ICT-31-2014: Human-centric Digital Age
Project number: 645043

A typology, method and roadmap for HUman-MAchine NEtworks

Deliverable D3.4

Typology-driven modelling and validation of design options

Editor(s)	Vegard Engen
Lead Partner	IT Innovation Centre, University of Southampton
Version	1.0
Date	19/05/2017
EC Distribution	Public

Project Number	H2020 – 645043
Project Title	HUMANE

Title of Deliverable	Typology-driven modelling and validation of design options
Date of delivery to the EC	19/05/2017

Editor(s)	Vegard Engen (IT Innovation)
Contributors	Juri Papay (IT Innovation) Eva Jaho (ATC) Stamatis Rapanakis (ATC)
Reviewer	Taha Yasseri (UOXF)

Abstract	The HUMANE project has developed a typology and method for characterising and analysing Human-Machine Networks (HMNs) in order to help the design process when new networks are being developed or existing networks are evolved. However, evaluating design options is a non-trivial task as networks can be complex and emergent behaviour can be difficult to predict. As an alternative to building and testing prototypes, we propose a simulation modelling approach that not only has a potential cost saving, but may also allow evaluation of scenarios that may otherwise be infeasible or difficult to test empirically, e.g., due to potential dangers involved. Grounded in the HUMANE method, we propose a modelling approach for network simulation using the agent-based modelling paradigm. We propose a Core HMN Model for describing networks that can be readily extended and used for simulation purposes of specific HMNs. We demonstrate the approach via two case studies: Wikipedia and Truly Media. The former provides a case study for introducing design-changes to a well-established HMN, while the latter provides a case study for evaluating design options while the HMN is being developed. As part of the design and evaluation phases of the HUMANE method, we pose some example design-oriented what-if scenarios for simulation modelling, demonstrate how the Core HMN Model can be used and extended, and discuss results from simulations.

Key-words	Human-machine networks; typology; modelling; simulation; ICT design

Versioning and contribution history

Version	Date issued	Description	Contributors
0.1	06/01/2017	Skeleton.	Vegard Engen (IT Innov)
0.2	22/02/2017	Added background, motivation and core HMN model.	Vegard Engen (IT Innov) Juri Papay (IT Innov)
0.3	02/03/2017	Added approach and details about the two example HMNs (Wikipedia and Truly Media) to section 4.	Vegard Engen (IT Innov) Eva Jaho (ATC) Stamatis Rapanakis (ATC)
0.4	09/03/2017	Further details and updates to Section 4.	Vegard Engen (IT Innov) Stamatis Rapanakis (ATC)
0.5	18/04/2017	Updated structure, introducing an implementation section (5). Integrated updates to Section 4.	Vegard Engen (IT Innov) Stamatis Rapanakis (ATC)
0.6	21/04/2017	Added content to Section 5 – analysis and implementation details. Also added abstract, executive summary and introduction.	Vegard Engen (IT Innov) Stamatis Rapanakis (ATC)
0.7	26/04/2017	Added results and conclusions for Truly Media simulation.	Stamatis Rapanakis (ATC)
0.8	12/05/2017	Added results and conclusions for Wikipedia simulation.	Juri Papay (IT Innov) Vegard Engen (IT Innov)
0.9	15/05/2017	Updated results and conclusions.	Juri Papy (IT Innov) Vegard Engen (IT Innov)
1.0	20/05/2017	Final version after internal review.	Vegard Engen (IT Innov) Juri Papay (IT Innov) Stamatis Rapanakis (ATC)

Definitions and abbreviations

Abbreviation	Definition
ABS	Agent-Based Simulation
DES	Discrete Event Simulation
KPI	Key Performance Indicator
HCD	Human-Centred Design
HMN	Human-Machine Network
UGC	User Generated Content
SLA	Service Level Agreements

Executive summary

The HUMANE project has developed a typology and method for characterising and analysing Human-Machine Networks (HMNs) in order to help the design process when new networks are being developed or existing networks are evolved. We understand HMNs as assemblages of humans and machines whose interactions have synergistic effects. That is, networks where the capabilities of humans and machines complement each other in ways that enable output which could not have been achieved by networks consisting solely of humans or machines.

The HUMANE typology provides the means of characterising and analysing HMNs according to 8 different dimensions, addressing, e.g., the agency of the actors in the network, their relationships and the way in which the network is organised/structured. By means of such characterisation, the HUMANE method also covers how implications can be assessed as well as identifying similar networks from which design knowledge and experience can be transferred. However, evaluating design options is a non-trivial task as networks can be complex and emergent behaviour can be difficult to predict. As an alternative to building and testing prototypes, which can be expensive, time consuming and limited in the types of scenarios that could be tested (e.g., due to potential dangers involved or specialist actors who would need to participate), we propose a simulation modelling approach to further exploring the network characteristics and evaluating potential network designs.

The simulation modelling approach proposed here is grounded in the HUMANE typology and method, leveraging the agent-based modelling paradigm. This provides a natural means of representing the network as interacting agents, capable of simulating emergent behaviour. We propose a Core HMN Model for describing networks that can be readily extended and used for simulation purposes of specific HMNs. The core model incorporates key concepts and properties from the HUMANE typology, designed to be extensible and applicable to any HMN.

We provide an approach to guide modellers in identifying opportunities for simulation modelling and developing simulation scenarios, utilising the outcomes of the existing steps of the HUMANE method such as network profiles, diagrams and implication analysis. We guide the creation and implementation of simulation scenarios, posing typical questions one should answer and demonstrate the approach via two case studies: Wikipedia and Truly Media.

Wikipedia is a well-known case study within the HUMANE project. Truly Media is a new HMN that is being developed as a collaborative platform for journalists working on verifying user generated content. This work stems from REVEAL, which is one of the original case studies in the HUMANE project. These HMNs provide two very interesting case studies, posing different needs and opportunities for simulation modelling. While Wikipedia is a well-established HMN with large quantities of historical data available, we can study this data to gain behavioural insights and inform more complex simulation modelling. In contrast, Truly Media is new HMN that is under development, so there is very limited data available. While this is a challenge, we propose ways of dealing with this case that is relevant to simulation modelling for other new HMNs.

For both of these case studies, we demonstrate the simulation modelling approach, following the HUMANE method by first outlining the objectives and challenges of the respective HMNs, discussing

the network profiles and diagrams. Then, as part of the design and evaluation phases of the HUMANE method, we pose some example design-oriented what-if scenarios for simulation modelling, demonstrate how the Core HMN Model can be used and extended, and discuss results from simulations.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURESs

1 Introduction

The final version of the HUMANE typology and method has been published in (Følstad et al., 2017), having undergone two iterations following the application and evaluation via case studies in the HUMANE project. The main aim of the typology and method is to support the analysis and design of Human-Machine Networks (HMN). HMN are understood as assemblages of humans and machines whose interactions have synergistic effects. That is, networks where the capabilities of humans and machines complement each other in ways that enable output which could not have been achieved by networks solely of humans or machines. Machines can be a range of agents, including, for example, services, sensors and robots. Or, in other words, any technology-based actor in the network that performs a function in the HMN in which they interact with and/or influence other agents - not merely transmitting information between other agents.

The typology serves to characterise HMNs on dimensions pertaining to the actors of the network, the relations between the actors, network extent and network structure. The method supports the profiling of HMNs along these dimensions, providing a mechanism for analysing implications of the network characteristics, identifying similar networks, and enabling the transfer of design knowledge and experience in the form of design patterns.

The HUMANE approach provides a framework that is well suited for high level analysis and strategic planning. The work proposed in this deliverable provides an approach to help evaluate design options in a quantitative fashion without the need for developing and testing prototype systems. The approach is grounded in the HUMANE method, incorporating simulation modelling as a way of performing what-if analysis that provides opportunities for, e.g., cost savings, exposing issues at design-time and testing scenarios that would otherwise not be feasible via real-life prototype testing.

In order to make the simulation modelling approach more accessible and help people identify and model key aspects of their respective HMN, we have proposed a Core HMN Model that allows users to describe a HMN in a way that can be used for simulation purposes. The core model incorporates key concepts and properties from the HUMANE typology, designed to be extensible and applicable to any HMN. We demonstrate the use of this core model for simulation purposes via two example simulation scenarios for Wikipedia and a new HMN that is under development called Truly Media.

Wikipedia is a well known case study within the HUMANE project. Truly Media is a new HMN that is being developed as a collaborative platform for journalists working on verifying user generated content. This work stems from REVEAL, which is one of the original case studies in the HUMANE project. These HMNs provide two very interesting case studies, posing different needs and opportunities for simulation modelling. While Wikipedia is a well-established HMN with large quantities of historical data available, we can study this data to gain behavioural insights and inform more complex simulation modelling. In contrast, Truly Media is new HMN that is under development, so there is very limited data available. While this is a challenge, we propose ways of dealing with this that is relevant to simulation modelling for other new HMNs.

Section 2 further discussing the background and motivation for this work, followed by a presentation of the Core HMN Model in Section 3. The approach and modelling scenarios for the two example

HMNs (Wikipedia and Truly Media) are presented and discussed in Section 4. Following this, details of relevant data analysis and implementation of the respective simulation models are given in Section 5. Results from example simulations are presented in Section 6 followed by conclusions and further work in Section 7.

2 Background and motivation

In this section, we discuss the background and motivations for the modelling and simulation work in the HUMANE project.

2.1 Motivation

The HUMANE typology and method have been developed within the project with the aim to aid the design processes of HMNs, observing issues with unsuccessful HMNs relating to a particular failure of taking account of the complex networks that are formed by the synergistic interactions between human and machine participants, as discussed in (Følstad et al., 2015, 2016). Modelling and simulating HMNs primarily forms part of the latter stages of the HUMANE method, at the point where different design options have been identified and prioritised. The design options can be investigated as "what-if" scenarios in order to perform quantitative evaluations before proceeding with implementation. The motivation for performing what-if analysis was reinforced in the HUMANE case studies (Pickering et al., 2016) as a way to examine scenarios not foreseen when initially going through the HMN design process. We will discuss some concrete simulation scenarios below in Section 4, but the general approach is to use simulation modelling as a way of determining the effects of a particular HMN design.

Simulation modelling bears the key benefit of being able to study the behaviour of a system (HMN in this context) without the resources (human and financial resources) required to build (and test) it. Evaluating design options via simulation, therefore, has the potential for cost savings, both for new and existing HMNs, in particular as models of a particular HMN can be re-used, adapted and expanded on for different what-if scenarios and developments that proceed to implementation. Simulation modelling may also allow designers to explore what-if scenarios that would otherwise be infeasible to do in real life, e.g., due to the dangers involved, one of the key points discussed in (Fritzson, 2004) and evident in one of the HUMANE case studies in particular; namely eVACUATE, for which it would be dangerous to test evacuation scenarios in real life, e.g., on cruise ships. As such, HMN designs can be put to the test via simulation to determine the potential effects in different scenarios, such as during an evacuation. Further, simulation modelling also comes with the opportunity to expose emergent and perhaps unanticipated behaviour, which may then be exploited (if positive) or avoided (if negative) when proceeding with implementation.

2.2 The HUMANE typology and method

The HUMANE typology was first introduced in D2.1 (Følstad et al., 2015) for the purposes of being able to characterise and analyse HMNs during a design process intended to form a part of the ISO standard on Human-Centered Design (ISO, 2010). The typology is structured according to 4 layers, each with 2 dimensions to describe the actors, their interactions, the network extent and structure. An overview of the typology is provided below in Table 1.

Table 1 – The layers and dimensions of the HUMANE typology.

Layer	Dimension	Description
Actors	1. Human agency	The degree to which human actors may have impact or cause change through open and diverse activities. *High levels of human agency mean:* People in the HMN typically can engage in many open and diverse activities towards self-decided goals, possibly aiming to influence others.
	2. Machine agency	The degree to which machine actors may have impact or cause change through open and diverse activities, as well as the extent they enable agency in human actors. *High levels of machine agency mean:* The networked machines typically can perform many open and diverse tasks, aiming to influence other actors in the network, possibly appearing human-like, and allowing human actors to do things they otherwise could not.
Relations	3. Social tie strength	The strength of typical relations between the human actors as nodes in the network. *High levels of social ties strength mean:* The social relations in the network typically hold characteristics of closeness such as intimacy, extended duration, and reciprocity.
	4. Human-machine relationship strength	The strength of the relation between humans and machines as nodes in the network *High levels of human-machine relationship strength mean:* The human actors in the network typically are trusting, dependent, and reliant on the machine actors.
Network extent	5. Network size	The number of actors as nodes in the network. *High network size mean:* The network has a large number of members and a dominant position in its market segment. A broad uptake of the network is required for intended network effects to be realized.
	6. Geo-graphical reach	The geographical extension of the network. *High levels of geographical reach mean:* The network covers large geographical areas, and in consequence typically have a character of transnationality and cultural diversity.
Network structure	7. Workflow inter-dependence	The levels of coordination and interaction required between the actors of the network. *High levels of workflow interdependence mean:* The desired results of the activities within the network typically require

Layer	Dimension	Description
		substantial interaction, coordination, and possibly also collaboration between its actors.
	8. Network organisation	The character of the network organisation with implications for predictability and emergence; specifically contrasting top-down vs. bottom-up organisation *High levels of network organisation mean:* The network is characterized by a top-down organisation, often with a hierarchical structure and centralized control. In contrast, low levels of network organisation indicate a flat organisational structure and substantial self-organisation

In order to model and simulation HMNs, the HUMANE typology forms part of the foundation that describes the network and properties that may influence the behaviour of agents within that network.

The first two dimensions describe the agents in the HMN, both human and machines. Machines can include a wide array of agents, such as sensors, software systems, autonomous cars and social robots. Each type of agent can have different levels of agency, ranging from the passive sensors (low agency) to more active machine actors, such as social robots (high agency). Agency provides a framework to describe the capacity for what the agents can do in the network, will relates to, e.g., their motivation for participating in the first place (can they do what they want/need?) and their potential behaviour (do they have a lot of freedom, e.g., to interact and influence other participants? Or are they constrained to fixed, predictable, activities?).

The next layer includes two dimensions describing the nature of relationships that may be observed within the HMN. Firstly, between humans, and secondly, between humans and machines. The relationships are characterised by, e.g., whether relationships in the HMN are reciprocal and whether people trust or even depend on machines. While the HUMANE typology has primarily been used to characterise HMNs at a higher level, the use of the typology for modelling and simulation purposes bring it down to the level of individual types of actors in the networks and the relationships between them. Further, the modelling needs to account for variations within what we may expect as a general dynamic between certain types of actors, as each individual actor may, for example, have different propensity to trust although we may state at a higher level that humans typically trust the machines in a particular HMN.

The two aforementioned layers are key to modelling and simulation. The next layer is the network extent layer, which represents scale of the HMN, but may not have a significant impact on the behaviour of the modelled agents in accordance with the two first layers. However, there are two considerations we highlight here: a) network size can affect the behaviour of the actors, e.g., by emergent behaviours leading to structural changes such as the creation of sub-communities (Shanga, Luoa, Lia, Jiaoa, & Stolkin, 2015), or b) increasing the geographical extent may introduce behavioural differences in segments of the actor population due to cultural norms (Tsvetkova, García-Gavilanes, & Yasseri, 2016).

In the final layer, we describe the network structure at a higher level in terms of the HMN being organised top-down or bottom-up. A bottom-up structure implies self-organisation, which may provide opportunities for flexibility, robustness and sustainability (Juris, 2012), as well as a greater potential for emergent behaviour (Francis Heylighen, 1989). There is, therefore a link between this dimension and agency (in terms of the capacity for what the actors can do in the network). Network interdependence describes relationships between actors at a higher level than in the 'relations' layer; by describing the nature of interactions that may need to take place within the HMN. That is, can actors do things independently, or do they do particular activities that depend on other actors?

As noted above, one of the aims in HUMANE is to provide ICT design support via the ISO standardised methodology on human-centred design (HCD) for interactive systems (ISO, 2010), which is depicted below in Figure 1. This methodology aims to include a human-centric perspective into the software development process. This HCD process comprises four cyclic phases, omitting the initial phase of planning (Maguire, 2001):

- **Context analysis:** understanding the environment for which the HMN will operate, identifying stakeholders, surveying existing users and establishing characteristics the HMN should support. This forms the basis for identifying requirements in the following phase.
- **User requirements:** requirements elicitation and analysis for the HMN, which includes *inter alia* stakeholder analysis and cost-benefit analysis, and establishing clear statements of design goals and benchmarks that the designs can be tested against.
- **Design:** an iterative process of producing design ideas, including mock-ups and possibly simulation of the system with the aim to rapidly seek feedback to progress the designs.
- **Evaluation:** prototypes from the previous phase are evaluated against the benchmarks established in the user requirements phase. The prototypes may be paper-based or software-based. The purpose is to measure and demonstrate how well the objectives have been met, as informing potential re-designs.

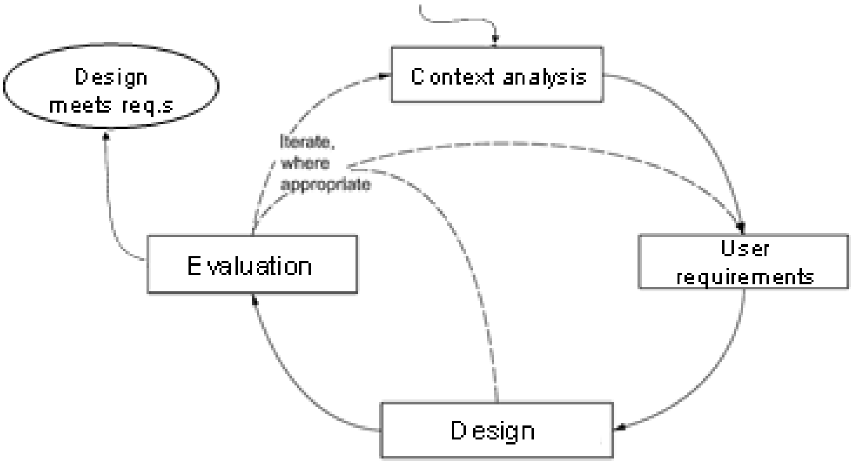
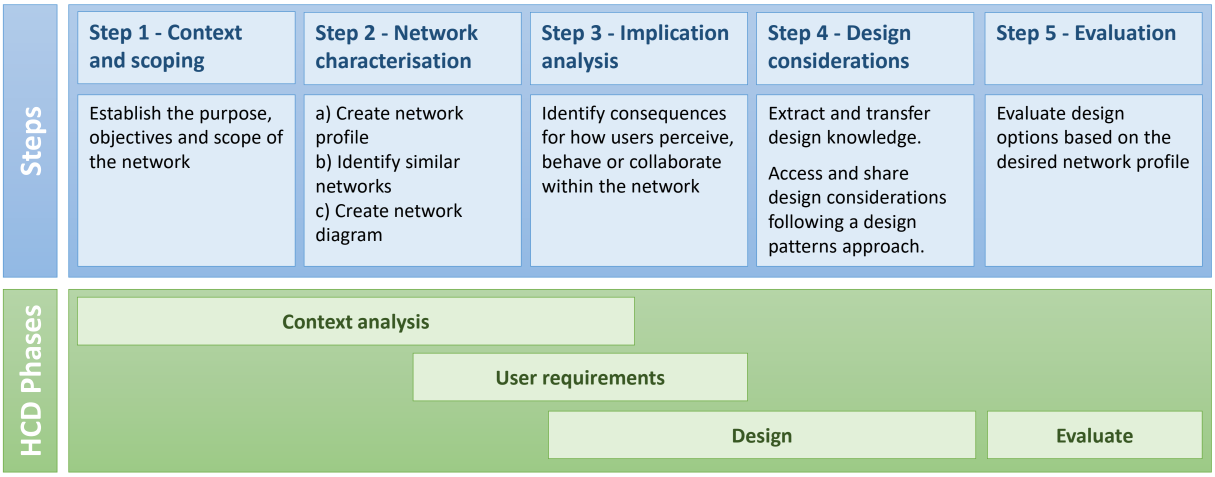


Figure 1 – Human-centred design process (ISO, 2010).

The HUMANE method starts with a similar step as the HCD methodology introduced above, with context analysis. There are actually three steps of the HUMANE method that address the context analysis, which is illustrated below in Figure 2. The five steps of the HUMANE method are outlined at the top of the diagram, and the phases of the HCD method at the bottom.



Steps				
Step 1 - Context and scoping	Step 2 - Network characterisation	Step 3 - Implication analysis	Step 4 - Design considerations	Step 5 - Evaluation
Establish the purpose, objectives and scope of the network	a) Create network profile b) Identify similar networks c) Create network diagram	Identify consequences for how users perceive, behave or collaborate within the network	Extract and transfer design knowledge. Access and share design considerations following a design patterns approach.	Evaluate design options based on the desired network profile

Figure 2 – The HUMANE method mapped to HCD methodology.

As described in HUMANE D2.3 (Følstad et al., 2017), the five steps of the HUMANE method are summarised as follows:

Step 1: Identify and describe the purpose of the HMN, broken down into objectives that can be used when assessing implications of design options, as well as evaluating the design(s) against. This step also includes a scoping exercise, to clarify who the actors are who are to be considered a part of the HMN. This can be particularly important for HMNs that may link with other networks.

Step 2: Characterising the HMN, primarily by creating a network profile using the HUMANE typology, potentially via the HUMANE Tool[1]. Using the aforementioned tool, this is also the step in which similar networks can be identified, from which design knowledge can be extracted from. In this step, a more structural view of the network can also be created, by producing an initial network diagram depicting the known agents and how they are connected to form the HMN. The step overlaps with the two first phases of the HCD methodology.

Step 3: Analyse the implications with respect to consequences of the network design based on the HMN profile and network diagram from Step 2. We have identified give categories of implication: motivation and experience; user behaviour and collaboration; innovation and improvement; privacy and trust; and underlying technical infrastructure.

Step 4: Extract and transfer design knowledge by analysing similar networks to the HMN under design (or re-design) with regards to potential design solutions they have implemented that may be of benefit. This involves accessing design patterns the similar networks have used.

[1] https://networkprofiler.humane2020.eu/

Step 5: Evaluate the design options against the desired network profile identified as a benchmark earlier in the method.

For further information on the HUMANE typology and method, readers are referred to (Følstad et al., 2017). We will return to both when discussing the Core HMN Model proposed in Section 3 and how this is used in Section 4.

2.3 Modelling and simulation techniques

Modelling and simulating HMNs is a challenge due to the complex non-linear and non-deterministic nature of the interactions that can take place. Individual actors in the HMN will behave according to a range of greatly interlinked factors, such as their agency, self-efficacy, trust, perception of risk, social norms and regulations (Pickering, Engen, & Walland, 2017). The relationship between such factors on the behaviour of actors in a HMN is still under active research. As such, modelling and simulation techniques rely on simplifying and scoping the problem where appropriate, in order to adequately reflect a particular HMN in order to answer a set of specific questions. Further, as noted above, an aim of this work in HUMANE is to simulate networks, using knowledge and characteristics embodied in the HUMANE typology (Følstad et al., 2015, 2016), forming part of the HUMANE method. As such, we have included a review of relevant modelling and simulation techniques that lend themselves to network-based modelling and simulation.

