Bachelor Thesis
15 ECTS

Cultural awareness in social agents in activity generation on a macro scale.

Alex Klein
3848523
Utrecht University

Supervisors:
Frank Dignum (UU), Steven de Jong (TNO), Ruben Smelik (TNO)
Second reader: Jan Broersen

February 18, 2016

Abstract

In this thesis a model for cultural awareness for agents in activity generation is proposed. A social network generation process is used to represent social connections necessary for realistic cultural behaviour among agents. Using need-based activity generation, that uses random utility maximisation, agents will plan their agendas for multiple days. Special activities are introduced to model dynamic decision making of the agents during the day, called dynamic activities, next to the more static activities that are planned for the whole day beforehand. Hofstede's model of culture is used to adjust the utility functions, to make the agents more cultural aware. An experiment is set up to show the effectiveness of the model, i.e. the agents' behaviour is changed according to the expectations of Hofstede's model, and it is shown why the results are non-conclusive.

Contents

1 Introduction

1.1 Background

In recent years there has been an increased interest in training through simulation for Defence. Because many modern military missions are performed in other countries, training needs to account for the various challenges that come with such missions. To make realistic and effective virtual training situations, these simulations need to reflect the possible environment that these mission may take place in. This includes the physical environment, the people acting in that environment and the behaviour these people have. The latter comes down to the cultural differences between those who perform the mission and the local population. Although work on modelling the physical environment and the population has been done to some detail, cultural affected behaviour has not been researched much. Although some research has been done on cultural affected agent-based systems, these are normally made for specific situations, with a limited number of agents [12] /[15].

Silverman et al. [22] did develop a simulation for military training purposes called NonKin, which incorporated culture in to the behaviour of the agents in a village. A trainee had to perform tasks to make the various factions, all with positive or negative attitudes towards other faction, work together for the greater good of the village. The aim of NonKin was to teach trainees about culture dependent interactions.

1.2 Research Objectives

In this thesis a model is proposed for culture based activity generation. The culture is based on Hofstede's dimensional model of culture. The simulation is implemented within the IDSA framework, developed by TNO [7]. Agendas are then created based on need-based activity generation that uses utility maximisation to plan activities. I will show how different cultures can affect the behaviour of the agent, by running the simulation with the cultural values of The Netherlands and Portugal and show that this has an impact on the way the agents lower their needs and how they plan their activities throughout a couple of weeks time.

Activity generation only plans activities at the start of every day, which makes then static for that day. Because people can decide to change the plans they made originally throughout the day, dynamic activities are added. These can overwrite activities that were already in the agenda.

1.3 Research Questions

The main research question is:

- How can culture be incorporated in activity generation?

Other questions that need to be answered are:

- What study of culture is best suited for modelling?
- How can activity generation be implemented, as well as dynamic activities?
- What aspects of culture can be modelled in activity generation?
- How can we generate social networks, such that they can model cultural adjustments?
- What kind of differences are expected from two different cultures?

1.4 Outline

In the first part existing research is covered: first culture studies and the reasons why Hofstede's model is used for modelling, as well as a description of Hofstede's model. This is followed by a detailed description on the IDSA system. Its components and the data and algorithms are covered. Current issues and some improvements are discussed as well.

Then, some improvements are discussed, necessary to model culture later on. First an improved social network is described. Although the IDSA system has a social network, it connects the agents randomly and does not take semantics in the world into account. Because culture is mostly reflected in social interactions among people, a realistic social network among the agents is needed. Then the need-based activity generation algorithm is discussed, where agents plan certain activities based on the needs they have. Special cases like needs for households as a whole, social activities and dynamic activities are also discussed. This followed by modelling of culture and how social interactions and activity generation is influenced by culture.

This is continued by an experiment where we will compare the behaviour of the agents between Dutch and Portuguese national cultural values. Results from the two simulations will show differences in planned activities and changes in needs over several weeks simulated time. Finally we come to the conclusion and possible future work.

2 Culture

Culture can be seen as the pattern of behaviour that emerges from the rules that underlie a group of people. As Hofstede [11] describes it: culture lies between people's primitive needs and their own personality. It could be seen as the personality of a group.

This section starts with a general overview of several cultural studies. This is by no means exhaustive, but it gives an overview on work that has been done in the field. The next section describes Hofstede's multi-dimensional model in more detail. In the last part an overview of the modelling of culture for simulations purposes is shown.

2.1 Cultural Studies

Since the 1960's there has been increasingly sophisticated research on culture. This comes from various fields, including sociology, economics and business. E. T. Hall was one of the first to describe culture in a multi-dimensional model [26].

Later, two major publications on culture were done by Trompenaars and Hofstede [11]. They have similar theories, but different aims. Trompenaars is a business consultant and his work is aimed at that. His work is not peer-reviewed and lack verifiable empirical evidence. Hofstede took a more scientific approach, his work is peer-reviewed and quantifiable. Both take the dimensional approach Hall had, just with different semantics.

In 2009 Solomon and Schell [24] published a modern dimensional approach, where most people from various cultures agree on its intuitive correctness. It is inspired by previous work of Hall, Trompenaars and Hosftede, but they reevaluated the dimensions to describe culture. Their work lacks however a description of their approach and the empirical evidence they base their theory on. Although they have quantified national culture to some extend, where every country has a score between 5 and 25 for every dimension, they place these in five buckets. So we would know all countries that have a value between 5 and 9, which is too limited to base a simulation on.

Because of the scientific background and quantifiability, Hofstede's work is the best work on culture for modelling. Hofstede's model is also been adjusted over the years to take new insights on culture into account.

2.2 Hofstede's Cultural Dimensions

Hofstede describes a total of six cultural dimensions [11]. These are power distance, individualism versus collectivism, masculinity versus femininity, uncertainty avoidance, long term versus short term orientation and indulgence versus restraint. The first four are based on a survey held among the employees of IBM in various countries. Long term versus short term orientation results from a similar survey, which was made by Chinese students, to make decrease the possible Western bias of the first survey. Indulgence versus restraint is mostly adopted from Minkov, who researched the World Value Survey.

Every national culture in their work has a relative value between 0 and 100. This means that an IND of 30 means that a culture is more collectivist than a culture with an IND of 60. It does not mean that the latter is twice as individualist as the first.

In the following paragraphs the six cultural dimensions are explained to more detail. Note that the two extreme ends of the dimensions will be discussed, to give an intuitive notion about the meaning of these dimensions. In real world cultures, they are more likely to lay somewhere in between these extremes. Also note that these are used to describe national cultures, not individual people. Individuals may differ a lot from the national values: the national values are the averages of the values of the people within these societies.

2.2.1 Power Distance (PDI)

Power distance refers to the way people deal with authority and power inequality. When the Power Distance Index (PDI) is high, people are more used to are large power distance. This means people will be more acceptable to the fact that their boss, parents or teacher are higher places than themselves. Contradicting a person that is seen as higher instance than you is not done. A quite strict hierarchical order among people means that they also expect differences in chances and resources between the powerful and powerless.

In low PDI cultures, people seek equality among all the people and power is distributed among the people. If there is a hierarchy, it is mostly formal and when people are working together they threat each other equally, regardless of the set hierarchy. For example, workers can call their boss out on mistakes or give suggestions.

2.2.2 Individualism versus Collectivism (IDV)

IDV is about how people define themselves: as an individual or as part of a group, "I" or "we". In a collectivist culture, when IDV is low, people form strongly bonded in groups, which can be family or another form of in-group. Family, friends and partners are predetermined from birth and unquestionable loyalty is expected within a group. Not following the group can be seen as betrayal and opinions of a person should be conform the opinion of the group.The interest of the group is more important than personal interest.

When IDV is high, people show the same amounts of respect to all other people. Time and effort should be spent on friendships to maintain them and throughout people's lives these friendships may change a lot. People are less connected in an individualist society than in collectivist societies.

2.2.3 Masculinity versus Femininity (MAS)

Masculine societies favour achievement, heroism and material reward. Stereotypes about the roles of men and women in society are strong: men should be macho and women should be caring. People tend to assertive.

In feminine culture people are more tender and caring. Society should take care of people who have it worse. Cooperation is the standard to solve problems and people are looking for consensus and comprises.

2.2.4 Uncertainty Avoidance (UAI)

UAI describes how people handle uncertainty and ambiguity. It also describes how people look on fate: can one control their own life, or does one follow their fate? When UAI is high, people avoid others who are different to them and situations they are uncomfortable or unfamiliar with.

In low UAI cultures people are more relaxed and more likely to do seek out new experiences and take a risk. Skills are more important than talent and can be improved by practice.

2.2.5 Long Term versus Short Term Orientation (LTO)

LTO describes how people perceive time. In long term oriented cultures people want result in the far future, even if that means that their will not be much gain on the short run. They also make more use of knowledge from the past to solve modern problems and traditions are seen as essential.

In short term societies, actions should have results fast, otherwise they are seen as not working. Financial profits are expected within a short period of time and problems should be solved in a quick fashion, even if that implies a similar problem will occur later on.

2.2.6 Indulgence versus Restraint (IND)

IND describes how people deal with their (primitive) needs. In indulgence cultures people will be comfortable fulfilling their needs. They tend to be more out going and sexuality is more accepted. The aim of life is on having fun and enjoying life.

In restraint cultures there are strict rules on what can and cannot be done to deal with one's needs and strict social norms about what behaviour is accepted.

2.2.7 Limitations

There are some limitations to Hofstede's model. The first is that the values in Hofstede's model are relative. This makes modelling cultures more difficult, because it is unclear how to implement functions using these values. Absolute values would have been better suited for computational modelling. Absolute values would make easier to quantify the impact of culture. For example, that an IDV 0f 20 is actually twice as individualist as and IDV of 10.

The addition of other countries resulted in some inconsistencies. The original countries were scaled between dimensional values of 0 and 100, but when new countries were added, some did not fit on this original scale, so some countries got values above the 100. This breaks assumptions that could be made about the values.

Finally, Hofstede's model describes cultural values, not their norms. These values can be used to explain the norms in a culture, but no information about the norms can be concluded, knowing the values. Norms are also based on variables like the climate and history of the country. For computational modelling it would be a useful addition to have the norms of a national culture in the same model as their values.

2.3 Agent-based Culture Models

There has been some research in modelling of cultural aware agents in the recent years, using Hofstede's model.

Hofstede et al. [12] modelled negotiating agents that make different decisions based on their national culture. They used utility functions for the decision process, which was influenced by the cultural values. They showed how the decision making of the cultural aware agents was in line with the expected behaviour based on Hofstede's multi-dimensional model.

