Progress Report

Matt Smith

April 18, 2013

Abstract

This project provides a proof-of-concept agent-based model for the problem of how social networking impacts on the behaviour of smoking cessation. By attempting to build a basic model of human smoking behaviours and defining interactions, this model allows for simulations using, in theory, any number of autonomous entities. Through simulations, it appears that the quantity and social location of humans in networks significantly impacts their effect within the graph and, by extension, the quantity of smokers present. Generally, the model shows promise as a proof-of-concept for further development both in the areas of quality of implementation and production of results.

Contents

1 Introduction **2**

1.1 Project Overview . . . 2

1.2 Project Rationale . . . 2

2 Research & Literature Review **4**

2.1 Overview . . . 4

2.2 Social Networking . . . 4

2.3 Smoking Cessation & Health . . . 6

2.4 Agent-Based Modelling . . . 7

2.5 Similar Work . . . 8

2.6 Summary . . . 8

3 Model Development & Implementation **9**

3.1 Overview . . . 9

3.2 Technologies & Tools . . . 10

3.2.1 Repast Simphony . . . 10

3.2.2 JUNG . . . 10

3.2.3 Java . . . 10

3.2.4 Gephi . . . 11

3.2.5 Github/Git . . . 11

3.3 Model Description . . . 11

3.3.1 Platform . . . 11

3.3.2 Social Network . . . 13

3.3.3 Agent . . . 16

4 Simulation Results & Model Analysis **23**

4.1 Overview . . . 23

4.2 Simulation Analysis . . . 23

4.2.1 Simulation Parameters & Agent Attributes . . . 23

4.2.2 Decision Tree Analysis . . . 23

4.2.3 Sampled Networks . . . 23

4.3 Model Analysis . . . 23

4.3.1 Commercial Analysis . . . 23

4.3.2 Model Analysis . . . 23

4.3.3 Further Improvements . . . 23

5 Conclusion **24**

Chapter 1

Introduction

1.1 Project Overview

A great deal of current research in the field of Computer Science focus on tackling real-world problems through a combination of separate theoretical areas, using abstract techniques to represent, merge and ultimate gain new information about these problems. Generally speaking, this project takes the fields of smoking cessation and social networking and, by mapping them to an agent-based model, tries to provide information about how social situations and conditions change the behaviour of someone in giving up smoking. On top of this, it aims to assess whether this would be useful as a starting point for commercial level approach to investigating smoking cessation by considering the implementation in terms of scalability, efficiency and more.

For each sof the constituent fields used within this project, there is the potential for an entire project in each. As such, the scope of this work was limited to an attempt to create a basic model of smoking related behaviours and social interactions. By doing this, it could act as a proof of concept for further work in emulating human behaviour or social networking, should the model indicate that it would be useful. At this point, it should be noted that social network is referred to in the context of all social interaction as opposed to just that of online networking.

Overall, the project is split into three sections, each building upon the previous. To provide an understanding of any existing useful work as well as where the gaps in the existing research lie, the relevant fields were investigated. This was followed up by the development of the model itself, with a focus on extensibility and maintainability, which involved mapping the problem domains of networking and human behaviour onto abstract representations. The final part of the project was to analyse this model by both considering its features and running sample simulations, the latter aiming to provide some new information about the interplay of smoking cessation and social networks.

1.2 Project Rationale

As described above, the project is focused on smoking cessation and in particular, what factors contribute to individuals to giving up. Smoking is a big problem for healthcare in many countries partly due to the expense managing it in 2005/06, the National Health Service (NHS) spent around 5.2 billion on costs directly related to smoking [REF PG 78 NHS paper]. Adding to this, many smokers are cited as wanting to give up though failing in their attempts [REF]. Understanding the social conditions through which smoking cessation is made more successful would be very useful from the point of view of the NHS and the individual.

Furthermore, due to the advent of services such as Facebook, Twitter and other online social networks, the concept of networking has become significantly more popular in recent years, although it is an ever-present factor in life [Ref USN pg 4]. Whilst studied for a long time in areas such as sociology, computer representations of these are relatively new. On top of this, there is a lot of interest in the impact of these networks on human behaviours; with reference to smoking cessation, an area of particular interest are that of health being affect due to those in close social proximity. Being able to investigate these effects and understand what changes their severity is a key factor in applying them to giving up smoking.

One approach to this is to perform a wide range of surveys, psychological and sociological studies to try to find out these conditions which could then be replicated in real social circles. Not only would this be very expensive, but the information gained in this kind of study depends on a number of circumstantial factors of the participants and as such, it may not be relevant everywhere, causing further expense and time in re-running this work. Using a computer model presents an alternative that, if developed accurately, can be adapted and tuned to simulate different situations. Ultimately, this project aims to provide an adaptable and configurable model of the effects of social networking upon smoking cessation such that new information can be gained about the problem and, with further development into human behaviours and social interaction, extending the model could provide more insight into how different situations can be manipulated to aid in giving up smoking.

Chapter 2

Research & Literature Review

2.1 Overview

Due to the variety of fields interacting within the project, understanding the relevant background to each of these areas is the key to creating a representative model. Specifically, investigating how social networks and personal health interact provide a good starting point in creating an agent-based model. This section will consider relevant and similar works to both justify and allow the project to expand upon the existing knowledge base.

2.2 Social Networking

As described previously, social networks, whilst always existing, have become a popular topic in recent years. In particular, applying an analytical approach to them has led to a formalisation. At the lowest level, networks are represented using mathematical graph theory [ref NetMark]. This means that individual entities within the network are shown as nodes, whilst relations between these entities are edges. Importantly, these edges can be either directed or undirected, which can change their meaning depending on context for example, directed edges could be used to show how one node likes another [ref USN 14]. By extension, a bidirectional edge represents some mutual relationship between the nodes. Building on this basic set of terms, a number of structural definitions emerge that begin to underpin the theoretical to the practical. A key one is that of triadic closure, which is where two nodes that share a common connection are, at some point in the future, more likely to be connected [ref NetMark pg 44]. This is particularly important in social networks when it comes to working out which pairs of people are going to form connections next. Extending this, the idea of strong & weak ties brings an extra aspect to the edges between nodes. A strong tie may be likened to a friendship whereas a weak tie to an acquaintance. Generally, the stronger ties are present in small, connected clusters whereas weaker ties link these clusters together [ref NetMark 46-48]. Again relating this to real-world examples, this is similar to groups of close friends being connected to others by acquaintances in other groups. Quantifying graph is done through a number of measures. One of these is betweenness, which is calculated by calculating a 'flow' through the graph edges of high flow are important since they carry the most traffic and as such have a high betweenness value. This can indicate the strength of a tie; for example, a weak tie is likely to have a high rate of flow since it is between two highly connected groups and is of high importance in joining these clusters [ref NetMark 66-67].

With these terms in mind, further concepts that more directly relate to social networks and human behaviours can be introduced. A key example of this is homophily, which is when groups of friends are similar. This similarity may be manifest itself in beliefs, interests, jobs or more [USN?]. Although

the term provides an overview, there are a number of mechanisms that underpin homophily. When fixed characteristics such as ethnicity are considered, selection is in play, which is when people choose those who they share the most in common with. In contrast to this, characteristics which are changeable such as interests or behaviours show how socialisation and social influence affect the person. The former is the process of individuals striving to bring them closer to others who have similar characteristics, whereas the latter is when existing connections to others cause changes to the behaviours or interests which is effectively opposite to selection[ref NetMark 81-2]. It should be noted that selection and social influence have an amount of interaction that can result in it being difficult to determine which aspect of homophily has contributed towards a connection.
Expanding upon influence within a graph representation of a social network, there are a number of approaches to emulating real-world influence between network people. Although there are many specialist models that attempt to recreate this, two basic approaches are:

- Linear Threshold, which is defined as *INSERT EQUATION HERE*. This is a basic representation of influence that can effectively be summarised as a node taking on behaviours that its neighbouring nodes also exhibit, depending on some predefined boundary. An example of this in a real social network could be that if more than half of someone's friends play football, they will also begin to play football.