A popular probabilistic method for simulation is a Markov chain (Grinstead & Snel, 1997), which is a random process in which changes in the state of the process occur with a transition probability that depends only on the current state, and not on the previous history of the process. Many systems can be cast as Markov chains, by designing the state model for the process in such a way that previous actions (where relevant) are captured by distinct states. For example, we might represent an agent (human or machine) in a HMN using several states representing different levels of engagement.

It is possible to further expand on the Markov chain, by linking other behaviour to the agent's state, e.g., the rate at which they will contribute to the HMN in each state. This allows one to calculate, e.g., the average rate at which users will join, leave and contribute to the HMN. However, this technique is detached from key concepts from the HUMANE typology, such as the two interaction dimensions (human-to-human and human-to-machine) and the network structure.

Markov chains are also closely related to another graph-based modelling technique known as compartment models (Godfrey, 1983), applied in social networks (online community analysis) (Tye, Fliege, & Avramidis, 2013). State models used in Markov chains and Compartment Models can also be used as a starting point for defining Discrete Event Simulations (DES), in which the Markov property need not hold. DES covers a range of techniques (Fishman, 2013), such as queue modelling, which has been used to model detailed processes of computer systems, such as resource optimisation for interactive service oriented multimedia applications that have soft real-time requirements defined in Service Level Agreements (SLAs, which are affected by user interactions/demand) (Cucinotta et al., 2010).

Agent based simulation (ABS) (Macal & North, 2010) can be considered a DES technique as it is driven by discrete sequences of events occurring over time, though some make a distinction between DES and ABS (Siebers, Macal, Garnett, Buxton, & Pidd, 2010). ABS lends itself well to modelling complex non-linear systems to capture emergent phenomena (Bonabeau, 2002; Polack & Stepney, 2005), which is well aligned with the HUMANE approach. For example, agent based simulation has been used for modelling the behavioural dynamics of people in online collaborative environments (Iñiguez, Török, Yasseri, Kaski, & Kertész, 2014), understanding the effects of change in online communities based on interactions of users when treating risk (Nasser, Engen, Crowle, & Walland, 2013) and changing community policies affecting user behaviour (Schwagereit, Gottron, & Staab, 2014; Schwagereit, Scherp, & Staab, 2011).

Whilst the concept of multi-agent system simulation has been laid out in the late 1940s, its application to simulation of real world processes become widespread from 1990 when the computational power offered by computers was no longer an inhibitor (Zambonelli & van Dyke Parunak, 2001). Since then, the multi-agent systems have become renown as a powerful simulation modelling technique (van Dyke Parunak, 1997) that offers the following advantages over the existing analytical tools (Bonabeau, 2002):

1. Capture emergent phenomena.
2. Provide a natural description of the HMN.
3. Offer flexibility as it is an extensible approach.

Complex non-linear systems are susceptible to produce emergent phenomena (Polack & Stepney, 2005). For example, cars produce traffic jams (Cetin, Burri, & Nagel, 2003), distributed systems generate trashing behaviour (Hogg & Hubermann, 1991) whereas human crowds herd (Arthur, 1994). Whilst these are behaviours that arise from the interactions of individual system parts, it is often very difficult, if not impossible, to accurately predict their emergence by looking at the architecture of individual elements at the design time (F. Heylighen, 1991; Wolf & Holvoet, 2004). The use of agent-based simulations allows to identify and understand the origin of these behaviours before they actually have a chance to endanger the real system operation. For example, in (Iñiguez et al., 2014), ABS was used to model the editing behaviour of people in Wikipedia leading to the emergent phenomenon of "edit wars", in which people continually revert or overwrite each other's contributions.

With agent-based modelling, the real system can be represented as a set of interacting components referred to as agents (Zambonelli & van Dyke Parunak, 2003). Such agents represent interacting, autonomous, parts of the simulated system such as people within a company organisation or production machines in a manufacturing system (Macal & North, 2010; van Dyke Parunak & Bruekner, 2004). In HUMANE, these would form various types of human and machine agents in the HMN. As in the real HMN, the agents have their own properties that enable them to conduct their own decisions based on the perceived state of the system (Jennings, 2001; Omicini, 2001). Agents and their behaviour are typically described by simple rules, but with diverse attributes and ability to influence each other (Macal & North, 2010). Properties of agents and their behavioural rules may be kept as simple as possible, yet enabling the simulation of complex emergent behaviours (Macal &

North, 2010). Given the natural mapping of the modelled system components into agents not only allows one to simplify the design of the system while also realising the full potential of the data a company may have about the users in their respective HMN. For example, in HMNs such as Wikipedia, historical data about the editors makes it possible to simulate them as agents with a set of behavioural rules imitating the actions of a real person. This way, the dynamics of the whole network can be reproduced (and validated) with the help of the available, historical, data.

Flexibility and extensibility can be identified on many levels (van Dyke Parunak, 1997; van Dyke Parunak & Bruekner, 2004; Zambonelli & van Dyke Parunak, 2003). For example, it is possible to increase the scale of the system by simply introducing more agents without any additional model refinements. Likewise, the complexity of the model can be modulated by altering the behavioural rules of individual agents or by enabling learning and adaptation. In here, small changes in the behaviour of individual system elements may result in a significant change in the global system output.

The capabilities of an agent are typically modelled as follows (Wooldridge, 2009):

a) ***Reactivity*** - agents are able to perceive their environment and respond in a timely fashion to changes that occur in it in order to satisfy their design objectives.
b) ***Proactiveness*** - agents are able to exhibit goal-directed behaviour by taking the initiative in order to satisfy their design objectives.
c) ***Social ability*** - agents are capable of interacting with other agents in order to satisfy their design objectives.

The above capabilities form part of what we refer to as agency, which is part of the HUMANE typology. We will return to this below, in Section 3. Further to the above, Macal and North (2010) add the environment itself, within which the agents can interact. In this context, the environment would be scoped to the HMN, in which agents could potentially interact with non-agenetic components, such as user-generated content.

As noted above, agent-based modelling is typically tailored and kept as simple as possible for the purposes of specific simulations. However, we have captured several core aspects of HMNs in the HUMANE typology (Følstad et al., 2015, 2016, 2017), which we use to characterise the agents, their interactions and structure of the network. While the HUMANE typology is high level, we propose a Core HMN Model below, which can be used as a foundation for simulations of HMNs. As discussed above, this modelling approach takes advantage of the extensibility of ABS.

3 Core HMN model

The Core HMN Model is a collection of generic classes that allows the core aspects of HMNs to be defined for the purposes of simulating the dynamics of HMNs to facilitate design-based what-if analysis. There are two key aims for this:

a) The model should be generic and flexible enough in its use so that it is applicable to any HMN.

b) The model should reflect the concepts identified in the HUMANE typology, allowing these key characteristics of HMNs to be modelled.

c) The model should be extensible, to be tailored to specific HMNs and respective simulation scenarios.

3.1 Overview

At the most basic level, a HMN can be considered as a collection of Nodes and Edges that are connected in a network (see Figure 3). We add to this basic layer, four different types of Nodes and the notion of a Connection (from a Node to an Edge).

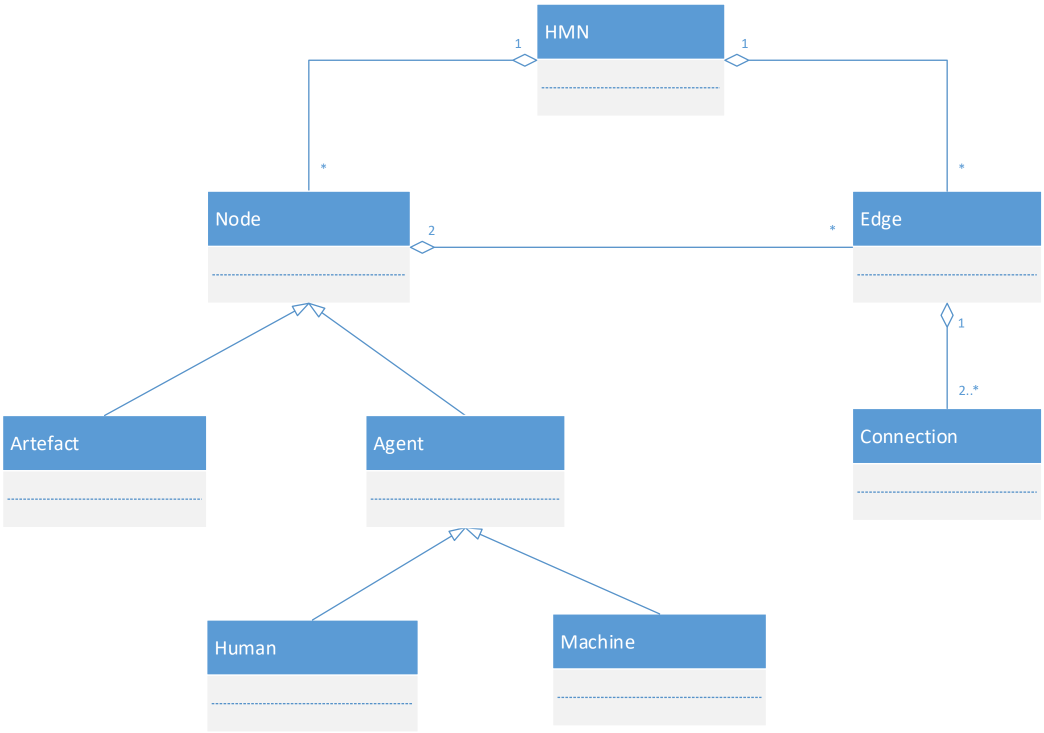


Figure 3 - Class hierarchy of entities of the Core HMN Model

A Node, or also known as a vertex, could be one of two types: an Artefact or an Agent. The key difference between these two is that the latter has agency and the former does not. Further, a node can create and interact with an artefact. For example, it can be a document or another type of content that agents can perform Create, Read, Update, or Delete (CRUD) operations on. Unlike artefacts, Agents can perform activities and influence other agents. Depending on whether we talk about "*conscious intentionality*" or "*programmed intentionality*" we can distinguish between Human and Machine agents and attribute agency to both as active participants in HMNs as per the HUMANE typology (Følstad et al., 2015, 2016, 2017).

An Edge is a link between two Nodes, signifying that there are one or more types of relationship between the two Nodes. The nature and properties of the respective relationships influence the interactions between the two agents, which is encapsulated within a Connection (from the Agent to the Edge). There can be up to two directed Edges between two Nodes, each with multiple Connections.

Each node is connected to the respective edges by Connection objects, which is associated with properties such as trust and trustworthiness. This is illustrated below in Figure 4.

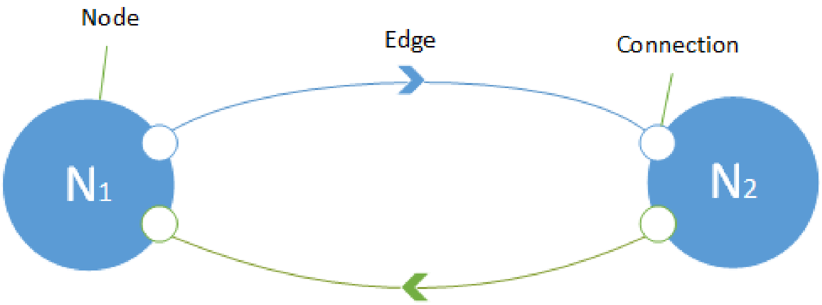


Figure 4 - Illustration of two nodes with their connections to two directional edges.

Typically, in network theory, a network only consists of Nodes and Edges. We have included a Connection class to reflect i) the possibility that there may be multiple relationships between two nodes, and ii) the possibility that the relationship properties from Node A to B may be different from Node B to A. For example, if nodes A and B are both human agents, one node may trust the other more than the other trusts them back. Consequently, their actions may differ when they interact with one another. To be able to model this, we have introduced a Connection class.

In the case of machine-to-machine connections, the Edges may represent a permanent, physical, connection, like a network link via an Ethernet cable. In human-to-human or human-to-machine connections, this generally reflects non-physical, non-permanent, connections. Take the example of an Edge between Machine nodes being a physical network link, machines can connect to this link in many different ways, using different ports or services. These different Connections may have different properties with regards to Quality of Service (QoS), for example, in terms of reliability, quality and speed. Further, as is the case in machine scenarios, there is typically a maximum number of connections possible, which can be represented explicitly with this modelling approach.

To help clarify a human-to-human connections, let us first consider a number of agents, which we may classify as either newbies or experts in a HMN for the purposes of this simple example:

- Agent A is a newbie.
- Agent B is a domain expert.
- Agent C is a domain expert.
- Agent D is a domain expert.

In addition to an Agent classification, such as that above, Agent properties may be described as, e.g.,

- Agent A has a propensity to trust other agents who are Experts.
- Agent B has a propensity to not trust other agents who are Newbies.
- Agent B and C have a propensity to trust other agents who are Experts (like Agent A).
- Agent D has a propensity to be hostile and mistrustful towards other agents who are Experts. This could be due to different beliefs or social background, but we shall keep it simple in this example.

There are Edges between Agent A – B, B – C, B – D and D – C, as depicted below in Figure 5. The Edges merely indicate that there is a communication/interaction channel between the Agents, but not the nature of the Connection between them.

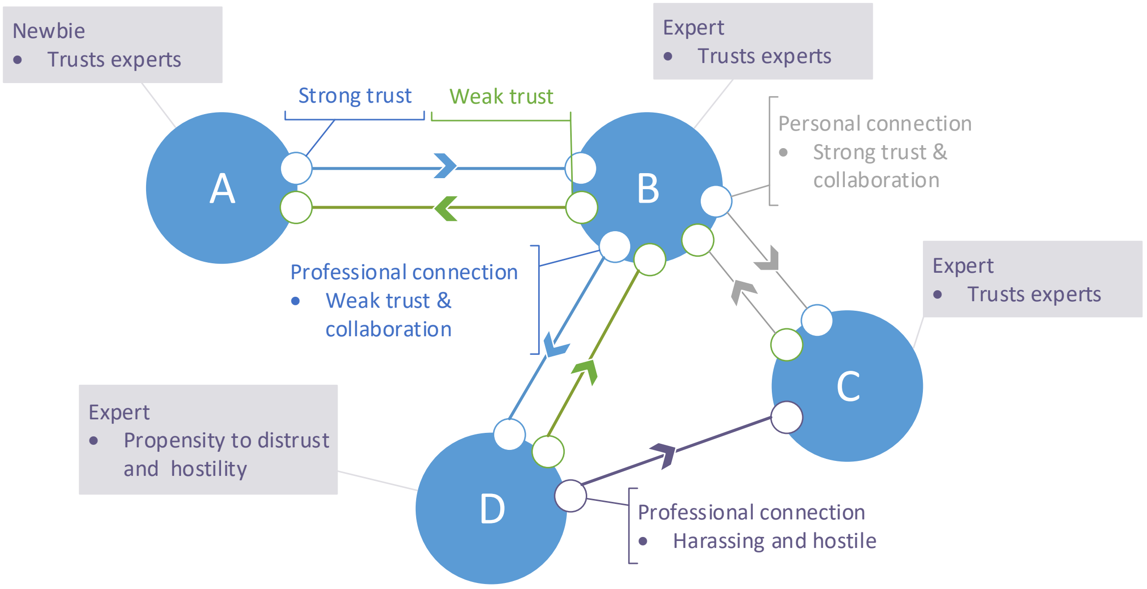


Figure 5 - Example relationships between four agents, depicting both professional and personal connections.

Consider the following Connection properties describing relationships between them that would influence how an Agent may interact with another:

- Agent A has strong trust in Agent B, C and D.
- Agent B, C and D have little trust in Agent B.
- Agent B has a trusting, collaborative relationship with Agent C.
- Agent B has a distrusting, non-collaborative relationship with Agent D.

The relationships between all agents are Professional, but we can also depict a Personal relationship between Agent B and C. For simplicity, only the personal relationship is depicted above in Figure 5. In the above example, the modelling of Connections, thus, allows us to reflect the case where Agent D has harassed Agent C professionally. Due to the personal connection between Agent A and C, this behaviour has influenced the relationship between Agent A and D, although agent A is generally trusting on other Experts.

While we can talk about propensity for certain behaviour at the Agent level, we still need a mechanism to describe the relationships between specific Agents. Further, in the example above we depicted two different types of Connections: professional and personal. This modelling approach could also be applied to distinguish the different ways in which Agents may interact with one another. Consider, for example, IBM Connections[2], which allows a range of ways in which people can interact and collaborate with each other, including the creation of distinct communities that people can engage in via blogs, bookmarks, email, wikis, forums and file sharing.

[2] http://www-03.ibm.com/software/products/en/conn

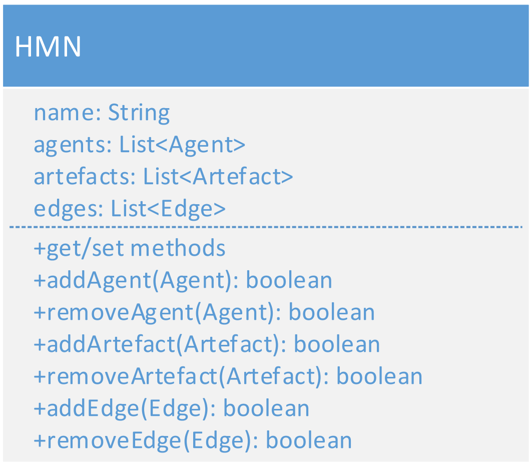
As noted above, this set of classes is considered core to any HMN. We can identify additional classes, such as different types of Machine agents, which may be present in multiple HMNs such as Sensor, Service, and Robot. However, we have scoped the proposed model here to what we consider applicable to any HMN. The intention is for this core model to be extended for specific HMNs and simulation scenarios. Below, in Section 4, we discuss extensions made for some modelling scenarios demonstrating the use of the model. One of the examples are based on Wikipedia, including, for example, User as a sub-class of Human, and Bot as a sub-class of Machine.

In the following sections, we describe the properties of each class.

3.2 Classes and properties

3.2.1 HMN class

The HMN class represents a collection of nodes and edges. This class contains information about the entire network and its topology (see Figure 6).



```
HMN
name: String
agents: List<Agent>
artefacts: List<Artefact>
edges: List<Edge>
+get/set methods
+addAgent(Agent): boolean
+removeAgent(Agent): boolean
+addArtefact(Artefact): boolean
+removeArtefact(Artefact): boolean
+addEdge(Edge): boolean
+removeEdge(Edge): boolean
```

Figure 6 - HMN class.

For implementation purposes, especially with scalability in mind, the class does not merely contain a master list of Nodes. We make use of two separate lists, one for agents and one for artefacts. This will make it more effective and efficient to, e.g., retrieve each type of node and calculate statistics of the simulated scenarios.

In addition to the edges, and nodes (agents and artefacts), the HMN can be given a name. However, it is worth noting that in this model, we do not directly support simulation scenarios involving agents belonging to multiple HMNs. However, this could be explored as an extension to the model in future work.

3.2.2 Node class

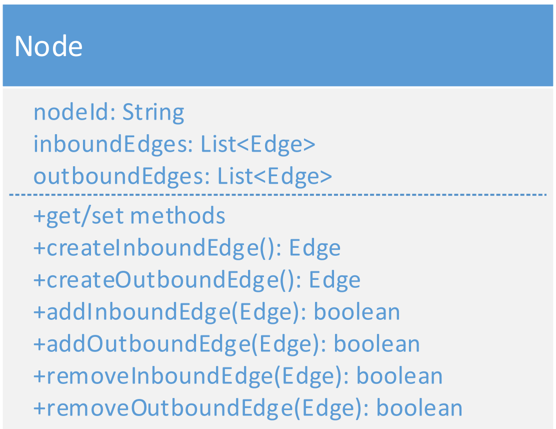

Node
nodeId: String
inboundEdges: List<Edge>
outboundEdges: List<Edge>
+get/set methods
+createInboundEdge(): Edge
+createOutboundEdge(): Edge
+addInboundEdge(Edge): boolean
+addOutboundEdge(Edge): boolean
+removeInboundEdge(Edge): boolean
+removeOutboundEdge(Edge): boolean

Figure 7 - Node class

A Node contains core properties such as an ID and lists of both inbound and outbound Edges from/to other Nodes. As such, this class' primary function is to allow a generic way of describing and working with the network structure, with methods such as adding and removing Edges.

3.2.3 Edge class

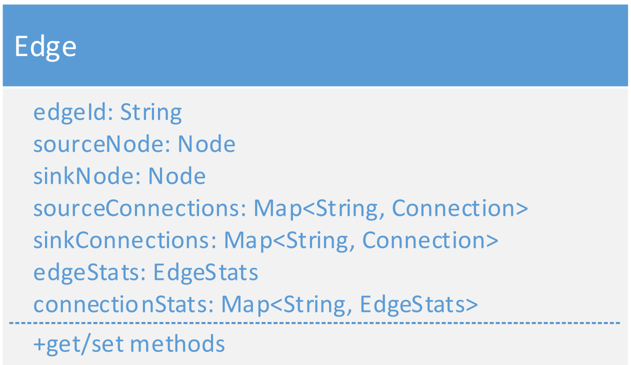

Edge
edgeId: String
sourceNode: Node
sinkNode: Node
sourceConnections: Map<String, Connection>
sinkConnections: Map<String, Connection>
edgeStats: EdgeStats
connectionStats: Map<String, EdgeStats>
+get/set methods

Figure 8 - Edge class

As discussed above, an Edge is directional. It has a source Node and a sink Node (the destination, or end point). In the overview diagram depicted above in Figure 3, we have implemented a direct association between a Node and an Edge. Conceptually, it would be natural to express that a Node is associated with an Edge via a Connection, as shown below in Figure 9. However, this conceptual model would be inefficient for simulation purposes, making it very laborious to, for example, i) determine which nodes a particular node is connected to, and ii) determine the types of connections between a pair of nodes. Another reason is that users of the core model may find the use of Connections for certain types of Nodes to be unnecessary, such as Edges between Agents and Artefacts. In this case, it is appropriate to model the notion that the Agent has some trust in the Artefact (e.g., a Wikipedia article), but the Artefact cannot experience trust back as it has no agency.