Mascarenhas et al. [15] showed how agents with different cultural values treat each other differently, using the Belief-Desire-Intent model to represent knowledge of the agents. They showed how these agents from different cultures react to someone they do not consider part of their "in-group": the people they consider part of their group, as part of their "we".

Schram [21] also used the the BDI model to model different reactions and effects to various actions among different cultures. The build a simulation where agents' beliefs are updated in reaction to actions of others, like leaving food on a plate. Within this simulation, different cultures have different reactions to such actions.

A complete serious gaming system for training purposes called NonKin was developed by Silverman et al. [22]. In NonKin a village of roughly 100 to 200 agents could be simulated. The agents would all have individual goals and opinions on other agents and keeping track of actions taken in the world. Agents are also part of factions, which also have relation between each other. A trainee has to make various factions work together for the greater good of the village. The interaction norms can be added to the agents, which applies to both interactions with the player, as well as other agents. These norms, together with the goals of agent, is how culture is modelled in NonKin.

3 The IDSA System

IDSA stands for Intend Driven Scenario Authoring [7] [14] and was developed at TNO, at the Modelling, Simulation and Gaming group. IDSA had several goals:

- Model a world and population based on data;
- All agents should have an agenda for a day;
- Non-scripted, dynamic events can be placed by the user. Agents will react to these events and will return to their normal behaviour, when the event is over.

IDSA is a multi-agent simulation where people will do their every day life, like going to work, bringing their children to school and doing groceries. This everyday life can be interrupted by events that are placed by the user. The system will construct these events at the moment the user places it: they are not scripted and should be different every time. Only one day is simulated each time, so an agenda is planned for one day only. For the demo the Dutch municipality of Rijswijk was modelled, which will also be the test-case for this thesis.

The IDSA can be seen as three parts: the simulation, the pre-processor and the event planner. The simulation takes care the behaviour of the agents at run-time, the visualisation and user input. The pre-processor consists of the world data model, the population generator and agenda planner. The world data model will construct the world from data, the population generator will model the population and their properties based on data and the agenda planner will give all the agents their agenda for that day. The event planner translates requests for events from the user to actual behaviour of the agents. It does this in two steps: first a planner makes a sequence of actions that need to be performed for the event. Second, a sampler looks for agents that need participates in the event.

The following sections will the describe the IDSA system in more detail. The first three sections cover the pre-processor: the world data model, population generator and agenda planner, respectively. Then the event planner is discussed in more detail and finally some improvements that have been added to make it more suitable for future modelling.

3.1 World Data Model

The world data model builds the world for the simulation. The environment model, which contains buildings, infrastructure, and outdoor areas such as parks and water, is derived from geographical open data sources, such as OpenStreetMap. It includes a functional description of each building or public area, which is represented in a hierarchy that features multiple inheritance. For example, a shop is inside and it is a place where people work and a place where people can buy products.

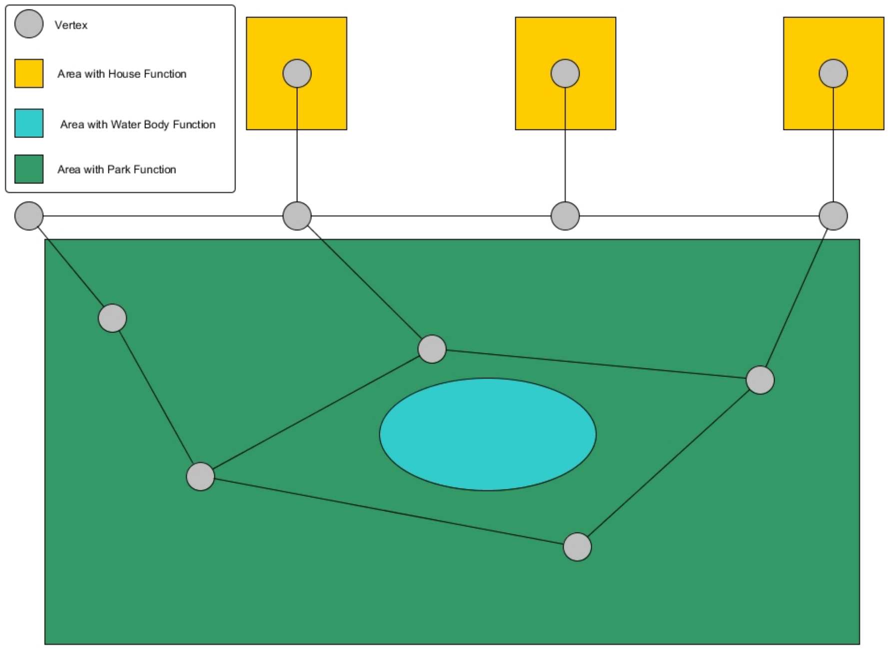

Figure 1: Simple example of the data structure of the world data model.

Each building or area also includes relevant environmental properties such as housing capacity, taken from the Dutch building register. In cases where data is insufficiently detailed, incomplete, or inconsistent, it is estimated based on heuristics (such as the size of a building), or, in some cases, manually provided.

3.1.1 Representation

The world is represented as a graph, where edges are roads where the agents can walk, and nodes connect these edges. Some nodes may be linked to an area, which can be any place that is of interest, like a house, shop, park or workplace. Most areas will only have one node, like houses and workplaces, but some have multiple nodes in their area, like parks where people can walk around. These areas will have a list of functions, as one area can have multiple functions, like a shop can be a workplace and a place to get groceries, but an area can also have multiple of the same function, because one area covers for example six apartments. These functions will have some properties, like surface area and capacity, which reflect the maximum number of inhabitants or workers.

3.1.2 Data

The world is based on data from OpenStreetMap and the Dutch building register Basisadministratie Adressen en Gebouwen (BAG). We use OSM to construct the graph, where the edges are the streets. Then OSM and BAG are combined to load the areas and these areas are then linked to the vertices of the graph. Then function descriptions are added to the areas based on a combination of BAG and OSM. The primary source is BAG, but sometimes BAG lacks a proper labels, so OSM is used to complement BAG data where necessary and possible. After this some buildings still have no function attached to them, so the lack function descriptions are added by making an educated guess what the function could be, by looking at the surface area or the surrounding buildings, because if a building is surrounded by only houses in a neighbourhood, it is very likely the unlabeled building is also an house.

3.2 Population Generator

The population generator creates the population based on data and places them in the world. First section will give a description of the agent, then the data that is used for the population is discussed. We continue with the algorithm used to model the population and finally the social network is discussed.

3.2.1 Agents

The agents in the IDSA system have a couple properties, being: gender, age, household type and household role. On top of this they have knowledge of who is part of their household and the location of their house. All of these properties are given to them by the population generator.

They also have a model stack, which is used to give the agents their behaviour. A behavioural model takes control over an agent and make the agent act out their behaviour. This can be simple behaviour, like just standing still or walking from one place to another, or more complex, e.g. following another agent at a distance. The model stack keeps track of the order the models need to be ran. The model at the top of the model stack is the one that is currently running or the one that needs to be started. When a model has finished it is removed from the stack and the next model is started.

Models can be pushed on the stack from two places: an activity in the agenda or an event that needs the agent. Every activity knows which model is needed and an activity is allowed to push its models onto the stack if the agent's model stack is empty. The agent will then do a look up in the agenda to see which activity is next and that activity is then allowed to push its model. Events on the other hand take control regardless of the models already on the model stack. As soon as the agent is sampled for the activity, new models are pushed on to the model stack and the event's models are therefor started to make the agent act in the event. When the models of the event are finished and popped from the model stack, the old models are checked for relevance. If the model can no longer be acted out, for example when the activity they belong to is already over or the agent cannot reach the activity on time, it is popped from the model stack.

3.2.2 Data

To model the population of Rijswijk, data from the Dutch Central Bureau of Statistics is used. This includes multiple data sources from the CBS. First, data about the local population is used. These are data tables about neighbourhoods, which covers up to a couple of thousand people. This includes fields for, but is not limited to, the number of males and females in neighbourhood, percentages for age categories and average household size. However, it lacks detailed conditional information, like how many men are in their twenties. We do know P(gender = male) and $P(15 < age < 25)$, but $P(15 < age < 25|gender = male)$ is unknown. This information is not available for the neighbourhoods, out of privacy considerations. Another issue with this data is that some values are rounded. For example, because of rounding of the values, the number of males plus the number of females might not be equal to the number of inhabitants.

To add this information we will assume that the national data about this is sufficient to bridge this gap in the data. Now we do know the conditional probabilities, but at the cost of having information that might not be completely in line with the real world. We will try to make this error as small as possible, as we will discuss later.

The national data contains two datasets that we will use together to make the households. The first dataset contains numbers for all combinations of gender, age category (in groups of ten years), household type and civil status. Household type can be Pair if they have children, In a relationship if they do not have children, Single parent or Single. The second dataset contains the number of children given the age and household type of the reference person of the household. If used together, we can construct one person from the first dataset and if necessary draw how many children this agent has and use the first dataset again to draw their partner and children.

3.2.3 Algorithm

There are two main algorithms found in the literature for population synthesis: Synthetic Reconstruction (SR) and Combinatorial Optimisation (CO).

SR will first construct one large data table that combines all the available data and estimates a probability distribution from that data table. The unknown conditional data is estimated using Iterative Proportion Fitting. It does this by iteratively estimating unknown fields between two data tables, which are guaranteed to convert to stable values.

CO first uses the the data on the largest scale, e.g. national, and constructs a population based on that large scale data, by doing Monte Carlo draws from the data. CO then uses the more local data to optimise the results from more general data to the local data, while maintaining the properties of the population from previous data sets.

We have chosen to use CO, because it gives more realistic results in general, at the cost of more computational time. Our population of 30 thousand people is relatively small, so the computational cost was not much of an issue. In our simulation we first use the national data to draw households, so we have agents with an age, gender, household type (e.g. Pair, Single Parent or Single) and their household role (mother, father, child or single). We make five times more agents than we want to fit in the world: we also keep a pool pf households that we can use for optimisation. Then all the houses get filled with households that fit within the capacity of the house. The next step is to optimise the population to the local data of the neighbourhoods. Per neighbourhood we keep track of the total absolute error (TAE), which takes into account differences in number of inhabitants, males, females and five age categories. A random household is chosen within the neighbourhood and one within the household pool and we swap these households if swapping them lowers the TAE. After a set number of iterations (1,000,000 per neighbourhood) we stop optimising. This gives us optimal results for all the neighbourhoods, although it does not gives us TAE of zero everywhere, because of the rounded values in the neighbourhood data.