- Independent Cascade, defined as a series of time-steps during which any 'active' nodes attempt to activate any 'inactive' neighbours with a certain probability. Should a node become 'active', it then tries to activate its neighbours, and so on. Once nodes have attempted neighbour activation, they cannot reattempt and as such, the process ends when no more activations are available. Relating this to human behaviour, it may be likened to someone trying to convince friends about an idea; they will not carry on attempting to convince if they fail, but if someone does adopt the idea, they may further spread it themselves.

Whilst these influence models are basic, they are useful when it comes to adapting them for a social network. For the purposes of this project, they are not directly applicable since different aspects of smoking and smoking behaviours may interact when it comes to influence spread, but they will serve as a basis.
Reference to influence paper needs to be here somewhere include on both
A final, useful aspect of social network research is that of generating or classifying the type of a network. The most basic type of network is a randomised one, such as that generated by the ErdsRnyi model [ref ERDOS], where edges between any given pair of nodes have an equal probability of existing. This, naturally, leads to most nodes having similar degree which, when compared to real-life networks, lack the hubs nodes which have a higher degree than the network average [ref ScienAm]. As such, random networks appear to be too far removed from what one might observe in nature and two other methods emerge as useful to the project small-world and scale-free networks.
Small-world networks are based on the concept of the small-world phenomenon, which is where human society exhibits a structure where the number of social connections between two people is, on average, quite low which indicates a high level of connectivity [ref Milgram]. More formally, small-worlds generally have high clustering and a low average path length, so aim to represent the effect observed by Milgram in a mathematical way the Watts-Strogatz method provides an approach to generate these networks [ref SW paper]. At a high level, given a starting set of nodes where each is connected to its neighbours, the algorithm considers rewiring edges based on a predefined probability. This allows for the formation of more realistic structures such as hubs within the network and could be used to investigate how small social groups are affected by smoking cessation attempts.
On the other hand, scale-free networks rely on the previously mentioned concept of hubs, who have a higher degree than the average node and are observed in a wide variety of situations, from computer networks, to business alliances and actors in Hollywood [ref SciAm]. A key principle in building these networks is that of preferential attachment, where nodes are more likely to connect to popular,

rather than unpopular nodes. Once more, this is seen in a number of situations such as web pages having a higher chance of linking to popular web pages than less well-known ones [ref SciEm bara]. Using the BarabsiAlbert approach to generate this type of network, the basic idea is that of starting with a simple, connected base and adding nodes incrementally, considering each other node as a connection candidate [ref BAStat]. The chance of each connection being made is relative to the degree of the current node where a higher degree increases the connection chance. The prevalence of hubs in this type of network is useful in judging the effect of influential members of a community.

2.3 Smoking Cessation & Health

To make an accurate attempt at mapping smoking behaviours to a social network simulation, the basic factors that affect how humans engage in smoking must be understood. There are two aspects to this both an analysis of the current smoking situation as well as how people go about trying to give up smoking. Furthermore, the impact of socialisation on the health of a person is another important factor in building this kind of model.

By looking at NHS smoking statistics for 2012 [ref NHS pg 13], a number of important pieces of information relative to a simulation. In Britain in 2010, 20% of the adult population were recorded to be smoking, with the average number of cigarettes a day being 12.7. Using the definition of 'heavy smoker' as someone who smokes more than 20 cigarettes a day [ref NHS 14] (and from this a light smoker being someone who smokes fewer than 20 cigarettes a day), it can be seen that the average smoker is not a 'heavy smoker'. Adding to this, a study in 2009 displayed that around 67% of smokers wanted to give up and of those questioned, people who had attempted to give up smoking in the last five years were more likely to want to repeat this effort [ref SmokOmn]. When it comes to commencing smoking, those who begin to smoke again state a number of reasons for their relapse including stress and their friends being smokers [ref NHS pg43], indicating a social aspect to smoking actions on top of the fact that those who are giving up are more likely to relapse than a non-smoker beginning to smoke.

Although often specific examples, factors contributing to both the commencement and cessation of smoking has been monitored. In developing countries such as Malaysia, smoking (and in this case tobacco chewing) is on the rise with around 61% of men being classed as smokers [ref Malay]. When analysed in respect to how these people began smoking, a variety of factors such as gender, ethnicity and alcohol consumption whereas only ethnicity of these was observed to be an influence in cessation. Whilst this is a single case and may be influenced by a number of socioeconomic factors, an important point to extract is that it is not necessarily a question of thresholds defining when a person starts and stops smoking. Instead, different factors seem to come into play for each action. More generally, research into cessation factors and their effect on relapse chances shows a multitude of factors that appear to contribute to failed attempts such as previous quitting attempts, presence of other smokers and behaviour/mood changes [ref UCL cess]. On the other hand, the work indicated a lack of effect by bodyweight/weight concerns and amount of cigarettes smoked. Although this was an internet based survey study, the observed aspects of the smoking behaviours are interesting as many different factors are involved but only some of these actually effect giving up, whist others only affect relapsing.

In terms of the social aspect of health, the Framingham Heart Study [ref Fram] investigates long term health concerns of a large social network. Further work has been conducted using the data from the previously mentioned study, particularly in regards to the spread of obesity [ref ObPap]. It was found that there are indications of social interaction playing a role in the way that obesity is present within a network this is crucial as it is an indication of health being affected by those with whom an individual interacts. It should be noted that the type of tie between persons was significant in its effect; for example, geographically close neighbours had little impact whilst a mutual friendship greatly increases the chance of those involved becoming obese.

2.4 Agent-Based Modelling

Centre to the project is the use of an agent-based modelling (ABM) approach to simulation. Fundamentally, it defines a series of agents with attributes, who have a set of behaviours ways they can interact with each other, with the aim being that information not initially provided to the system may emerge. Furthermore, agents should display autonomy, that is that they can function without input from outside the model and that they are social, allowing others the ability to influence their behaviour [ref REPAST PAPER]. With this in mind, it can be seen how the agents can represent humans, with their behaviours and interactions being mapped to smoking given the importance of autonomy and socialisation within a model.

On the whole, this technique brings about a number of advantages over other modelling techniques it allows for emergent phenomena, a more natural way of modelling systems and flexibility [ref ABM methTech]. The first is of particular importance since other methods, for example mathematical models, may be bound by strict limits which can in turn limit their possible results to those expected. Autonomy and simple interactions of agents means that beyond the starting state, any number of an extremely large set of end-states can be reached. As such, with elements of randomness involved, unexpected situations can arise in ABMs. On top of this, the descriptiveness of an ABM is important. As mentioned above, the agents are defined in terms of basic behaviours and interactions which are easily relatable to real-world actions. It is arguably easier to break complex systems into small sub-behaviours than attempt to model the entire environment for not only behaviours, but for all participants. The flexibility of this approach adds to this since once this description is decided upon and implemented, it is easy to add more agents, modify the behaviours and so on without having to redefine the whole model.