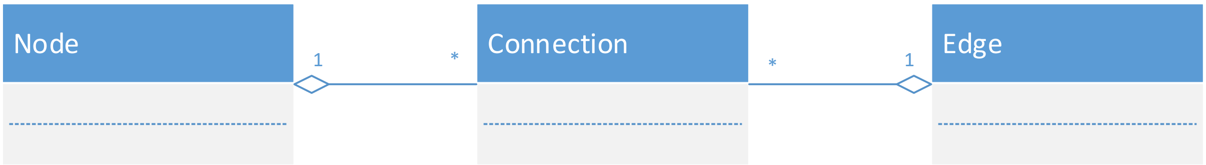


Figure 9 – Conceptual relationship between Node, Connection and Edge.

In the implementation of this Core HMN Model, there are two Maps of Connections. One for source Connections and one for sink Connections. The key for each of the two maps (String) represents the name of the type of Connection, e.g., “professional” or “personal”. It could also reflect specific means of communication, such as a “forum” or “wiki”. The key has been purposefully left very open, as the uses can vary significantly in different HMNs. For more details on the Connections, please see Section 3.2.4, below.

The remaining properties facilitate interaction statistics, which are encapsulated with in a class called EdgeStats. This class has got two properties:

- The *numberOfInteractions* represents the level of interaction from the source node to the sink node along the Edge.
- The *interactionStrength* represent the strength of the relationship between two nodes. We are not prescriptive about how this is calculated in the model, but it may be a function of the *numberOfInteractions* parameter and the number, or type, of connections between the two agents. For example, a “personal” relationship may have a greater weight than a “professional” relationship.

The model includes a Map of EdgeStats objects (connectionStats), which allows users to differentiate interaction statistics according to the type of connection, using the key as the name of the connection type, as for the two connection maps discussed above. In addition to this, the class also includes another instance of EdgeStats that can be used for accumulating the stats across the different connections. This is also to provide the opportunity for quicker retrieval of information, should users of the model wish to use this property as such.

3.2.4 Connection class

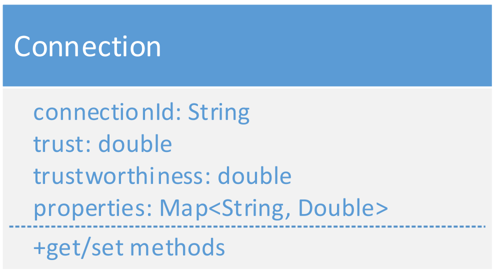


Figure 10 – Connection class

This class represents a Connection from a Node to an Edge, as discussed above. The class contains a property to give the Connection an ID, and two specific trust-related properties; trust and trustworthiness. Trust is one of the key foci in the HUMANE project, which influences how people behave in a HMN (Dwyer, Hiltz, & Passerini, 2007; Jones & Leonard, 2008; McKnight, Carter, Thatcher, & Clay, 2011; Thatcher, McKnight, Baker, Arsal, & Roberts, 2011a). This is reflected in the HUMANE typology, particularly in the two dimensions of the interaction layer (Følstad et al., 2015). However, it is an area that is still under active research (Pickering et al., 2017), in order to understand the relationship between trust and the behaviour in HMNs, let alone how to quantify such properties. However, we include these properties to facilitate a numeric or probabilistic approach to quantifying such values, should it be possible in the respective HMN.

As noted before, Edges are directional, and a Connection from the source Node to the destination/sink Node along the Edge encapsulates properties determining how they may interact with the destination/sink on the basis of properties like trust. Similarly, Connection objects associated with the destination/sink Node encapsulates properties determining how they may behave with regards to the interactions they receive from the source Node.

While the Connection class includes a Map data structure, allowing users of the model to add any kind of numeric parameter given a unique name, trust and trustworthiness are included as explicit properties, for the reasons mentioned above. Their use warrants explanation, as the use of these properties depends on whether the connection is for a source or sink node of a directed edge.

From the perspective of a SOURCE node:

- They have trust in the SINK node.
- Their trust is based on the trustworthiness of the SINK node.
- Their trust influences their interactions towards the sink along this edge and connection type, e.g., they may not even choose not to interact.

From the perspective of the SINK node:

- They have trust in the SOURCE node.
- Their trust is based on the trustworthiness of the SOURCE node (as somebody/something interacting with them).
- Their trust influences how they may react to the incoming interaction from the SOURCE node.

Therefore, if this is a SOURCE connection type:

- Trust is associated with the SOURCE node.
- Trustworthiness is associated with the SINK node.

And, conversely, if this is a SINK connection type:

- Trust is associated with the SINK node.
- Trustworthiness is associated with the SOURCE node.

3.2.5 Artefact class

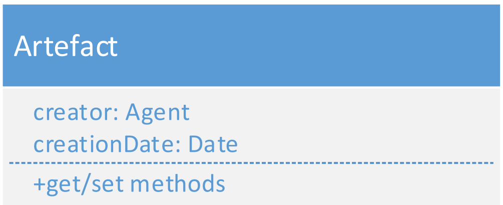


Figure 11 - Artefact class

The Artefact is a passive Node that Agents can create and interact with. Artefacts do not have agency. They could represent a wide range of things, such as a Wikipedia page, an online forum thread, a file, etc. Properties include the Agent (Human or Machine) that created the Artefact and the date time stamp in which it was created. In addition to these properties, the Artefact class inherits the properties from the Node class (see Section 3.2.2).

This class is intended to be extended by specific Artefact type of classes by users of this core model and may not be used directly. For example, a Wikipedia Article or a Tweet, as we will see in the Wikipedia and Truly Media cases discussed in Sections 4 and 5.

3.2.6 Agent class

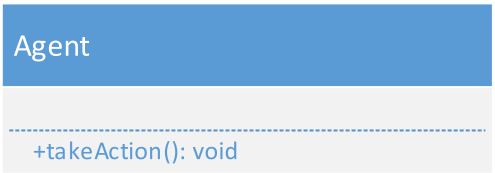


Figure 12 - Agent class

Unlike Artefacts, an Agent has agency and can perform activities in the HMN. There is a relationship between these two classes, though, as Agents can perform CRUD activities on Artefact objects.

This class has no properties. It inherits all properties from the Node class (see Section 3.2.2). It does, however, define one (abstract) method that is important for simulation purposes; takeAction(). This utilises polymorphism, allowing generic simulation code to call this method on the various types of Agents, which may trigger them to take one or more action in the HMN. We will see examples of this in Section 4.

3.2.7 Human class

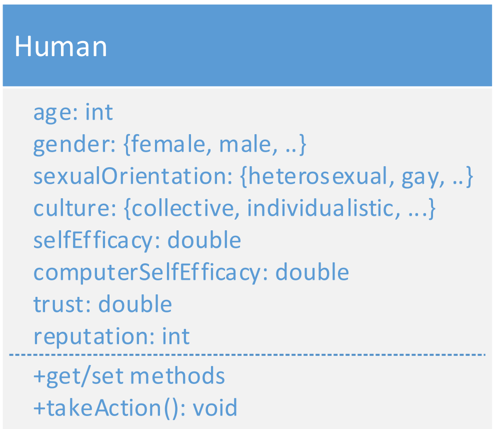

Human
age: int
gender: {female, male, ..}
sexualOrientation: {heterosexual, gay, ..}
culture: {collective, individualistic, ...}
selfEfficacy: double
computerSelfEfficacy: double
trust: double
reputation: int
+get/set methods
+takeAction(): void

Figure 13 - Human Class

The Human class is an extension of the Agent class. A distinguishing feature of the Human Agents is "*conscious intentionality*", which essentially means that the actions taken are non-deterministic; depending on mental states, cultural background, beliefs, etc. Modelling human beings is a very complex task, and significant simplifications are expected for simulation modelling purposes (in line with the discussion above in Section 2.3). Here, we include key properties that we have exposed via the HUMANE typology or related research, which are likely to affect the way in which they may interact with other Agents or Artefacts. The properties may also determine how other Agents may interact with them. One example of this is discriminating behaviour based on gender or sexual orientation. However, for simulation scenarios where these factors are not relevant, such parameters are expected to simply be omitted.

We have already discussed trust in Section 3.2.4 for the Connection class. We do include a trust property here as well, which with the intention to reflect a Human's general propensity for trust. Basic human properties include age, gender and sexual orientation. For both gender and sexual orientation, we have only included a subset of options as the options are vast[3] and unlikely to be widely applicable in simulation scenarios which users can extend, should they require:

- Gender: FEMALE, MALE, AGENDER, ANDOGYNE and OTHER.
- Sexual orientation: HETEROSEXUAL, BISEXUAL, GAY, LESBIAN, ASEXUAL and OTHER.

We have included culture, but also simplified this concept to:

- COLLECTIVE, INDIVIDUALISTIC and OTHER.

We choose this level of granularity in line with findings from (Tsvetkova et al., 2016) on Wikipedia, showing that people behave particularly differently whether they belong to a collective or individualistic culture. If finer granularity is needed, users of the core model can extend this as needed.

The Human class includes a property for reputation, which is a common mechanism that may be made explicit in HMNs to help build trust (Schwagereit et al., 2011). For example, in e-commerce,

[3] See, e.g., http://www.telegraph.co.uk/technology/facebook/10930654/Facebooks-71-gender-options-come-to-UK-users.html and http://www.uua.org/lgbtq/identity for lists of gender types and sexual orientations.

customers can rate both products and sellers, such as for eBay, Amazon and Etsy. The ratings become a symbol of the reputation of the seller or product, influencing the trust customers may then experience. Researchers have explored reputation mechanisms in types of HMNs that traditionally have not included reputation mechanisms, such as Wikis (Dencheva, Prause, & Prinz, 2011) to help increase contribution quantity and quality.

The two final properties of this class, self-efficacy and computer self-efficacy, which are core constructs affecting the behaviour of people in HMNs in a recently proposed model by Pickering et al. (2017). Bandura (1977, 1982, 2012) defines self-efficacy as an individual's belief in their ability to be able to achieve a given objective. This notion has also been applied to technology (Marakas, Johnson, & Clay, 2007; Thatcher, Zimmer, Gundlach, & McKnight, 2008) and its acceptance (Mun & Hwang, 2003). Computer self-efficacy refers to the belief that one can achieve goals via the help of computers (technology) (Thatcher et al., 2008). These properties can, therefore, be used to model how different people may engage in the HMN, in particular with regards to interactions with machines. For example, Thatcher et al. (Thatcher, McKnight, Baker, Arsal, & Roberts, 2011b) attempt to model the relationship between self-efficacy and the intent to explore technology (i.e., utilise its functions) and Pickering et al. (2017) model the relationship between (computer) self-efficacy, trust, and agency as core constructs affecting the behaviour in an HMN in more general terms.

As noted above, the core model exposes the properties that are seen to affect the behaviour of human agents in a HMN. However, determining the values of some of these properties, such as trust and self-efficacy are beyond the scope of this model, as this is an ongoing research area.

This class is intended to be extended by specific Human classes by users of this core model. We will see some examples of this in Section 4.

3.2.8 Machine class

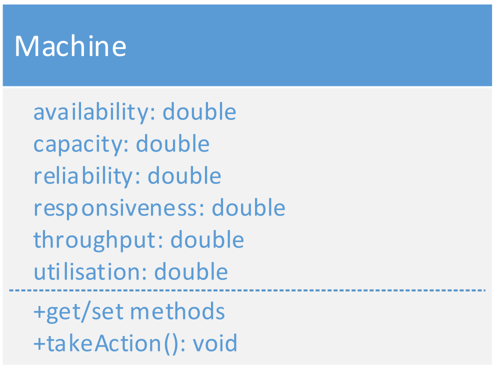


Figure 14 - Machine class

Similar to Human agents, Machine agents can perform activities in the HMN, but the actions taken by the machine can typically be considered deterministic for very specific purposes (Tsvetkova et al., 2015). The increased autonomy, complexity and stochastic nature of machines has, however, led to an updated definition of machine agency, seeing Machine actors as increasingly active and influential participants in HMNs (Engen, Pickering, & Walland, 2016).

A machine is typically more predictable in their behaviour than a human agent, but there can be a wide variety of machines with different properties, e.g., ranging from sensors (largely passive) to social robots (active and anthropomorphic). Hence, only what is considered core and common properties are included here, which include typical non-functional requirements for Machines[4]:

- Availability: accessibility of the machine, i.e., if it is online, able to take connections, etc.
- Capacity: resources that the machine possesses.
- Reliability: probability of the machine functioning correctly.
- Responsiveness: the ability and speed at which a machine responds to requests.
- Throughput: numbers of task the machine can complete in a given time interval.
- Utilisation: percentage of using the available capacity of the machine.

This class is intended to be extended by specific Machine classes by users of this core model. We will see some examples of this in Section 4.

4 Modelling approach and scenarios

In this section, we discuss an approach to using the Core HMN Model presented in Section 3 as part of the HUMANE method to evaluate design options.

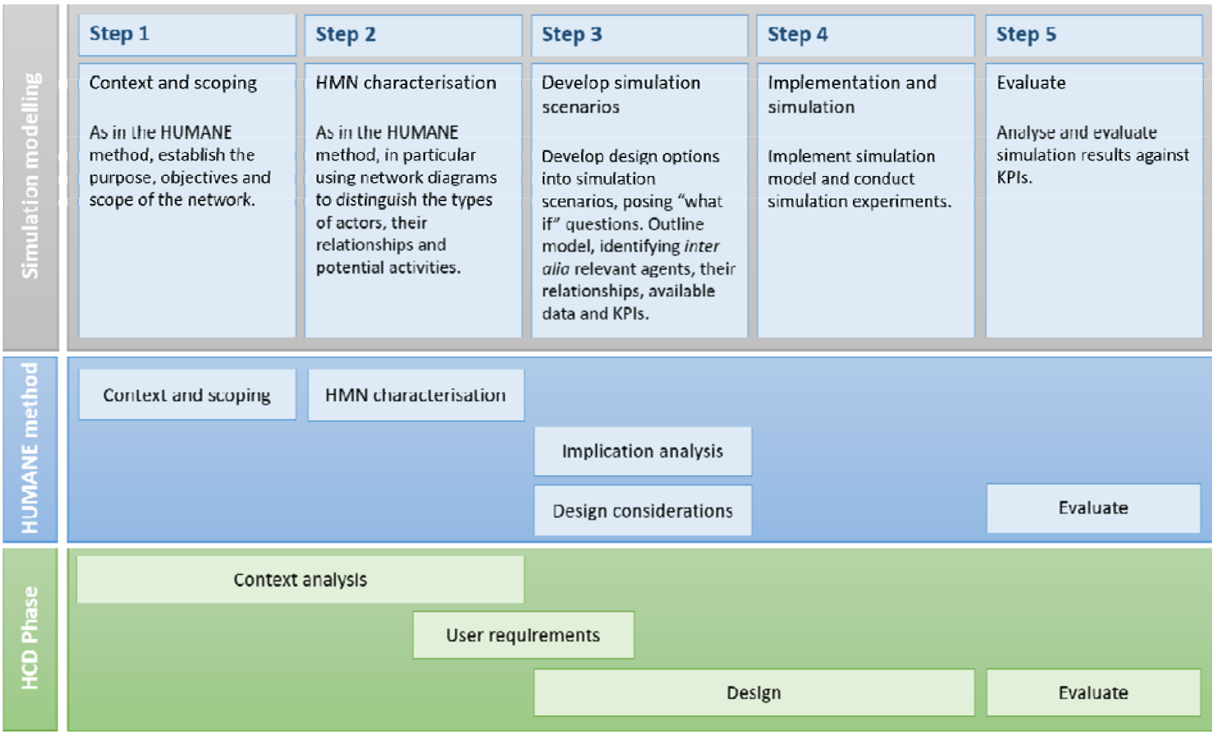
The purpose of the Core HMN Model is to facilitate the modelling process, providing an extensible foundation to build on. As such, there are two key things to note here, which we have touched on previously: a) the artefact, human and machine classes can be extended for specific HMNs / simulation scenarios; b) all the properties of the connection, artefact, human and machine classes do not need to be used as they may not be relevant or even possible to determine in all simulation scenarios. Regarding the latter point, new properties may be introduced, some of which may reflect the essence of properties in the core model, but encoded differently, e.g., to utilise the data available for the particular HMN. For example, if historical data are available for a particular HMN, behavioural probabilities may be extrapolated without needing to model the complex influences on those probabilities such as trust and trustworthiness - which is still under active research and is non-trivial to quantify, as discussed previously.

4.1 Approach

We have introduced the HUMANE typology and method above in Section 2.2. For more details, readers are encouraged to see (Følstad et al., 2017). For the modelling and simulation of a HMN we also take a formal approach, which consists of the five steps of the HUMANE method: setting the context and scope of the HMN, creating a HMN profile and network diagram, developing designs we want to simulate through defined simulation scenarios, and finally evaluating the design.

[4] We do note that the interpretation and use of some of the machine properties may be different for specific machines. For example, we are purposefully non-specific about how ‘resources’ may be defined for the ‘capacity’ property.

An overview of simulation modelling approach is given below in Figure 15. This overview summarises five steps, illustrating how this relates to the steps of the HUMANE method and the HCD methodology (as discussed above in Section 2.2).



	Step 1	Step 2	Step 3	Step 4	Step 5
Simulation modelling	Context and scoping As in the HUMANE method, establish the purpose, objectives and scope of the network.	HMN characterisation As in the HUMANE method, in particular using network diagrams to distinguish the types of actors, their relationships and potential activities.	Develop simulation scenarios Develop design options into simulation scenarios, posing "what if" questions. Outline model, identifying *inter alia* relevant agents, their relationships, available data and KPIs.	Implementation and simulation Implement simulation model and conduct simulation experiments.	Evaluate Analyse and evaluate simulation results against KPIs.
HUMANE method	Context and scoping	HMN characterisation	Implication analysis Design considerations		Evaluate
HCD Phase	Context analysis	User requirements	Design		Evaluate

Figure 15 – Overview of the simulation modelling approach.

<u>Step 1 – context and scope of the HMN:</u>

- This initial step of the HUMANE method includes describing the purpose and objectives of the HMN and establishing the scope of the HMN.
- Setting the objectives is important in order to a) identify and frame issues as problems that should be addressed, b) identify the design opportunities that may positively affect the objectives, and c) evaluate the efficiency of the HMN in the simulation output using specific Key Performance Indicators (KPIs). For example, if an objective is to foster collaborative work, the simulation scenarios should be evaluated on the basis of whether any potential designs actually improve collaboration among the actors in the HMN. As such, collaboration becomes a KPI for simulations that may be conducted in the later steps of the method.
- In addition, if the HMN connects to other networks, it is important to establish the scope of the HMN and the interfaces to the other networks.

<u>Step 2 - create HMN profile and network diagram:</u>

- Both the HMN profile and the network diagram are needed to identify properties for a simulation. In particular, the HMN profile sets the high-level framework (properties of actors and network), while the network diagram helps to distinguish the types of actors, their relationships and potential activities, which the higher-level profile does not reflect.

Step 3 & 4 - develop designs and simulation scenarios:

- Different design options may emerge and be identified as promising, warranting further evaluation. The design options should ideally be associated with problems or opportunities identified in step 1, and may be further elaborated following implication analysis in Step 3 of the HUMANE method.
- Promising design options can, at this stage, be developed into simulation scenarios. First, this should be framed as "what if" statements that can be evaluated in the final step. For example, in addressing a problem with fostering collaboration, some design changes could be defined as:
    - *What if we change the policy on interactions to X.*
    - *What if we increase machine agency?*
- The "what if" examples above are very high-level. They would need to be further detailed to the level of actions of human and machine agents (examples of which we shall see in the following sections). Further, the simulation scenarios need to be detailed in terms of which actors and what activities to simulate. As discussed earlier, a simulation model would typically be simplified and tailored to specific questions, rather than attempting to model the complexity of the entire HMN. Typical questions that help in this scoping exercise are:
    - *Who are the relevant agents?* Initial input to this would be the diagram from Step 2.
    - *What are the relationships between the agents?*
    - *What are the relevant activities the agents can do? How frequently are they likely to perform the different activities? What are the interactions possible, e.g., can all agents interact with all others? Are there interaction patterns, e.g., can/will some agents only interact with certain types of agents?*
    - *What are the key properties that may influence how the agents interact?* For this, the properties of the Core HMN Model should serve as a guide. *For example, is the cultural property of human agents important? What about reputation? Do some agents act differently towards others based on properties like gender and sexual orientation?*
    - *What data are there for setting properties associated with the behaviour of the agents? For example, how frequently would an agent perform a particular action? Or, for reputation of human agents, is this something that can be extracted from data?*
    - *What are the KPIs of the simulations?* KPIs may have been identified in Step 1, but may need to be adapted to quantitative measures that are possible to generate via a simulation model.

Step 5 - evaluate:

- Implement simulation scenarios to determine the design impacts on the HMN objectives defined in step 1. Evaluate design options using the KPIs defined above for the different "what if" scenarios that may be simulated.

Finally, it's worth noting that the evolution of HMNs is stochastic in nature. As such, simulation models should be run multiple times to capture statistical measurements of the outputs. Following this, we give examples of applying this approach to two HMNs: Wikipedia and Truly Media.

4.2 Wikipedia

Wikipedia is a free online encyclopaedia hosted by the Wikimedia Foundation, which offers a platform for people to create and edit entries on any subject only limited by imagination and regulations established by the community of editors. Wikipedia's main features are described by its "5 pillars" (Wikipedia, 2015):

1. Wikipedia is an encyclopaedia: It combines many features of general and specialized encyclopaedias, almanacs, and gazetteers.
2. Wikipedia is written from a neutral point of view: We strive for articles that document and explain the major points of view, giving due weight with respect to their prominence in an impartial tone. We avoid advocacy and we characterize information and issues rather than debate them.
3. Wikipedia is free content that anyone can use, edit, and distribute: Since all editors freely license their work to the public, no editor owns an article and any contributions can and will be mercilessly edited and redistributed.
4. Editors should treat each other with respect and civility: Respect your fellow Wikipedians, even when you disagree. Apply Wikipedia etiquette, and don't engage in personal attacks. Seek consensus, avoid edit wars, and never disrupt Wikipedia to illustrate a point. Act in good faith, and assume good faith on the part of others.
5. Wikipedia has no firm rules: Wikipedia has policies and guidelines, but they are not carved in stone; their content and interpretation can evolve over time. Their principles and spirit matter more than their literal wording, and sometimes improving Wikipedia requires making an exception.

Wikipedia runs on the MediaWiki platform (MediaWiki.org, 2017), which is a free, open source, project developed in PHP. The most prevalent type of actor in Wikipedia is the 'editor', human participants who create and maintain content. In addition to the human agents, there are bots (derived from the word "robot") that can also create and edit pages. They are mainly in existence to help maintain Wikipedia, being able to process and analyse pages, performing mundane and repetitive tasks far quicker than any human editor could. For example, there are reported vandalism attempts that have been detected and rectified within seconds (Nasaw, 2012). Bots may flag updates that human editors can make, e.g., to improve the quality of pages, or automatically make updates as in the aforementioned case.