3.2.4 Social Network

Between the agents is a social network, which indicates which agents are friends and could do activities together. The social network is constructed based on the work of Toivonen et al. [25]. The algorithm uses five steps:

1. Make an initial network of N_0 agents;
2. Find for an agent that is not yet in the network an average of m_r initial contacts, where $m_r \geq 1$;
3. Then find the agent on average m_s secondary contacts. Secondary contacts are contacts of the initial contacts, where $m_s \geq 0$;
4. Connect the agent to the initial and secondary contacts and connect the contacts to the agent;
5. Repeat step 2 tot step 4 until all agents are in the network.

Agents are explicitly connected to their household members. This approach has nice properties, because we can easily adjust the average number of contacts and the clustering by adjusting m_r and m_s respectively.

However, this algorithm leaves out many details and does not take into account semantics in our world model. People do not know their neighbours or their coworkers (at least, it is very unlikely) and no information about the agents themselves. In the real world, similar people are more likely to become friends, where in this case we could use age and gender. Later we will discuss improvements made to the social network to include these semantics.

3.3 Agenda Planner

When the world has been constructed and the population synthesised, all agents need to get an agenda for one day. These activities should not overlap and the agent should have enough time to walk from activity to activity.

First we will cover what activities are and how they are planned. We continue with the constraint satisfaction that makes sure activities do not overlap. Third, the multipliers, which control how frequently a certain activity gets planned, are covered. Then the use of the agendas during the run-time simulation is discussed and finally the issues with the current system and what could be improved.

3.3.1 Planning Activities

An agenda is an ordered list of activities, where the activities are ordered by starting time (or ending time, because we assumed that activities do not overlap in time). An activity is a tuple a, l, p, s, e, where a is the name of the activity, l is the location of the activity, p is a group of agent that participate in the activity, s is the start time and e is the end time. The tuple also indicates the order in which the variables are given their value in the planning process.

The agenda planner has an order in which activities are planned. First mandatory activities are planned, which are, in order: parental duties, work and school, and dinner. After these activities have been planned, all the other optional activities are planned, like taking a walk, doing groceries and visit friends. Parental duties were introduced to cope with the fact that very young children cannot yet do activities on their own and need to be accompanied by some form of supervision. So parents will bring young children to school and pick them up again later. This is planned before even work is planned to ensure that at least one of the parents is available to take care of the parental duties. Then work and school is planned. We will refer to both school and work as work, because they are planned in a similar fashion. Some agents will not have these planned, because they are retired or just have a day off. Then dinner is planned, if the agent does not already have plans around dinner time. Then all the other activities are planned around these mandatory activities.

To plan a specific activity, all the variables in the tuple a, l, p, s, e are instantiated. We pick a certain activity to plan. How these are picked will be covered under the multipliers. So we already know a. Then a location l is selected. Normally a search through the graph is done to find nearby locations that fit the description given by the activity we are planning, e.g. a shop when an agent is planning to do groceries or a workplace when planning work. A timeframe that is still available is picked (timeframes will be covered under constraint satisfaction) for further planning. From the list of possible locations one is picked that is close enough to the location of the previous activity. If no previous activity has been planned yet, the house of the agent is used as the previous location. The same holds for the activity that comes after the activity that is being planned. If more agents should take part in the activity the list of contacts of the agent is used to search for available people, otherwise one agent is the only participant. Lastly the start and end time are picked within the selected timeframe, such that the agent can still make it to the activity and to the next activity.

Then the activity is inserted in the right place in the agenda of the agent. Figure 2 shows a graphical description of the agenda planner.

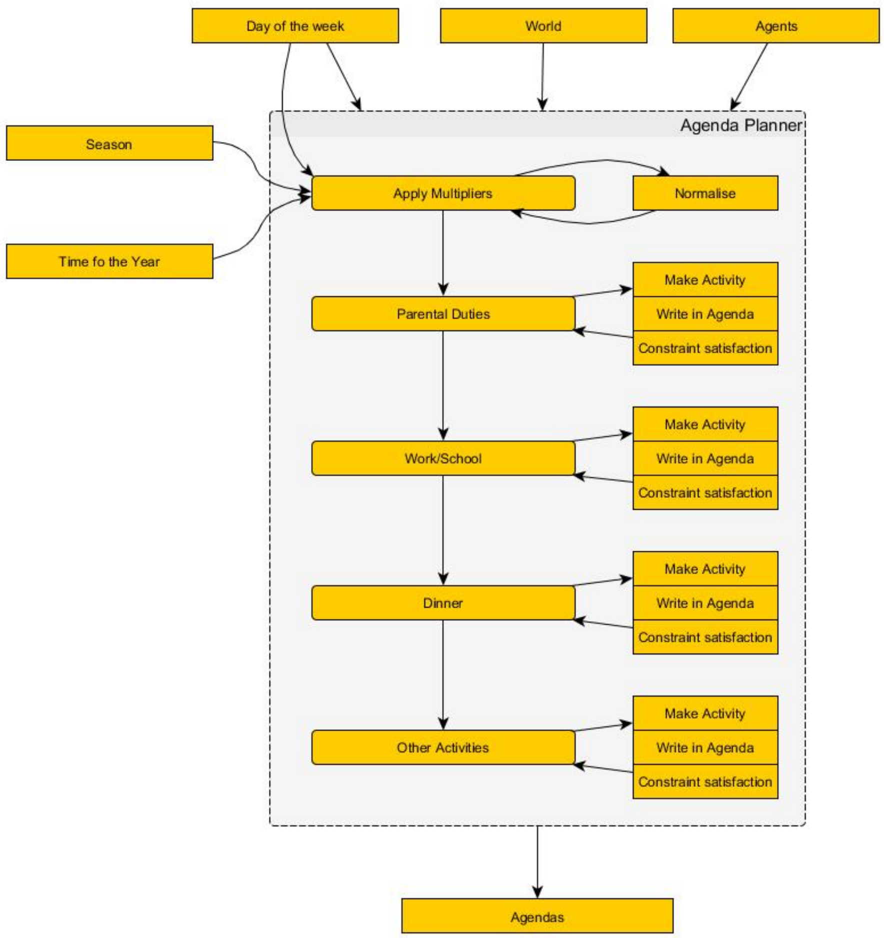

Figure 2: Diagram of the agenda planner.

3.3.2 Constraint Satisfaction

When planning activities, each activity that can still be added to the agenda keeps track of the times it can be planned. It should not be possible that more than one activity is planned on a certain moment in time. To make sure such temporal clashes in the agenda do not happen, constraint satisfaction, based on Forward Checking [**?**], is used.

Every activity has a list of timeframes in which it can be planned. A timeframe is a tuple $< s, e >$ where s is the start of the timeframe and e the end. So for example, doing groceries is can be planned as long as the grocery shop is still open, which might be between $8:00$ and $18:00$, which means that the initial timeframe for grocery shopping is $< 8:00, 18:00 >$. If another activity is planned on a time that overlaps with the timeframe of the activity, the timeframe is adjusted. This way it is no longer possible to plan two activities at the same time. So, continuing our example, if another activity is planned that starts at $7:00$ and ends at $10:00$, the timeframe for grocery shopping is shortened to $< 10:00, 18:00 >$. A special case is when an activity is planned within the the timeframe. This makes that the timeframe has to split. So if an activity is planned from $12:00$ to $14:00$, the timeframe will split and the grocery shopping will keep track of both timeframes: $\{< 8:00, 12:00 >, < 14:00, 18:00 >\}$. Either of those timeframes can be used for planning, but it is not possible to plan any activity outside of its timeframes.

Every activity also has minimum and maximum duration, d_{min} and d_{max}. If a timeframe becomes shorter than $d_{min}$the timeframe is removed from the list of timeframes. If the list of timeframes becomes empty, the activity can no longer be planned.

3.3.3 Multipliers

When selecting activities for planning, there is normally a chance it will be planned (except for parental duties, which must be done by one of the parents). So if an activity is selected and if a generated random number between 0 and 1 is lower than the chance, the activity gets planned. Otherwise the activity is no longer available for planning.

Because of the difference age and gender among the agents, some agents should be more likely and some less likely to plan certain activities. For example, elderly people are less likely to go to work, but might be more likely to go for a walk in the park that day. For this we use multipliers. We can adjust these chances with these multipliers to make agents with certain characteristics more likely or less likely to plan certain activities. A special case is the multiplier of 0, which makes an activity no longer available for planning. This can be used to make sure young children cannot go to work, but have to go to school.

These multipliers can also be used in a global sense. For example, summer will make it globally more likely for agents to go to the park and less likely in the winter. This is season based. The alternative is more culturally based, like fewer working people and fewer children going to school on holidays.

3.3.4 Use of Agendas

During the simulation, agents will do look ups in their agendas. When an agent's modelstack becomes empty, when the agent has finished the previous activity and is not involved in an event, the agent will look for the next activity. The model from the activity will be pushed onto the agent's modelstack, which will start and end at the time as stated by the activity, and a walking model to the location of the next activity is pushed on top of that. The agent will then start by walking to the next location and when they arrive they will start the next activity.

The same hold for the first activity of the day. If there is no next activity, i.e. the agent just performed the last activity of their agenda, the agent will push a walking model to their house, where they will stay until the end of the day.

3.3.5 Issues

There were a few issues with this system. The first being the multipliers, because the resulting behaviour was not directly predictable when changing the multipliers. Intuitively, halving the multiplier should half the number of planned activities. But because of other multipliers, for example from the season, and normalising probabilities later among the activities, this was not the case and iteratively doing trial and error to find the right parameters was the best solution.

Another problem with this system is that it is not suitable to support multi-day activity generation. It will be able to generate activities for a next day, but the activities planned will be inconsistent. For example, when work is planned, a location for a job is found semi-randomly. So when we plan activities for the next day, another location will most likely be selected. This problem will in part be addressed in the next section.

Lastly, an issue that comes from the fact that agenda planner was developed specifically in the knowledge that Rijswijk was the use case. This means that the agenda planner works in a very Dutch or Western fashion. Because of this, it will give unrealistic scenarios if the IDSA system needs to model different parts of the world. The main focus of this thesis is to lower this bias and make the IDSA system more suitable to also model other areas, populations and cultures than the Dutch.

3.4 Improvements

Since the initial development of IDSA, some improvements have been made to the system to make it better scalable for further expansions. These improvements will be necessary for the additions later described. I will discuss two improvements: first, a dynamic spatial data structure and, second, work allocation.

3.4.1 Dynamic Spatial Data Structure

Within the world data model, there already was a static spatial data structure. This static grid made it fast to look up where nearby building were, which was used for visualisation: agent that entered a building would no longer be drawn, until they left that building. This was a grid with 20m by 20m tiles and in every tile the buildings that are (partially) located in that tile are saved. Later we want agents to be able to meet their contacts in the street, even if it was not planned. So if two contacts are within a certain radius of each other, some form of social interaction will occur.