Obviously, this does not come without disadvantages. One of the key issues with all modelling approaches is that a 'general model' cannot be constructed so the model is only useful for its original purpose [ref UCL paper]. By extension, this means that the model has to focus on one area of behaviour (which could be very wide itself), thus removing those which are considered external. This can limit the results of the model since many 'external' aspects may have some effect on those modelled. In addition to this, mapping some actions to an ABM is difficult, particularly those in humans such as subjectiveness [ref ABM methTech]. When understanding the results of simulations, this kind of omission from the model must be considered since these behaviours can have a major impact on the course of events; for example, irrational choices by one agent might cause a 'butterfly effect' over the course of the rest of the run which would change the results damatically.

Although it is a relatively new approach to modelling, there are a number of examples which demonstrate that it is widely applicable. In a similar capacity to this project, work has been done to model how viruses spread through humans [ref IEEE paper]. Specifically, the inclusion of real-life data allows the simulation to be built and set up using a realistic base, with the aim of understanding how governmental decisions affected the H1N1 epidemic in Mexico. A particular finding of this study was that a lot of agent-based models use survey data as a basis resulting in a lack of representation of the way in which humans move over time to handle this, phone records were used to build in this aspect of behaviour.

Furthermore, ABM has also been used to investigate how emergency response can work optimally; from terrorist attacks to floods, the approach can be used to understand how current processes can be improved and new actions can be added [ref emerPap]. The two approaches can apply to any ABM approach to understanding how to affect human behaviour. It is noted that for systems which propose such changes where human life is at stake, an amount of verification and validation of the model must be carried out. This emphasises the fact that for the data to be relevant to real-world situations, the simulation must display and acceptable degree of similarity.

2.5 Similar Work

To conclude this section, it is worth considering other, similar pieces of work as these can be useful in informing the direction of the project. It does appear that the particular combination of areas that this project is using has not been widely explored, but there are a number of examples that display some common features.

The first is an analysis of how epidemiology, the study of disease spread, can be analysed by using a social network and agent-based modelling approach [ref epid]. By combining these approaches, the researchers found that it allowed a much more complete view of the field than by studying individual effects alone. This is due to the agent-based approach that provides the opportunity to define interactions and behaviours, many of which can be handled at once. Furthermore, the inherent ability of ABMs to model social interaction means that the social aspects of epidemiology can be investigated more thoroughly. Although they were found to be useful, it was also the case that the researchers emphasised the need for validation, as mentioned above, and that the scope of the model needs to be restricted to avoid the high level of complexity associated with modelling humans.

Closer to the aims of this project, there are a number of pieces of research in regards to the relationship between social networks and smoking [ref combo paper]. In particular, a three-part approach is taken, one focussing on modelling addiction and cessation as functions, another considering influence and the final one looking at generating realistic networks. Whilst going into much greater detail that this project intends, it is useful to note that a probability based approach is used within the modelling within which many different aspects of human behaviours and character are factored in[ref pap1]. When it comes to influence research, it was found that targeted audiences and indirect influence can be the very effective when attempting to change behaviours, but this was also relative to how a person reacts [ref pap2]. In general, the model uses a combination of peer pressure and health when it comes to influencing the decision. [ref 3]

2.6 Summary

Chapter 3

Model Development & Implementation

3.1 Overview

Based on the research detailed above, it was clear to see that the agent based model (ABM) approach was a good match to the problem domain at a high level, the social network could be represented by a graph, the humans being agents (and as such nodes in said graph), with connections between them mapping relationships and being edges within the graph. The model, including details as to how the final version was reached, is below. It was decided during the conception of the project that it should stand up for comparison against a commercial approach. Due to this, a number of design plans were laid in relation to the structure. In general, the solution was written using a 'platform' approach so that, as well as being easier to maintain, it was extensible. To accommodate this, good programming practice was followed by ensuring functional separation and suitable abstraction of methods meaning sections such as decision tree branches or interaction rules could be changed or removed without much effort. Adding to this, a number of input and output methods are supported. Whilst the system incorporates a limited number of ways of generating graphs as social network bases, it also supports sampling other graphs and importing pre-defined ones. Whilst this will be explained in more detail below, it does allow for a great deal of flexibility when it comes to experimenting with how the ABM reacts to non-standard environments. In regards to outputs, two main formats were used. The system can, at any given step in the simulation, feed a list of all agents and their attribute values at that particular moment as well as the social network graph for that step. This is important for analysing how the network changes over time intervals. Furthermore, if necessary, the system can reveal console output to detail what is happening at a given moment though this is more a debugging feature. Throughout the project, a series of existing technologies and toolkits were utilised; given that graph and simulation techniques were heavily incorporated into the model, using libraries for these was key since they are both complex fields within themselves. Furthermore, by using these tools, more time could be spent on the development of the solution itself rather than the back-end representation and management of data. Due to the project being research driven, a lot of time was spent referencing existing information so more traditional development methodologies were not directly applicable in both the initial research section and the development of the model, a loose spiral approach was taken, meaning that there were a number of short cycles, each analysing the work so far, understanding what needs to change then working on those changes [ref spiral]. In particular, this was very useful during the creation of the final model since upon each group of changes, the effect on the behaviour of the model needed to be checked to ensure that it didn't become imbalanced. Although the balancing will be detailed below, issues can stem from the fact that the simulation involves autonomous interaction of many agents, meaning a small change can result in wide-scale effects within the model, often unexpected. Controlling these changes is important as developing on top of an unbalanced model results in subsequent issues later in the project.

3.2 Technologies & Tools

3.2.1 Repast Simphony

Repast (Recursive Porus Agent Simulation Toolkit) is an agent-based modelling framework which provides a number of tools to simplify the process of developing agents and the environment in which they react. Scheduling tools, a simulation engine and more are provided whilst Repast Simphony provides extra features to further aid in the fast development of ABMs by 'wrapping' the features of Repast, allowing for more detailed representation of networks, a more interactive development environment and interaction with tools such as Weka or JUNG [ref RepQnA]. By utilising this framework, a great deal of time was saved in the project a wide range of tools such as a scheduler and a network representation would need to be developed otherwise, meaning a large section of the work would be dedicated to writing and testing this. Instead, Repast handles all this through its APIs. Furthermore, by providing implementations of networks, MORE HERE Whilst it is vast and well constructed framework, using it is not without a steep learning curve since the provided documentation is minimal. As a result, an amount of experimentation is required to use some of the more specialist features, for example incorporating user parameters into the GUI. Largely, this is due to the fact that the user base is naturally limited to those carrying out agent-based simulations and as such the support community is limited to this group.

3.2.2 JUNG

JUNG is the Java Universal Graph Network/Graph framework which provides a series of data structures and related methods for storing and manipulating graphs and networks. At the lowest level, it provides a series of abstract definitions meaning that custom classes for nodes and edges can be used to construct said graphs. In particular, a large number of graph metrics such as betweenness centrality can be calculated via the library. Whilst the sole purpose of this tool is to represent graphs, it was not used to handle the main social network representation due to the fact that using a JUNG graph in the Repast Simphony environment would require a conversion on every step. This is a computationally expensive task, especially once the graph exceeds around 100 nodes so would heavily impact the performance of the ABM itself. Instead of this, it was used to monitor the social network at intervals, using the inbuilt metric algorithms to report on any undesirable behaviours such as excessive clustering.