4.2.1 HMN objectives and challenges

The objectives Wikipedia can be summarised as follows:

a) Collection of the sum of human knowledge.
b) Providing the sum of human knowledge freely to the Internet users.
c) Facilitating cross-cultural synergy towards collaborative work.
d) Advocating the free sharing culture.
e) Developing infrastructure for fostering large-scale collaborative work.

For the purposes of this report, we will focus on a particular issue that affects objective c) in particular, namely that of edit wars. Despite the second and fourth points of the “five pillars” of Wikipedia, as outlined above, the editing behaviour of people in Wikipedia has given rise to the phenomenon referred to as “edit wars”, in which people continually revert or overwrite each other’s contributions (Iñiguez et al., 2014). This, in turn makes it harder to reach a consensus, affecting the reliability and quality of information on Wikipedia.

4.2.2 Profile and network diagram

The profile of Wikipedia has been discussed in (Følstad et al., 2015), but we revisit it here in order to demonstrate how the dimensions of the typology may inform the modelling and simulation work. A spider diagram of the Wikipedia profile is depicted below in Figure 16.

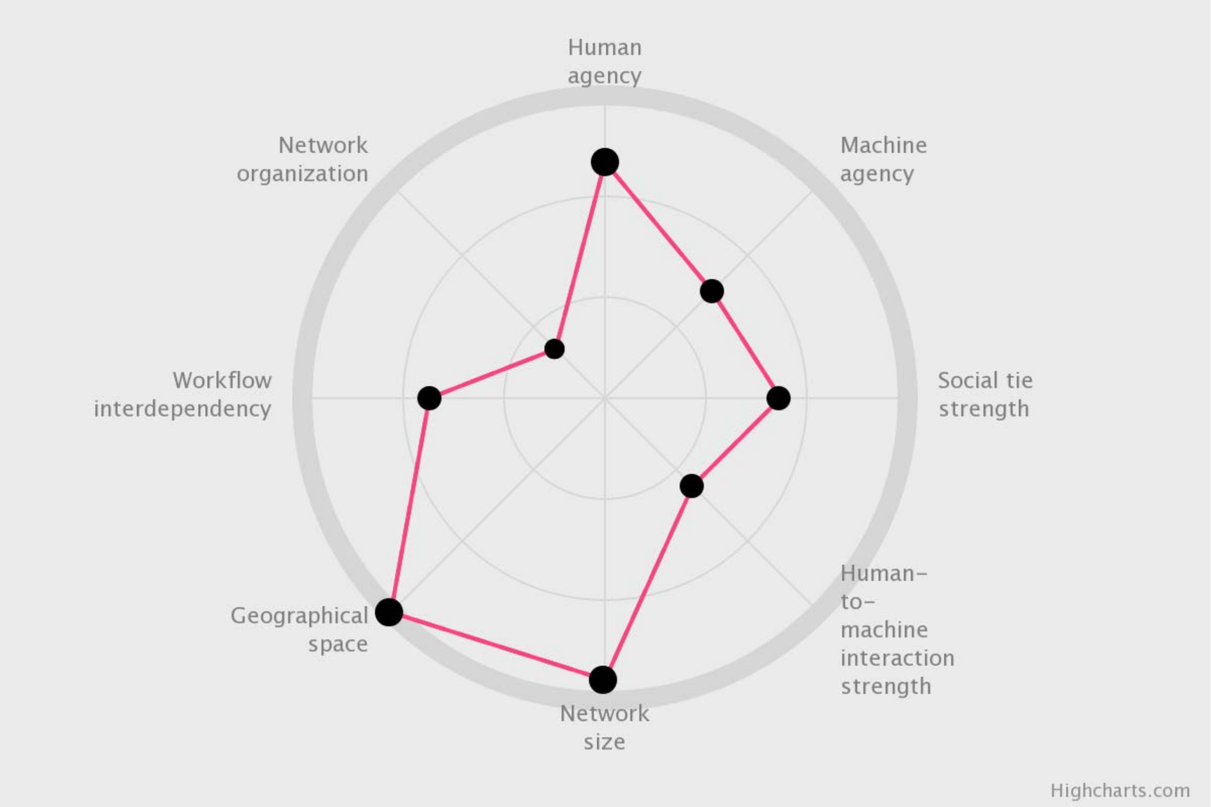


Figure 16 – Wikipedia profile.

Human and machine agency: human agency is high, as people have a lot of freedom and flexibility in expressing themselves and creating content on Wikipedia. Machine agency is intermediate. Although bots can do the same activities as humans, they are limited in the creation of content and are mainly concerned with maintenance tasks. A question to consider here is whether there are any opportunities for increasing machine agency further? For example, enabling bots to detect and help facilitate the resolution of edit wars.

Social tie strength: the relationships between the human editors in Wikipedia is weak, meaning that they interact infrequently with each other and are not invested in forming relationships. The interactions that take place are, naturally, focused on editing documents. This may not be something that ought to be changed, as weak ties are seen to facilitate the transmission of information (Watts & Strogatz, 1998) and help generate creative ideas (Burt, 2004).

Human-to-machine interaction strength: this is fairly low, and many users of Wikipedia may not even be aware of the bots and their activities. Bots do an important job at maintaining Wikipedia, but humans are not dependent on them and can override the bots of needed. Increasing the dependency, for example, may rather have negative impacts on the HMN, e.g., reducing the flexibility and adaptability, and perhaps stifling the creative crowdsourcing process. Although the impacts may be negative, simulation modelling could be used to explore scenarios to get a better feel for the emergent behaviour that could arise from changes addressing this dimension.

Network size: this is massive in terms of consumers, editors and bots. As it is already at such a large scale, Wikipedia will have already had to deal with infrastructure to be able to support the network. Possible simulation scenarios may investigate new infrastructure models, however. Further, simulation scenarios could explore the resources required for running the bots, perhaps in terms of determining the minimum number of bots required to perform the maintenance tasks in Wikipedia effectively and efficiently enough.

Geographical space: this is global, though the majority of users are from western countries. Recent research has shown that the cultural background of the editors does affect their behaviour (Tsvetkova et al., 2016). As such, different strategies for addressing the edit wars may be explored for the different language editions of Wikipedia. That is, an approach that is effective on the English (western, individualistic, cultures) edition may not work well on the Asian (collective) language editions.

Workflow interdependence: this is intermediate as editors can chose to contribute independently, but do collaborate with both humans and machines to produce content for articles on Wikipedia. However, there's no dependency between the agents *per se*; there's no requirement for co-ordination and a single person can effectively create and maintain an article completely independently. However, for complex issues, collaborative effort is needed. As such, mechanisms can be explored via simulation modelling to help facilitate collaborative work more effectively.

Network organisation: bottom-up, i.e., self-organised. This is core to Wikipedia and would likely change. There are few regulations, which are set by the editor community itself. Although a popular expectancy was that Wikipedia would not succeed on the grounds of its network organisation, it has proven to be a successful approach (Spek, Postma, & Herik, 2006). In terms of increasing or ensuring a sufficient level of quality of content that consumers can trust in, Wikipedia have added functionality over the years. For example, including references for statements in order to ensure quality content. Further, bots help detect and flag cases where references are needed. Most recently, Wikipedia has banned The Daily Mail newspaper on grounds it is considered an unreliable source of information (Jackson, 2017). However, these changes has not made significant changes to the network organisation, which remains bottom-up. The decision to ban The Daily Mail as a source stemmed from a consensus reached by Wikipedia editors.

A network diagram for Wikipedia is illustrated in Figure 17, depicting the key actors we are interested in here, namely the contributors (editors) and bots. There are contributors with administrative rights as well as consumers who may never create or edit content on Wikipedia. Although the consumers make up the largest user group, they are not involved in the edit wars problem, which we focus on in this example.

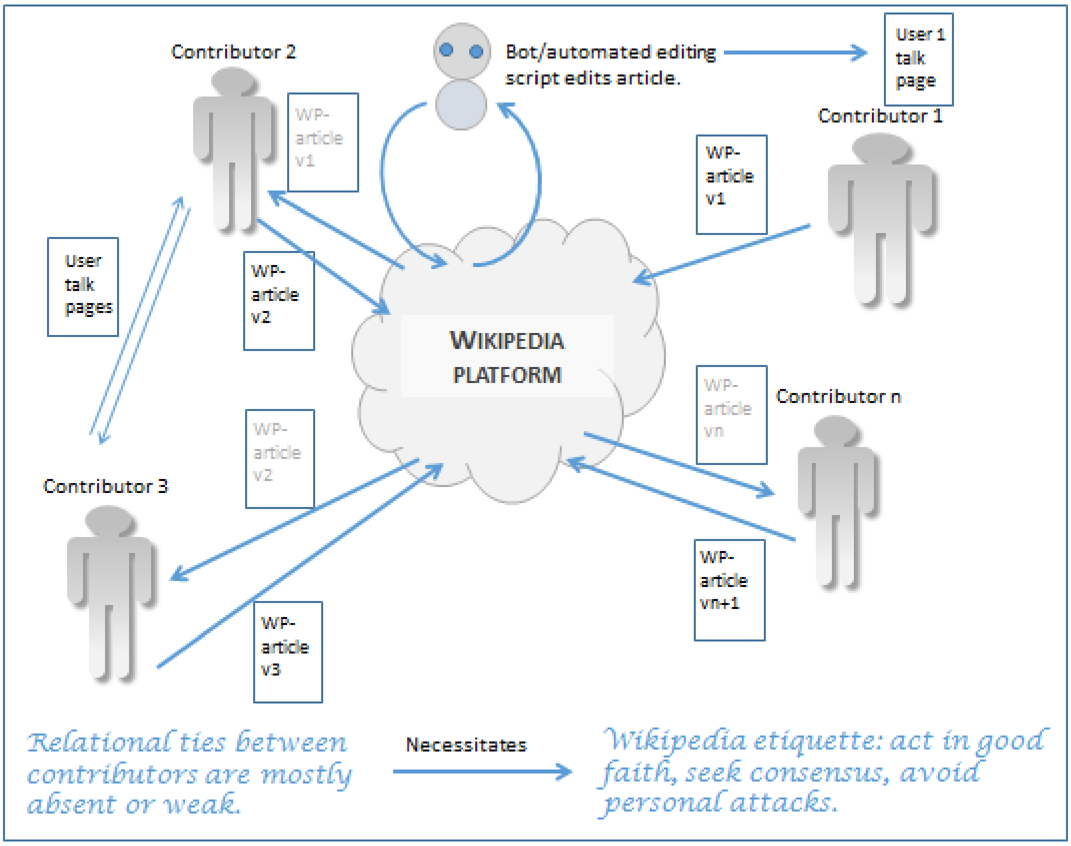


Figure 17 – Wikipedia network diagram.

The diagram depicts of different human contributors edit pages on the Wikipedia platform. Some of them may discuss a particular page (contributors 2 and 3) via its respective talk page. Contributors may also be anonymous users, in which case their contributions are logged and attributed to their IP address. As noted before, bots can do most things that human actors can, and are programmed to perform activities such as detecting vandalism, fixing grammar and spelling mistakes, cross-linking articles, and checking links.

4.2.3 Simulation scenarios

Following the above sections, we have identified edit wars as a focus for simulation modelling. Based on the discussion of the HMN profile, we have indicated key dimensions relevant to this: human and machine agency, and human-to-machine interaction strength. Although, at this point, further investigation, especially via data analysis, is needed, we can identify the following high-level what-if hypothesis:

> *What if we increase the agency of bots to be able to detect and help facilitate the resolution of edit wars?*

While the detection of edit wars is arguably trivial to implement, enabling bots to help facilitate the resolution of edit wars raises a range of interesting questions. First, what are the mechanisms that could be used or introduced to help facilitate the resolution of the edit wars? To simply encourage discussion on Talk Pages? Give warnings if agents do not resolve the edit wars within a certain time frame? Second, how would agents involved in an edit war react to a bot intervening? This would most likely depend on the type of agents involved, human users or bots, as well as individual characteristics of the human users. Further, there may also be a difference in how anonymous users generally behave compared with registered users.

There could be several design solutions for this in terms of how bots may help facilitate the resolution of the edit wars. In order to propose design solutions, further data analysis is needed to understand the problem, particularly with regards to the possible distinctions between types of agents who are involved in edit wars and how they may respond to a bot intervening. As such, we would also benefit from knowing more about who are typically involved in edit wars, for how long, and ideally also why they engaged in an edit war in the first place. That is, are there agents with specific intentions to troll others by engaging in edit wars? Do some agents engage in edit wars with multiple people? Do some agents engage in edit wars with the same agent(s) multiple times across different articles?

The above questions relate to the initial steps of the simulation modelling approach outlined in Section 4.1, scoping the problem. As the focus here is on edit wars, we can scope the modelling of the actors to those who contribute to Wikipedia – not the vast majority of users who merely consume information (reading articles). Of the contributors, we can distinguish between humans and bots on the one hand. Humans can further be distinguished as registered or anonymous users.

All agents are able to interact with all others. But there more interesting question, as posed above, is that some agents may only interact with some and not with others. This can be based on the type of agent or their perceived reputation, for example. That is, people may naturally have a higher propensity to trust a bot reverting one of their contributions over an anonymous user. Understanding such characteristics is part of data analysis of historical Wikipedia data, which is presented in Section 5.1.

In terms of the relevant activities agents can perform in Wikipedia, this includes: creating articles; editing articles; reverting contributions (by others or oneself); and discussing on talk pages. We note here that an edit war is actually an emergent phenomenon stemming from the ability to revert contributions. A simple definition of an edit war is when two agents mutually revert each other. For example, agent B reverts agent A and then agent A reverts B (see Figure 18**Error! Reference source not found.**). This mutual re-reverting can carry on over a long period of time.

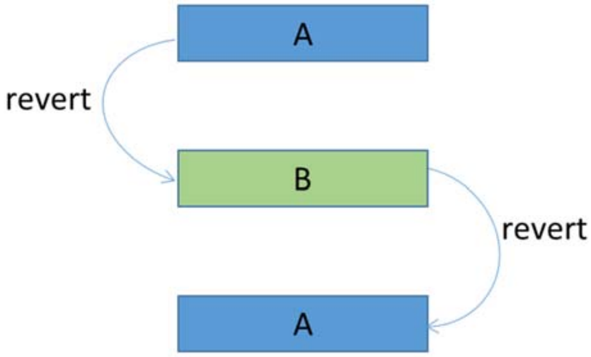


Figure 18 - Mutual reverts

One of the opportunities for simulation modelling of Wikipedia is that the HMN has been running for many years, generating large quantities of data that are possible to download and use. As such, we can analyse the behaviour of groups of agents, as discussed above, as well as determining individual behavioural patterns and properties of each agent in the system. The latter really takes advantage of the agent-based simulation modelling approach, as discussed earlier in this document.

The datasets used have been prepared by previous researchers (Iñiguez et al., 2014), and are available online[5]. The data are available in a text file for a particular language edition of Wikipedia. The datasets contain logs of the activities on different Wikipedia articles, including a timestamp, the revision ID of the article, a flag to indicate whether the activity was an edit or a revert, and the ID of the user performing the activity. Note that the dataset does not contain information about the content of the articles, nor the activity of users on the articles' talk page. For the proof of concept discussed here, we have based the analysis and simulation modelling on the Simple English edition[6] of Wikipedia. This version of Wikipedia uses simple English words and grammar so that it is more accessible, e.g., to children and adults who are learning English. This dataset contains 74,880 articles and 147,076 agents (25,037 registered users, 121,636 anonymous users, and 403 bots).

KPIs for simulations focusing on edit wars include: the number of edit wars; the average number of reverts in edit wars; and the average duration of edit wars. In order to validate the simulation model, other performance metrics include: the number of articles created; the number of edits; and the number of reverts. Note that we do not consider here mechanisms that may reduce the occurrence of edit wars, although we may can hypothesise that this may be affected by the design changes for enabling bots to help facilitate their resolution.

The first simulation scenario is to establish a ground truth, to test a model of Wikipedia as it is. Following this, scenarios for increasing the agency of the bots can be explored. Such simulation scenarios may involve exploring thresholds for parameters, such as when a bot intervenes in an edit war.

[5] http://wwm.phy.bme.hu/light.html
[6] https://simple.wikipedia.org/

4.3 Truly Media

Nowadays, it is possible to "report" and share information at any time from almost anywhere in the world. This is regularly done before professional journalists arrive at the scene of an event (e.g., riots at a demonstration or a natural disaster). As a consequence, journalists increasingly turn to Social Media to find both news and background information to add to their situation assessment and, subsequently, their reporting. The ability to verify content in an easy, transparent and fast way is becoming more and more desirable, especially when taking into consideration (a) the sheer quantity of content found in Social Media, and (b) the fact that a lot of content consists of hoaxes, rumours or deliberately misleading information (e.g. propaganda, fake news, and other untrue statements).

Truly Media is an online collaboration platform for journalists working on User Generated Content (UGC) verification tasks. Truly Media includes both machine-generated recommendations about the validity of UGC, as well as recommendations from journalists about this content. More specifically, Truly Media allows journalists to use UGC for their reporting in an easy and trustworthy manner. The platform provides for easy aggregation of UGC, management of UGC items, real time collaboration, a clear verification workflow, semi-automated verification tools and sharing of verification tasks and information between news desks or even between multiple media organisations.

Truly Media focuses on Twitter, which is the most widely used platform for breaking news and is used by journalists for their daily work. Twitter has transformed the way breaking news are transmitted, by letting people play the role of journalists and report directly from the scene of an event. However, the use of UGC content for news has also accentuated problems of incomplete, poorly edited or fake information, or contradicting posts by different users. Truly Media offers innovative tools to address the challenge of validating UGC, combining both humans and machines. For the verification analysis, Truly Media integrates TruthNest[7], a commercial service providing all the verification evidence to the users and guide them to discover the truth that may be hidden behind the content.

The basic content verification modules for the platform were developed in the course of the REVEAL project (one of the HUMANE case studies). The work of ATC and Deutsche Welle[8] carried out in REVEAL served as a basis for the conceptualization of a collaborative UGC verification platform, Truly Media[9], which is now being developed partly with funds secured from the Google Digital News Initiative's Innovation Fund.

The fact that Truly Media is currently under development, means that HUMANE's feedback in the design of Truly Media platform is easier to integrate. Therefore, the results from this simulation will help us to improve the design of the platform.

4.3.1 HMN objectives and challenges

[7] http://www.truthnest.com/

[8] Both partners from REVEAL consortium

[9] Co-developed by ATC SA and Deutsche Welle (DW) within the Verify.Media Project (http://ilab.atc.gr/projects/verifymedia), funded by Google's DNI fund.

The objectives of Truly Media can be summarised as follows:

a) Facilitating perceptions of trustworthiness.
b) Helping users collaboratively verify UGC.
c) Providing smart visualisations for cross-checking evidence.
d) Developing infrastructure for fostering real-time collaboration.
e) Facilitating journalistic workflows.
f) Discovering people with expertise in specific areas.

For the purpose of the HMN modelling and simulation, we will focus on the second objective, i.e. how to help users collaboratively verify UGC. The need for collaboration tools to verify content from Social Media has risen in the news media industry. One of the main tasks that should be accomplished with the use of those tools is the merging of the verification process output performed by different journalists on certain social media posts. The output might be produced from users that work in different teams or even different organisations. The ability to manage conflicts in the verification process of a certain social media post and to propose solutions to its users is an important part of a verification platform, and will be addressed in the simulation scenario.

4.3.2 Profile and network diagram

In order to profile Truly Media, we followed the step-by-step process of the HUMANE Network Profiler (https://networkprofiler.humane2020.eu/). The derived output diagram is depicted below in Figure 19.

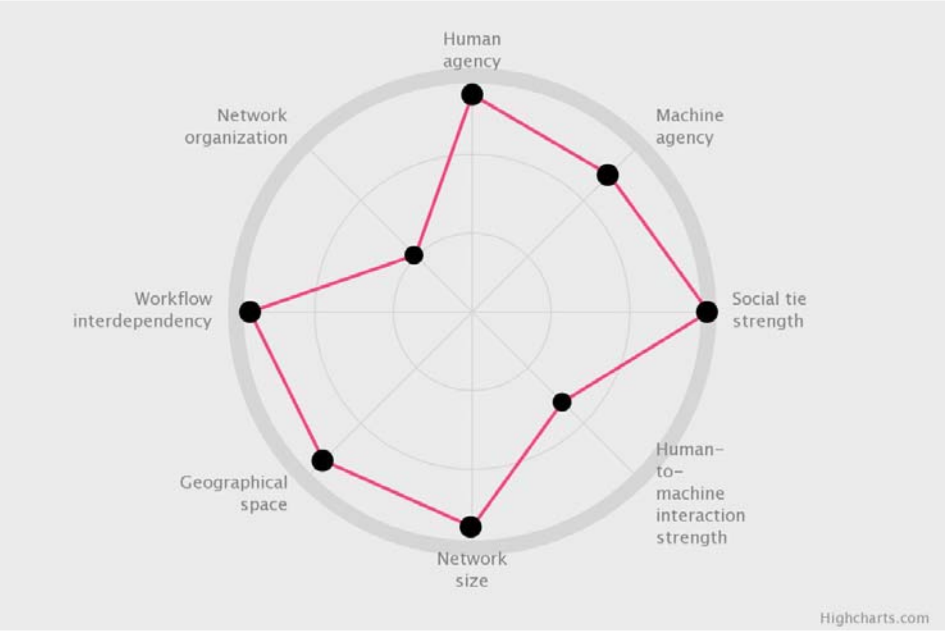


Figure 19 – Truly Media profile.

Human and machine agency: human agency is high, as journalists have a lot of freedom and flexibility in expressing their views regarding the verification of a UGC. Machine agency is also high. Machine actors are software agents that are called during a conflict in the verification process. Machine agents provide validation results that are taken into account by human users when performing their own assessments.

Social tie strength: this is high, as there is collaboration between journalists in the same team, across different groups in an organization and between verification teams of different organizations, who can interact with each other by providing comments and recommendations on the same content.

Human-to-machine interaction strength: this is intermediate; journalists have a constant interaction with the machine part of the system for setting preferences and content filters, creating topics, browsing the presented results, and providing feedback. Furthermore, journalists can follow the machine output for verification analysis, or ignore it and review UGC themselves.

Network size: As Truly Media is under development, the large size that is set in the profile refers to the potential use of the platform by a large number of users. Limitations to the expansion of the service can occur because of language, as Truly Media is currently offered only in English. In addition, although the system can support a large number of users, scalability issues (processing power, capacity, etc.) should be addressed when the number of users significantly increases.

Geographical space: Similarly to the network size, the geographical size is large, as Truly Media can be used anywhere in the world. Even though languages other than English are not currently supported, different media organisations from all over the world would be interested in participating in such a platform for verification purposes.

