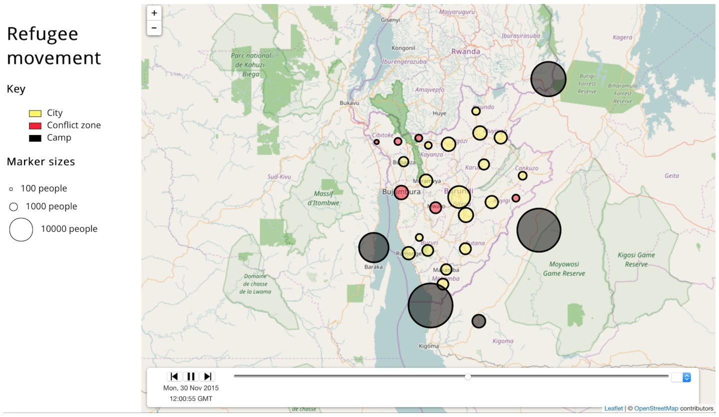


Figure 6: Visualflee representation.

5 CONCLUSIONS

Chandra and Grabis (2003) state that computational modeling with an automated construction helps to avoid errors and inaccuracy in the simulation process. It is because an automated simulation can embed all stages of modeling into a user-friendly interface providing an opportunity to construct models. We presented a vision to automate the construction of refugee movement simulations from start to end, combining data from a range of publicly available sources and highlighting five key stages, and relying on existing automation tools. To summarize, Table 4 illustrates an overview of the current state and the ultimate aim of our conventions and contributes to automation of refugee simulation.

The major contribution of this paper is to rapidly and efficiently predict refugee movements and potentially to save refugee lives. As the automated agent-based simulation modeling will allow governments and non-governmental organizations (NGOs) to regulate the distribution of humanitarian resources, as well as ability to examine and incorporate implications of policy decisions. Consequently, an automated simulation model for refugee movements will intend to provide an exploration of parameters forcing people to flee for policy purposes. The automated simulation approach is essential to systematically investigate the effect of policy decisions through simulation, as manual simulation construction is simply too labor-intensive.

Although our vision is developed specifically for constructing simulations of refugee movements, we argue that several components of the design can be adopted for other purposes. In particular, the first two steps (also highlighted in Figure 3) could be reused in different contexts to provide a geographic environment for other purposes, such as the simulations of longer-term migration or international conflicts.

Table 4: An overview of the current state and the ultimate aim of our conventions for an automated construction of refugee simulation.

Conventional	Our current state	Ultimate aim
Manual extracting and converting UNHCR data	manual extraction of CSV files for camps, automated conversion to simulation datasets	automated mining of UNHCR portal + conversion to simulation data and parameters, given a selected conflict.
Manual downloading and filtering ACLED data	manual download of XLS files for conflict zones, automated conversion to simulation datasets	automated downloading of ACLED data with necessary information + conversion to simulation data given a selected refugee situation.
Manual searching and extracting population distributions	manual search of location populations for conflict situations, automated conversion to simulation datasets	automated mapping of population numbers + conversion to simulation data give a selected country
Manual searching and determining of geographical data and distances	manual search of latitudes and longitudes, as well as distance calculations between conflict areas, camps and intermediate towns	automated determining of geographical data and distances between two locations for a selected conflict situation
Manual constructing of network maps	automated detection of location, coloring to specific location type but manual link of locations, sign of distances	automated construction of network maps + conversion to a selected simulation scenario

ACKOWLEDGEMENTS

This project has received partial support from the Science Hackathon Geneva 2017 hosted by CERN and the Collaborations Workshop 2017 organized by the Software Sustainability Institute. DG has received partial support from the European Unions Horizon 2020 research and innovation programme under grant agreement No 671564 (ComPat project). We are grateful to Dr Mario Antonioletti, Dr Andy South, Dr Joanna Leng and Davide Alocci, who contributed in parallel and gave insights into an automation construction of refugee modeling. We also thank Dr Edward Smith and Dr Thomas Kluyver for their contribution in developing a visualization platform within the Visualflee project.

REFERENCES

Michael Alder 2001. “An Introduction to Pattern Recognition: Statistical, Neural Net and Syntactic Methods of Getting Robots to See and Hear”. http://index-of.co.uk/Artificial-Intelligence/An%20Introduction%20to%20Pattern%20Recognition%20-%20Michael%20Alder.pdf.