3.2.3 Java

Java was the main language of the project since a large number of graph toolkits are available such as Repast Simphony and JUNG. On top of this, it provides a large set of standard libraries providing data structures and algorithms useful in maintaining order within the model. Furthermore, due to the established nature of these libraries, they are considerably faster and more reliable than implementations specifically for this project would be. As a side-effect of using Java, the object oriented nature of it lends itself strongly to modular design, reinforcing the platform-based approach described previously and allowing properly designed classes and methods to be swapped in and out as required. Whilst Java does have a number of positives, the nature of the language brings some disadvantages. Repast Simphony, designed for single workstations or small clusters [ref?], is implemented in and works with Java, but should the model require scaling to larger simulations, it would require reimplementation in C++ to be used with Repast HPC [ref]. Although this is not ideal, the usage of Java simplifies the model creation since a HPC version requires much greater involvement in scheduling and other low-level operations meaning that this project can serve as a proof of concept.

3.2.4 Gephi

Gephi, a graph visualisation tool, was very useful in understanding how the networks within simulations were changing over time through the use of a number of visualisation methods. It provides the ability to view a number of metrics such as average path length, formatting based on node attributes or viewing graphs using differing layout methods. In particular, viewing graphs using layout algorithms was very useful since it allows an easy, high-level comparison between two graphs, giving an understanding about clustering, hub nodes and size.

3.2.5 Github/Git

At a more managerial level, Git, and more specifically GitHub was used for version control and source-code management. The project was maintained in one repository meaning that commits formed different versions. This was very useful during the development of the decision tree and network reconfiguration sections of the agent as it required comparison between versions and monitoring the changes made as it progressed. In addition, using this tool meant that development across a number of separate computers was made much easier and in the case of a series of changes not working as planned, allowed for the rolling back of the code to a given point.

3.3 Model Description

Overall, the model can be split into two main functions the agent, representing humans within the network, and the network, representing the social connections between agents. Whilst this is the core, there are a number of ancillary functions such as the platform tools and monitoring agents which provide a wide array of features to make the simulation balanced and simple to use.

3.3.1 Platform

The platform consists of a suite of tools that aid in both the operating and interacting with the model. Broadly, these can be separated into Input/Output, Graph Tools, Simulation Monitoring and Statistics & Constants.

Input & Output

In order to facilitate analysis of the model and the results it produces, the system can input and output in a variety of manners. For input, the aim was to allow users to have a choice in the way that they set the model up whilst output provides feedback as to the simulation state at a given moment. An example of an input requirement is the ability to use a real-world friendship graph which can then be simulated upon, whereas for output, it is useful to be able to graph the change in attributes of nodes over a number of simulation intervals. To handle the importing and exporting of graphs, support for GraphML is included. GraphML is a way of describing graphs using XML and allows for many types of graphs and importantly, attributes relating to edges and nodes [ref?]. Given the specific needs of this project, existing GraphML generators were not suitable so the Java XML library JAXB was utilised to create a tool that, given a Repast network, would output the XML representing the graph and attributes of constituents of said graph then output it to a GraphML file. In addition to this, the tool can import GraphML as either a complete network (i.e. all attributes stated for nodes and edges) or as a barebones graph. For the former, this is simply converted to a network that can be simulated upon, whereas the latter generates the missing attributes in the same way that creating agents on a system-generated graph would use. This means that as long as a valid graph is provided, it can be used in a simulation. Another format supported for the importing of graphs into the system is a CSV importer that specifically handles datasets from the Social Network Analysis Project at Stanford University [ref?]. This is of the form of a list of edges between node IDs, from which a barebones graph can be constructed and filled with attributes in the same manner as a GraphML import. In regards to exporting CSV files, a 'snapshot' tool was created that, at a given simulation step or interval, can output all attribute values for all agents in the simulation. As mentioned earlier, this is very useful when it comes to graphing how attributes change over the course of a period of time as well as debugging balancing issues such as feedback loops.

Graph Tools

A number of services to handle graph operations have been built to maximise the compatibility and usability of the model whilst the rationale behind the generation and sampling methods chosen will be detailed in the Social Network section of this document, the tools will be discussed here at a high level. For generation of graphs, two methods are provided: scale-free and small-world. With each of these methods, the number of nodes required is provided, alongside a number of parameters specific to the generation method. For scale-free graphs, the BarabsiAlbert (BA) model [ref] is used. It, given a connected graph with more than one node in it (which is generated here by creating a user-specified number of nodes then randomly adding edges until it is connected), sequentially adds nodes, each one being evaluated against all the other nodes in the graph with a chance for each to be connected by an edge. This probability is calculated using [INSERT EQ] meaning that edges with a higher than average degree attract more new edges. Once generation is complete, any disconnected nodes are removed since they do not add to the simulation which means that for this method, the user provided number of nodes required is an upper bound, not an exact figure. To aid in the investigation of how existing social features within graphs affect the outcome of simulations, the generator has a feature that allows a base graph to have the BA model applied to it. It should be noted that due to the fact that the base graph does not have connectedness enforced, the output may not be truly scale-free. Small-world graphs are generated using the Watts-Strogatz model [ref]. This method (K), number of nodes (N) and a value beta then constructs a ring lattice of N nodes, connected to K neighbours (split between either side of the node) then circulates this lattice, evaluating all edges and rewiring them to connect the current node to a random node k with a probability of beta. Usually, this method constructs graphs of high edge density meaning that they have specific use when it comes to analysing them as a network; again, this will be described in detail later. Whilst the sampling of other graphs was not a focus of the tool development, the ability to provide a JUNG graph and take a sample of this is provided. The method chosen was snowball sampling [ref] which is a form of a breadth-first sampler, picking a random node and adding layer by layer of nodes until the required number of nodes are available. Although useful, the nature of the sampler means that graphs produced centre around one node which in turn, may represent a very small part of the entire network. Many other sampling algorithms that give better representations of the network-at-large and are easy to include within the system.

Simulation Monitoring

As will be explained later, during development it was noticed that in some situations, the network suffered from excessive clustering causing most nodes to collapse into an incredibly densely connected group. This prompted the creation of WatchMan, which, although operates as an agent within the simulation, does not interact with the network or any of the other agents. At a user-set interval, the agent can export the current network to a GraphML file, output all attributes of all agents to a CSV file and calculate the network metrics. These metrics are the average clustering coefficient for the entire network, percentages of smokers and those giving up and local clustering coefficients on a series of random points. The aim of the metrics is to provide some insight as to the stability of the network so that if necessary, external intervention can be made.

Statistics & Constants

To aid in making the model easy to adapt, constants from many aspects of the system are combined into one location for easier editing. This means that simulation parameters such as the means and standard deviations of randomly generated attributes and the maximum number of cigarettes that can be smoked by an agent is all done from a central location. In a similar vein, a number of statistics tools are included in the system. At the most basic level, normally distributed numbers with a given mean and standard deviation can be generated for use anywhere within the model, especially when creating agents. As stated in the description of WatchMan, statistics based on the network are calculated using the clustering coefficient algorithm that is part of JUNG, both averages for the whole graph and a number of randomly selected points are used. The former is useful for highly connected graphs such as small-world model ones, whereas the latter is useful in graphs of a scale-free structure, where there may be areas of very high clustering surrounded by less dense areas. To aid in determining what nodes are causing these cases of clustering, a list of high clustering coefficient nodes is returned so that should any graph modifications be required, the ideal nodes are available.

3.3.2 Social Network

As discussed in the Literature Review, the social network representation within this model is key to the results being relevant and useful. On top of this, the development process revealed a number of aspects of social network modelling that were not initially expected and had to either be worked around or have the model adjusted to work with them. The following section details both the development process and the final solution to the representation of a social network.