The grid has been extended, such that the old functionality for the static objects is maintained, and the location of the agents can be dynamically updated. Now every tile has information about both the static objects and the agents that are on that tile. Every time an agent moves, it will send an update to the grid and the grid will remove the agent from one tile and place it to the next tile, if that is necessary. Every agent can also send a request to find all agents that are within a certain range. The agent can then use that to look for the contacts within the agents that are close.

3.4.2 Work Allocation

In the IDSA system, workplaces and schools were selected for an agent when they started to plan a work or school activity. This was good for the IDSA system, because it only simulated one day, so continuity about the workplace or school was not necessary. But if the system is extended to support multi-day simulation, workplace consistency have to be considered. Workplace allocation is moved to the population generator because of this, and the workplace/school is added to the properties of the agent. After the agents have their houses allocated, workplace allocation is started. For every household, we give all the adults a workplace and, if there are children, allocate all children to one and the same school. Elderly people have a smaller chance of having work, which decreases as the agent is older than the retirement age.

Schools are selected based on the distance from the house to the school

The chance to select a school is using the following heuristic:

$$h_{a,w} = \frac{1}{2+EuclidDist(house_a, s)}.$$

Where a is an agent and s is a school. These heuristical chances are then normalised among all schools.

Work allocation uses a heuristic based on the the capacity of the workplace and the distance from a worker's house tot the workplace. Every workplace will try to get workers. The chance of a workplace selecting a specific agent is

- 0.5 if the euclidean distance is smaller than 400m;
- 0.2 if the euclidean distance is smaller than 1000m;
- 0.1 otherwise.

Potential workers are picked at random and then the decision is made A workplace will keep on searching for workers until its capacity is reached. Allocation stops when the

After work allocation, all agent have a workplace, a school or are otherwise unemployed/retired and their workplace is therefor null. Because the workplace/school is stored by the agent, it can be used for consistency of workplaces between multiple days.

3.5 Agents and Cultural Values

Agents in the simulation need to have their own set of cultural values. However, we cannot simply assign the national values to the agent as their cultural values, because then we would ignore the fact that a national culture is based on the cultural values of its individuals and the national cultural values are only the averages of these of the individuals.

We will assume that each cultural value is normally distributed, where the mean is equal to the national value. The decision for a normal distribution is that most people will be conform the national cultural values and a minority has values far from the national "standard". Each agent will then get a normal distributed random cultural value assigned, where the standard deviation is set to an arbitrary chosen value of 20.

4 Social Network

National culture is for a great part reflected in social interactions between the members of the society. Dignum et al. [8] explains the importance of social connections and interactions between realistic social agents. A realistic social network is essential for realistic cultural-aware agents and the behaviour that is expected of them.

As stated before, the social network model used in the IDSA system, based on the work of Toivonen et al. [25], does not take into account the semantics of the agents, like similar agents being more likely to know each other, or the world, like neighbours and colleagues knowing each other. Social network models should, as a network, hold the following properties [23]:

- x contacts per agent on average;
- c amount of clustering between the agents.

The exact values for x and c are dependent on what kind of social network we are modelling and which relationships we consider to include in to social network. In most social networks, the clustering is significantly higher that in random networks. Even though Toivonen's model holds nice properties for the resulting social network (it can be specified to reach specific values for x and c), it does this randomly among the agents, without any information other than the agents themselves.

Edmonds [9] showed the importance to link the physical, topological world to the social networks. This link to the physical world, together with agent similarity, can be found in a model Arentze et al. [2]. They use random utility maximisation (RUM) to construct the social network. RUM means that a random component is added to the utility function to account for possible real-world phenomena that are not explicitly added to the model. This will ensure than two agents, even though they are equal in the model, they will behaviour differently. It would be unrealistic to assume that these two agent should behave the same: this would not happen in the real-world. The utilities will take agent similarity and topology into account and threshold values that will be used as criteria to connect agents. These threshold values need to be estimated, because they are not known a priori.

In the rest of this chapter the social network generation process of Arentze et al. is discussed. First the generation itself and its utility functions, then the process of parameter estimation and the evaluation functions used for the estimation process. For this thesis pseudo code is added for clarity, as this was not trivial to make from the original papers.

4.1 Social Network Generation

The model for social networks assumes that the probability of friendship between two persons depends on the evaluation of the utility the two individual expect to gain from such a relationship. This depends on three elements:

1. Homophily;
2. Geographic distance;
3. Presence of common friends, or transitivity.

Homophily, or similarity between agents, will make it more likely that agents that are similar, e.g. having similar interests, become friends. Distance is the link to the physical world. The probability of friendship will decrease when agents life far apart from each other: neighbours and colleagues are more likely to know each other. When geographic distance increases, the probability to be friends should decrease, therefor the utility should decrease. When two persons have common friends they are more likely to know each other, because they could have been introduced or all had some sort of common history together. When common friends are present, the utility should increase. This results in the following utility function:

$$U_{ij} = U_{ij}^H + U_{ij}^D + U_{ij}^C + \epsilon_{ij}$$

Where U_{ij}^H, U_{ij}^D and U_{ij}^C are the utility components for homophily, geographic distance and the presence of common friends, respectively. To account for the non-observable properties that do exist in the real-world, but are not present in the model, the error term ϵ_{ij} is introduced. For now, we will assume that the utility between two agents is symmetric, i.e. $U_{ij} = U_{ji}$. Note that this can be adjusted to model different perceptions on the relationship.

A person can be friends with a limited number of people, because maintaining relationships takes time and effort. To decide who will become friends, there is a threshold on the utility value. When the utility for both people are above the threshold of both people, they are connected:

$$C_{ij} = \begin{cases} 1 & if\ U_{ij} \leq U_i^* \ and\ U_{ji} \leq U_j^* \\ 0 & otherwise \end{cases}$$

Where C_{ij} is the connection between agent i and j and U_i^* is the threshold of agent i. The threshold represents both the "cost" to become friends, as well as the opportunity that two persons have met.

Then, for every combination of two agents the utility is computed and the threshold is applied: when the utility is greater than the threshold the agent are connected. The resulting network is the social network.

4.1.1 Clustering

Because social networks have a higher clustering than random networks, a special mechanism in needed to obtain this clustering. Clustering is defined as:

$$C = \frac{3 * \#triangles}{\#triples}$$

A triangle are three nodes that are all connected to each other. A triple is two edges connected by one node. Clustering can be modelled by using a clustering utility component U_{ij}^C. However, the mutual friends are not yet known completely during the generation of the network, so U_{ij}^C cannot be used during generation of the network. However, clustering can also be modelled in the threshold value of the agent. So a second round is added to the generation algorithm and this time the threshold is defined as:

$$U_i^* = \begin{cases} U_i^* - \theta & if\ C_{ij} = 1 \\ U_i^* & otherwise \end{cases}$$

Where θ is a threshold lowering constant that favours clustering. The first round can also be seen the same as the second round, but with $\theta = 0$. Because of this, the second round only changes utilities for pairs of agents who have mutual friends, so only these cases have to be considered. It is possible to add more rounds, similar to the second round. This will not be done in the simulation discussed here, because a high enough clustering can easily be reached with one additional round.

Below is the pseudo code for the primary and secondary rounds.

function PRIMARYROUND
    for all Agent a **do**
        for all Agent b **do**
            if $a \neq b$ AND $U(a,b) > U_a^0$ AND $U(b,a) > U_b^0$ **then**
                $Connect(a,b)$
            end if
        end for
    end for
end function

function SECONDARYROUND
    for all Agent a **do**
        for all Contact c of a **do**
            for all Contact d of c **do**
                if $a \neq d$ AND $U(a,d) > U_a^0 - \theta$ AND $U(d,a) > U_b^0 - \theta$ **then**
                    $Connect(a,d)$
                end if
            end for
        end for
    end for
end function

4.1.2 Utility Functions

The social utility is composed of two parts: homophily and distance.

$$U_{ij} = U_{ij}^H + U_{ij}^D + \epsilon_{ij}$$

Homophily is defined here as

$$U_{ij}^H = 100 - 10 * (gender_i \neg gender_j) - \frac{|age_i - age_j|}{2} - \Delta cv(i,j)$$

This means that the homophily component has a maximum value of 100. A penalty of 10 is given when i and j have a different gender. Then the utility is lowered by half the age difference, because usng the total age difference appeared to be too aggressive on the utility. Lastly is the difference in cultural values between the two agents.

The distance utility is defined as

$$U_{ij}^D = 100 - 50 * (work_i \neq work_j)$$
$$-min(50, EuclDist(house_i, house_j)/20)$$

Again, the maximum value for the utility is 100. If the agent do not work at the same place, the utility gets a penalty of 50. And a penalty of 1 is given for every 20 meters people live apart from each other, with a maximum penalty of 50.

How realistic the social network will be depends on these utility functions. These utilities are an indication for a social network, yet are not empirically justified.

4.1.3 Computational Demands

With the simulation there are $30,000$ agents, which need to be connected in a social network. The first phase requires $\frac{N(N-1)}{2}$ steps, where N is the number of agents. This is quite a large workload and will increase quadratically as N grows larger. However, to reach x and c it is not necessary to evaluate all pairs of agents: considering only a part of the agents can be sufficient. Arentze et al. introduces a step size, such that for every agent i, i only considers agent j such that:

$$j = i + 1, i + 1 + y, i + 1 + 2y...$$

This will limit the number of evaluations in the first phase to $\frac{N(N-1)}{2y}$. However, this also implies that it is now impossible to consider any other agent than the agents selected by the steps size. This could be too limiting, because a large portion of the population can no longer become connected. This why the step size will not be a set value, but a random integer r such that $1 \leq r \leq 2 * stepsize$ for every step. This way all agent combinations are still possible, yet the expected step size is equal to the set step size. This means that the speed up still holds, because the expected value for the step size is still the previously set step size, therefor resulting in an expected number of evaluations of $\frac{N(N-1)}{2y}$.

The second phase is less demanding, because only a limited set of the agents have to be evaluated. Unless the average number of contacts is very high, the second step should behave linear in complexity, thus making the first phase the bottleneck.