Workflow interdependence: this is high as journalists collaborate with both humans and machines to edit the verification results of a UGC. While the human agents act independently during the verification process, their actions are affected by the values of the other users.

Network organisation: The structure for publishing and organising content in Truly Media is bottom-up, in the sense, that the human users themselves provide input, which can be used to assess the main criterion, i.e. the validity of content. Although machine agents provide an automatic assessment and values for different metrics, even a single human user suffices to classify the content as verified or non-verified.

A network diagram for Truly Media is illustrated in the following figure, depicting the key actors we are interested in here, namely the journalists and the machines for verification analysis.

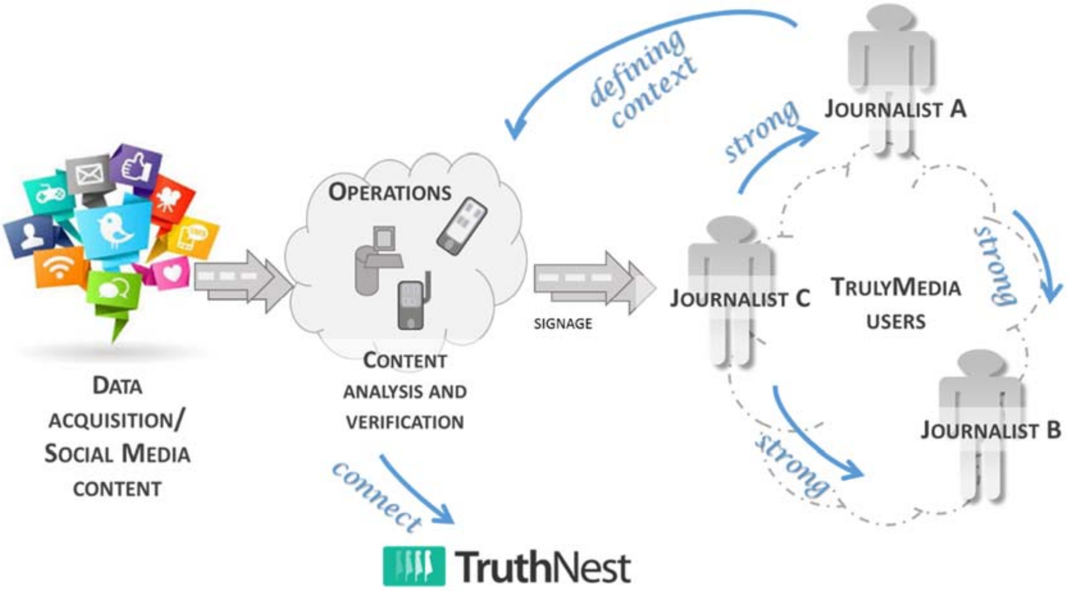


Figure 20 – Truly Media network diagram.

Each user of Truly Media logs into the system with their social media account (currently Twitter). The user can view all information about a post as received by social media (currently only Twitter posts are crawled through Truly Media). The user groups posts into collections and cooperates with other users on the verification of those posts. They can also view recommendations of other users about whether the post is verified or non-verified, as well as information about the users that performed these evaluations.

4.3.3 Simulation scenarios

As discussed above, we will address one of the objectives of Truly Media via simulation modelling, namely “helping users collaboratively verify UGC”. The collaboration is the key in the success of the verification process as people from different backgrounds, working remotely and often nonparallel must cross check a big amount of news and their related information. Using the HUMANE typology, we create a model that describes the verification workflow and in the simulation scenarios we examine how a conflict resolution tool would improve the overall agreement on the verified posts. In this way, the users are helped to reach a consensus and maintain high quality standards on the verification accuracy.

The simulation modelling will also help the Truly Media designers to decide on how to implement several functionalities. The Truly Media platform is still under development and new features are added every week. The output of the simulation will assist not only the software design process but also the organization of the verification activities. Should each verification team member work sequentially and verify the posts processed by another colleague previously, like proofreading, or would it be preferable to work in parallel and verify the same tweets concurrently? The management of the verification team can benefit from the simulation modelling because the actor’s interactions are analysed in detail and the performance indicators are revealed.

The main simulation scenario is that 20 users verify a corpus of 1.000 posts. Each user assigns a value to each post, whether they consider its information verified or not. The user decision depends on their action probabilities which are the same for all the users of the group. Users are divided on a group of experienced users and a group of occasional users. Initially, a ground truth is established where every tweet is assigned a value of verified or not and a user's weight is calculated. When a user verifies a post and its value differs from that of a previous user, they have the operation to cooperate with the other user. The possibility of collaborating and reaching a conclusion is estimated.

In the normal operation phase, if no agreement is reached, the conflict resolution tool is called. The tool proposes a verification value that is determined mainly on the users' profiles and the past verifications on that post. The user can follow the tools recommendation and all the verification activity on the post is recorded and visible to them.

The probabilities used in the simulations are the user action's related probabilities (accept, reject). These probabilities do not come from real data, but are set arbitrarily to correspond to the common editing policies (cooperative, authoritarian, unresponsive). Real Twitter accounts are used in the simulations that correspond to different types of users (e.g. experienced journalists, plain users). An experienced user is considered to have high credibility, has a long history of evaluating information and keeps an active list of sources that help him in the verification task. A plain user has no significant verification experience, his ability to assess the information correctly is limited and typically works under the supervision of another user.

As a performance metric, we measure the absolute value of agreements minus the number of disagreements for each post verified. Each post will be verified by 20 users. At the end of the process, we measure the agreements and disagreements for each post, their absolute difference.

If the difference is low or close to zero, the evaluations on a post are equally separated/ divided which is not ideal in terms of verification consensus. If the value is high, the evaluations tend to agree (whether accept or reject the verification) that is desirable in terms of users' evaluation agreement. For every tweet, we could measure how much the verifications converge:

$$|agreements - \ disagreements|$$

From a performance perspective, we are interested in calculating the related sum for the whole corpus of tweets:

$$\sum_{i=1}^{i=n} |agreements_i - disagreements_i|$$

The higher the value, the better. It should be demonstrated that with the use of the tool, less conflicts occur on average. We make the assumption that the user will follow the tool recommendation in all cases.

5 Analysis and implementation

In this section, we discuss data analysis work that has informed the modelling exercise as well as the model implementation.

5.1 Wikipedia

As noted above, to inform the modelling exercise for Wikipedia, further data analysis was needed. Based on this, we have proposed model extensions and discuss behavioural rules for the agents, and the simulation workflow.

5.1.1 Data analysis

As noted above, we have based this simulation modelling example on the Simple English[10] edition of Wikipedia. The dataset we base this work on has been created by previous researchers (Iñiguez et al., 2014) and made available online[11]. The dataset covers the activity of users and bots on Wikipedia articles in the period of 18/05/2001 to 17/10/2012, comprising 74,880 articles and 147,076 agents (25,037 registered users, 121,636 anonymous users, and 403 bots). Although the first human activity was recorded in May 2001, it took over 2 years before the first bot was introduced (11/10/2003).

5.1.1.1 Overview

The dataset we base all the analysis and modelling on is a simple flat text file listing articles and their respective activity log. Each activity entry contains a date time stamp, a flag to indicate if the entry was a revert (1) or not (0), an integer for the revision ID, and the ID of the agent. Below is an example entry for a single article:

```
Last_Friday_Night_(T.G.I.F)
^^^_2011-10-26T04:14:50Z 1 2 ArthurBot
^^^_2011-10-26T04:12:59Z 0 3 September_1988
^^^_2011-10-26T03:44:47Z 0 2 ArthurBot
^^^_2011-10-26T03:28:53Z 0 1 September_1988
```

In the above example, we see that ArthurBot has reverted an entry by September_1988, taking the article back to revision number 2. Note that at this point, this is not an edit war. However, should September_1988 chose to revert ArthurBot back, it would then become an edit war.

Due to the anonymous nature of Wikipedia, although registered users and bots have nick-names, there is a lot of personal information we cannot use, which the Core HMN Model accommodates. For example, we cannot set the age, gender and sexual orientation of the agents. We can make assumptions about the culture on the basis of the language edition that is used, but as we are scoping this proof of concept to a single language edition this property is not relevant. Further, we are unable to determine properties like self-efficacy, but we do address trust and reputation below.

[10] https://simple.wikipedia.org/
[11] http://wwm.phy.bme.hu/light.html

Although the dataset we can work with is simple, we can produce the meta-data that is of interest to simulation modelling. We can create a view of each agent, extrapolating when they joined (first time they contributed), when the left (last time they contributed), and accumulate the number of articles created, edits, reverts and edit wars they have been involved in. Based on this information, we can analyse and interpret patterns of behaviour of specific agents, such as whether they create a large number of articles and whether they engage in edit wars. We have also generated weekly snapshot information about the activities of the agents, which allows us to estimate probabilities of their ongoing activities within the simulation based on bootstrapping information from historical data.

We can also analyse the reverting and edit wars behaviour in more detail. We can extract across all the articles information about agents reverting each other, and whether the reverts were part of an edit war or not. We have also extracted summary data on edit wars specifically, including their start and end dates, how long they lasted for (in days), how many reverts the wars comprised, and which agents were involved in the respective edit wars.

This information help us determine both properties as well as behavioural rules that we can implement in the Wikipedia simulation model. Further details of the analysis leading to this is given below in the following sections. As part of the open access initiative in the project, we have provided all the data used for the simulation modelling online, accessible at Zenodo[12]: https://doi.org/10.5281/zenodo.573223

5.1.1.2 Agent activity

Analysing the activity of the agents, we observe that the lifetime of both human and machine agents varies significantly. In terms of the human agents, they are dominated by anonymous users. As anonymous activity logging is done by recording the user's IP address, this is not possible to conclusively associate this with a unique individual. That is, a single IP address could be representing the activity of multiple users, e.g., when behind an organisation's firewall. Similarly, a single individual may contribute from different locations and through different devices at different times, recorded as different IP addresses. As such, it is not surprising that we find that 87% of the human users are only active for a single day. For these reasons, we have aggregated the anonymous users in Wikipedia into a single representative agent in the simulation model.

The activity levels of agents (users and bots) vary significantly as well, and we clearly observe how some agents focus on specific tasks. While bots are configured for specific purposes, such as detecting vandalism, they naturally have a skewed activity profile towards reverting. However, we see such skewed profiles among human users as well. A few sample users are included in Table 2, below. For example, there are users who are active but never (or rarely) create new articles, such as CommonsDelinker and Creol. There are very active users, such as Nameless_User who do not revert or engage in edit wars much, compared with Razorflame, despite a similar number of edits. Finally, we also observe some users who have a high proportion of reverts and edit wars such as PseudoOne.

[12] The access to the data used for the Wikipedia simulation modelling is restricted by an embargo period until the end of December 2017.

However, we observe that such users do not have a considerable amount of activity before they are presumably detected as vandals and are blocked. In most cases, from what we observe, users like Club_Penguin are only active for a day. PseudoOne was active for approximately 5 months, though sharing a similar profile.

Table 2 – Sample users exhibiting different activity profiles.

Username	Number of articles	Number of edits	Number of reverts	Number of edit wars
Nameless_User	5109	14768	161	2
Razorflame	2033	17920	3757	125
CommonsDelinker	0	2985	20	0
Creol	345	29646	3449	41
PseudoOne	0	21	18	9
Club_Penguin	0	16	7	4

The observations above form a good illustration of the benefit of the agent-based modelling approach taken here, as each agent can be initialised according to their historical activity. As such, we reflect how some agent create more pages than others, etc. Furthermore, we also observe that the majority of contributions to Wikipedia are done by a small proportion of agents. For example, merely 0.1% of all users are responsible for all content contributed to the Simple English language edition of Wikipedia for the time period we have data for (18/05/2001 to 17/10/2012).

5.1.1.3 Edit wars behaviour

The total number of articles in the dataset for the Simple English language edition of Wikipedia is 79376. From this, 1508 are affected by edit wars[13]. Most articles have only a single edit war. However, there are pages in which several edit wars are going on (up to 13), which is depicted below in Table 3 (left). We also did an analysis of the number of times each agent engaged in edit wars. For example, there are 1270 agents who are involved only in one edit war and there is one agent (GoblinBot4) who is involved in 161 edit wars. We also observe that there are pairs of agents who repeatedly engage in edit wars with one another. As such, we can introduce this observation into the simulation model, by introducing a bias towards engaging in edit wars with agents there's a prior history of edit wars with. We can also extract information on which agents typically start the edit wars and who has the last revert (and, thus, we can say they finish the edit war).

[13] See Section 9.3 for details on how the edit wars were extracted.

Table 3 – Statistics for the number of articles that have 1 – 13 edit wars (left) and the number of pairs of agents that have 1 – 29 edit wars (right).

Number of Edit wars	*Number of Articles*
1	1177
2	211
3	52
4	30
5	10
6	10
7	6
8	4
9	4
10	1
11	1
12	1
13	1

Number of Edit Wars	*Number of pairs of agents*
1	1740
2	100
3	32
4	10
5	5
6	2
7	3
8	2
9	2
10	2
11	1
13	1
14	1
16	1
29	1

In terms of implementing behavioural rules enabling the emergence of edit wars, we also hypothesise that there are differences in edit war emergence depending on the type of agent. If we take Humans (H) and Bots (B), we have four possible scenarios.

- H2H – Human to Human
- H2B – Human to Bot
- B2H – Bot to Human
- B2B – Bot to Bot

In the above breakdown, the first letter in the three-letter abbreviation refers to the type of agent who did the initial revert. The third letter refers to the type of agent who was reverted. Once the agent who initially was reverted decided to revert back, this became an edit war. Further, we can break down the Human class into Registered (R) and Anonymous (A) Human users, leading to a breakdown of 9 different classes. An overview table is given below.

Table 4 – Overview of edit war classes

Agent class	Number of edit wars	Proportion of edit wars
B2B	**237**	**10.44%**
B2H	**290**	**12.77%**
B2A	195	8.59%
B2R	95	4.18%
H2B	**62**	**2.73%**
A2B	4	0.18%
R2B	58	2.55%
H2H	**1682**	**74.06%**

A2A	22	0.97%
A2R	172	7.57%
R2A	860	37.87%
R2R	628	27.65%
Grand Total	**2271**	**100.00%**

We observe that the majority of edit wars are between human users (H2H) – at 74.06%. Most of which are between Registered and Anonymous users (R2A), followed by wars between Registered users (R2R). In the former case, most of the edit wars are started when a Registered user reverts an Anonymous user (R2A). The reverse is rare.

Of the remaining 25.94% of the edit wars, only 2.73% of the edit wars are between Humans and Bots (H2B). That is, specifically where a Human has decided to revert a Bot, the Bots rarely revert back to start an edit war. However, nearly 13% of the edit wars occur when a Bot reverts a Human (mostly anonymous users). 10.44% of the edit wars are between Bots.

If we look further into the temporal aspects of edit wars, the table below shows statistics for the number of days edit wars last. Edit wars that last less than a day are indicated by zero, which was the case for the 4 edit wars between Anonymous users and Bots (A2B).

Table 5 – Duration of edit wars in days

Agent class	Average Duration	Max Duration	Sum Duration	% Sum Duration
B2B	**14.41772152**	**177**	**3417**	**34.33%**
B2H	**3.468965517**	**573**	**1006**	**10.11%**
B2A	3.969230769	573	774	7.78%
B2R	2.442105263	62	232	2.33%
H2B	**29.41935484**	**429**	**1824**	**18.33%**
A2B	0	0	0	0.00%
R2B	31.44827586	429	1824	18.33%
H2H	**2.202734839**	**327**	**3705**	**37.23%**
A2A	0.363636364	7	8	0.08%
A2R	0.755813953	31	130	1.31%
R2A	1.356976744	263	1167	11.73%
R2R	3.821656051	327	2400	24.12%
Grand Total	**4.38221048**	**573**	**9952**	**100.00%**

On average, edit wars last just over 4 days. The edit wars are generally longer between Humans (registered users) and Bots (R2B) – just over 30 days on average. The longest edit war between a human and bot is 429 days. Less so between humans (327 days). However, while the average duration between humans (H2H) is only 2.2 days, compared with 29 days between humans and bots (H2B) and 14 days between bots (B2B), we saw above that the vast majority (almost 75%) of the edit wars were between humans (H2H). Therefore, the accumulated duration of edit wars is significantly higher.

Table 6 – Number of reverts in edit wars.

Agent class	Average #reverts	Max #reverts	Sum #reverts	% reverts
B2B	**2.620253165**	**16**	**621**	**9.58%**
B2H	**2.806896552**	**58**	**814**	**12.56%**
B2A	2.687179487	15	524	8.09%
B2R	3.052631579	58	290	4.48%
H2B	**2.725806452**	**8**	**169**	**2.61%**
A2B	2	2	8	0.12%
R2B	2.775862069	8	161	2.48%
H2H	**2.898929845**	**35**	**4876**	**75.25%**
A2A	2.454545455	5	54	0.83%
A2R	2.325581395	7	400	6.17%
R2A	3.059302326	18	2631	40.60%
R2R	2.851910828	35	1791	27.64%
Grand Total	**2.85336856**	**58**	**6480**	**100.00%**

In terms of the number of reverts in edit wars, the table above presents summary statistics. We can observe that the average number of reverts in edit wars across the different agent classes is 2.85. On average, all agent classes remain close to this figure. However, there are outliers, which is shown by the maximum number of reverts figures. While the Bot to Human (B2H) edit wars have slightly less reverts on average (2.81), the maximum number of reverts in an edit war is 58. This does not, however, imply a long duration – as seen above (3.47 days on average compared with the global average of 4.38 and H2B at 29.42 days). Worth noting that there are more reverts between registered users and bots than for anonymous users.

The Human to Human (H2H) edit wars have slightly more reverts than the global average (2.89), but have less long-running edit wars (max 35). While we observed that the most edit wars were between Registered and Anonymous users (R2A), followed by wars between Registered users (R2R), the R2R edit wars are longer though comprise less reverts on average.

We can introduce these observations into the model, indicating a typical bias between types of agents engaging in edit wars, as well as the typical duration of edit wars between types of agents.

Reputation is a property of the core model that we explored. Wikipedia does not implement an explicit reputation mechanism, but users on Wikipedia have pages associated with their nick-names, which other users can view. On these pages, anybody can view, for example, the number of edits the particular user has made. The activity level of a user, thus, could give an impression of reputation. As such, we hypothesised that this would influence the number of edit wars an individual was in, i.e., users of a higher reputation (more edits) are in less edit wars. However, we found no direct correlation between users' edit and edit war history. For brevity sake, we omit the details of this analysis here, but we note that a possible reason for the lack of correlation may be that few users may look at the profile of other users who revert them, or if they do, there are other factors that may play a stronger role in assessing the reputation that are not available in the dataset we have worked on. Further, if an edit war is based on, for example, opposing views, which is something addressed by

(Iñiguez et al., 2014), the reputation (or perception thereof) may play an insignificant role. This is something that would require further investigation, but falls outside the scope of the proof of concept for the purposes here.

5.1.2 Model extensions

The extensions to the core model are shown below in Figure 21 (in grey). There are additional classes used in the implementation that are omitted as they are not of particular relevance here, e.g., for encapsulating information, collecting statistics or performing input/output operations. For the same reason, some properties of the classes are omitted in the discussions below.

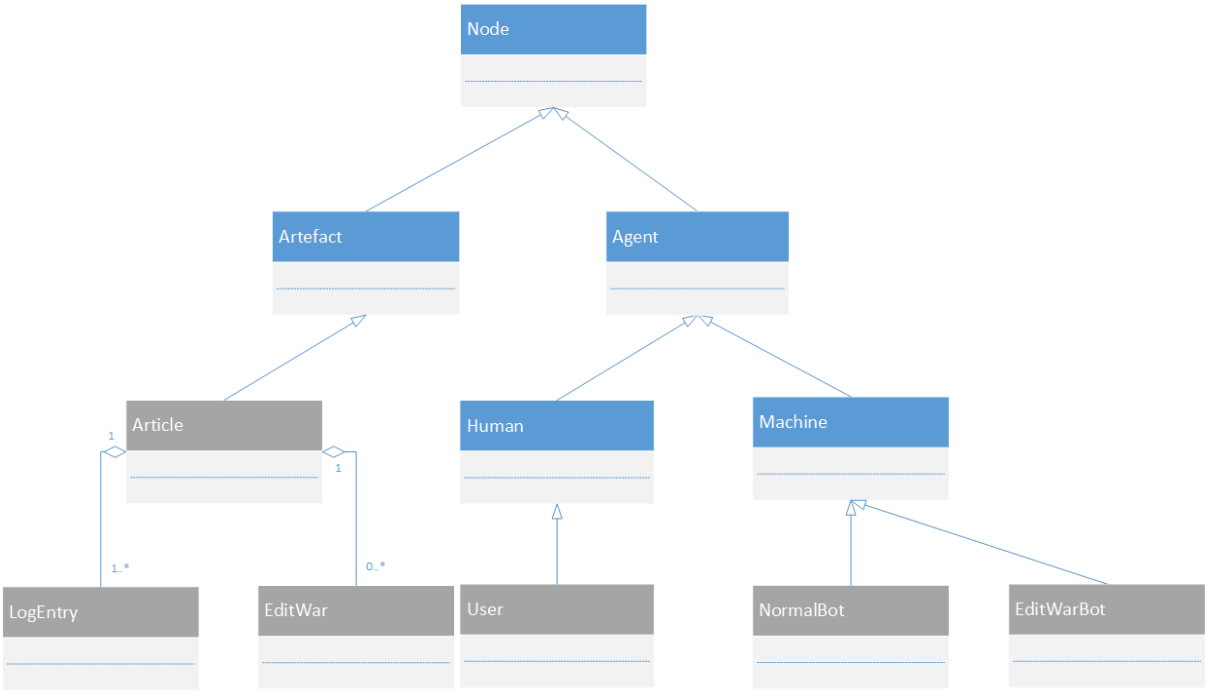


Figure 21 – Model extensions for Wikipedia.

The extensions made to the Core HMN Model are all part of the Node hierarchy. We have extended the Artefact class by introducing an Article class. We have extended the Human class with a Wikipedia User class (representing human contributors) and the Machine class with two bot classes – a NormalBot (reflecting the bots that currently exist in Wikipedia) and new bot we introduce here, EditWarBot, which has the capabilities of identifying edit wars and interacting with users involved in the edit wars to help facilitate their resolution.

We use the rest of the Core HMN Model as is. The Edge class is used to establish relationships between agents who revert each other. We also considered creating Edges between users who edited the same page. However, this was not required for the purposes of the model and just introduced unnecessarily scalability issues. For example, just the first six articles generated approximately 450,000 edges.

The properties of the classes introduced for the Wikipedia simulation are discussed in Section 9.1 (part of Appendix A). Here we will outline the classes in more general terms. The Article class reflects the article's name and contains a log of all the contributions to it. Each log entry is dated, indicating what type of action was performed (create, edit or revert) and which agent performed the action. The article also keeps track of edit wars taking (or have taken) place on it.