Representation

At the most basic level, the way in which the network is represented provides the foundation for the model. As found in the research section of the project, a graph provides the functionality needed to hold both humans and the connections between them at a sufficient level of abstraction. This level of abstraction is important since modelling relationships accurately and with a lot of detail is very difficult due to a wide variety of internal and external factors to said relationship being able to change it, either gradually or suddenly. As is common within ABMs, nodes within the graph represent humans whilst edges map to some relationship between two humans. It was decided that the graph should be directed to better model a relationship undirected edges only indicate a connection, whereas directed edges reveal much more. Working on the principle that a friendship is not necessarily mutual since one person may consider another a friend whilst the other may not reciprocate. Due to this, an edge from person A to person B represents 'has a relationship with' and by extension 'influences'.

Influence

The concept of influence is modelled in a manner similar to a probability; it is a weight on an edge with a value between 0 and 1, where 0 represents no influence and 1 represents absolute influence. In general, this makes it easier to calculate multi-hop influence as well and influenced attributes and in doing so, drastically reduces the complexity of operations carried out on a regular basis. Furthermore, a fundamental aim of this model is to keep the complexity to a minimum. Rather than emulate relationships and their influences at a more realistic level, this basic approach was judged more reasonable since it would not require further complexity when it involves it interacting with other elements of the model. Importantly, this implementation of influence remaps the concept of positive and negative opinions of a person to a linear scale, for example a person may dislike another, causing them to not want to take on their traits. As such, a negative figure should be viewed as one which has a low influence value and a positive one being seen as holding a higher influence. Multi-hop influence within the model aims to map to the idea of 'friends-of-friends', with the belief being that should an individual's friend be influenced by a third-party, that party should, in turn, influence the individual in some way. In the graph, this is represented by a series of edges from one node to another, as seen in figure X. The extent to which this influence acts on the individual is a matter for tuning of the model and will be discussed in the Agent section of this document. Obviously, the influence at one hop is simply the weight of the influencing edge, whereas to calculate the influence at n hops, the formula seen in figure Y is used. By multiplying the influence of each hop together, a likeness to weakening influence across the mutual friends is incorporated along with the method also giving a probability-alike value that can be used when calculating influenced attributes [ref?]. A common situation may be that there are a number of routes to a given destination node from a source and as such there many be a number of possible influence values for this 'friend-of-a-friend'. Once again, simplicity is maintained by taking the highest value this is based on the assumption that the higher value represents a chain of stronger influences (i.e. friendships) and due to this, a person is likely to opt for that input rather than the ones of weaker influence [ref]. Influenced attributes, in this model, are defined as how an individual perceives another's actions affecting themselves, with the intention being that the influence provides some form of weighting against the actual value of the attribute. As will be described later during the Neighbourhoods part of the Agent section, the idea is for an agent to have an 'ideal figure' to which they move towards, where the attributes for this figure are calculated using averages of these influenced attributes. In regards to calculating the attributes, the general formula used can be seen in fig A. There were a number of changes that were required for some data types but, since these depend on the specific attribute they will be described in the Agent section. Fig A influence x attribute value From both multi-hop influence and influenced attributes, it can be seen that there is not a direct representation of negative opinion causing a person to act in an opposite manner to the figure they dislike. This is a move away from a realistic behaviour as generally, it would be expected for someone to avoid behaving like a person that they do not get along with, though including this would require some form of state in the relationships between nodes to be maintained. Since it was decided that weighted edges were a simple but expressive method, storing state on these is difficult and requires a great deal of extra computational complexity on each turn to determine the current state as well as how this effects the influence.

Graph Generation & Sampling

As described in the Tools section, a number of graph generation and sampling methods are available to provide networks for use within a simulation. Since this system is designed to simulate upon smaller networks than a commercial environment might, using large, real-world networks is not ideal. This is because networks of many thousands of nodes, each requiring many operations to work out influence, attributes and decisions, causes simulations to last for a very long time. As such, an alternative was required that could provide a number of realistic but smaller networks for the simulations. Scale-free networks were included due to their regularity of appearance within social circles [ref SF] typically, they provide a low to medium edge density across nodes, with a number of hub nodes. This is particularly useful since it allows the investigation of 'social hubs' within a network to see whether this kind of figure has a specific effect on the smoking behaviours of those around them. A disadvantage with this type of network is that, on the examples generated for this project, nodes tended to have a fairly low degree. Since edges represent relationships here, it could be argued that it is not representative of a realistic group. To counter this, nodes within this graph could be considered as groups of humans (and in effect a small-world network) or simply a less dense friendship group, such as an office of workers where ties are more likely to show acquaintance than friendship. As an aside, this type of graph was most useful for testing the model on since it clearly exhibited any signs of imbalance or lack of change over simulation steps. Example Scale-free network here Small-world networks are another useful inclusion the main aim with this type of network was to be able to simulate the effects of groups of close friends like one might find in a club or society. In general, a small-world network displays high connectivity relative to the number of nodes present (assuming that the mean number of edges it is generated with is of reasonable size). The key downside of this method is that due to the raised level of connectivity, the chance of small feedback loops forming is very high. This can cause large-scale imbalance within the simulation due to nodes influencing each other to more extreme behaviour and thus exuding this influence to their respective neighbours. As such, this type of network should be used with caution and balanced carefully to avoid these situations when small-world networks are used, the relevant balancing will be stated. Example Small-world network Snowball sampling was the chosen method due to its simplicity in terms of implementation. Due to the nature of this sampling method, one of a breadth-first search from a random node, it characteristically produces graphs that have one major hub from which most edges emerge. Due to the fact that it can be run on graphs sampled from real-world social networks, the product is often a mix between what might be expected of small-world and scale-free networks. Whilst this was not particularly useful for developing the model upon due to the often high level of connectedness, it does allow for a much wider range of tests to be carried out using the system. Example Snowball sample

Graph Stability

In the early stages of development a regular occurrence was for the graph to, due to actions undertaken by agents, exhibit very high clustering coefficients for a number of nodes. Due to the complexity of many agents interacting on a regular basis, it was too difficult to balance this entirely through careful parameter selection and agent actions. In order to maintain this a balance, the WatchMan was used to intervene should the graph become too clustered. The way in which intervention was judged necessary had an effect on the workings of the network and as such, two different methods were implemented and tested. The first of these takes the average clustering coefficient of all nodes in the graph and, if above a system parameter threshold, the 10% of the nodes in the graph, ordered by highest clustering coefficient first, has each edge (both in and out) considered for removal with a 50% chance. Whilst an artificial way to maintain balance, it did provide a more stable graph since by thinning out edges on the nodes central to the clusters, feedback loops are removed which helps to slow the compression of the network. The other method was to chose 10% of the nodes from the graph at random and calculate the average clustering coefficient at one hop for each. If over a separate system parameter threshold to the above, they are stored as a locally high clustering coefficient. Each of these nodes is then considered in a similar manner to the above method, in that they have their edges considered for removal with the chance of this happening at 50%. This is a more targeted approach that aims to prevent strongly bound clusters forming at an earlier stage by again thinning out their ties. In testing, the value of the threshold was revealed to be very important as due to the ‘early stage’ prevention of heavy clustering, the removal had the ability to become overzealous and make the graph sparse. By increasing the threshold for a node to be classed as a locally high clustering coefficient, this method is reserved for only extreme cases. To illustrate the effects of these stability measures, FIGURE X displays how a non-stabilised graph may appear after HOW MANY steps compared to the stabilised version at around HOW MANY steps. Even though the external intervention does mean that the model loses some faithfulness to real-world interaction in respect to balance, the fact that the network can provide much more useful results by running for longer simulations outweighs this. Furthermore, to have the model balance itself through the interactions themselves would require a significantly more involved representation of agent networking which is beyond the scope of this project. FIGURE X non-stabilised graph at crunch, then stabilised at 100k steps