Allan, R. 2010. “Survey of Agent Based Modelling and Simulation Tools”. Technical Report Version 1.1, Science and Technology Facilities Council (STFC) Daresbury Laboratory, Daresbury, Warrington.

Amblard, F., A. Geller, M. Neumann, A. Srbljinovic, and N. Wijermans. 2010. “Analyzing Social Conflict via Computational Social Simulation: A Review of Approaches”. In *Complex Societal Dynamics - Security Challenges and Opportunities*, edited by K. Martins, D. Matika, and A. Srbljinovic, 126–141. Amsterdam.

Boano, C., A. Rottlaender, S. Alberto, and F. Viliani. 2003. "Bridging the Gap: Involuntary Population Movement and Reconstruction Strategy". Technical report, Centro Regionale dIntervento per la Cooperazione, Reggio Calabria, Italy.

Bonabeau, E. 2002. "Agent-based Methods and Techniques for Simulating Human Systems". In *Proceedings of the National Academy of Sciences of the USA*, Volume 99, 7280–7287.

Castle, C. J. E., and A. T. Crooks. 2006. "Principles and Concepts of Agent-based Modelling for Developing Geographical Simulations". In *CASA Working paper series*, Volume 110. University College London, London, UK.

Chandra, C., and J. Grabis. 2003. "A Data Driven Approach to Automated Simulation Model Building". In *Proceeding of 18th European Simulation Symposium*, edited by A. Verbraeck and V. Hlupic, 26–29.

Collins, A. J., and E. Frydenlund. 2016. "Agent-Based Modeling and Strategic Group Formation: A Refugee Case Study". In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Z. T. Huschka, and S. E. Chick, 1289–1300: IEEE.

Curcin, V., and M. Ghanem. 2008. "Scientific Workflow Systems - Can One Size Fit All?". In *Biomedical Engineering Conference, CIBEC*, 1–9. Ciaro International: IEEE.

Davenport, C., W. Moore, and S. Poe. 2003. "Sometimes You Just Have to Leave: Domestic Threats and Forced Migration, 1964-1989". *International Interactions* 29 (1): 27–55.

Disney, G., A. Wiśniowski, J. J. Forster, P. W. F. Smith, and J. Bijak. 2015. "Evaluation of Existing Migration Forecasting Methods and Models". Technical report, Centre for Population Change, Southampton, UK.

Faist, T. 2000. *The Volume and Dynamics of International Migration and Transnational Social Spaces*. Oxford: Oxford University Press.

Gebbie-Rayet, J., G. Shannon, H. H. Loeffler, and C. A. Laughton. 2016. "Longbow: A Lightweight Remote Job Submission Tool". *Journal of Open Research Software* 4 (1): 1–8.

Gleditsch, N. P., P. Wallensteen, M. Eriksson, M. Sollenberg, and H. Strand. 2002. "Armed Conflict 1946-2001: A New Dataset*". *Journal of Peace Research* 39 (5): 615–637.

Groen, D. 2016. "Simulating Refugee Movements: Where Would You Go?". *Procedia Computer Science* 80:2251–2255.

Groen, D., A. P. Bhati, J. Suter, J. Hetherington, S. J. Zasada, and P. V. Coveney. 2016. "FabSim: Facilitating Computational Research Through Automation on Large-scale and Distributed e-Infrastructures". *Computer Physics Communications* 207:375–385.

Huang, Y., M. D. Seck, and A. Verbraeck. 2011. "From Data to Simulation Models: Component-based Model Generation with a Data-driven Approach". In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, 3724–3734. Phoenix, USA: IEEE.

Itani, M. A., U. D. Schiller, S. Schmieschek, J. Hetherington, M. O. Bernabeu, H. Chandrashekar, F. Robertson, P. V. Coveney, and D. Groen. 2015. "An Automated Multiscale Ensemble Simulation Approach for Vascular Blood Flow". *Journal of Computational Science* 9:150–155. Computational Science at the Gates of Nature.

Junjie, L. L., K. M. Carley, and S.-F. Cheng. 2016. "An Agent-based Approach to Human Migration Movement". In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 3510–3520. Arlington, Virginia: IEEE.

Lopez-Lucia, E. 2015. "Early Warning Models for Irregular Migration". Technical report, Governance Social Development Humanitarian Conflict (GSDRC) Applied Knowlegde Services.

Macal, C. M., and M. J. North. 2009. "Agent-based Modeling and Simulation". In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 86–98. Austin, Texas: IEEE.

Macal, C. M., and M. J. North. 2010. "Tutorial on Agent-based Modelling and Simulation". *Journal of Simulation* 4 (3): 151–162.

Mernik, M., J. Heering, and A. M. Sloane. 2005. “When and How to Develop Domain-specific Languages”. *ACM Computing Surveys* 37 (4): 316–344.

O’Brien, S. P. 2010. “Crisis Early Warning and Decision Support: Contemporary Approaches and Thoughts on Future Research”. *International Studies Review* 12 (1): 87–104.

Raleigh, C., A. Linke, H. Hegre, and J. Karlsen. 2010. “Introducing ACLED-Armed Conflict Location and Event Data”. *Journal of Peace Research* 47 (5): 651–660.

Schmeidl, S. 1997. “Exploring the Causes of Forced Migration: A pooled time-series analysis, 1971-1990”. *Social Science Quarterly* 78 (2): 284–308.

Shami, S. 1996. “Transnationalism and Refugee Studies: Rethinking Forced Migration and Identity in the Middle East”. *Journal of Refugee Studies* 9 (1): 3–26.

Sokolowski, J. A., and C. M. Banks. 2014. “A Methodology for Environment and Agent Development to Model Population Displacement”. In *Proceedings of the 2014 Symposium on Agent Directed Simulation*, 3: Society for Computer Simulation International.

Sokolowski, J. A., C. M. Banks, and R. L. Hayes. 2014, 12. “Modeling Population Displacement in the Syrian City of Aleppo”. In *Proceedings of the Winter Simulation Conference 2014*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 252–263. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Suter, J. L., D. Groen, and P. V. Coveney. 2015. “Chemically Specific Multiscale Modeling of Clay-polymer Nanocomposites Reveals Intercalation Dynamics, Tactoid Self-assembly and Emergent Materials Properties”. *Advanced Materials* 27 (6): 966–984.

Swaminathan, J. M., S. F. Smith, and N. M. Sadeh. 1998. “Modeling Supply Chain Dynamics”. *Decision Sciences* 29 (3): 607–632.

UNHCR 2010. “Convention and Protocol. Relating to the Status of Stateless Refugees”. http://www.unhcr.org/uk/3b66c2aa10.

UNHCR 2017. “Figures at a Glance”. http://www.unhcr.org/uk/figures-at-a-glance.html.

Wilde, M., M. Hategan, J. M. Wozniak, B. Clifford, D. S. Katz, and I. Foster. 2011. “Swift: A Language for Distributed Parallel Scripting”. *Parallel Computing* 37 (9): 633–652.

Yilmaz, L. 2007. “Tools for Conflict and Peace Studies”. *Social Science Computer Review* 25 (1): 48–60.

Yilmaz, L., T. I. Oren, and N. Ghasem-Aghaee. 2006. “Simulation-based Problem-solving Environments for Conflict Studies”. *Simulation & Gaming* 37 (4): 534–556.

AUTHOR BIOGRAPHIES

DIANA SULEIMENOVA is a Ph.D. researcher in the Department of Computer Science at Brunel University London. Her current research focuses on forced migration models and developing an automated simulation modeling technique to predict refugee movements. Her email address is diana.suleimenova@brunel.ac.uk.

DAVID BELL is a Lecturer in Pervasive Informatics at Brunel University London. He holds a BSc (Hons) in Computing and Electronics (Swansea), a MBA (Warwick) and a PhD in Information Systems (Brunel). His research interests lie at the interface between software, ubiquitous hardware and the human users in development of new forms of data-driven interaction and marketplaces that use Cloud for scale, big data for intelligence and semantic technologies for modelling. His email address is david.bell@brunel.ac.uk.

DEREK GROEN is a Lecturer in Simulation and Modelling at Brunel University London, and a visiting Lecturer at the Centre for Computational Science, University College London. He obtained his PhD in 2010 at the University of Amsterdam. His main research interests are in high-performance computing, multiscale computing, blood flow simulation and refugee modelling. Derek is PI for Brunel in two EU-funded projects (ComPat and OpenMultiMed) and has published more than 50 peer-reviewed papers. His email address is derek.groen@brunel.ac.uk.