The User and NormalBot agents introduced in the Wikipedia model are nearly identical as humans and machines can do the same actions: creating new articles, editing existing articles, reverting contributions (from oneself or others), and taking part in edit wars. The only difference between the User and NormalBot classes is a flag in the User class to indicate whether the respective user is an anonymous user or not. Some of the key properties of these classes affecting their behaviour in the HMN are probabilities extracted from the historical data for creating, editing and reverting articles.

The complexity of these classes (agents) are encapsulated within the takeAction() method, implementing the behavioural rules discussed in Section 5.1.3. Edit wars are modelled as an emergent phenomenon that occur when mutual reverting takes place between two agents.

5.1.3 Behavioural rules

In our model, users and (normal) bots are able to perform three different core actions: (1) creating an article, (2) editing an article, and (3) reverting an article. From the latter, there is also an opportunity to engage in edit wars. In addition to this, given our scenario outlined above in Section 4.2.3, we introduce a new bot with increased agency, capable of monitoring edit wars and notifying the agents involved in these wars. This is a non-punitive approach to reducing the duration of edit wars, which is motivated by increasing the reliability and quality of information in Wikipedia (as discussed in Section 4.2.1).

5.1.3.1 Creating and editing articles

Based on the historical activity levels of each agent, broken down by weekly snapshots, we can calculate rates of creating articles or editing articles, scaled to the discrete time intervals of the simulation. As discussed below in Section 5.1.4 each "tick" of the simulation is an hour, at which point agents have the opportunity to take actions.

We have kept the model simple by calculating rates based on an average activity level from the past *N* number of snapshots. We can configure *N* to determine a value that the best approximates the activity observed in the dataset. In preliminary experiments, we found *N = 10* to be a reasonable estimate. That is, we consider the activity of each agent over the past ten weeks in order to determine what they will do in the future during the simulation.

Again, keeping the model simple, when an agent edits an article, we randomly select an article from the HMN to be edited.

5.1.3.2 Reverting and edit war

For modelling edit wars we have considered the interaction between different types of agents, statistical data extracted from Wikipedia logs and stochastic variables. In terms of data required for the simulation each agent maintains information about:

- List of reverts against the agent
- List of edit wars that the given agent is involved in
- Notifications received from the *Edit War Bot*

The following we establish the rules for handling reverts and edit wars.

1) *Rules for handling reverts*

In the case of a revert, the agent needs to decide whether to accept the given revert or to re-revert (see Figure 22). Re-reverting an article creates an edit war that will continue as long as the end date is not set.

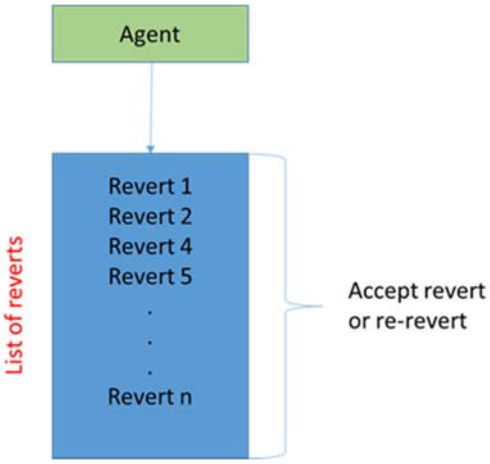


Figure 22 – Dealing with individual reverts

When we deal with reverts we consider the following options:

a) *If the revert is not part of edit war*

The question is to decide what percentage of reverts will be re-reverted, i.e., turning into an edit war? We have looked at the historical data collected over ten years and calculated the ratio of the number of edit wars and reverts. This is an approximation, but provides some indication on what percentage of reverts can we expect to become part of an edit war. For example, anonymous agents are the more likely to enter into an edit war compared to the registered agents (see Table 7).

Table 7- Percentage of reverts that are part of an edit war

	Number of agents	Reverts	Edit wars	Edit wars /Reverts
Anonymous agents	121637	10481	1275	0.12
Bots	403	22675	826	0.036
Registered agents	25034	113752	2441	0.021

The agent may decide to re-revert; in this case enters into an edit war. The ratio of edit wars and reverts done by anonymous and registered agents is about 14%. The logic for creating new edit wars can be described by the following pseudo-code:

```
if (revert part not of edit war)
probabilityOfEditWar = calculate the probability for the revert to become part of edit war
    if (randomNumber <= probabilityOfEditWar){
        Create edit war
```

b) *If a revert is part of an ongoing Edit War*

If a revert is part of an ongoing edit war, then the agent can either continue the edit war by re-reverting again or end the edit war by setting the end date according to the last revert. Here we need to compare the duration of current edit war and the average duration of edit war characteristic for a type of interaction (see Table 5):

```
find the edit war for the current revert
get the duration of edit war
if(B2B agents)
        if(durationOfEditWar<= durationB2B)
                Re-revert
        else
                Finish edit war
if(B2H agents)
        if(durationOfEditWar <= durationB2H)
                Re-revert
        else
                Finish edit war
if(H2B agents){
        if(durationOfEditWar <= durationH2B)
                Re-revert
        else
                Finish edit war
if(H2H agents){
        if(durationOfEditWar <= durationH2H)
                Re-revert
        else
                Finish edit war
```

2) *Finishing the agent's own current edit wars*

Each agent maintains a list of current edit wars; the question is when to end these edit wars. The options that we consider are as follows:

a) We may assume that whenever an agent logs in and performs some activity (i.e. editing, creating a new page or reverting) the agent also checks the list of own edits wars and finishing them all by setting the final date.
b) The other option is that the list of edit wars is cleaned up depending on a stochastic variable. For example, we may assume that with a probability of 0.1 the agent will clean up own edit wars.

5.1.3.3 EditWarBot intervention

We have introduced an *EditWarBot* class, as seen above in Section 5.1.2. This class is a stripped-down version of the NormalBot class. This bot cannot create, edit or revert content in Wikipedia. Instead, it monitors current edit wars in the network. If an edit war duration exceeds a threshold *T*, the bot sends a notification to the agents involved in the war simulating a message with instructions to end the edit war. As a policy for the *EditWarBot* we have set the threshold *T* as 4 days based on the observation that edit wars on average are resolved after 4.38 days (see Section 5.1.1.3, above). This policy is motivated by aiming to allow agents the time to resolve the edit wars between themselves before taking intervention.

When the agents receive a notification from the bot to end an edit war, there are several modelling options to consider:

a) In each tick the agent check if there are any edit wars in the notification list and ends them immediately. This is the case when an agent unconditionally follows the instructions from the *EditWarBot*.
b) The other case is when the agent does not immediately end the edit war but only with some probability.

5.1.4 Simulation workflow

A simulation model has been implemented in Java following a sequential discrete event based approach. The simulation first undergoes a bootstrapping phase, reading in data to initialise the population of both humans (users) and machines (bots) in the HMN, as well as artefacts (Wikipedia articles). Based on the historical data properties of the agents and articles are set, reproducing the state of the HMN at a specific point in time. From this point in time, a simulation starts, as per the following pseudo code:

```
snapshotStartDate = simulationStartDate
snapshotEndDate = snapshotStartDate + snapshotDuration (1 week)
currentDate = snapshotStartDate
do {
       for each agent {
             agent.takeAction()
       }
       currentDate + timeIncrement (1 hour)
       if currentDate > snapshotEndDate
             collect and save statistics
             snapshotStartDate + snapshotDuration (1 week)
             snapshotEndDate + snapshotDuration (1 week)
} while (currentDate < simulationEndDate)
```

As discussed in Section 5.1.2, the actions the agents can take include: create a new article, edit an article, and revert a contribution. The latter may form part of an edit war, but we emphasize here that this is an emergent phenomenon that can happen according to the behavioural rules outlined in Section 5.1.3.

The above simulation workflow is repeated multiple times due to the stochastic nature of the agents so that statistical figures can be extracted, such as the mean and standard deviation. As seen above in the pseudo code, we collect and save statistics for regular snapshot periods, which we have set to 1 week. Once the simulation end data has been reached, statistics are calculated and exported to disk.

5.2 Truly Media

The Truly Media use case is modelled in terms of the HUMAN typology. The core classes Human and Machine are used to represent the users and the tool respectively. To represent specific properties of the Truly Media use case, the TwitterUser class has been defined that extends the Human class. It contains fields such as a TruthNest contributor score, a belief vector that describes the acceptance/ reject probabilities of the user, a list of actions performed by the user, a list of its relevant posts and date fields. The takeAction method is overridden by the TwitterUser class.

If no users exist in the database (because the simulation is run for the first time), the users are created from a predefined list and we search Twitter for their posts. We store their posts (if they were found) and update the database in any subsequent simulation run. The point is to have user posts available (collected in previous runs) to be able to calculate parameters of the simulation.

The Machine class is extended by the Bot class. The Bot class overrides the takeAction function where the conflict resolution tool is implemented. The utilization field of the Machine class is interpreted as the frequency of the tool usage.

The simulation code defines an HMN class object. In this object are added/set the agents which are the TwitterUser class objects and the Bot class object that represents the conflict resolution tool. Furthermore, Edges are defined between the users and are added to the network. Since all the users can communicate in the platform, a number of $\frac{n*(n-1)}{2}$ edges are created. For the 20 users of the simulation, this corresponds to 190 edges. The Edges also define a Connection object and its trust field is set with the probability that the two users will collaborate and agree.

The class VerificationForm extends the Artefact class and represents a form that contains the posts verification data. Each post is assigned a ground truth in the ground truth phase, so a relevant field is defined. A list field that represents the series of user actions on this form is also defined. The VerificationForm object is added to the network object.

During the simulation, the users verify the posts by calling the takeAction method. The probabilities of the user actions are calculated and according to them, the user decision is formulated. The HMN object contains all the information about the simulation. Additional classes have been added

(SimulationResult, Statistics) to save the simulation result in the database and present it. More information on the custom classes is provided in Appendix B.

The simulation implementation is a Spring Boot Java Maven project. The motive behind using Spring Boot framework was the ease to create a stand-alone application and the embedded Tomcat server it offers. In this way, it would be easier to design a front-end page (e.g. animation) that will show the simulation execution or results. This has not been implemented since our focus was on using the HUMANE typology, but it could be done in the future.

The basic workflow of the simulation is that 20 users verify 1000 posts without and with the use of the conflict resolution tool. The conflict resolution tool is configured with three settings. The simulation execution is performed in two phases. In the first phase, the “ground truth” phase, a user weight is calculated and each post is assigned a value whether it is verified or not (ground truth). The user weight is needed for the second phase, where it is being used by the conflict resolution tool. Furthermore, for each user, its TruthNest contributor score is calculated, as well as the age/sex (through Stylometry module). All this information is stored in a Mongo database.

In the second phase, the users verify the posts but with the use of the conflict resolution tool. If a user has a conflict with a previous user’s verification of a post, the probability of reaching an agreement is computed and an action is taken. This is implemented with a sample of an enumerated distribution that has been set up with the respective probabilities. If the output of the action is that the users still disagree, the conflict resolution tool is called. According to its configuration, the results vary. A performance metric is computed for each configuration and is saved in the database.

The data used have been 100 posts from 10 known media agencies. They were collected via the Twitter REST API, formulating a corpus of 1000 tweets. The users of the simulation are retrieved from the TruthNest platform. Half of them are experienced, renowned journalists with active social media presence and the other half are plain users with no remarkable social media activity. In this way, we have a balanced group of users who are representative of the Truly Media users.

The Truly Media platform is in beta version and no significant user data (e.g. user activity logs) exist. If historical data was available, we could analyse them and use more features of the HUMANE typology. We had to select the features of the HUMANE typology that were appropriate for our scenario given the absence of historical data. For the lack of simplicity, a detailed, fine grained representation of the user interactions has been avoided. That would also complicate the implementation and probably slow down the execution time. Our focus has been to use the HUMANE typology classes in a clear way and present a consistent model.

The simulation implementation provides logging, exception handling and execution performance information. The logging appears on the console output and on a log file. The code can be easily debugged and the information about the processing progress and steps is displayed during the simulation. The total execution time is also presented after every execution. The simulation results are stored in a database to be retrievable and compared.

One approach to speed up the simulation execution is to avoid storing intermediate results in the database. Especially during the verification process of 1000 forms, the saving of the object representing the form requires a significant time. The objects are saved in the database only when the simulation is completed and the results have been generated. Therefore, the simulation interruption should be avoided.

After each simulation completion, we do not remove the Twitter users from the database. Only reset the user specific fields that are modified after each run (e.g. user actions list). The TruthNest score for a user will not be modified significantly during a day. Calling the external services (TruthNest, Stylometry) in every execution delays the simulation. Furthermore, we do not remove the saved posts of the users after each execution. The TruthNest and Stylometry services require at least one previous post and we should keep them stored.

5.2.1 Ground Truth Case

In the basic scenario, the “ground truth” case, the user’s actions are determined by the user’s action probabilities (belief vector), without any tool intervention. For example, a user might have 60% probability to accept an evaluation and 40% probability to reject it. When applying verifications on a corpus of 1000 tweets, on average he will accept 600 evaluations and reject 400 evaluations. Through collaboration with the disagreeing users, the final number of rejections will be lower.

In the simulations, the 20 Truly Media users are divided equally into two groups, the plain users and the expert users. Each group has the same probability to accept/ reject the previous evaluations (belief vector). The accept probability is higher for the expert’s users group, because they are more experienced in the verification process, achieving less conflicts. The users verify all the tweets and the output of the process is used to compute the user weights, used in the normal operation configurations.

5.2.2 Normal operation phase

In the normal operation phase, the user can also collaborate with the other users, with whom he disagrees. We calculate the probability that the users will finally agree. In case of disagreement, the verification tool is called and a verification value is proposed. In order to compute this probability, it is needed to prior determine the past activity of the users in the ground truth case.

The conflict resolution tool will be configured in three settings which will be compared. In general, each post has been assigned a value during the ground truth case (verified/ no verified). When the tool is called, it computes the majorities of the verifications for a post and proposes a verification value. There are different ways in which the majorities are calculated.

5.2.2.1 Majority based decision

The simplest tool setting is to decide based on the majority of the verifications in the ground truth phase. Let’s define with $h(i, j)$ the assessment of user i for post j.

$$h(i,j) = \begin{cases} 1, user\ i\ regards\ post\ j\ as\ verified \\ 0, user\ i\ regards\ post\ j\ as\ unverified \end{cases}$$

The plain majority vote is the percentage of the users who consider post j as verified:

$$\frac{\sum_{i=1}^{N} h(i,j)}{N} \cdot 100\%$$

If the percentage is greater than 50%, the tool will propose the post j as verified. For example, if 12 users regard it as verified from the total of 20 users who verified it, the tool will propose the value verified to the user (12/20 > 0.50). This is an arbitrary choice that can be modified. Especially the cases of marginal majority could be the focus of further experimentation.

5.2.2.2 Weighted majority based decision

In the previous setting, we considered that each user has the same validity, therefore the decision of each user has the same weight. However, this is not very consistent with the actual case. We could find the users who have a greater validity and assign their decision a greater weight. One approach would be to measure a user "correctness" weight during the ground truth case:

$$a_i = \frac{number\ of\ evaluations\ of\ user\ i\ in\ accordance\ with\ the\ ground\ truth}{total\ number\ of\ evaluations\ of\ user\ i}$$

In this way, the users with the most correct evaluations will have a greater user weight related to the users with less correct evaluations. The weight will be a number in the internal of [0, 1].

In this case, the assessment function $h(i,j)$ where user i verifies post j is defined as:

$$h(i,j) = \begin{cases} a_i, user\ i\ regards\ post\ j\ as\ verified \\ -a_i, user\ i\ regards\ post\ j\ as\ unverified \end{cases}$$

The weighted majority can be calculated as the sum of the assessment function values:

$$\sum\nolimits_{i=1}^{N} a_i$$

If this is positive, the users who regard the post as verified have a greater weight than the users who regard the post as unverified and the tool will assign the verified value. In case of zero value we regard it as verified.

5.2.2.3 TruthNest weighted majority based decision

Similar to the previous setting, we could calculate the users "correctness" weight by using the TruthNest contributor score. The TruthNest contributor score is a number between [0, 4] that is formulated by taking into consideration the Twitter social media profile of the user. The higher, the more credible the user is considered. However, in order to compute this score, we attempt to retrieve a tweet of the user through the Twitter REST API. If he has not posted a tweet the last 7 days, no tweet will be found and the TruthNest score cannot be calculated. In this case, we assign a default TruthNest score that corresponds to a user with seldom social media activity and a few followers.

In this case, the assessment function $h(i,j)$ where user i verifies post j is defined as:

$$h(i,j) = \begin{cases} score_i, user\ i\ regards\ post\ j\ as\ verified \\ -score_i, user\ i\ regards\ post\ j\ as\ unverified \end{cases}$$

The TruthNest weighted majority can be calculated as the sum of the assessment function values:

$$\sum_{i=1}^{N} score_i$$

If this is positive, the users who regard the post as verified have a greater weight than the users who regard the post as unverified and the tool will assign the verified value. In case of zero value we regard it as verified.

5.2.3 Probabilities estimation

During the simulation, the user actions are determined by respective probability functions. There are three probability functions defined. The probability that a user will verify/no verify a post and the probability that a user will collaborate with the users with whom he disagrees and reach a consensus. These probabilities are explained in the following sections.

5.2.3.1 User actions probability

The user actions probabilities are set according to the group where the user belongs and are fixed through the simulation. The group of the experienced users has greater acceptance probability than the group of the plain users. This is to indicate that an experienced user is more likely to interact with the other users in the platform and agree on a verification value, in relation to a less experienced user. Furthermore, other users tend to agree with him if they appreciate his work and consider him credible. In the scenarios executed, the difference in the acceptance rate is 10% (60% acceptance probability for plain users, 70% for experienced users).

5.2.3.2 Consensus probability between users

During the verification process, when a user disagrees with the verification value of a post, he can communicate with the users with whom he disagrees. In a real case scenario, this would be done either via chatting, especially if the users with whom he disagrees are logged in and available in the platform or via a mail message. Truly Media offers both collaboration mechanisms. The users can cooperate and agree on the verification value of the post. If they do not agree, the platform will use the conflict resolution tool to propose a verification value.

The most common reason behind the disagreements are different opinions on the verification of a post. In some cases, collaboration is not possible because not all users are connected in the platform. Another common case is the lack of adequate information to justify the verification decision. The Truly Media users fill forms and share their sources as part of the verification process. If a decision is not supported with relevant sources, other users might reject it.

We can model the probability that a user will collaborate with the users with whom he disagrees and reach an agreement. This probability depends on the following factors:

- The personal characteristics of the users, how possible it would be for a user to follow a certain user's judgment. We will use the respective property of the Human class (trust).

- The availability of a user in the platform. Truly Media offers asynchronous verification and all users are not logged in at the same time. Certain users might not be reachable at a certain time.

Considering two users (i, j) the two probabilities could be described at a higher level as:

$$Trust(i, j) * Availability(i, j)$$

The trust between two users could be determined by examining the past occurrences of agreement between these users. If no data is available, we could set a default value (e.g. 80% trust probability). A relative high value is expected because most Truly Media users are colleagues who belong to the same organization. The modelling approach would allow us to claim that different people will have different levels of trust. We could model how trustworthy the users are and their culture in the HUMANE typology. Currently the trust default value modelling is left as future work.

$$Trust(i, j) = \begin{cases} \dfrac{agreements}{agreements + disagreements} \\ 0{,}80\ (no\ data\ available) \end{cases}$$

Since the verification tasks are typically performed by users who belong to corporate verification teams and operate in an organized fashion, there is a high probability that all members are available at the same time. Therefore, we could set the availability probability to a high value e.g. 90%. It should be stated that the availability probability can be easily computed, but is also left as future work.

$$Availability(i, j) = 0{,}9$$

This probability can be computed if we monitor the date of the verification values entered by the users (when they entered the values) and the time they were connected in the platform. We can calculate how many of the users (that performed a verification) were concurrently connected in the platform.

Finally, if we take under consideration the number k of disagreeing users, the probability of reaching an agreement is formulated as:

$$P(agreement\ of\ k\ users) = Trust(1,2) * \ldots * Trust(k-1, k) * Availability^{k-1}, k > 1$$

6 Results and discussion

In this section, we discuss the results from the simulation models described above for the Wikipedia and Truly Media HMNs.

6.1 Wikipedia

For the Wikipedia case study, we simulate the baseline and the edit war bot scenarios. The baseline simulation allows to validate the results produced by the model against the data extracted from Wikipedia logs. Once the model is validated we can introduce a special bot called *EditWarBot* that monitors edit wars and also sends out notifications to agents involved in edit wars. The purpose of the simulation is to investigate whether the introduction of *EditWarBot* can affect the duration of edit wars.

The KPIs that we are interested in are the duration and number of reverts in edit wars. The other indicators are the number of new articles created, edits, reverts and edit wars. These indicators are used for checking the accuracy of simulation model.

The key parameters of simulation configuration are:

- Start date of simulation: 04/04/2011
- Number of simulation days: 28
- Duration of snapshots: 7 days
- Number of snapshots used for calculating probabilities: 10
- Number of simulation runs: 100
- Agent activity threshold: 30 days
- Aggregating anonymous users: y/n
- Presence of Edit War Bot: y/n
- Edit war duration threshold: 4

One of the important decisions for Wikipedia simulations is to aggregate the anonymous agents into a single entity and focus on the registered agents. There are several reasons supporting this decision, such as:

a) Anonymous agents are usually short lived, less than a day
b) The same agent might be behind different IP addresses
c) There can also be several agents or even an organisation behind a single IP address
d) By aggregating anonymous agents, we can keep the number of agents fixed and also speed up the simulations

6.1.1 Performance

As noted above, there are a large number of actors and artefacts in Wikipedia, which presents a performance and scalability challenge. We already discussed that it was not scalable or necessary to generated edges between all agents who had edited an article. However, we generate edges between agents who have reverted each other (and, thus, also who have been in edit wars with each other). To address the performance aspects of the simulation, we explored some model modifications.

The first modification was to aggregate all anonymous users into a single, representative agent. The second modification was to filter out any agents who have not been active for a certain period, as well as articles that had no activity in the same period. For the latter case, please note that for any agents initially excluded who had contributed to an article that somebody else had been active on within the set period, they are re-introduced.

The performance data reported on below in Table 8 are from 10 simulation runs, for 4 weekly snapshots, starting 03/01/2011. This was executed on a Dell Latitude E5450 with an Intel Core i5-5200U 2.20GHz CPU, 512GB 7200 RPM hard drive and 16GB RAM, running Windows 10.

Table 8 – Comparison of model modifications on performance.