3.3.3 Agent

The agent is the most significant section of the project development, in both operation and time taken to build. Within this section of the ABM, decisions are made about how to change behaviours, influence is acted upon and connections to other nodes are reconfigured; since the social network provides the framework within which the agents act, a lot of the ability to tune the model arises through various aspects of the agent. It is split into two sections: attributes, the data that defines the agent and by extension, the human, and actions which are the functions that an agent can perform in a simulation step. The basic principle for the agent is that within their local neighbourhood, i.e. the nodes that surround them, influence is used to generate an 'ideal person'. This figure is a set of attributes that is effectively the average of this neighbourhood and to which the agent in question would consider the ideal. Using both these attributes and a number of metrics about the surroundings, the agent then follows a decision tree which provides the chance for an attribute change. On the back of this, the agent can then reconfigure their social connections based on their current attributes, scoring themselves against others in their neighbourhood. Throughout the development of the model, testing was carried out by running the simulation on a scale-free graph and observing how it behaved. In general, the expected behaviour was for the network to avoid collapsing into a feedback loop and a high average clustering coefficient whilst there being some change within the graph. This was considered to be a display of a balanced model, which is an aim of the project an example of both a balanced and imbalanced model can be seen in FIGURES X, Y, Z FIGURE X BASELINE FIGURE Y BALANCED FIGURE Z - IMBALANCED

Attributes

A number of attributes were considered with the intention of modelling human smoking behaviour as accurately as possible, however this is a difficult notion due to the lack of detailed information relating about smoking cessation. A large part of the information that does exist, such as NHS statistics [ref], is based on survey information which in turn may bring an element of bias. Generally speaking, the attributes were added on an ad-hoc basis in the early stages of development; once an idea of the kind of factors involved in smoking cessation were known through research, specific ones were implemented on the basis of simplicity and usefulness within the model. Initially, a number of attributes were considered that extended the smoking cessation decisions into areas of lifestyle such as alcohol consumption [ref malay] and stress. Early on in development, it was decided to avoid using these since the methods to simplify and represent them would remove a large part of the usefulness for example, representing stress would require two functions, one to map produce values for stress caused externally to the system and one for stress as a result of the actions chosen. This is a complicated endeavour since modelling stress in itself is field worthy of research. A similar decision was made for alcohol consumption as attempting to model factors such as social smoking when under the influence of alcohol is difficult and again could be the focus of a modelling project itself. The attributes chosen for the agent can be seen in table X. The most basic attribute is that of isSmoker, which simply shows if an agent is currently a smoker or not. This is coupled with smokedPerDay, representing the number of cigarettes smoked per day, form the fundamental smoking behaviour for the model. The number smoked per day is very important since statistics indicate that heavy smokers, those smoking over 20 cigarettes a day, are less likely to quit than lighter smokers [ref NHS] and as such, this is crucial in deciding if an agent should give up. As can be seen in the table, there is an upper cap on this value; since feedback loops are almost inevitable in the system - and may be natural in real-world networks this cap prevents one feedback loop causing excessive damage to the rest of the environment. In early tests without the cap, it was not uncommon for agents to be smoking upwards of 1000 cigarettes per day due to a few nodes influencing others with their higher than average smoking rates, in turn spreading this throughout the network. Although the cap is artificial, it in some way maps to a physical and reasonable limit of cigarettes smokeable in a day. Health is another important inclusion as the quality of someones health impacts their views of smoking and their likeliness to continue [ref NHS]. Although only a basic method of including health in the model, using a value between 0 and 1 means that it can act in a similar manner as a probability. In a commercial model, this attribute could be implemented in a much more detailed way, including types of illness, how developed an illness is and a more accurate function of how health quality changes over time. There is a strong interplay between the status of the agent as a smoker and the change in health. Since the health in this model is in respect to smoking-related illness, being a smoker decreases health slightly on each turn, whereas being a non-smoker increases it. Three separate attributes handle the process of giving up. As indicated in the research for this project, those who are in the process of giving up smoking are likely to relapse and restart smoking [ref NHS] and being able to incorporate for how long someone has been giving up models the process more accurately. The isGivingUp field stores whether an agent is in the process of giving up, and can only be true when that individual is not a smoker. Furthermore, to simulate the giving up period, stepsSinceGiveUp is set to 0 whenever a giving up attempt is started and incremented on every simulation step. Obviously, a point arises where the individual no longer considers themselves to no longer be someone 'giving up' and instead just a non-smoker. The simulation does this by limiting the number of steps someone 'gives up' for, and at that point sets isGivingUp to false so that the effect of attempting to quit no longer impacts on the choice to relapse into smoking. A final aspect is that of the number of attempts at giving up someone has again indicated by the NHS statistics, those who have attempted to give up smoking and failed before are more likely to fail again [ref NHS]. By tallying the number of giving up attempts to date (incremented every time an agent changes isSmoker from false to true) this can be factored in to the decision tree. The final attributes are those of sociability and influenceability, or how easy the person is influenced. Again values between 0 and 1, the former attempts to represent some aspect of willingness to form new social connections. Although not directly related to smoking cessation, it adds an individual trait to agents when it comes to reconfiguring their connections. The latter is used as an extra factor in deciding when agents change their behaviour, i.e. give up/begin smoking, to model the idea that in order for those who reject most influence would require a lot of sustained pressure to take on the influenced behaviour. In addition to this, it provides two extra attributes for comparison of agents, specifically in terms of assessing similarity as sociable people are more likely to interact with others who are also sociable [ref?]. In terms of assigning values to these attributes, the normal distribution was widely used for numerical attributes. Due to the abstraction of the attributes representing largely unquantifiable concepts with decimal numbers determining appropriate values for the means and standard deviation was a matter of balance for the model. As such, by setting the mean and standard deviation sets for all attributes as system parameters, they can be adjusted through different simulations, allowing for analysis into how this affects the network. Generally, opting for a mean of around 0.5 and a standard deviation of 1 provides a reasonable spread of values across the agents. For the boolean attributes of isSmoker and isGivingUp, system parameters specify the probability for each being true with uniform random numbers being tested against them. Again, this allows for the user to investigate different starting states of the simulation with regard to the number of smokers/non-smokers present.

Action Overview

In regards to the 'rules' section of this agent, there are three parts to each simulation step. First, the agent calculates a series of metrics in regards to its surrounding nodes, then uses these as part of a decision process organised into a decision tree. This allows the agent a chance to modify its own attributes and from there reconfigure its connections to other nodes, if necessary.