4.2 Parameter estimation

When the social network generation is started, the threshold value and theta are not known, because they cannot be analytically derived. They can be estimated however. Let x^* be the expected average number of contacts of the agents, c^* the expected amount of clustering. We can run the network generation and compute the actual x and c, and update the threshold u^0 and θ accordingly. This way it is possible to iteratively estimate the right values for u^0 and θ. This can be done with the following procedure:

1. Initialise the thresholds as $u^0_1 = u^0_0$ and $\theta_1 = \theta_0$ and $t = 1$;
2. Run the social network generation using u^0_t and θ_t;
3. Compute x_t and c_t;
4. If $x_t < x^* - d_1$ then set $u^0_{t+1} = u^0_t - s_1$
   else if $x_t > x^* + d_1$ then set $u^0_{t+1} = u^0_t + s_1$;
5. If $c_t < c^* - d_2$ then set $\theta_{t+1} = \theta_t - s_2$
   else if $c_t > c^* + d_2$ then set $\theta_{t+1} = \theta_t + s_2$;
6. If $u^0_{t+1} \neq u^0_t$ or $\theta_{t+1} \neq \theta_t$ then set $t := t + 1$ and repeat from 2.

Where d_1 and d_2 are, respectively, how far off the x and c can be from the expected values. s_1 and s_2 are the adjustment parameters, i.e. how much the threshold and θ are adjusted. In the actual implementation s_1 and s_2 are starting off high and every iteration are divided by 1.5 until a minimum value is reached. This is to speed up the search for the right value, by narrowing down the possible values (similar to binary search). So during the parameter estimation, the threshold and θ will first "jump" around, . Because of the influence of the threshold and θ on each other, minimum values for s_1 and s_2 are used, to make sure the values can "walk" to the right values, if the right values were not yet found. This process is similar to simulated annealing.

4.2.1 Evaluation Functions

The algorithms to find the number of triples is quite simple, but the algorithm to find the number of triangles in a network is not trivial. Here are these evaluation functions discussed.

The number of triples in a network is found by looking locally at every node. Because of the definition of a triple (two edges connected by one node) we know that every pair of outgoing edges from a node is a triple. Therefor, the number of triples is defined as:

$$\#triples = \sum_{i=0}^{n} \frac{e_i(e_i-1)}{2}$$

Where e_i is the number of outgoing edges of node i.

The triangles are found by an algorithm described by Schank [20]. Below is the pseudo code to find the triangles in the social network:

```
function Triangles(Agents)
    t ← 0
    for all agent a in agents do
        for all contact c of agent do
            if id_c < id_a then
                for all agent i in the intersection of a's and c's contacts do
                    if id_i < id_c then
                        t ← t + 1
                    end if
                end for
            end if
        end for
    end for
return t
end function
```

For every agent it looks at the triangles the agent starts, by checking for mutual friends among all of its contacts. If such a mutual friend is found, the number of found triangles is increased.

To make sure triangles are not counted multiple times, the algorithm only counts a triangle if the agents appear in decreasing order of their ids. So if three agents, 2, 5 and 11, are in a triangle, then only 11, 5, 2 is considered a valid triangle; 5, 2, 11 is not.

4.2.2 Sampling

Because doing the parameter estimation on the whole population is too demanding, it is possible to use only part of the population and use only these agents to estimate the parameters. So part of the population is sampled randomly and a network only for that part is constructed and evaluated. When the parameters have approximated the right values, the threshold value and θ is used or the final social network for the entire population.

5 Need-Based Activity-Generation

All agents need an agenda for the whole day. The agenda planner used in IDSA could generate agendas for one day, on a statistical basis. Because culture needs to be modelled, a different approach is needed to model culture within activity generation. The statistical approach does not model much of a cognition of the agents and its outcomes are too opaque. To model culture realistically, it is necessary that agents have a decent level of cognition to model decision making of the agent.

Research about activity generation is mostly done in the field of travel prediction and transportation. Although the simulation purpose discussed in this thesis, i.e. simulation for training, is different from transportation, the approaches used for transportation simulations are useful for training purposes as well. It gives a framework for pattern of life behaviour of the agents on a macro-scale. This means that the agents will go through their daily-life, where agents will create patterns, even though no individual agent has any knowledge about these patterns.

The previous agenda planner was based on the work of Kitamura [13] and was based on statistical sampling of activities. The model discussed in this section is developed by Arentze et al [4] [1] [5] [3], together with the work of Nijland[16]. Arentze's model is specifically useful for culture modelling, because of its use of needs and utilities at its core. Using these needs and utilities agent decide which activities to conduct and which are better to postpone to another day. How agents perceive these needs and how these utility functions are defined can be used to model cultural aware agents. These utilities can be seen as a model for the thought process of a person who has to decide what activities to plan the next day.

In the first part the activity generation process is discussed. This includes the needs, the growth functions and the utility functions. Second, the activities themselves are cover: how they are defined and some special cases of activities. Finally, the sequencing process, which will schedule the activities into the agent's agenda. Similar to the section about the social network generation, pseudo code was added for this thesis for clarity.

5.1 Activity Generation

The big assumption in need-based activity generation is that people conduct activities to fulfill certain needs. An agent has a set of needs and a set of activities it can choose from and every activity has an influence on the needs of the agent. An activity generates utility based on the influence it has on the agent's needs. This utility is used by the agent to decide which activity conduct. The utility for an activity a is defined as:

$$U_a^t = U_a^0 - \sum_i \Delta B_{ai}^t$$

This is also called the episode utility (utility gained from one episode of a specific activity) and is noted as U^{ep}. Where U_a^0 is the need independent utility for activity a, ΔB_{ai} is the difference a makes to need i and t is the start time of the activity. U_a^0 can incorporate preferences to conduct an activity on certain days. ΔB_{ai} can be defined as:

$$\Delta B_{ai} = \begin{cases} b_{ai} B_i^t & if\ b_{ai} \leq 0 \\ b_{ai}(B_i^{max} - B_i^t) & if\ b_{ai} > 0 \end{cases}$$

Where b_{ai} is the potential of activity a, B_{ai}^t the need i on time t and B_{ih}^{max} is the maximum value need i can take. Because the needs can work on any arbitrary scale, B_{ih}^{max} can be set to any arbitrary value. In this thesis $B_{ih}^{max} = 100$ is assumed for every need. The potential $-1 \leq b \leq 1$ of an activity on a need i is the influence it has. So if $b > 0$ it will increase the need and $b < 0$ will decrease the need, which is a property of the activity. The potential b is defined as

$$b_{ai} = (\frac{b_a^0}{1+exp(\beta_a[\alpha_a - D_a])})$$

Where D_a is the duration of a and b_a^0, α_a and β_a are activity specific constant. α_a gives an indication about the normal duration of activity a. The function for b_{ai} describes a s-shaped function, which inflection point is defined by α_a and β_a indicates the gradient at the inflection point. b_a^0 is the base potential of the activity and are pre-defined.

5.1.1 Needs

All agents have a set of needs, which grow automatically over time and can be affected by the influence of conducted activities. The growth function G for a need i is defined as:

$$G_i(B_i^t, D) = \frac{B_i^{max}}{1+(\frac{B_i^{max}}{B_i^t} - 1)exp(-\gamma_i D)}$$

Where D is the duration since the last update and γ_i is the growth rate of need i. When an activity is conducted, it will update the needs according to the ΔB_{ai} defined for it. If for a need i there is no change made by the activity, then i will increase according to the growth function. The update function after the activity is conducted then:

$$B_i^{t+D_a} = \begin{cases} B_i^t + \Delta B_{ai} & if\ \Delta B_{ai} \neq 0 \\ G_i(B_i^t, D_a) & if\ \Delta B_{ai} = 0 \end{cases}$$

Nijland found that the needs the motivate people to conduct activities can be reduced to six core needs:

- Rest;
- Social contact;
- Physical exercise;
- Being outdoors;
- Entertainment;
- New experiences.

These six needs are added to our agents. These six needs only apply to individual persons. Agents will also share needs among their household members. In addition to the personal needs, three household activities will be added to this:

- Daily goods;
- Housekeeping;
- Non-daily goods;

Household needs are defined per agent, and not household-wide, because possible difference in perception among the household members. When an activity is conducted by one of the agents, all agent in the household benefit from this. Agents can also decide that another agent should perform the household activity. This means that agents within a household should find consensus on who will do the household activities.

5.1.2 Utilities

Before activities can be planned, a few concepts need to be clear. We will assume that for planning only the biggest influence over all the needs is used. This more realistic, because for people it would be too cognitive demanding to consider all side-effects from an activity. First concept is the utility of time (UoT), which is the utility per time unit of an activity. This is the utility of the activity divided by the duration of the activity plus any travel time:

$$UoT_a = \frac{U_a^t}{D_a + D_a^{trav}}$$

To give a notion of urgency to the activity generation, a pattern utility U^{patt} is used. The pattern utility reflects how long it takes the primary need to reach the the level it had before the activity was conducted. This way the agents will be able to plan ahead.

$$U_a^{patt} = \frac{U^{ep}}{\Delta T}$$

$$\Delta T = -\frac{1}{\gamma_i} \ln\left(\frac{B^{max}}{B^x} - 1\right) \frac{B^x + \Delta B}{B^{max} - (B^x + \Delta B)}$$

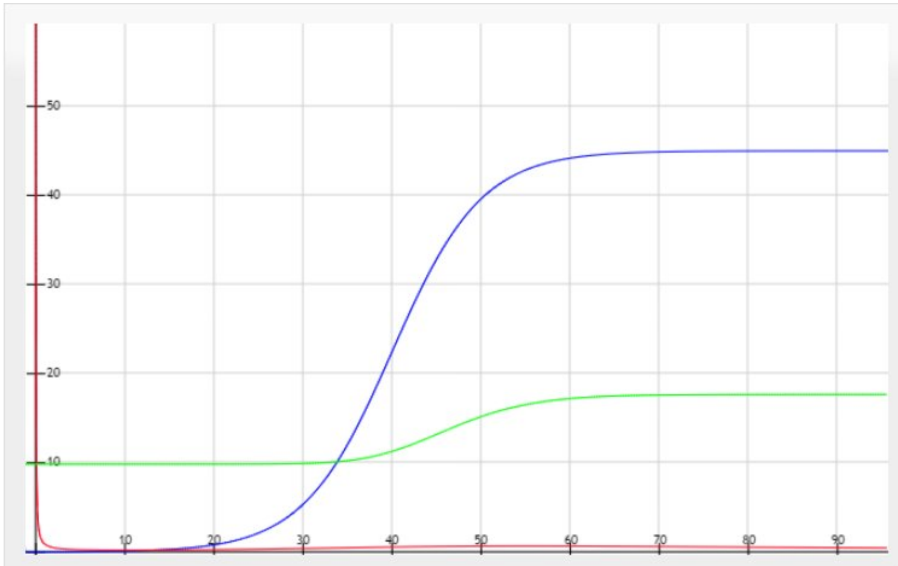

Figure 3: Indication of utilities: U^{ep} : blue, UoT : red, U^{patt} : green

$$\Delta B = b^0 B^x = \frac{b^0 B^x}{1+\exp(\beta(\alpha D))}$$

Give the formulas given above, we can now define when to conduct an activity. A threshold on the UoT, UoT^*, is defined for every agent-day combination. The threshold represents the scarcity of time on a day for an agent. The higher the time pressure, the higher the threshold. When $UoT_a > UoT^*$, the activity can be selected for that day. From the activities that reach the threshold, the one with the highest pattern utility is selected. Then, the episode utility, utility of time and pattern utility is computed again for the other activities and this is done until no activity is left that reached the threshold.