	No modifications	Aggregate anonymous users	Aggregate + filtering (180 days)	Aggregate + filtering (30 days)
# agents	147076	25440	7000	5229
# articles	74880	74880	63524	59451
# edges	78208	15258	1832	381
Bootstrap time	00:00:30	00:00:15	00:00:15	00:00:13
Simulation time	00:00:45	00:00:08	00:00:02	00:00:02
# sim articles (mean | error)	136 | 0.04	168 | 0.18	158 | 0.11	150 | 0.06
# sim edits (mean | error)	5782 | 0.26	6602 | 0.24	6576 | 0.22	6436 | 0.18
# sim reverts (mean | error)	427 | 0.18	453 | 0.12	447 | 0.11	451 | 0.00

The performance gains in aggregating the anonymous users is very significant; reducing the bootstrapping time by 50% and the simulation time by 82%. While the bootstrapping time remains more or less the same by filtering out agents and articles who have not been active within a certain period (displaying 180 and 30 days above), the simulation time is further reduced to merely 2 seconds. These performance gains can be easily understood by observing how the number of agents and edges reduce significantly in both cases.

In terms of the simulation results, we observe less relative error when aggregating anonymous users, except for the creation of new articles. As our focus is on reverting behaviour leading to edit wars, this is more important. However, said that, when we filter out inactive agents and articles for which there has not been any activity for the past 30 days, the results for the number of new articles is nearly the same 4% error versus 6%.

6.1.2 Ground truth

Before we can run and analyse simulation results for making changes to Wikipedia by introducing the *EditWarBot* discussed above, we need to determine if the model of Wikipedia without modifications is accurate. This is the so called "ground truth", for which the objective is check whether the modelling approximations and behavioural rules driving the agents' activities reflect the observations in the historical data. To do this, we use a part of the dataset to bootstrap the simulation (initialising the population of agents and their properties) and then run a simulation within a time period immediately after for which we still have historical data so we can make a comparison of the results. That is, we bootstrap the simulation with data from time t_0 to t_1, and then simulate from time t_1 to t_2.

For the simulation we have selected a four week period, starting on Monday 4th April 2011. The historical data for the four weeks is provided below in Table 9, as well as the four weeks prior to the simulation period.

Table 9 - Data extracted from Wikipedia logs

Simulation period	Snapshot start date	Articles	Edits	Reverts	Edit wars
Data before the simulation snapshot	**07/03/2011**	224	8567	623	13
	14/03/2011	183	8051	573	13
	21/03/2011	174	8150	628	14
	28/03/2011	154	7344	691	20
Data for the simulation snapshot	**04/04/2011**	177	10839	637	19
	11/04/2011	198	9487	542	15
	18/04/2011	214	12071	462	17
	25/04/2011	231	8061	469	18

The results produced by the model are presented in Table 10. The configuration settings for the simulation as follows:

a) The ten previous weeks were used for bootstrapping, i.e., to calculate the probabilities for agent's actions such as creating new articles, editing and reverting existing articles.
b) The simulation for each week was repeated 1000 times and the relevant statistics calculated.

Table 10 – Predictions produced by the model

Snapshot start date	Number of new articles	Number of edits	Number of reverts	Number of edit wars	Mean edit wars reverts	Mean edit wars duration
04/04/2011	204	7480	697	19	8	70.23
11/04/2011	203	7479	590	18	9	55.29
18/04/2011	204	7477	590	16	10	45.79
25/04/2011	204	7477	590	16	10	38.86

The relative error between the data extracted from Wikipedia logs and the result produced by the simulation model is provided below in Table 11.

Table 11 - Relative error

Snapshot start date	Number of new articles	Number of edits	Number of reverts	Number of edit wars
04/04/2011	0.15	0.31	0.09	0.00
11/04/2011	0.03	0.21	0.09	0.17
18/04/2011	0.05	0.38	0.28	0.04
25/04/2011	0.12	0.07	0.26	0.13

The relative error (see Table 11) between the data extracted from Wikipedia logs (see Table 9) and the results produced by the model (see Table 10) varies depending on the type of activity and the weekly snapshot periods. We were able to simulate the emergence of edit wars with a very high accuracy. On average, we achieve a 91.5% accuracy in the results for the emergence of edit wars. In one snapshot period it was even 100%.

In case of number of edits and reverts the relative error for two weekly snapshot is around 30%. The main reason is that in the model we approximate the agents' behaviour based on their past actions. Provided that the data changes "smoothly" this approximation works well, however in case of spiky function this approach is less accurate. The data for the period before and after the simulation snapshot in Table 9 clearly indicates that the numbers for edits and reverts are not changing smoothly, hence the error of approximation.

6.1.3 Increasing the machine agency

In this section we analyse how the introduction of the *EditWarBot* impacts the duration of edits wars in terms of number of reverts and days. In regard to the *EditWarBot* we need to consider the following factors:

a) *Threshold for notifying the agents about edit wars* – for the simulation we set the threshold to 4 days. This is the average duration of edit wars extracted from the statistical data. If the edit war goes on longer than 4 days the bot sends a notification to the waring agents to end the edit war. Using the same threshold for all edit wars is an approximation, for the future work we could consider to differentiate between edit wars and set different thresholds.

b) Reactions of agents' influences the rate at which edit wars are resolved. In this case various scenarios may be considered:
   - The agent after receiving a notification immediately finishes all edit wars in which the agent is involved in.
   - The agent finishes the edit war only with some probability. We explore this option.

For describing the behaviour of agents in regard to the notifications from *EditWarBot* we have adopted the following approximation*:*

> Whenever an agent is logged in, they check whether there is a notification from *EditWarBot*. In case there is a notification, the agent ends the edit war in which the agent is involved in with a probability of 10%.

We ran a simulation over the same time period as the ground truth, above. The simulation for each snapshot was repeated 1000 times and the averages calculated. The results obtained by running the *EditWarBot* is presented in Table 12.

Table 12 – Results produced by the model when the EditWarBot is introduced

Snapshot start date	Number of new articles	Number of edits	Number of reverts	Total edit wars	Mean edit wars reverts	Mean edit wars duration
04/04/2011	204	7477	698	19	8	69.24
11/04/2011	204	7476	591	11	7	6.75
18/04/2011	203	7476	591	7	7	7.67
25/04/2011	204	7478	590	6	6	8.08

We see a reduction in the duration of edit wars in terms of both reverts and days. In the first snapshot, however, there is little difference whether the *EditWarBot* is running or not. The reason for this observation is that there are five long running edit wars (see Table 13) at the beginning of the simulation. In particular, two edit wars have been going on for a long time: 70 and 476 days, which increase the mean edit wars duration significantly. With the *EditWarBot*, they are typically finished within the first snapshot, so their duration is accounted similarly to that of the ground truth case.

Table 13 – Long running edit wars before the start of simulation snapshot

Article name	Start date	Duration(days)	Reverts
Car	13.12.2009	476	4
Sun_Microsystems	23.03.2011	11	2
Charles_IV_of_Spain	22.03.2011	12	2
Charlotte's_Web_(1973_movie)	23.01.2011	70	2
Pervez_Musharraf	03.03.2011	31	11

For each weekly snapshot we have investigated the statistical significance of the results when introducing the *EditWarBot*. We do this on both the number of reverts (see Table 14) and duration (see Table 15) of edit wars. We have used the 2-tailed student t-test (alpha=0.05) according to which if $t_{statistic} > t_{critical}$ then we reject the null hypothesis of there being no significant difference.

Table 14 – Statistical significance of results with the EditWarBot on the number of reverts in edit wars

Snapshot	$t_{statistic}$	$t_{critical}$	Difference between data
04/04/2011	0.50	1.96	not significant
11/04/2011	17.81	1.96	significant
18/04/2011	25.76	1.96	significant
25/04/2011	31.46	1.96	significant

Table 15 – Statistical significance of results with the EditWarBot on the duration of edit wars

Snapshot	$t_{statistic}$	$t_{critical}$	Difference between data
04/04/2011	0.66	1.96	not significant
11/04/2011	46.29	1.96	significant
18/04/2011	37.39	1.96	significant
25/04/2011	36.74	1.96	significant

For the first week there is no significant difference between the data measured with or without running *EditWarBot*. In the snapshot for the following week the duration of edit wars decreases because the long running edit wars are finished.

6.2 Truly Media

The performance metric used in Truly Media simulation scenario is the total absolute difference of the agreements minus the number of disagreements for each post verified.

$$\sum_{i=1}^{i=n} |agreements_i - disagreements_i|$$

As explained, the higher the sum the more the evaluations converge on a verification value. We compare the ground truth (no usage of the tool) with the conflict resolution tools different settings. In the simulations, we use different values for the agreement probability, e.g. the probability that the users will reach an agreement via collaboration if they disagree. If agreement is reached, the tool is not called in the normal operation phase.

The simulations have been run 100 times for a certain agreement probability. The minimum, maximum, mean and standard deviation of the runs in presented in the following tables. The values of the agreement probability are 0.50, 0.60, 0.70, 0.80 and 0.90. By focusing on the maximum mean we can find the tool setting that was the most appropriate for the specific setting.

	Min	Max	Mean	Standard deviation
Majority score	17032	17736	17404	163
Weighted majority score	17060	17946	17445	174
TruthNest majority score	16966	17796	17385	166
Ground truth score	11408	11656	11572	77

Table 16 – Scores with agreement probability 0.50

These are the simulation results for agreement probability 0.60:

	Min	Max	Mean	Standard deviation
Majority score	17274	18012	17670	144
Weighted majority score	17086	17960	17644	145
TruthNest majority score	17382	18078	17684	146
Ground truth score	12670	13198	12945	148

Table 17 – Scores with agreement probability 0.60

These are the simulation results for agreement probability 0.70:

	Min	Max	Mean	Standard deviation
Majority score	17648	18352	18016	151
Weighted majority score	17694	18280	18024	136
TruthNest majority score	17670	18388	17975	143
Ground truth score	14148	14624	14263	132

Table 18 – Scores with agreement probability 0.70

These are the simulation results for agreement probability 0.80:

	Min	Max	Mean	Standard deviation
Majority score	17934	18666	18339	136
Weighted majority score	18050	18672	18333	132
TruthNest majority score	17994	18672	18351	129
Ground truth score	15666	16060	15901	114

Table 19 – Scores with agreement probability 0.80

These are the simulation results for agreement probability 0.90:

	Min	Max	Mean	Standard deviation

Majority score	18336	19008	18666	115
Weighted majority score	18342	18932	18655	124
TruthNest majority score	18302	18910	18678	124
Ground truth score	17446	17696	17545	90

Table 20 – Scores with agreement probability 0.90

Both the tool configurations give comparable results regarding the tools efficiency, as the mean score reveals. The standard deviation is low; the score does not fluctuate significantly among the results. It is evident that the use of the tool improves significantly the chances of reaching a consensus. This applies for all the configuration settings of the tool. Even a simplistic approach on the tool recommendation design rises significant the chances of an agreement.

Another conclusion is that the higher the agreement probability, the less useful the conflict resolution tool is because the collaborating users agree and it is not called often. In the following table, we show that when the agreement probability rises, the benefit of using the tool is reduced.

Agreement	Majority	Weighted majority	TruthNest Majority
50%	50.4%	50.7%	50.2%
60%	36.5%	36.3%	36.6%
70%	26.3%	26.3%	26%
80%	15.3%	15.3%	15.4%
90%	6.4%	6.3%	6.4%

Table 21 – Score improvement compared to ground truth (mean score)

The comparison of the performances of the three settings of the conflict resolution tool does not favour any specific setting. It seems that a decision based on the majorities, which is easily computable, is enough. The weighted majority setting has slightly the worst scores, and the TruthNest majority settings appears to be marginal better in most cases. However, relying on an external service (in case of TruthNest) and adding more complexity/ an extra point of failure, does not seem to compensate enough in terms of performance.

We could note several reasons to interpret these results:

1. Not all users have a TruthNest score, so its impact is low. If no TruthNest score can be calculated (no tweet posted the last week), a default (low) TruthNest score is assigned. In the simulations, this happened for half of the users, almost for all the plain users and some experienced users. Even those experienced users were assigned a low score. This fact reduced significantly the influence of the TruthNest contributor score.

2. The user weight a_i that has been used to calculate the weighted majority score did not differ significantly between the users. The difference between the highest and the lowest user weight is only 0.12. Therefore, its impact was low and the plain and weighted majority settings did not differ significantly.

3. The probability that a user will accept/ reject a tweet is determined by its belief vector. These probabilities are fixed in all the simulation phases and the tool is called only when a conflict occurs and no agreement is reached. In other words, only when a disagreement insists. Since this probability is the same in all configuration settings, the performance score will be similar in all settings.

The user's weights did not differ significantly in each of the configurations. Their value in case of agreement/ disagreement was equivalent. For example, if we assign the agreement with +1 and the disagreement with -1, or we assign the agreement with +10 and the disagreement with -10, the result will be the same. We could assign an extra weight if certain conditions occur (e.g. if the user is Twitter verified).

7 Conclusions and further work

We have proposed a core model of HMNs and an approach to using this for the purposes of simulation modelling in order to help determine the impact of HMN designs. We have demonstrated the approach via two proof of concepts. One in which we investigate increasing machine agency in Wikipedia for addressing the emergent phenomenon of edit wars. Another in which we investigate different degrees of machine agency and trust relationships in a new HMN that is under development at the time of writing called Truly Media. Via these case studies, we have demonstrated that the Core HMN Model is both generic and extensible. It has been used successfully in two very different HMNs; one well-established and large-scale HMN with available historical data and one in which there is very limited data available as is under development. Further, the model has been implemented in two different simulation frameworks, which also demonstrates its applicability. Below we further discuss conclusions and further work for the two case studies.

7.1 Wikipedia

In this deliverable we have developed a proof of concept simulation model that simulates the emergence of edit wars in Wikipedia. Wikipedia is a highly dynamic and unpredictable environment, which is described well by the following quotation:

> *"Wikipedia is a complex system in which of millions of individually unpredictable editors collaborate in an unscheduled and virtually uncontrolled fashion"*[14]

Being aware of the complexity of Wikipedia, the focus was on developing a model that would reduce the complexity, yet being able to produce meaningful results that we can interpret and justify. The data produced by baseline simulations allowed to validate the rules and parameters used for modelling edit wars. Once we were satisfied with the accuracy of the baseline model we have explored how machine agency could be increased by introducing a bot (*EditWarBot*) with the capability to detect and help facilitate the resolution of edit wars.

[14] http://wikilit.referata.com/wiki/Cooperation_and_quality_in_Wikipedia

For the Wikipedia modelling, we explored a quantitative approach to utilise the historical data available on this HMN. The initial analysis of logs allowed us to gain insight to the working of Wikipedia, informing the simulation model. For example:

a) The majority of agents are short lived and they are active only for one day.
b) The work is not equally distributed; about 10% of agents do the heavy lifting and the rest are passive most of the time.
c) The information extracted from log data show that about 2% of articles are affected by edit wars.
d) The duration and intensity of edit wars depends on the type of interacting agents. For example, about 74% of edit wars are between human agents.

We incorporated these observations into a model, which was able to predict the emergence of edit wars with a 91.5% accuracy on average (as high as 100% for some time periods). By introducing the *EditWarBot*, we also observed a significant reduction in the duration of edit wars, which was the aim when increasing the machine agency in order to help agents reach a consensus in order to increase the reliability and quality of information in Wikipedia.

As noted above, there are aspects of Wikipedia that we have simplified for this proof of concept. For future work, we note the following aspects that could be improved upon. For example, the model that we have developed is not deterministic and the probability of events occurring have been described by stochastic parameters. However, the model makes use of average rates, which fails to account for the variation that we observe from week to week in Wikipedia. As such, the natural ebb and flow of activity within Wikipedia was not possible to capture accurately, observing significant variation in certain time periods for particular types of activity, such as editing articles.

We have established a set of rules and for modelling the activities of agents, achieving simulation results that are close to what we observe in the real HMN. However, for future work, we can introduce a notion of influence, rather than relying as heavily on statistical information, such as the bias that agents may have in their behaviour, which we explored in the data analysis discussed in Section 5.1.1. That is, modelling how some agents may be biased towards engaging in edit wars with other agents of a particular type or where there is a history of past edit wars.

The modelling was also limited by the available information in the data set we used. Further work could expand on the model by taking into account activity on articles' talk pages. This is a natural place for human agents to discuss and resolve edit wars, and information about this would help in different ways. For example, to help determine if agents in edit wars show a willingness to resolve edit wars. It would also help an EditWarBot determine an appropriate action to take, such as not interacting with agents who are already working towards reaching a consensus.

7.2 Truly Media

The simulation modelling approach of the Truly Media case with the use of the HUMANE typology gave us insight in several aspects of the verification process. The influence of several factors has been identified and practical feedback has been provided.

One of the conclusions is that the conflict resolution tool configuration does not play the most important role in the verification process. Whether we use a specialized service to evaluate the users' credibility, such as TruthNest, or a simplistic approach based on the past user actions, the effectiveness of the tool is almost the same. Focusing on an in-depth user profile analysis to unveil unique users' characteristics should not be the priority of the Truly Media designers.

The use of the conflict resolution tool did reduce significantly the conflicts. The conclusion is that its frequent use, whenever a disagreement occurred, improves the verification process. The verification tools output could be used more frequently, even in cases of agreement, to increase the confidence of the users about their choice.

An obvious remark is that if the probability that a user will collaborate with the users with whom he disagrees and reach an agreement is high, the overall tool performance is high. A review on the simulation model definition should indicate how the verification process should be managed by an organization. The posts should be verified by users working independently but in parallel, so that they can communicate directly in case of disagreements. For example, a pair of employees (consisting of at least one experienced employee) verifying posts concurrently is preferable than having them working sequentially. Furthermore, the members of the verification team should work in pairs and not change frequently. It is much easier to cultivate appreciation and trust between two colleagues who know each other and rely on each other's professionalism, than to expect from employees with no relations to agree and achieve synergy.

The process of modelling the Truly Media case allowed us to study the verification process at several different levels of abstraction. By approaching the problem at a higher level of abstraction, we were better able to understand the behaviors and interactions of all the components within the system. Therefore we are better equipped to counteract the complexity of the overall system. For example, the use of the TruthNest service adds complexity without compensating with a significant improvement in the conflict resolution. On the other hand, the existence of historical data (on users past evaluations) is of high importance and the Truly Media designers should concentrate on that.

The conflict resolution tool can be improved further by taking into consideration the conclusions of the previous paragraph and enhancing the simulation model. The Truly Media platform should be updated to support the proposed functionalities. This task can be easily accomplished as the changes in the system architecture are minor. The recommendations will result in a more accurate description of the system.

One of the major problems that we had to overcome is the lack of real user activity data. The platform is in beta version and no sufficient data could be retrieved. As soon as the data are available, we should perform a data analysis to underpin our hypothesis. The user action probabilities and the probabilities values in general will correspond to the real use cases. A data analysis is required to estimate these values.

Another restriction is the nonexistence of older posts for almost half of the simulation users. Without posts, we cannot compute the TruthNest scores and the Stylometry related fields (age/sex). The Truly Media platform should be updated to search periodically for posts of its users and store them. The most recent posts will be used by the conflict resolution tool. Gradually posts of all the users should exist in the platform.

The HUMAN typology offers a range of fields that could be used to represent behavioural models. If the Truly Media platform logs the user activities/ interactions in detail, we could define a behavioural model and represent in greater detail the user interactions. We could record for example how often the users use the collaboration mechanisms of the platform and the series of his actions during the validation process. This information is also useful for the evolution of the Truly Media platform itself.

The usage of the Stylometry module and the age and sex fields in the Human class would be interesting. A behavioural model from the fields of human machine interaction/ psychology/ ergonomics could be applied. Especially the age field is important since it strongly relates with the user behaviour. Combined with other user characteristics, it could reveal aspects of the user's personality and lead to an improved user profile modelling.

With the use of the HUMAN typology and method we can understand the processes necessary for developing and maintaining the Truly Media HMN. The internal complexity of the network and the connections between humans and machines have been analysed. We focused on characteristics of relationships such as trust and reputation, and proposed a conflict resolution tool to address the problem of conflicts during the verification process. We learned that the use of a Social Media credibility assessment service (TruthNest) does not significantly reduce the conflicts. Improved results would be achieved by reorganising the verification workflow (e.g. users working in pairs). The increased understanding of the HMN allows us to redesign it according to the findings and further evolve it.

8 References

Arthur, W. (1994). Inductive reasoning and bounded rationality. *American Economic Review*, *84*, 406–411.

Bandura, A. (1977). Self-efficacy: toward a unifying theory of behavioral change. *Psychological Review*, *84*(2), 191. http://doi.org/10.1037/0003-066X.37.2.122

Bandura, A. (1982). Self-efficacy mechanism in human agency. *American Psychologist*, *37*(2), 122. http://doi.org/10.1037/0003-066X.37.2.122

Bandura, A. (2012). On the functional properties of perceived self-efficacy revisited. *Journal of Management*, *38*(1), 9–44. http://doi.org/10.1177/0149206311410606

Bonabeau, E. (2002). Agent-based modeling: methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, *99 Suppl 3*(90003), 7280–7. http://doi.org/10.1073/pnas.082080899

Burt, R. S. (2004). Structural holes and good ideas. *American Journal of Sociology*, *110*(2), 349–399.

Cetin, N., Burri, A., & Nagel, K. (2003). A large-scale agent-based traffic microsimulation based on queue model. In *Swiss Transport Research Conference*.

Cucinotta, T., Checconi, F., Kousiouris, G., Kyriazis, D., Varvarigou, T., Mazzetti, A., … Stein, M. (2010). Virtulised e-Learning with Real-Time Guarantees on the IRMOS Platform. In *IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*. IEEE.

Dencheva, S., Prause, C. R., & Prinz, W. (2011). Dynamic Self-moderation in a Corporate Wiki to Improve Participation and Contribution Quality. In *The 12th European Conference on Computer Supported Cooperative Work*. Springer.

Dwyer, C., Hiltz, S. R., & Passerini, K. (2007). Trust and Privacy Concern Within Social Networking Sites: A Comparison of Facebook and MySpace. Retrieved June 6, 2015, from http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1849&context=amcis2007

Engen, V., Pickering, J. B., & Walland, P. (2016). Machine Agency in Human-Machine Networks; Impacts and Trust Implications. In *HCI International*. Toronto, Canada: Springer.

Fishman, G. (2013). *Discrete-Event Simulation: Modeling, Programming, and Analysis*. Springer.

Følstad, A., Eide, A. W., Pickering, J. B., Tsvetkova, M., Gavilanes, R. G., Yasseri, T., & Engen, V. (2015). *D2.1 Typology and Method v1*.

Følstad, A., Engen, V., Mulligan, W., Pickering, B., Pultier, A., Yasseri, T., & Walland, P. (2017). *D2.3 The HUMANE typology and method*.