Neighbourhood Actions

In the network, every agent connected to one or more other node has a neighbourhood. This is defined to be the set of nodes that can be reached within n hops, where n is a parameter to the simulation. Through the development process, it was shown that the value of n affects the result of the simulation and depends on the size of the graph examples of this can be seen in FIGURE X, where n=2 stays stable compared to the clustering of n=3. TABLE X shows the metrics calculated in this stage. FIG X diags for network crunching etc Aside from the 'ideal figure' attributes that are calculated for the current agent, a number of figures that describe the current basic state of those in the neighbourhood are calculated. Percentages of smokers and those giving up are particularly useful in creating the effect of peer-pressure within decisions and are also a possible representation of what the agent would be surrounded by in their life. This can be furthered by including the influence of the nodes into these percentages, which can be seen in FIGURE Y which enables the system to attempt to model situations where a person's susceptibility to peer pressure depends on how influential those surrounding them are; should they be of low influence, it is unlikely that the person would adopt a behaviour that they are displaying. FIG Y comparison of normal and influence percentage As described previously, the compound influence across a multi-hop route can be calculated so from this, the influence to each node within the neighbourhood can be given. Using this information, the 'ideal figure' for this neighbourhood is then produced using influenced attributes by effectively averaging attributes over all of the nodes in the neighbourhood. The way in which influence attributes are calculated is slightly different for booleans but in general, the formula in FIGURE Y is used. Booleans instead have values of true mapped to 1 and false to -1 to allow for influence to be worked into the 'ideal figure'. Once summed over the whole neighbourhood, a negative value indicates false and positive true. Notice that the sum of influence * attribute is divided by the total influence, not the number of nodes. This is because, through experimentation, division by the number of nodes resulted in very low values for influenced attributes whereas using the sum of influence gave typically more reasonable figures. Due to this, it is less a straight average, instead more of a weighed one based on the influence of a node. It should be noted that this will never exceed the values that would be generated by simply averaging out attributes since that would be a case where all nodes were at an influence of 1.DOUBLE CHECK THIS. Another point to note is that when calculating with smokedPerDay, cases of non-smokers (and in turn those who smoke no cigarettes) were excluded to avoid artificially lowering the average.

Decision Tree Actions

In order to change their own attribute values, an agent must go through a series of decisions to choose which value to change. The aim of this approach is to model a human decision making process in a logical and maintainable way, so that decisions can be rearranged, inserted or removed easily. In general, more important decisions are placed higher in the tree since they are more likely to be reached by an agent than any decision lower in the tree, by which point the decision in question becomes one in many. Due to this, when importance of decisions is discussed, those of higher importance will be higher in the tree than ones of lower importance. Throughout the development, a major focus was to uphold balance within the tree. This means that where possible, feedback loops causing extreme attribute values should be minimised and excessive clustering within the network should be avoided. Both of these had to be monitored by testing different versions of the decision tree, with changes being made as necessary. Discuss balancing within tree To begin with, an approach was taken to combine a number of attributes into calculations that can be seen in FIGURE Z, giving the tree in FIGURE P. Generally, the tree was kept fairly shallow in favour of producing values similar to probabilities or range checks. For example, in Figure Z, (b) shows how the decision is made based on whether the agent attributes are within 10% of the influenced attributes if so, the number of cigarettes smoked is adjusted. As each decision required such a combination, to build a sufficient series of decisions the same parameters would have to be formed in different ways for each choice. It became clear that this approach was flawed as it lent itself into collapsing many separate attribute choices into one. This meant that control over how each parameter effects the end choice was lost and also made it very difficult to add new decisions as most were already represented in some way in the existing steps. Furthermore, using combinations of attributes led to extra complexity being added, making it unclear exactly which attributes are causing different effects. Since a substantial part of this project is to understand what the model is showing, difficulty in isolating attributes and judging their effects is problematic. Due to this, the decision tree was heavily separated out into single attribute decisions that lead into one another. FIGURE P original dec tree with combined figs FIGURE Z initial probability calcs Examples of the results of the transformation that took place can be seen from FIGURE A to FIGURE B, where each separate part of the combined decision is put into a 'level' of the new tree. By doing this, not only is the tree much more structured and manageable this way, but decisions can be traced through each branch, revealing how different set-ups can affect the decision making. The overall result of this stage was to move from FIGURE P to FIGURE S. The values on each edge represent the thresholds that need to be matched for that branch to be followed. Figure A combined decision Figure B separated decision FIGURE S v0.9 dec tree (not full one) Also to note is the inclusion of an irrational choice this is to add a human characteristic into the simulation and introduce a small element of unpredictability. The probability of an irrational choice is a simulation parameter so can be changed easily. Importantly, the probability of an irrational choice should relate to the number of nodes in the network, since if the probability is kept the same a higher number of nodes, each making a decision every turn, will result in more irrational choices. The method of inclusion is to simply override the decision in question by logically or-ing it with a function that returns if an irrational choice is to happen. This function is a uniform random decimal number generator, only returning true when the random number is less than the probability. All branches eventually arrive at one of two final decisions, shown in FIGURE E; one decision is to either give up smoking or adjust how many the agent smokes per day, and the other is to begin smoking or adjust their willpower value. Different thresholds can be provided to these decisions in that if the threshold is exceeded, the agent will give up smoking/begin smoking. These calculations can be seen in Figure E1; the intention for the giving up threshold is to compare the input against the number of others in the neighbourhood who are either also giving up or not smoking, with the latter giving a higher threshold due to having similar but not quite the same behaviours. This maps to a 'group mentality' approach to giving up, in that someone is more likely to give up if those around them are also doing the same. A similar concept applies for starting smoking, where the threshold instead is based on those who are smokers should the person begin smoking, they start by smoking the influenced attribute value of the number of cigarettes. In addition to the threshold being exceeded, the agent must generate a probability less than their influenceability value, to emulate a humans potential resistance to influence. Both also have the ability for an irrational choice to override the decision. FIGURE E end decisions FIGURE F end decision formulas FIGURE E1 equations for thresholds During the course of development, some agents were borderline between smokers and non smokers. This meant that they would switch between the two on a regular basis, potentially causing others in the neighbourhood to do the same. In an effort to this, a measure was introduced that counts the number of turns since the last change and produces a probability, seen in EQUATION E2, that slows down this rate of change by guarding the attribute swap from/to a smoker with this probability against a uniform random number. EQUATION E2 turns since last change formula For the giving up decision, labelled as A, either the process of giving up begins (described in the attributes section of this document) or the number of cigarettes smoked per day is adjusted. The latter is done by comparing the influenced attribute value for cigarettes smoked per day against a percentage threshold the threshold is a system parameter and creates a range around the agent's own attribute. Should the influenced attribute be outside of this range, the agent has its attribute moved towards the influenced value by half of the influenced value. This provides a gradual change. In terms of B, a small amount of willpower is removed should they relapse to smoking, otherwise it is added. EQUATIONS E3 attribute changes Although this simplifies the decision-making process beyond that of human cognition, it allows a basic model of peer pressure and susceptibility to general influence. There is the potential for improvement through swapping this decision series with a more accurately modelled one would be straightforward. For the purposes of this model, it provides sufficient complexity and configurability since the thresholds can be set through the decision tree as part of the tuning process. The final decision tree had a number of extra 'decision layers' added to attempt a more accurate mapping of a real decision process; it can be seen in figures X AND Y. The aim of this was to provide a more configurable and realistic decision-making model than the previous version, whilst keeping each attribute in a separate layer. FIGURE C shows at which point in the tree any attributes are changed (outside of the end-states A and B), with FIGURE E displaying the decision values and whether there is a chance of an irrational choice. For the currently smoking arm of the tree, a number of changes were made. The ability to give up in the 'normal' smoker range (here 10 to 15 per day, but this could be adjusted) was only allowed by an irrational choice. The rationale behind this is that whilst lighter smokers are likely to be more able to go without smoking [ref 43 NHS], gradually reducing their own consumption until they quit completely. On the other side of this, heavy smokers are likely be stronger in their resolve against giving up smoking and as such, their willpower may important decision factor than their health due to the habit being harder to break. Health, however, is deemed more important for lighter smokers since any change in the number of cigarettes smoked is likely to have more of an impact than for a heavy smoker. Furthermore, the fact that influenced health is used aims to represent how being less healthy than those in the neighbourhood would provide more encouragement of the person to attempt to give up, with the other side of this being that if someone smokes and is of above average health, they will not feel as much social pressure with regards to said health. For the non-smokers, the tree was expanded to include a number of new layers. If the agent hasn't failed at giving up before, then the willpower is considered most important as it represents their ability to continue the process of giving up. Health is included as a sub-point, since if someone is more healthy than their neighbours they might see that as a reason to continue their giving up effort. For those who have tried to give up a few times (between 1 and 5), the percentage of non-smokers/those giving in the area dictates which route they should take which then leads into a decision based on willpower. The idea here is that since the agent has relapsed to smoking a number of times, the encouragement of the neighbourhood may have a greater effect on their own actions. In addition to this, the willpower decision is based on the influenced attribute for willpower, so the agent is effectively comparing their own resilience to those who surround them if they are more resilient then it is assumed that they require more of their neighbours to be smoking to cause a relapse. If many (in this case more than 5) attempts at giving up have been made, a crowd mentality approach is used, where the giving up status of those around the agent dictates how susceptible they are to relapsing again. Influenced health is included since willpower appears to be irrelevant to these individuals; being more healthy than those in the neighbourhood leads to a lower chance of relapsing. Those who are not currently giving up are diverted to a more simplistic tree as they do not need their decision to be affected by how many times they have tried to quit before. Given that they have displayed the willpower to not smoke in the first place, it is assumed that this is a less important factor than their health. In terms of health, the agent's own value is compared to the influenced attribute in an effort to model an attitude of health-awareness; if someone is of good health relative to others then they are less likely to risk it by smoking, whereas if they are of below average health, they may perceive smoking to make little difference. The end-state decisions for each branch have a percentage threshold stated the values shown on the diagram have been selected through a process of trial and error with the aim of providing as balanced a simulation as possible. Since some decision routes are more popular than others, the percentages need to account for this. With this in mind, they also represent the relative effect of that branch on the chance of giving up. An example is that if someone is a light smoker (¡ 10 cigarettes a day), has never tried to give up before and has higher than the influenced health, then it would be reasonable for them to be hard to convince to give up as they don't smoke a lot and are, compared to their social connections, quite healthy. To show this, that particular end-state requires more than 70% of their neighbours to also be giving up before they will do the same. FIGURE X AND Y tree separated into two FIGURE C table of attribute changes & where they happen FIGURE E table of decisions & whether they have an irrational choice. DOUBLE CHECK THE DIAGRAM, THERE ARE A FEW BOXES WRONG CHANGE TO 10-15 In general, whilst this configuration of the decision tree was settled upon for performing simulations using the ABM, extra levels could easily be added or decisions rearranged. Ideally, this kind of decision-making process would be developed alongside in-depth psychological studies into the field, so that an accurate and expansive tree can be built. Because this was beyond the scope of the project, a more ad-hoc approach was used by designing the tree in accordance to model balance and real-life comparison, leading to some assumptions about human behaviour having to be made.