Note that minimising needs is not equivalent to maximising utility. An agent will get more utility when the needs are high. Using the pattern utility will make the agent postpone the activities that do not give a high enough pattern utility, i.e. if it is better to let the need grow for a bit longer.

5.1.3 Optimal Duration

To find the optimal duration, we need to find the duration that maximises the pattern utilty, with the constraint that $UoT > U^*$. In figure 3 we see how the various utilities behave: the blue line is the episode utility, the green line the pattern utility and the red line is the UoT. As seen in figure 3, the UoT will reach a maximum somewhere when the episode utility is still increasing. When the threshold is used, we need to find the highest value the pattern utility can reach as long as UoT is above the threshold. This leads to the following algorithm to find the optimal duration.

function FINDOPTIMALDURATION
$dur, prevDur \leftarrow 1$
$UoT, prevUoT \leftarrow computeUoT(dur)$
$Upatt, prevUpatt \leftarrow computeUpatt(dur)$
$increased \leftarrow false$
$aboveThreshold \leftarrow false$
for ; true; dur++ **do**
$UoT \leftarrow computeUoT(dur)$
$UPatt \leftarrow computeUpatt(dur)$
if $\neg increasedANDprevUoT < UoT$ **then**
$increased \leftarrow true$
end if
if $\neg aboveThresholdANDincreasedANDprevUoT >= UoT$ **then**
if $prevUoT < U^*$ **then**
$break//cannotreachthreshold$
else
$aboveThreshold \leftarrow true$
end if
end if
if $(aboveThresholdANDUpatt < prevUpatt)ORUoT < U^*$ **then return** $prevDur//bestUpattfound$
end if
$prevUoT \leftarrow UoT$
$prevDur \leftarrow dur$
$prevUpatt \leftarrow Upatt$
end for
return dur
end function

In words, the algorithm means this: start the duration at 1. Then increment the duration, until the threshold is reached. Then increment the duration until the pattern utility start decreasing. This means that the maximum pattern utility has been found. If the UoT starts decreasing before the threshold has been reached, the search is stopped.

Some special cases need to checked before. Always decreasing UoT would cause the algorithm to keep on going. This can be prevented by checking whether at $duration = \alpha$ the UoT is increasing. At α is the inflection point of U^{ep}. So if the UoT is not increasing there, it never will.

Another case is when the threshold is always lower than the UoT. For this we use to longest possible duration (the whole day) and check if that UoT (the lowest that is possible on a day) is at least lower than the threshold. If not, than any duration is sufficient (we return α as the duration then).

5.2 Activities

To see all the activities used for this theses, see Appendix A. All activities and their respective properties are all listed there.

One special activity needs special attention, which is slack time. Slack time can be seen as the leisure time someone has when doing nothing particular at home after work. It is an activity that lowers the need for rest and is the default activity: when the time of a day has not been completely used for activities, the rest of the time is automatically slack time.

To simplify the model, all children younger than twelve are not simulated. In the agenda planner in IDSA this was a big problem to tackle, because of the dependencies parental duties create. For this thesis, which focuses on modelling culture, this would only complicate the model. Although an interesting thing to model, parental duties are beyond the scope of this thesis.

5.2.1 Order of Planning

Just like in IDSA's agenda planner. the activities here are planned in a certain order. First all mandatory activities are added to the agenda of the agent. This includes work and school, as well as sleep. Sleep could be made a personal activity with a high need-independent utility, so it should be always planned, but to guarantee all agents would go home and sleep during experiments, sleep was made mandatory. The duration and starting time of these activities are pre-defined. For work and school this is specified in a contract. That contact is created when work is allocated and dictates which days and how long the agent is working and what time they have to start.

Second all household activities are planned, either to the agent themselves or to another household member. Because inconsistencies may arise, an exchange phase is done to reach consensus.

Lastly all social and personal activities are planned. Every time an activity is added to the agenda, the needs of the agent are updated. A schematic example of the activity generation process can be seen in Figure 4.

Planning stops when there are no longer activities that exceed the threshold or when there is no more time left that day.

5.2.2 Household Activities

Household activities are a special case, because one agent can do the activity and the need of all household members is lowered. The household activities need to be divided among the household members and consensus on who will conduct an activity needs to be reached. For household activities a special utility function is used

$$U_a^t = U_a^0 - \sum_i \Delta B_{ai}^t - \omega * \sum_j \Delta B_{aj}^t$$

Where i are personal needs and j are household needs. ω is a parameter than can change an agent's value for the household needs relative to their personal needs. An $\omega < 1$ makes an agent more selfish and an $\omega > 1$ can be seen as more altruism.

To reach consensus about the household activities an exchange phase is used. This is done slightly differently from how it was proposed in the original model.

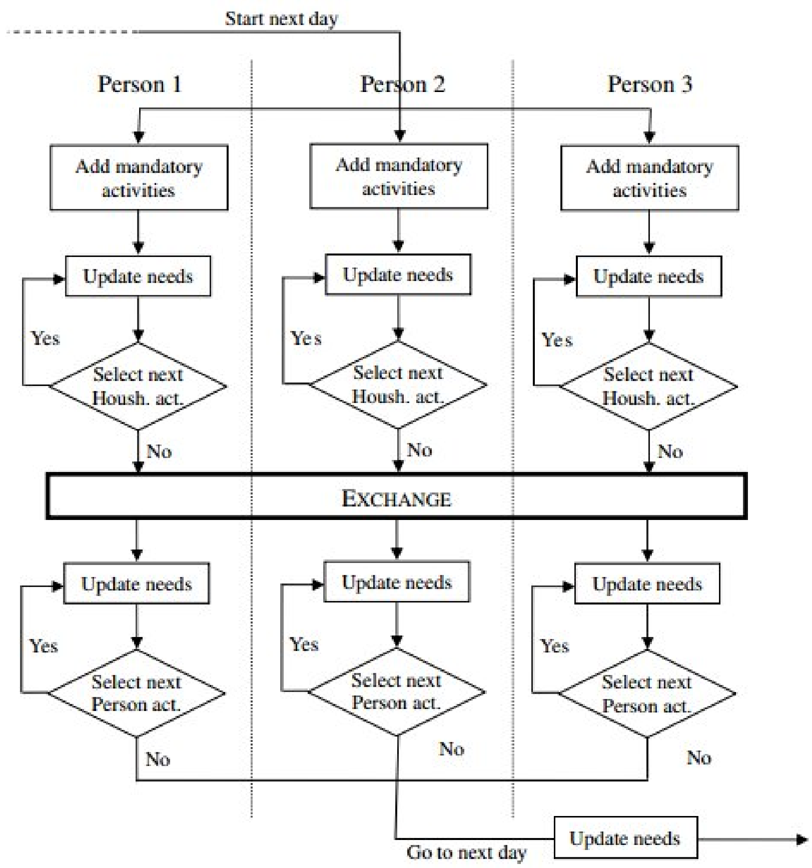

Figure 4: The activity generation process as described by Arentze et al. [4]

Here only the agents who plan a certain household activity are considered to do the household activity. Then the one that has the highest utility is selected to do the activity, if the UoT exceeds their threshold.

5.2.3 Social Activities

Social activities are a special kind of personal activities, because it involves planning it with another agent. Inspiration has been taken from the social activity protocol of Ronald et al. [18]. Key for planning social activities is that both agents need to be able to conduct the activity, but only one agent can initiate the planning of the social activity: the agent with the lower id is allowed on initiate it. This will have no influence on availability, because both agents have to able to plan the activity, so it should not matter which agent initiates.

If both agents can conduct the activity, the social activity is planned.

5.2.4 Dynamic Activities

Activities made at the start of the day are static for that day: these activities will not change when during the simulation itself. To model dynamic changes of the agenda of the agent, dynamic activities are introduced. This happens when a person changes their original agenda and will do a different activity, if this activity is more appealing than the original agenda.

During the simulation, the contacts can meet each other on the street, even if they did not plan this, but just run into each other. In these situations, they can decide to override their planned activities and plan a social activity at that moment. Mandatory activities cannot be overwritten. Other activities can be overwritten if the UoT of the dynamic activity exceeds the UoT of the planned activity. If this is the case for both agents, the dynamic activity can be placed in the agendas of both agents. There may be a utility bonus or penalty for every agent for dynamic activities, because of social pressure or the urge to maintain the planned agenda.

A taboo list is used to make sure people do not keep on conducting dynamic activities. When an agent meets a contact, the contact is added to this taboo list, to ensure that the two agents will not keep on trying to plan an activity. If this would happen, they would keep on trying to plan the activity until the random component is high enough that the UoT will exceed the threshold. This taboo list is emptied at the end of every day.

This procedure could be extended to other forms of dynamic activities beside social interactions. A good example of this could be window shopping. It is easy to imagine someone not going home for leisure, but goes window shopping until it is time for the next activity. To not complicate the system when adding these dynamic activities, we will leave other forms of dynamic activities out in this thesis.

5.3 Threshold Estimation

The threshold for all agent-day pairs are not known a priori, because of the influence the thresholds have on each other between days. Like in the social network generator, the thresholds can be estimated by running the algorithm and adjust the thresholds iteratively.

The same procedure as described by Arentze is used. For some period of time (normally a couple of weeks) agendas are generated and based on the difference between the UoT of activities (UoA) and the UoT of slack time (UoS). The idea here is that, if more utility could be gained by doing nothing (slack activity) than doing activities, the threshold should be raised, so only activities with higher UoT will pass. If slack time has much lower UoT, the threshold is lowered, so the activities will also get lower UoT. This is because most of the time, UoT of an activity will be just above threshold.

The following procedure is used per household to find the optimal thresholds.

1. Initialise U^* for each agent-day combination;
2. Generate agendas for each agent given the the current U^*;
3. Calculate UoS and UoA for each day of the week;
4. If $UoS < UoA$, decrease U^*, if $UoS > UoA$, increase U^*. The difference must be greater than a certain value;
5. If no changes have been made: stop
6. Go to step 2.

If thresholds can only be adjusted one way, the algorithm should terminate. So if a threshold increased first, it can only increase, and if it decreased first, it can only decrease. Because of mandatory activities on certain day, the scarcity of time on those days increases, therefor the threshold increases. This might affect days afterward, so these thresholds may be lowered, so the agent can still do the activities they need to do to fulfil all needs.