Følstad, A., Engen, V., Yasseri, T., Gavilanes, R. G., Tsvetkova, M., Jaho, E., … Pultier, A. (2016). *D2.2 Typology and Method v2*.

Fritzson, P. (2004). *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley & Sons Ltd.

Godfrey, K. (1983). *Compartment models and their application*. Academic Press.

Grinstead, C., & Snel, J. (1997). *Introduction to Probability*. Americal Mathematical Society.

Heylighen, F. (1989). Self-organization, Emergence and the Architecture of Complexity. In *Proceedings of the 1st European Conference on System Science* (pp. 23–32).

Heylighen, F. (1991). Modelling emergence. *World Futures: The Journal of General Evolution, Special Issue on Creative Evolution*, 1–10.

Hogg, T., & Hubermann, B. A. (1991). Controlling chaos in distributed systems. *IEEE Transactions on Systems, Man and Cybernetics*, *21*, 1325–1332.

Iñiguez, G., Török, J., Yasseri, T., Kaski, K., & Kertész, J. (2014). Modeling social dynamics in a collaborative environment. *EPJ Data Science*, *3*(1), 7. http://doi.org/10.1140/epjds/s13688-014-0007-z

ISO. (2010). *Ergonomics of human–system interaction — Part 210: Human-centred design for interactive systems*. Geneva, Switzerland: International Organization for Standardization.

Jackson, J. (2017). Wikipedia bans Daily Mail as “unreliable” source. Retrieved March 2, 2017, from https://www.theguardian.com/technology/2017/feb/08/wikipedia-bans-daily-mail-as-unreliable-source-for-website

Jennings, N. (2001). An agent-based approach for building complex software systems. *Communications of the ACM*, *44*(4), 35–41. Retrieved from http://dl.acm.org/citation.cfm?id=367250

Jones, K., & Leonard, L. N. K. (2008). Trust in consumer-to-consumer electronic commerce. *Information & Management*, *45*(2), 88–95. http://doi.org/10.1016/j.im.2007.12.002

Juris, J. S. (2012). Reflections on# Occupy Everywhere: Social media, public space, and emerging logics of aggregation. *American Ethnologist*, *39*(2), 259–279.

Macal, C. M., & North, M. J. (2010). Tutorial on agent-based modelling and simulation. *Journal of Simulation*, *4*(3), 151–162.

Maguire, M. (2001). Methods to support human-centred design. *International Journal of Human-Computer Studies*, *55*(4), 587–634.

Marakas, G. M., Johnson, R. D., & Clay, P. F. (2007). The evolving nature of the computer self-efficacy construct: An empirical investigation of measurement construction, validity, reliability and stability over time. *Journal of the Association for Information Systems*, *8*(1), 15.

McKnight, D. H., Carter, M., Thatcher, J. B., & Clay, P. F. (2011). Trust in a specific technology: An investigation of its components and measures. *ACM Transactions on Management Information Systems (TMIS)*, *2*(2), 12. http://doi.org/10.1145/1985347.1985353

MediaWiki.org. (2017). MediaWiki. Retrieved March 1, 2017, from https://www.mediawiki.org/wiki/MediaWiki

Mun, Y. Y., & Hwang, Y. (2003). Predicting the use of web-based information systems: self-efficacy, enjoyment, learning goal orientation, and the technology acceptance model. *International Journal of Human-Computer Studies*, *59*(4), 431–449. http://doi.org/10.1016/S1071-5819(03)00114-9

Nasaw, D. (2012). Meet the “Bots” That Edit Wikipedia. *BBC News*. Retrieved from https://scholar.google.co.uk/scholar?hl=en&q=Meet+the+%27bots%27+that+edit+Wikipedia&btnG=&as_sdt=1%2C5&as_sdtp=#1

Nasser, B., Engen, V., Crowle, S., & Walland, P. (2013). A novel risk-based approach for online community management. In *The Eighth International Conference on Internet and Web Applications and Services (ICIW)*.

Omicini, A. (2001). SODA: Societies and infrastructures in the analysis and design of agent-based systems. *Agent-Oriented Software Engineering*, 185–193. Retrieved from http://link.springer.com/10.1007/3-540-44564-1_12

Pickering, J. B., Engen, V., Jaho, E., Sarris, N., Yasseri, T., Tsvetkova, M., … Følstad, A. (2016). *D3.3*

Report on second set of case-studies.

Pickering, J. B., Engen, V., & Walland, P. (2017). The Interplay Between Human and Machine Agency. In *HCI International*.

Polack, F., & Stepney, S. (2005). Emergent properties do not refine. In *REFINE workshop, Electronic notes in Theoretical Computer Science*.

Schwagereit, F., Gottron, T., & Staab, S. (2014). *Micro Modelling of User Perception and Generation Processes for Macro Level Predictions in Online Communities*.

Schwagereit, F., Scherp, A., & Staab, S. (2011). Survey on Governance of User-generated Content in Web Communities. In *Web Science Conference*.

Shanga, R., Luoa, S., Lia, Y., Jiaoa, L., & Stolkin, R. (2015). Large-scale community detection based on node membership grade and sub-communities integration. *Physica A: Statistical Mechanics and Its Applications*, *428*, 279–294.

Siebers, P. O., Macal, C. M., Garnett, J., Buxton, D., & Pidd, M. (2010). Discrete-event simulation is dead, long live agent-based simulation! *Journal of Simulation*, *4*, 204--210.

Spek, S., Postma, E., & Herik, J. Van den. (2006). Wikipedia: organisation from a bottom-up approach. *Research in Wikipedia, on the …*. Retrieved from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2097682

Thatcher, J. B., McKnight, D. H., Baker, E. W., Arsal, R. E., & Roberts, N. H. (2011a). The Role of Trust in Postadoption IT Exploration: An Empirical Examination of Knowledge Management Systems. *IEEE Transactions on Engineering Management*, *58*(1), 56–70. http://doi.org/10.1109/TEM.2009.2028320

Thatcher, J. B., McKnight, D. H., Baker, E. W., Arsal, R. E., & Roberts, N. H. (2011b). The Role of Trust in Postadoption IT Exploration: An Empirical Examination of Knowledge Management Systems. *IEEE Transactions on Engineering Management*, *58*(1), 56–70. http://doi.org/10.1109/TEM.2009.2028320

Thatcher, J. B., Zimmer, J. C., Gundlach, M. J., & McKnight, D. H. (2008). Internal and external dimensions of computer self-efficacy: An empirical examination. *Engineering Management, IEEE Transactions on*, *55*(4), 628–644. http://doi.org/10.1109/TEM.2008.927825

Tsvetkova, M., García-Gavilanes, R., & Yasseri, T. (2016). Dynamics of Disagreement: Large-Scale Temporal Network Analysis Reveals Negative Interactions in Online Collaboration. Computers and Society; Physics and Society. Retrieved from http://arxiv.org/abs/1602.01652

Tsvetkova, M., Yasseri, T., Meyer, E. T., Pickering, J. B., Engen, V., Walland, P., … Bravos, G. (2015). Understanding Human-Machine Networks: A Cross-Disciplinary Survey. *arXiv Preprint*.

Tye, E., Fliege, J., & Avramidis, T. (2013). Macro-level risk management for online communities. In *The 26th European Conference on Operational Research*. Rome, Italy.

van Dyke Parunak, H. (1997). “Go to the ant”: engineering principles from natural multi-agent systems. *Annals of Operations Research*, *75*, 69–101.

van Dyke Parunak, H., & Bruekner, S. (2004). Engineering swarming systems. In *Methodologies and Software Engineering for Agent Systems* (pp. 341–376).

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of “small-world” networks. *Nature*, *393*(6684), 440–2. http://doi.org/10.1038/30918

Wikipedia. (2015). Wikipedia:Five pillars. Retrieved March 1, 2017, from https://en.wikipedia.org/w/index.php?title=Wikipedia:Five_pillars

Wolf, T. De, & Holvoet, T. (2004). Emergence and self-organisation: a statement of similarities and differences. In *Engineering Self-Organising Systems*. Retrieved from https://trac.v2.nl/export/7711/rui/projects/UnleashCulture/UnleashCulture3/Emergence and Self-Organisation, similarities and differences.pdf

Wooldridge, M. (2009). *An Introduction to MultiAgent Systems* (2nd ed.). John Wiley & Sons.

Zambonelli, F., & van Dyke Parunak, H. (2001). Signs of a revolution in computer science and software engineering. In *2nd Italian Workshop on Objects and Agents*.

Zambonelli, F., & van Dyke Parunak, H. (2003). Towards a paradigm change in computer science and software engineering: a synthesis. *The Knowledge Engineering Review*, *18*(4), 329–342. http://doi.org/10.1017/S0269888904000104

9 Appendix A: Modelling details for Wikipedia

Add text/content, e.g., details about implementation, simulation method, how parameterisation was done, simulation results, etc.

9.1 Wikipedia simulation data

Here we provide some snippets of data generated from the source dataset in order to be used for the simulation modelling.

Table 22 – Example Wikipedia user overview statistics.

Username	Join date	Date of last activity	Number of articles created	Number of edits	Number of reverts	Number of edit wars
Anchoviesit	08/08/2011	08/08/2011	0	2	0	0
Kramnella	27/05/2011	27/05/2011	0	1	0	0
Lulubermudez	09/04/2008	09/04/2008	0	2	0	0
Jou46	19/07/2006	14/09/2006	30	221	1	0
Chacor	12/10/2006	08/07/2007	0	4	0	0
Kaiser_matias	05/12/2004	06/12/2010	39	180	4	0
Auntieruth55	09/06/2010	09/06/2010	0	1	0	0
Vandal_account	25/03/2007	25/03/2007	0	11	0	0
Plommespiser	14/03/2010	24/03/2010	0	3	0	0
Imhotep	14/08/2008	16/08/2008	0	2	1	1
Gracelyn100	20/01/2010	20/01/2010	0	1	0	0

Table 23 – Example overview Wikipedia snapshot statistics.

Snapshot start date	Number of active users	Number of active bots	Number of new pages	Number of edits	Number of reverts	Number of edit wars
04/01/2010	767	50	408	9485	654	13
11/01/2010	812	42	213	7527	694	11
18/01/2010	815	45	102	7070	699	10
25/01/2010	893	39	144	6908	746	12
01/02/2010	854	44	137	10150	738	16
08/02/2010	880	46	122	7676	767	25
15/02/2010	896	44	153	7019	640	11
22/02/2010	1051	44	120	7414	829	18
01/03/2010	964	41	141	8173	791	10

Table 24 – Example reverting history for Wikipedia.

Article	Date	Reverter	Revertee	PartOfEditWar
Viking_program	2011-10-27 12:56:32Z	DJDunsie	DJDunsie	0
Gluten-free_diet	2011-10-19 14:08:46Z	The_Rambling_Man	Racepacket	0
Flower_of_Scotland	2011-10-18 12:27:06Z	Normandy	Normandy	0
HM_Prison_Pentridge	2011-10-16 06:07:48Z	Peterdownunder	Tassedethe	0
Hardwood	2011-10-24 13:22:51Z	Courcelles	152.26.41.253	0
Hardwood	2011-10-24 12:52:12Z	GoblinBot4	152.26.41.253	0
Zacarias_Moussaoui	2011-10-24 13:41:06Z	Katarighe	Katarighe	0
Zacarias_Moussaoui	2011-10-09 22:24:00Z	Djsasso	Auntof6	0
Zacarias_Moussaoui	2011-08-05 12:11:33Z	Iqinn	41.136.228.94	0
Zacarias_Moussaoui	2010-10-04 23:37:18Z	Wayne_Slam	72.92.0.35	0
Zacarias_Moussaoui	2010-10-04 23:37:02Z	72.92.0.35	Gfoley4	1
Zacarias_Moussaoui	2010-10-04 23:36:14Z	Gfoley4	72.92.0.35	1
Zacarias_Moussaoui	2010-05-23 07:55:39Z	5_albert_square	76.95.131.7	0

Table 25 – Example overview Wikipedia edit wars.

Article	Start date	End date	Duration (days)	Number of reverts	Agent 1 (first reverter)	Agent 2
Muzaffarabad	2009-09-09 09:21:34Z	2009-09-09 09:48:31Z	0	3	Mercy	00jayhind
WrestleMania_XXVI	2009-02-25 23:35:15Z	2009-02-27 20:57:35Z	1	2	SuperSilver901	3bulletproof16
WrestleMania_XXVI	2009-02-24 03:10:18Z	2009-02-26 00:03:50Z	1	2	3bulletproof16	Truco
Homeopathy	2008-08-13 04:54:40Z	2008-08-13 04:55:14Z	0	2	Werdan7	Adam_Cuerden
This	2010-01-25 05:06:11Z	2010-01-30 16:24:17Z	5	2	Thijs!bot	Adrenalin
Christianity	2006-04-16 23:13:52Z	2006-04-16 23:16:11Z	0	2	Aflm	Danielle_Cunio
Zeus	2006-04-23 18:32:22Z	2006-04-23 18:48:40Z	0	3	Aflm	Teagan
Charley_Uchea	2008-01-11 18:23:51Z	2008-02-04 19:55:06Z	24	3	AnemoneProjectors	Alexbot

For the edit wars details, we extracted further information that was omitted from the above tables due to space restrictions. In the summary snapshot statistics (Table 23), we include details of the edit wars according to four conditions:

- Condition 1: ongoing (started before snapshot and continues)
- Condition 2: started and finished within the snapshot
- Condition 3: started within the snapshot, but did not finish yet
- Condition 4: started before the snapshot, but finished within the snapshot

We also labelled the edit wars statistics in Table 25 according to the type of agents who were involved in the edit wars. We distinguish between human users and bots on the one hand, but also between anonymous and registered users. This information was used in the data analysis presented above in Section 5.1.1.3.

9.2 Extensions to Core HMN Model

Details of the Article class are shown below in Figure 23, depicting the use of two key classes introduced to reflect entries in the article's log and any edit wars taking (or having taken) place on the respective article.

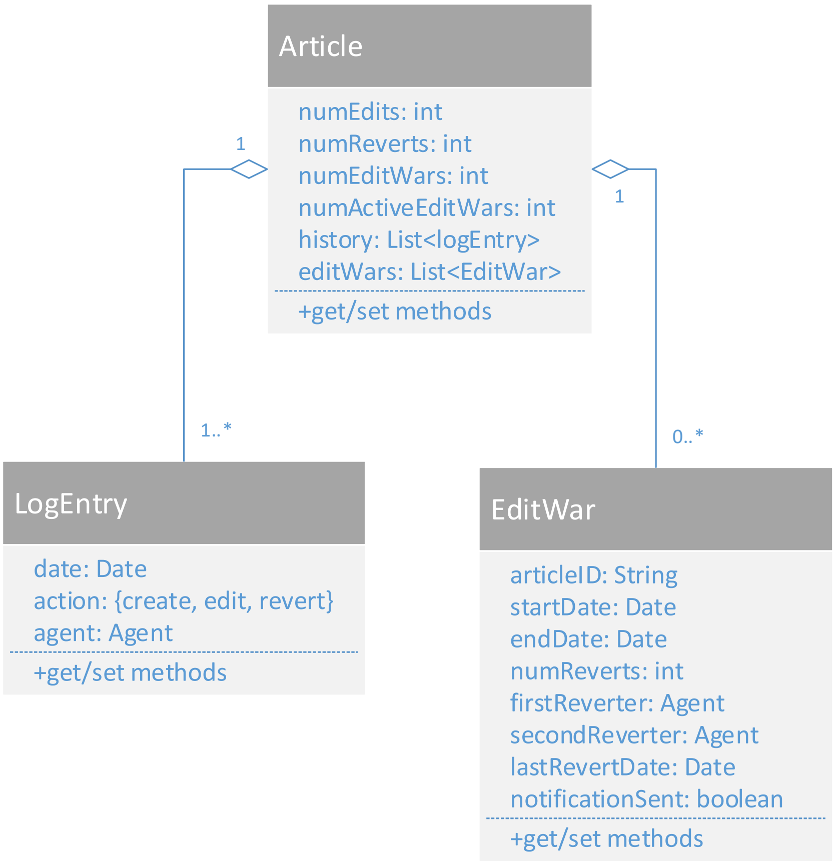


Figure 23 – Article class details.

The User and NormalBot agents introduced in the Wikipedia model are nearly identical as humans and machines can do the same actions: creating new articles, editing existing articles, reverting contributions (from oneself or others), and taking part in edit wars. The only difference between the User and NormalBot classes is a flag in the User class to indicate whether the respective user is an anonymous user or not.

The parameters of these classes include: the date in which they joined and when they were last active; probabilities for creating new article, editing existing articles and reverting; a map of all the articles the respective agent has contributed to (the key is the article ID); a list of edit wars they are or have participated in; the number of times they were the first to revert in an edit war, and how many times they were the last; and finally, a list of edit wars they have received notifications from the EditWarBot about ending.

The complexity of these classes (agents) are encapsulated within the takeAction() method, implementing the behavioural rules discussed above in Section 5.1.3. As such the only property of the EditWarBot is a list of the edit wars taking place in Wikipedia.

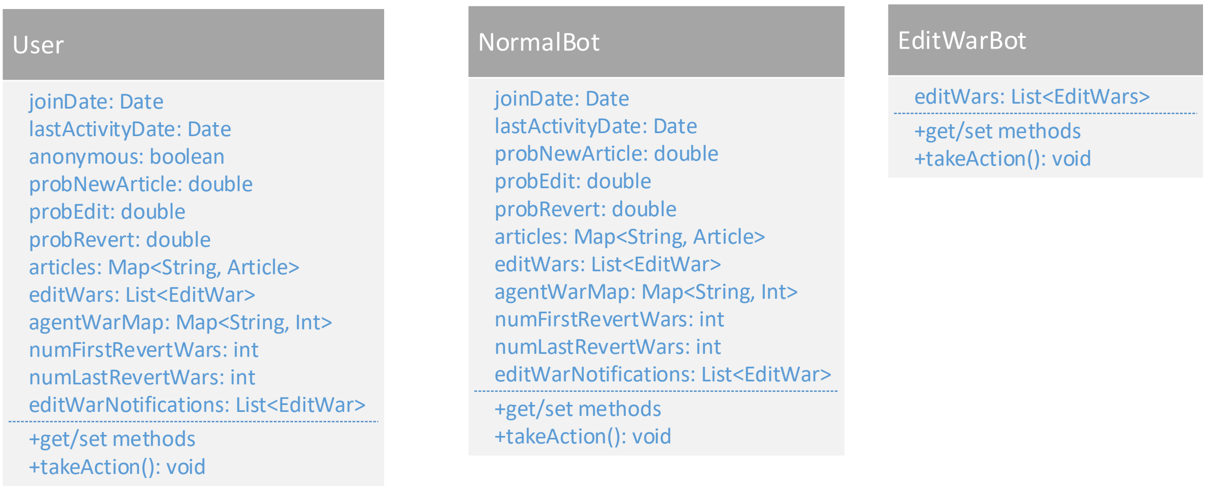


Figure 24 – User and Bot class details.

9.3 Extracting edit wars

Information about edit wars was extracted from the Wikipedia log files. The following section is part of the logfile for article “Powderfinger” that clearly illustrates an example of an edit war:

```
^^^_2010-01-14T13:20:56Z 1 134 GoblinBot4
^^^_2010-01-14T13:20:52Z 1 136 Doktor_Nooo
^^^_2010-01-14T13:20:41Z 1 134 GoblinBot4
^^^_2010-01-14T13:20:38Z 1 136 Doktor_Nooo
^^^_2010-01-14T13:20:14Z 1 134 GoblinBot4
^^^_2010-01-14T13:20:09Z 1 136 Doktor_Nooo
^^^_2010-01-14T13:19:37Z 1 134 GoblinBot4
^^^_2010-01-14T13:19:33Z 0 136 Doktor_Nooo
```

The first item in each row is the date of activity, the second is the type of activity that can be 0 (for edit) and 1(for revert), the version number and agentID. We can analyse the log from the last row and moving upwards. On the last row we can see that version 136 was created by *Doktor_Nooo* and 4 seconds later it was reverted by GoblinBot4 to version 134. About 32 minutes later *Doktor_Nooo* reverted GoblinBot4 and went back to version 136 of the article. The algorithm that extracts edit wars for each article creates a list of agents who mutually reverted each other.

10 Appendix B: Modelling details for Truly Media

10.1 Truly Media core model usage

The basic classes of the HUMAN typology are used to model the Truly Media use case. We override the Human and Artefact classes to describe the user class and the verification form class. Edge classes between users are defined and with the use of the encapsulated Connection class the trust between them is represented. The Machine class is also used to describe the verification tool. It does not represent the system as a whole, but the tool. Connections between the Machine class and the user's class are defined.

In the following we present the fields used by the major classes.

Human class

age	Filled by using the Stylometry module
gender	Filled by using the Stylometry module
trust	The user "correctness" weight a_i calculated through the ground truth
reputation	TruthNest contributor reputation score normalized in the interval [0, 1]

Machine class

utilisation	How often the tool was called

Connection class

trust	$Trust(i, j)$ in case of Human to Human connections

The Artefact class is also being overridden and all its fields are being used to represent the verification form. Other HUMANE core classes are also being used implicitly or explicitly (Edge, Node). We also present some basic custom classes.

TwitterUser class

insertionDate	Insertion date of the user in the platform
tweets	List of user posts
belief	The belief vector of the user (action probabilities)
userActions	List of performed actions by the user
contributorScore	TruthNest contributor score

Verification class

formId	Unique id of the form
edits	List of edit operation on this form (verification values)
editsSummary	Holds the number of accepts/ disagreements on this form
groundTruth	The ground truth as defined in the learning phase

SimulationMode class

GROUND_TRUTH(0), MAJORITY_VOTE(1), WEIGHTED_MAJORITY_VOTE(2), TRUTHNEST_WEIGHTED_MAJORITY_VOTE(3)	enum class that represents the tools configuration

SimulationResult class

sumOfDifferences	The performance metric
createdAt	Date creation field
mode	SimulationMode of the execution

10.2 Stylometry Profiling

The motivation behind the use of the Stylometry module has been to fill the age and sex fields of the Human class. While these fields are not being currently used in the simulation model, it would be interesting to show that the fields are meaningful and that they could be included in a future update of the model.

The Stylometry module provides predictions of the age and gender of authors, based on their posts. In the context of REVEAL[15] project it is applied in streaming batches of tweets. It is heavily based on the analysis of the lexical and contextual content of tweets, in order to find discriminating features to be used as indicators for the different age classes and gender.

The underlying algorithm has been tested in practice as part of the Author Profiling Challenge[16] for PAN16[17] where it was placed first on average for the English Language, thus being among the state-of-the-art models for this task.

[15] Reveal project: https://revealproject.eu/

[16] Author Profiling Challenge: http://pan.webis.de/clef16/pan16-web/author-profiling.html

[17] PAN: http://pan.webis.de/