Connection Reconfiguration Actions

The final stage of the agent actions is to reconfigure their connections to the neighbourhood. By this stage, the agent has assessed the neighbourhood and made changes to their attributes where appropriate so based on this new information, the agent can now look for any relationships to others who are now too dissimilar or for new matchings that are closer to their interests. From the initial research and development stage of the project, the chosen method of comparing agents was to generate a scoring algorithm. This way, any two agents could be compared at any point and a figure to represent their similarity produced. Different attributes will have different weightings when it comes to comparing agents. As the algorithm is intended to imitate the process of a human looking to their circle of connections for other similar people, someone's willpower is likely to have less impact than whether they have the same smoking behaviours. To account for this, the scoring algorithm uses different score values for each attribute this can be seen in TABLE X. Another factor in the weighting is that of how the difference is scored. The algorithm offers three options, the formulas for which can be seen in FIGURE X: percentage difference, attribute comparison and linear scoring. Percentage difference is most commonly used since it provides a scaled score within one interval plus and minus of the current agent's attribute score. Obviously this means slightly different things for different attributes but in general, the higher scoring someone is in an attribute, the more accepting they are of others; an example might be that light smokers would consider someone who smokes a few more cigarettes a day more to be of similar difference to themselves as a heavy smoker would to someone who smokes half their intake. Attribute comparison is a more straightforward mechanism, simply handing out an amount of points should two attributes be the same, or in the case of stepsSinceGiveUp, below a certain value. Finally, linear scoring is only used by influenceability and is a way to incorporate the susceptibility of someone to peer pressure into the score. No scoring is done against the other agent for linear scoring, since it is based on the attributes of the current agent only. A disadvantage of this method is that it greatly simplifies what humans might look for in social ties as it assumes that they are only interested in those who behave similarly. Obviously this is not the case in reality though for the sake of avoiding modelling this connection seeking in a high level of detail, the assumption will stand for the project. If required, the scoring method could be substituted for one which accounts for one which handles the interplay of personality in a more detailed manner. TABLE X lists attribute, max score, min score, comparison type, FIGURE X- pct diff formula, attribute comparison & linear scoring. Once a score has been calculated and normalised to between 0 and 1, this can be used to determine if the two agents should form a connection, remove an existing one or do nothing. The method for this is to have two simulation parameters, ConnectionScoreRemoveBound and ConnectionScoreAddBound, which dictate the boundaries for action to be taken. The remove boundary must be lower than the add and if the score is below this value, the edge is remove. Should the agent score be above the add boundary, then the edge is added. New edges have their influence set through a normal distribution with a mean of FIG A. 0.5 is deducted to avoid the mean exceeding 1, but also serves to move the value from one above the boundary (meaning if the score was set to be the influence mean, all edges would probably be of high influence), to a more central position. Experimentation has shown this to give a more reasonable distribution of influences. FIG A formula for adding edge ND It should be noted that as part of the effort to keep the model balanced, limits were imposed on the number of connections that each node can have. Should another edge be added when the node has reached its limit, the lowest influence edge is removed in favour of the new connection. Once again, the limit a simulation parameter so can be changed to investigate the effects that is has, though testing indicated that it helped avoid clustering and excessive connection-forming with other nodes. Especially important in this area is the boundary values. By lowering the upper bound to 0.65, the graph significantly increased its chance to become very dense and turn into a tight cluster, whereas a raising it to 0.7 maintained a much better balance and avoided the clustering, as seen in FIGURE Y. FIGURE Y show mild change in upper bound In order to introduce external influences to a neighbourhood, every agent has a small chance to form a random connection to some other agent in the graph. This is deliberately set to have a very low occurrence rate as testing showed that setting the probability to be any more than 0.002% of sociability lead to regular addition of edges, effectively rendering the scoring algorithm useless and resulting in an incredibly dense graph. By making this a rare event, it attempts to model the day-to-day chance of a new influence entering a human's life i.e. that of meeting a new person or taking on a new role-model. Over the course of the simulation, there is a chance that nodes may become disconnected from others. As it is slightly unrealistic for humans to have absolutely no social connections, the model reconnects them to a random node in the network with a 30% chance of connection. This is not a certain reconnection since to reach a disconnected state, the edge must have held attributes such that it fell below the removal boundary of its neighbour nodes.

Chapter 4

Simulation Results & Model Analysis

4.1 Overview

4.2 Simulation Analysis

4.2.1 Simulation Parameters & Agent Attributes

4.2.2 Decision Tree Analysis

4.2.3 Sampled Networks

4.3 Model Analysis

4.3.1 Commercial Analysis

4.3.2 Model Analysis

4.3.3 Further Improvements

Chapter 5

Conclusion