An interesting property of this mechanic is that normally all activities have roughly the same UoT on a day, as will the UoT of slack time.

5.4 Sequencing

When all activities have been selected, they have not yet been ordered and inserted into the agenda of the agent. This is done be the sequencing phase. For this we use a greedy heuristic.

When all activities have been planned, they already are in order of priority: first the mandatory activities, the household activies, social activities and finally personal activities. They have already been added in order of pattern utility (that is what is maximised).

So first a summed timeframe is made. It uses the timeframes from IDSA. Every activy has a certain timeframe in which it can be planned. The summed timeframe adds them together, so it is clear which times of the day are more demanded: these have higher values in the summed timeframe.

When looking for a time in the agenda, it tries to minimise the overlap with other activties. So if at a first moment on the day three activities could planned and on a second moment it is the only one that uses that time as available time, it will pick the latter. Finding the best time is done by sampling a set number of times possible for the activity (given its timeframe, duration and walking time) and select the best one. Then constraint satisfaction is done to make sure other activities cannot pick that time anymore.

Sequencing is done until no more activities can be added to the agent's agenda, either because all activities have been planned or no activities can be planned anymore.

Because slack time is never planning in the activity planning phase, it is added at the end of sequencing. Slack time is added to the agenda when there are empty places in the agenda that are not necessary for walking to another activity and the agent has enough time to get home and have a decent amount of time for leisure (here set at 15 minutes). This process is called padding, because the agenda is "padded" with slack time.

5.4.1 Social sequencing

Social activities are a bit more complicated to sequence, because there are more agents involved. This is solved by using a similar greedy approach. An agent will first check if the other agent has already sequenced the social activity. If not, the agent will sequence the activity like any other activity. If the other agent already did sequence, the agent will see of they can make that time. If the agent can, it will place the social activity in their agenda. If not, the agent will remove the social activity in the other agents agenda.

6 Modelling Cultural Agents

The modelling of culture will be limited to two of Hostede's dimensions: individualism versus collectivism (IDV) and uncertainty avoidance (UAI). Within the scope it is not possible to model all dimensions, so IDV and UAI were selected, because they have the greatest impact on activity generation. By applying changes to the social network and the activity generation, both can reflect the cultural values of the individual agents. Between two cultures the individual choices might not seem very different, but the overall pattern will show that one culture behaves differently to another on a macro-scale.

Simply changing the cultural values of the agents does not automatically imply the agents are cultural aware or that their behaviour is correctly adjusted. It is not trivial how the utility adjustments are quantified, in order to make the behaviour of the agents follow the expected behaviour, based on Hofstede's model.

Note that Hofstede's dimensions only apply to the values of people, not the norms in their society. The values are what underlies the decision making on the cultural level. The norms that a culture follows may be explained by the cultural values, but cannot be prescribed by them. This will place limits on the modelling capabilities of culture in the multi-agent system discussed in this thesis.

6.1 Social Network

Culture is shown in great part in how people interact with each other and how they are connected. This is why the social network is influenced by culture.

6.1.1 Individualism versus Collectivism

In collectivist cultures is a stronger difference between in-group and out-group. We assume here that this stronger difference can be modelled as more tight groups among the people. These tight groups are the clusters in the social network. To model this difference between these two ends of this cultural dimension, a multiplier on the θ is used, to influence clustering.

Adjusting the θ for the agents will give different clustering among the agents, but the clustering that has been set in the social network generation process, c^*, will not change. This means that no change will emerge between cultures. Therefor, c^* needs to be adjusted with the culture as well. For this, a collectivist bonus (or individualist penalty) can be used to also adjust the c^*. this adjustment should be based on the national cultural values, because the change in c^* applies to the entire population as a whole.

The exact implementation of theta for agent a is

$$\theta_a = \theta^0 - \frac{(IDV_a - 50)}{50} = \theta = \theta^0 - \frac{(IDV_a)}{50} - 1$$

Where θ^0 is the base θ used in the social network generator and IDV_a is the IDV of agent a. This would make 50 the default value and the influence is changes as IDV increases or decreases. The 50 in the denominator is determines the impact of IDV. An impact of 100 would make the impact on θ too aggressive. Both 50's in the quotient could be replaced by other values, if this would give a more realistic results, that could be empirically found in the real world.

6.1.2 Uncertainty Avoidance

When UAI is high, people tend to avoid risk and ambiguous situations. This also applies to their social life. People with similar ideas are more likely to know each other than people who are very different. This is because people with different ideas are kept at a distance. This means that the higher the UAI, the more people want to avoid others who are different to them.

In a culture with high UAI, homophily is high: homophily is the similarity between agents. So when the utility between agents is computed, the effect of homophily compared to the distance utility will become greater. When UAI decreases, this utility bonus can be relaxed.

So if people have a very low UAI, the balance between U^H and U^D are set to the normal value, which means they both have a weight of $\frac{1}{2}$. The relative weight of U^H should increase when UAI increases. In the implementation, the social utility is computed as follows:

$$b = \frac{1}{2} + \frac{1}{4}\frac{UAI_a}{100}$$

$$U = bU^H + (1-b)U^D + \epsilon$$

Where b is the balance between U^H and U^D. This will scale b roughly between $\frac{1}{2}$ and $\frac{3}{4}$. This way the relative importance of homophily will increase when the UAI increases. It would be unrealistic if someone would only consider homophily to determine who they befriend. Also note that the utility represent the possibility that two agent met, which has to include distance.

6.2 Activity Generation

Activity generation is the main source of behaviour in the simulation: although the behaviour models could be implemented in such a way that more detailed behaviour is performed by the agents, but this is beyond the scope of this thesis. Because most of the modelled cognition of the agents resides in the activity generation process and its utility functions, this is the place to insert culture.

6.2.1 Individualism versus Collectivism

In collectivist cultures people see themselves as part of a group and identify themselves less in terms of their individual. In the simulation the group is assumed to be the family or household of an agent. This means that, based on the examples given by Hofstede, the needs of the household becomes more important than the personal needs in a more collectivist culture. A direct link can be made between IDV and ω in the activity generation process. Recall that

ω is the relative importance between household needs and personal needs. When $\omega < 1$ agents become more selfish and $\omega > 1$ corresponds to more altruism.

An average value for IDV, i.e. 50, should result in an $\omega = 1$: personal and household needs are equally important. When IDV increase, ω should decrease, and if IDV decreases, ω should increase. The following function is used to map IDV to ω:

$$\omega_a = max(0, \frac{100 - IDV_a}{50})$$

This scales ω linearly between 0 and 2, given that $0 < IDV < 100$. Because cultural values are randomly selected from a normal distribution with mean the national value, an agent specific cultural value can be above 100 or smaller than 0. Because it is not realistic that an agent gains utility for increasing household needs, ω has a minimum of 0: otherwise an agent would not be selfish anymore, but sadistic for gaining utility for making the life of their household members worse. When $\omega > 2$ such a problem will not occur: then they just really like doing household activities.

6.2.2 Uncertainty Avoidance

We will make the following assumption for UAI in activity generation: the need for new experiences is lower in a culture with high UAI: people do not really seek new experiences and like to stay in their comfort zone. This can be implemented in two ways: increasing how activities that fulfil the need for new experiences gets decreased when such an activity is conducted, or by lowering the growth speed of new experiences. This should have similar results. The latter is chosen here, because it seems more logical: activities do not change based on someones perception, but how great their need is for such an activity is strongly affected by their perception. Because of this, only the the growth speed $\gamma_{newExperience}$ is altered by UAI.

$$\gamma_{NE} = \gamma^0_{NE} * \frac{100 - UAI_a}{100}$$

Where γ^0_{NE} is the growth speed for new experiences.

Another place where UAI has an influence is dynamic activities. When UAI is high, people do not want to change their plans and change their agenda. This can be represented by a need-independent utility, which implemented as follows:

$$U^0_{dynamic} = x - UAI/x$$

Where x can be scaled to an appropriate value, which is dependent on the properties of the dynamic activity. X was not specifically determined for this thesis, because of technical difficulties, discussed later in this thesis.

7 Implementation and Experiment

All components discussed in this thesis were implemented in Java as an expansion on the IDSA system. In this section two subjects are discussed. First the implementation of the culture and some issues that come with it. In the second part an experiment is set up to determine the effect of the culture modelling.

7.1 Implementation of Culture

There is a problem when using utilities and culture: it is very likely that cultural influences are all over the programming code. If anything needs to be changed, functions in various classes need to be updated. For the implementation for this thesis this was partially circumvented by placing all methods that use cultural values in one class. All these methods were declared static and public so that it was always accessible.

This way all cultural function are still cluttered all over the code, but if changes need to be made, it is much easier find the right function. If only one cultural dimension need change, an easy search in one file can be done. This makes maintaining a project using cultural utility functions a lot more clear.

7.2 Experiment

To determine the effect of the different cultures on the behaviour on the population as a whole, the simulation is ran twice with different national cultural values: once with Dutch and once with Portuguese national cultural values. These two cultures differ mostly on IDV and UAI [11].

During the simulation, the overall need of the agents and the number of planned activities for each sort of activity will be plotted. In order to show that the current system works, the results have to be in line with what Hofstede's model of culture describes to be different between these two cultures. The expected results are described in the previous section, for both the social network and the activity generation.

Throughout the simulation, every 10 simulated minutes the system writes all needs, summed over the agents. At the end of every day the total amount of activities planned that day is also written away, with a special field for the number of planned dynamic activities.

During the experiment, all other variables will remain the same. Although there might be differences in tokens of activities that people can plan between the two cultures, these are left out of the experiment, to show how the model works. In a realistic setting, different activities are considered in different cultures. These different norms are also not part of Hofstede's model of culture.

8 Results and Discussion

Unfortunately, both the social network and the activity generation were too unstable to properly conduct the experiments.

For the social network, there was too much variance between the final values obtained in parameter estimation and the final result. A lot of times the difference between parameter estimation and the final result were more than the allowed margins set for the parameter estimation. This causes so much noise that the effects from the cultural values cannot be determined: the noise is greater than the effects of the cultural model.

The greatest problem here is the step size that is used in sampling compared to the step size that is used for the whole population. When just dividing the step size by the sample size, the results of the final network is far from the expected values. The relationship between the step size used during sampling and the step size when generating the final network is yet to be found.

A similar problem occurred in activity generation. The number of planned activities was varying greatly. Adding the cultural model to this gave no significant change, because, again, the noise was bigger than the expected result. Activity generation itself works: it is just not consistent over multiple days and runs.

The inconsistent results can come from the models for the social network and activity generation themselves, or from the errors in the implementation of the models. The results are not conclusive. It is still possible that the cultural modelling proposed in this thesis works, but given the state of the system at the moment, it cannot be said for certain.

It is possible that there are mistakes in the implementation, causing the results to be varying so much. To test if this approach could work, the output of both the social network and activity generation needs to become more stable and consistent. More detailed utility functions and a better understanding of their impacts could help to reduce the variance. If it can be shown that certain quantifications of the utility functions give more consistent results, it might be possible to show the whether or not the cultural utility adjustment follow result in behaviour that follows Hofstede's model.

9 Conclusion

In this thesis a model for cultural agents has been proposed. This is done within a RUM social network generation process and RUM need-based activity generation. The basis for the cultural model is based on Hofstede's dimensional model of culture, because of its scientific background and quantifiablity. A special kind of activity has been introduced: dynamic activities, which happen when agent meet each other on the street. Individualism versus collectivism and uncertainty avoidance were modelled in social network and the activity generation, by changing the utility functions based on an agent's cultural values.

The model was implemented within the IDSA system. An experiment was set up to show how two different cultures, The Netherlands and Portugal, would give differences in the social network and planning behaviour of the agents, that could be explained by Hofstede's model. However, the results were not consistent enough to draw conclusions from.

For future research, a thorough analysis of the implementation is needed, followed by debugging. It is also possible that utility functions are not suited to model culture, because culture affects various parts of the decision-making process. Culture affect which activities can be planned, how they are planned and the execution of the activities. All these steps are combined in one function, which might not be expressive enough. Another way is the use of BDI models, as used by Mascarenhas et al. [15].

In case this model does give the expected results, a empirical foundation for the quantification of the utility functions used in the model could give more insight into cultural modelling.

The IDSA system and its event planning can now be extended to use the agents utilities. Sampling can be done more selectively, e.g. when a procession is sampled, a threshold on utility can be used to select the agents that are more likely to interrupt their activities. This could in turn be influenced by uncertainty avoidance for example. This way interesting and complex dependencies could emerge.

10 Acknowledgments

I would like to thank the Modelling, Simulation and Gaming group at TNO The Hague for letting me do my thesis within their group. Without their support it would not have been possible to do my bachelor thesis on such a complicated subject. Especially I would like to thank Steven de Jong and Ruben Smelik, who I could always ask questions regarding my thesis, helped me in my thought process and showed me that seemingly difficult problems can be approached. Thanks to Freek van Wermeskerken, for helping me with the more mathematical problems, when my own understanding was not enough.

I would like to thank Frank Dignum, for the freedom he gave me regarding this thesis and for help on the scientific parts. Special thanks go to Pieter van Nispen for getting me started with the culture studies and to Theo Arentze for his time to explain the need-based activity generation process to me.

Last, but definitely not least, I would like to thank my family and friends for understanding and support throughout, during my entire degree, and especially my thesis. Thank you.

11 References

[1] T. Arentze, D. Ettema, and H. Timmermans. Estimating a model of dynamic activity generation based on one-day observations: method and results. *Transportation Research*, Part B: Methodological(45(2)):447–460, 2011.

[2] T. Arentze, M. Kowald, and K. Axhausen. An agent-based random-utility-maximization model to generate social networks with transitivity in geographic space. *Social Networks*, 35(3):451 – 459, January 2013.

[3] T. Arentze and H. Timmermans. A learning-based transportation orientated simulation system. *Transportation Research*, Part B(38):613–633, 2004.

[4] T. Arentze and H. Timmermans. A need-based model of multi-day, multi-person activity generation. *Transportation Research*, Part B: Methodological(43(2)):251–265, 2009.

[5] T. Arentze and H. Timmermans. A dynamic model of time-budget and activity generation: development and empirical derivation. *Transportation Research*, Part C: Emerging Technologies(19(2)):242–253, 2011.

[6] R. L. Axtell. Effects of interaction topology and activation regime in several multi-agent systems, July 2000. Working paper no. 12.

[7] S. de Jong, A. Klein, R. Smelik, and F. van Wermeskerken. Integrating run-time incidents in a large-scale simulated urban environment. 2015.

[8] F. Dignum, G. J. Hofstede, and R. Prada. From authistic to social agents. *AAMAS 2014*, January 2014.

[9] B. Edmonds. How are physical and social spaces related? cognitive agents as the necessary glue. Technical report, Centre for Policy Modelling, 2003.

[10] L. Hamill and N. Gilbert. Social circles: A simple structure for agent-based social network models. *Journal of Artificial Societies and Social Simulation*, 12(2):3, 2009.

[11] G. Hofstede, G. jan Hostede, and M. Minkov. *Cultures and Organizations: Software of the Mind*. McGraw Hill Education, 3rd edition edition, 2010.

[12] G. J. Hofstede, C. M. Jonker, and T. Verwaart. Cultural differentiation of negotiating agents. *Springer*, February 2010.

[13] R. Kitamura, C. Chen, R. M. Pendyala, and R. Narayanan. Micro-simulation of daily activity-travel patterns for travel demand forecasting. *Transportation*, 27:25–51, 2000.

[14] A. Klein. Internship report tno population modelling: Project intent driven scenario authoring, November 2015. Utrecht University and TNO Modelling, Simulation and Gaming.

[15] S. Mascarenhas, N. Degens, A. Paiva, R. Prada, G. J. Hofstede, A. Beulens, and R. Aylett. Modeling culture in intelligent virtual agents: From theory to implementation. *AAMAS 2015*, October 2015.

[16] L. Nijland. *A need-based approach to dynamic activity generation*. PhD thesis, Eindhoven University of Technology, January 2011.

[17] A. S. Rao and M. P. Georgeff. Bdi agents: From theory to practice. *Technical Note 56*, April 1995.

[18] N. Ronald, V. Dignum, C. Jonkers, T. Arentze, and H. Timmermans. On the engineering of agent-based simulations of social activities with social networks. *Information and Software Technologies*, 53:625 – 638, January 2012.

[19] S. Russell and P. Norvig. Artificial intelligence: A modern approach. *Third Edition*.

[20] T. Schank and D. Wagner. Finding, counting and listing all triangles in large graphs, an experimental study. University of Kalrsruhe, Germany.

[21] J. Schram. Culturally affected behaviour in belief-desire-intention agents. diploma thesis, Tilburg University, 2011.

[22] B. G. Silverman, D. Pietrocola, N. Weyer, R. Weaver, N. Esomar, R. Might, and D. Chandrasekaran. Nonkin village: An embedded training game generator for learning cultural terrain and sustainable counter-insurgent operations. *Agents for Games and Simulations*, pages 135 – 154, 2009.

[23] T. A. Snijders. The statistical evaluation of social network dynamics. *Sociological Methodology*, January 2001.

[24] C. Solomon and M. S. Schell. *Managing Across Cultures: The Seven Keys to Doing Business with a Global Mindset*. McGraw Hill Education, May 2009.

[25] R. Toivonen, J.-P. Onnela, J. Saramki, J. Hyvnen, and K. Kaski. A model for social networks. *Physica*, (371):851 – 860, 2006. Helsinki University of Technology.

[26] P. van Nispen. Culture encyclopedia.

A Activities

	Location	Monday	Tuesday	Wednesday	Thrusday	Friday	Saturday	Sunday
Sleep	Home	22:00- 8:00	22:00- 8:00	22:00- 8:00	22:00- 8:00	22:00- 8:00	22:00- 8:00	22:00- 8:00
slack time/Leisure time	Home	5:00-5:00	5:00-5:00	5:00-5:00	5:00-5:00	5:00-5:00	5:00-5:00	5:00-5:00
Sport	Sport	7:00-23:00	7:00-23:00	7:00-23:00	7:00-23:00	7:00-23:00	7:00-23:00	7:00-23:00
Walk in the park	Park	7:00-23:00	7:00-23:00	7:00-23:00	7:00-23:00	7:00-23:00	7:00-23:00	7:00-23:00
Museum	Museum		9:00-17:00	9:00-17:00	9:00-17:00	9:00-17:00	9:00-17:00	9:00-17:00
theatre/cinema	Theatre	19:00-23:00	19:00-23:00	19:00-23:00	19:00-23:00	19:00-23:00	19:00-23:00	19:00-23:00
Visiting café/bar	Café	19:00-1:00	19:00-1:00	19:00-1:00	19:00-1:00	19:00-3:00	19:00-3:00	19:00-1:00
Going out for dinner	Restaurant	17:00-21:00	17:00-21:00	17:00-21:00	17:00-21:00	17:00-21:00	17:00-21:00	17:00-21:00
Visiting friends	Contact's Home	8:00-00:00	8:00-00:00	8:00-00:00	8:00-00:00	8:00-00:00	8:00-00:00	8:00-00:00
Receiving visitiors	Home	8:00-00:00	8:00-00:00	8:00-00:00	8:00-00:00	8:00-00:00	8:00-00:00	8:00-00:00
Work/School	Work	Contract	Contract	Contract	Contract	Contract	Contract	Contract
Daily-Shopping	Shop	8:00-20:00	8:00-20:00	8:00-20:00	8:00-20:00	8:00-20:00	8:00-18:00	
Housekeeping	Home	5:00-5:00	5:00-5:00	5:00-5:00	5:00-5:00	5:00-5:00	5:00-5:00	5:00-5:00
Non-daily shopping	Shop	8:00-20:00	8:00-20:00	8:00-20:00	8:00-20:00	8:00-20:00	8:00-18:00	

Figure 5:

	α	β	primary influence	Daily-goods	House-keeping	Non-daily goods	Rest	Social Contact	Physical Execise	Being Outdoors	Enter-tainment	New Ex-periences
Sleep	440	_	Rest				-1					
slack time/Leisure time	200	0.01	Rest				-1					
Sport	55	0.08	Physical Execise						-1			
Walk in the park	83	0.053	Being outdoors						-0,3	-1		
Museum	150	0.053	New Experiences				0,1					-1
theatre/cinema	150	0.053	Entertainment				0,2				-1	
Visiting café/bar	180	0.053	Social Contact					-1			0,5	
Going out for dinner	70	0.053	New Experiences				0,2	0,3				-1
Visiting friends	75	0.044	Social Contact					-1				
Receiving visitiors	75	0.044	Social Contact					-1				
Work/School	510 or 240	_	(-)Rest				0.8					
Daily-Shopping	40	0.1	Daily-goods	-1			0.2					
Housekeeping	180	0.011	Housekeeping		-1		0.4					
Non-daily shopping	105	0.027	Non-daily goods			-1						
				0 otherwise								

Figure 6: