Exploring Dynamic, Multi-level Interactions within an Organization: An Agent-based Modeling Approach

Overview, Design Concepts, Details, and Human Decision Making

Bianica Pires[1], Joshua Goldstein[1], Emily Molfino[2], Kathryn Schaefer Ziemer[3], Gizem Korkmaz[1], Mark Orr[1]

[1]Biocomplexity Institute of Virginia Tech
[2]U.S. Census Bureau
[3]Ipsos Public Affairs

June 4, 2018

This document provides an overview of model structure using the ODD+D (Overview, Design Concepts, Details, and Human Decision Making) protocol initially developed by Grimm et al. (2006) and later extended by Müller et al. (2013) to include human decision-making. Table 1 outlines the protocol and provides example guiding questions. Following the protocol, in Section 1, we provide an overview of the model. In Section 2, the general concepts underlying model design are discussed. In Section 3, we explain the details of the model's implementation.

Table 1. The ODD+D protocol and example guiding questions (source: Müller et al., 2013).

Structural elements		Example guiding questions
1. Overview		
	1.1 Purpose	What is the purpose of the study?
	1.2 Entities, State Variables and Scales	What kind of entities are in the model? By what attributes are these entities characterized?
	1.3 Process Overview and Scheduling	What entity does what, and in what order?
2. Design Concepts		
	2.1 Theoretical and Empirical Background	Which general concepts, theories, or hypothesis are underlying the model's design at the system level or at the level(s) of the submodel(s)?
	2.2 Individual Decision-Making	What are the subjects and objects of the decision-making?
	2.3 Learning	Is individual or collective learning included in the decision process?
	2.4 Individual Sensing	What endogenous/exogenous state variables do individuals sense and consider in their decisions?
	2.5 Individual Prediction	Which data do the agents use to predict future conditions?
	2.6 Interaction	Are interactions among agents and entities assumed as direct or indirect? On what do the interactions depend?
	2.7 Collectives	Do the individuals form or belong to aggregations that affect and are affected by the individuals?
	2.8 Heterogeneity	Are the agents heterogeneous? If yes, which state variables and/or processes differ between the agents?
	2.9 Stochasticity	What processes (including initialization) are modeled by assuming they are random or partly random?
	2.10 Observation	What data are collected from the ABM for testing, understanding and analyzing it, and how and when are they collected?
3. Details		
	3.1 Implementation Details	How has the model been implemented?
	3.2 Initialization	What is the initial state of the model world?
	3.3 Input Data	Does the model use input from external sources such as data files or other models to represent processes that change over time?
	3.4 Submodels	What, in detail, are the submodels that represent the processes listed 'Process overview and scheduling'?

1 OVERVIEW

We developed an agent-based model that simulates the dynamics of an organization within a hospital setting. The ABM captures the local interactions of individuals in time and physical space, individual attitudes, and the feedback system between interactions and attitudes and their impact on organizational dynamics, including the formation of subcultures, the diffusion of attitudes, and the sharing of knowledge.

Figure 1 illustrates a conceptual diagram of the model. A central feature of the framework is the agent. The agents represent the autonomous, heterogeneous, and interacting employees of a hospital. Agent movement occurs over the physical environment of the hospital, which is facilitated through the use of GIS and made dynamic through ABM. Furthermore, this pattern of movement is combined with SNA to create the agents' contact network, providing empirically-based input into the development of the organization's social network. Generation of explicit social networks are facilitated through the use of GIS to place agents in time and space, demographic data to provide the agents with unique characteristics (e.g., profession), and observational data to simulate the agents' daily routines. The emergence of social networks gives us insight into the informal roles that emerge at the individual level and subcultures that emerge at the group level within an organization. Another feature of the framework is the agents' theoretically-based attitude towards knowledge sharing, which drives behavior. In this case, the behavior is the decision to share knowledge. As a feedback system, individual attitudes dynamically provide input into the creation of the social networks.

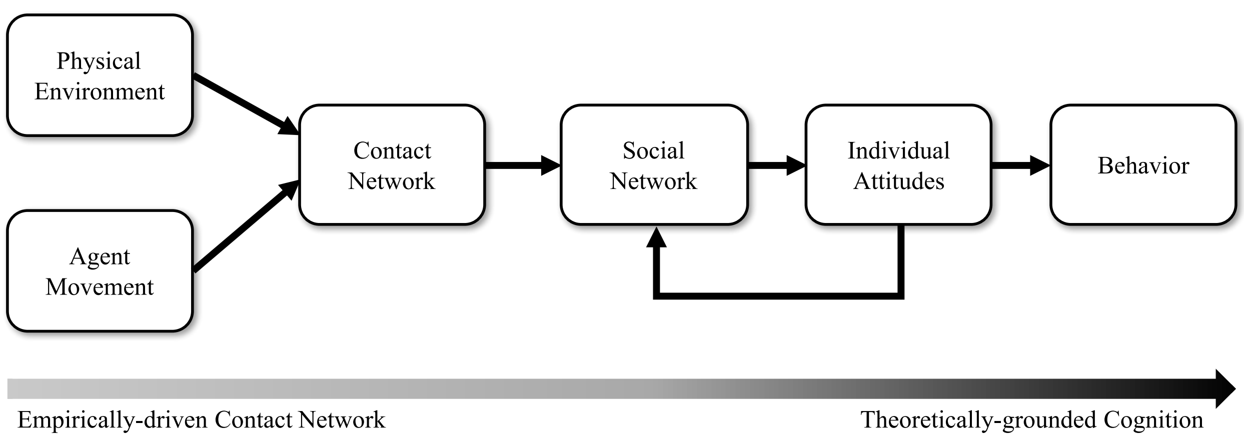


Figure 1. A conceptual diagram of the agent-based model. The physical environment and agent movement across the hospital creates the agent' contact network, providing the empirically-based input into development of the social networks. The agent's social context and prior attitude can drive attitude formation, which influences behavior. As a feedback system, individual attitudes dynamically influence the social network.

We drew on the literature to develop three models of attitude diffusion and knowledge sharing: (1) an independent cascade of knowledge sharing, (2) a linear threshold model of attitude diffusion, and (3) a cognitive model of attitude formation. These models are outlined in Table 2 and described in more detail in Section 3.4. The first model is the simplest and is driven by the dynamics of prior cascade models of information diffusion. The second model builds on the first one by using a threshold function to determine individual attitudes, which can subsequently impact knowledge sharing dynamics. Finally, the third model, is the most complex in its incorporation of a cognitive model of attitude formation and change. The cognitive model accounts for the agents' attitude state (e.g., beliefs around knowledge sharing) and external social context (e.g., social network), the combination of which forms the current attitude and in turn drives behavior.

Table 2. Models of attitude diffusion and knowledge sharing.

Model	Attitude Formation	Knowledge Sharing
Model 1: Cascade model of knowledge sharing	None.	Diffusion with probability p.
Model 2: Threshold model of attitude diffusion	Attitude changes using a threshold value, where $A \in \{-1, 1\}$.	Diffusion with probability p occurs only when $A = 1$.
Model 3: Cognitive model of attitude formation	Attitude formation using the Reasoned Action Approach, where $-1 \leq A \leq 1$.	Diffusion probability as a function of A.

At the beginning of the simulation, one or more agents are selected to have some piece of knowledge. Knowledge in the ABM is an abstract construct representing some piece of information, know-how, or expertise. In the context of a hospital an example of knowledge is an understanding of hand hygiene practices. As agents decide whether to share, knowledge spreads through the network. An agent will make the decision to share (or not) knowledge when it is in contact with another agent. Note that through the agents' individual interactions attitudes can change over the course of the simulation. Thus, an agent's decision to share tacit knowledge changes as that agent's attitude on knowledge sharing evolves. This creates a feedback system as changes in one's attitude after an interaction can effect changes in another, which in turn impacts individual behavior, and potentially, the strength of social ties in the dynamic networks.

1.1 Purpose

The purpose of the model is to explore how individual-level interactions in time and physical space and individual attitudes interact to influence the formation of subcultures and the emergence of informal roles, all of which impact organizational processes such as knowledge sharing. An ABM is developed that incorporates theories of attitude formation and change with dynamic social networks over a physical environment for this purpose.

1.2 Entities, State Variables and Scales

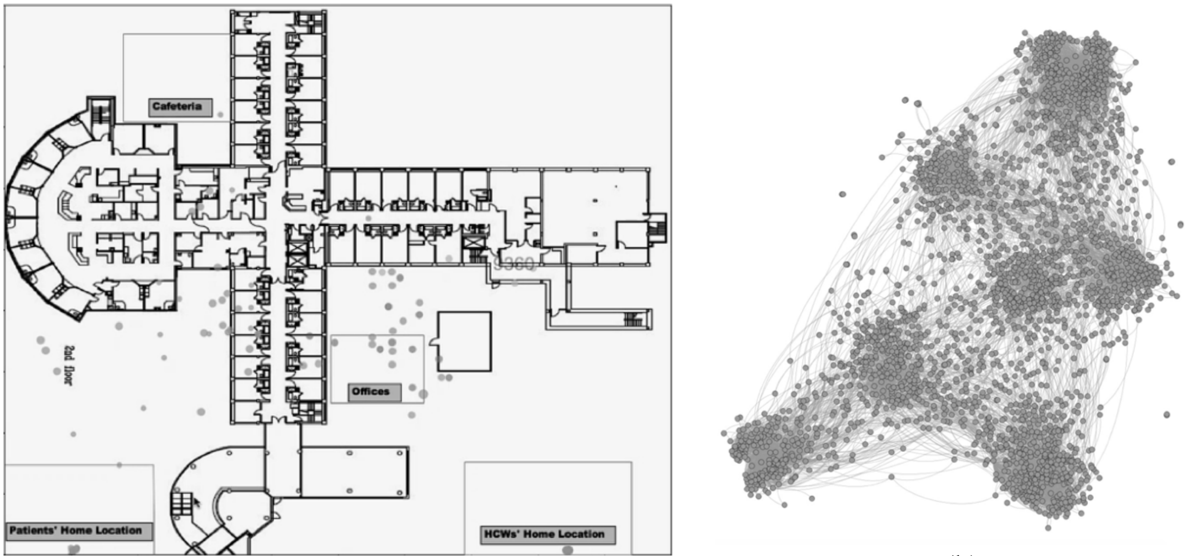
The model contains the following entities, from highest to lowest hierarchical scale: (1) the physical environment and population, (2) the contact and social networks, and (3) the agents, which represent the healthcare workers of a hospital.

The physical environment and population. We utilized data collected for an empirically-based simulation model that explored the potential outbreak of healthcare acquired infections in a hospital in southwest Virginia (Jiménez, 2014). The study was conducted by The Network Dynamics Simulation Modeling Laboratory (NDSSL) of the Virginia Bioinformatics Institute at Virginia Tech (Adigaa et al., 2015). The physical environment is the physical layout of the hospital in southwest Virginia. The hospital contains nine floors and over 1,000 locations, such as patient rooms and employee lounges. The population is the 2,127 synthetic healthcare workers of the hospital. The synthetic individuals represent 30 different healthcare professions (e.g., physicians, nurses, nurse assistants, social workers, physical therapists). The synthetic information model developed by Jiménez (2014) includes not only the synthetic population of the hospital but also the hospital layout and details on their movements (i.e., activity schedules) across the hospital over the course of 200 days. Table 3 provides a summary of the parameters/inputs used in the framework to represent the physical environment and the synthetic population. The data used to create the synthetic information model are discussed in more detail in Section 3.3.

Table 3. Population and environment parameters/inputs used in the framework and the attitude diffusion and knowledge sharing model(s) associated with that parameter/input.

Parameter/Input	Description	Model(s)	Reference
Number of agents	The number of healthcare workers in the hospital was estimated to be 2,127.	1, 2, 3	(Jiménez, 2014)
Agent professions	Agent professions (e.g., nurse, physician) as provided by the synthetic information model of the hospital.	1, 2, 3	(Jiménez, 2014)
Activity schedules	The pre-determined activity schedules of the agents (start time, end time, location) provided in the synthetic information model.	1, 2, 3	(Jiménez, 2014)
Hospital rooms and location types	The hospital rooms and location types based on the hospital's physical layout.	1, 2, 3	(Jiménez, 2014)
Number of agents that have the knowledge	The number of agents that have the knowledge at model initialization.	1, 2, 3	User settable
Number of agents seeded with positive attitude	The number of agents seeded with a positive attitude towards sharing knowledge at model initialization. All other agents have a neutral attitude.	3	(Orr, Thrush, & Plaut, 2013)

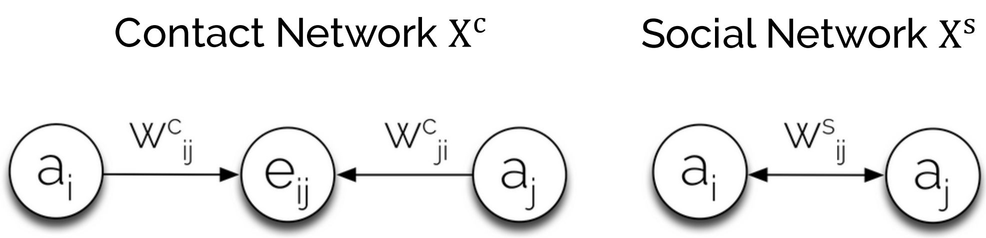
<u>*The contact network and social network*</u>. Physical proximity in addition to homophily and social influence are important drivers of social networks. We operationalize physical proximity in our model as a "contact"—an event where two or more agents are at the same physical location (e.g., nurses' station) at the same time. Agents' activities patterns (schedules) across both physical space and time allows for the development of the agents' contact network X^c, a two-mode affiliation network where agents a_i and a_j participate in the same event e_{ij}. The weight of the tie w_{ij}^c is a function of the duration of event e_{ij}. The average number of activities and contacts in the hospital simulation during the course of 30 days is 8,234 and 164,176, respectively. As agents move about the hospital, we can visualize these contacts through the network diagram in Figure 2.



(a) (b)

Figure 2. The hospital simulation. (a) The daily activities in the hospital for agents on a floor of the hospital. Each dot represents an agent. (b) The contact network diagram of the hospital population over one month. Agents are represented by the dots and contacts between agents are represented by the lines (source: Jiménez, Lewis, & Eubank, 2013).

The social network X^s is a one-mode network between agents a_i and a_j, where the weight of the tie w_{ij}^s is a function of the contact network, profession homophily, and attitude homophily. A notional representation of the contact network X^c and the social network X^s are visualized in Figure 3. Implementation of the contact and social network are discussed further in Section 3.4.



(a) (b)

Figure 3. The notional two-mode and one-mode networks. (a) A two-mode affiliation network X^c where agents a_i and a_j participate in the same event e_{ij}. The weight of the ties $w^c(w_{ij}^c = w_{ji}^c)$ is a function of the duration of event e_{ij}. (b) A one-mode social network $X^s$with agents a_i and a_j, where w_{ij}^s is the weight of the tie between agents.

<u>*Agents*</u>. The agents in the model represent the 2,127 healthcare workers of the hospital in southwest Virginia. Agents have at least three attributes: their profession, their pre-determined activity schedules for the course of the simulation, and a binary attribute that indicates whether they have the knowledge. Agent attributes as they relate to individual attitudes and knowledge sharing vary depending on the model being run. Table 4 shows the set of agent parameters and

the model associated with the parameter. Sections 3.2 and 3.4 provide details on the initialization process and the use of these parameters in the simulation. Not included here are parameters related to the learning processes of the cognitive model (Model 3), these are outlined in Section 2.3.

Table 4. Agent parameters used in the framework and the attitude diffusion and knowledge sharing model(s) associated with the parameter.

Parameter	**Description**	**Model(s)**
Profession	The healthcare discipline assigned to the agent.	1, 2, 3
Activity Schedule	The agent's pre-determined schedule for the course of the simulation, including start time, end time, and location of the activity.	1, 2, 3
Knowledge	A binary variable $K \in \{0, 1\}$ indicating whether the agent has the knowledge.	1, 2, 3
Attitude	In Model 2, attitude is binary variable $A \in \{-1, 1\}$. In Model 3, attitude is a continuous variable $(-1 \leq A \leq 1)$.	2, 3
Attitude Threshold	Threshold values are heterogeneous across agents and are drawn from a normal distribution.	2
Success probability	The probability of sharing knowledge with an interacting agent.	1, 2
Activation of valence units	A vector of length 20 that stores the current value of the agent's valence units. The first 10 items represent the values of the agent's positive intentions, the last 10 items represent the values of the agent's negative intentions towards knowledge sharing.	3

1.3 Process overview and scheduling

The model proceeds in one minute time steps. While employee schedules were provided by second, a minute allows us to capture the individual interactions and activity patterns that are important to the development of social networks (Torrens, 2014) and at the same time, maintains the computational feasibility of running the model. Figure 4 illustrates the model's key processes (discussed further in Section **Error! Reference source not found.**). Agent behavior is broken out into five sub-models discussed in Section 3.4: the *Activity Scheduler*, the *Dynamic Contact Network*, the *Dynamic Social Network*, the *Attitude Formation and Change Model*, and *Knowledge Sharing*.

At the start of the simulation, agents run the *Activity Scheduler*. The *Activity Scheduler* pulls information from a database of pre-determined schedules, including the start time, end time, and location of the agent's current activity. It then searches for any other agents who are at the same location, at the same time. If other agents are present, the contact network is updated, which consists of either creating a new tie (if one did not exist) or updating an existing tie. The strength of the contact tie, in addition to agent attributes, is then used as input into the computing the strength of the social tie. In the case of Models 2 and 3, the agent will then evaluate its attitude in the *Attitude Formation and Change Model* based on interactions with other agents. Next, the agent will determine whether to share knowledge with one of the interacting agents in the *Knowledge Sharing* submodel. At completion of the activity, agents will evaluate their next activity by re-running the *Activity Scheduler*. In addition to this process, agents will periodically evaluate the need to decay any ties in their social network.

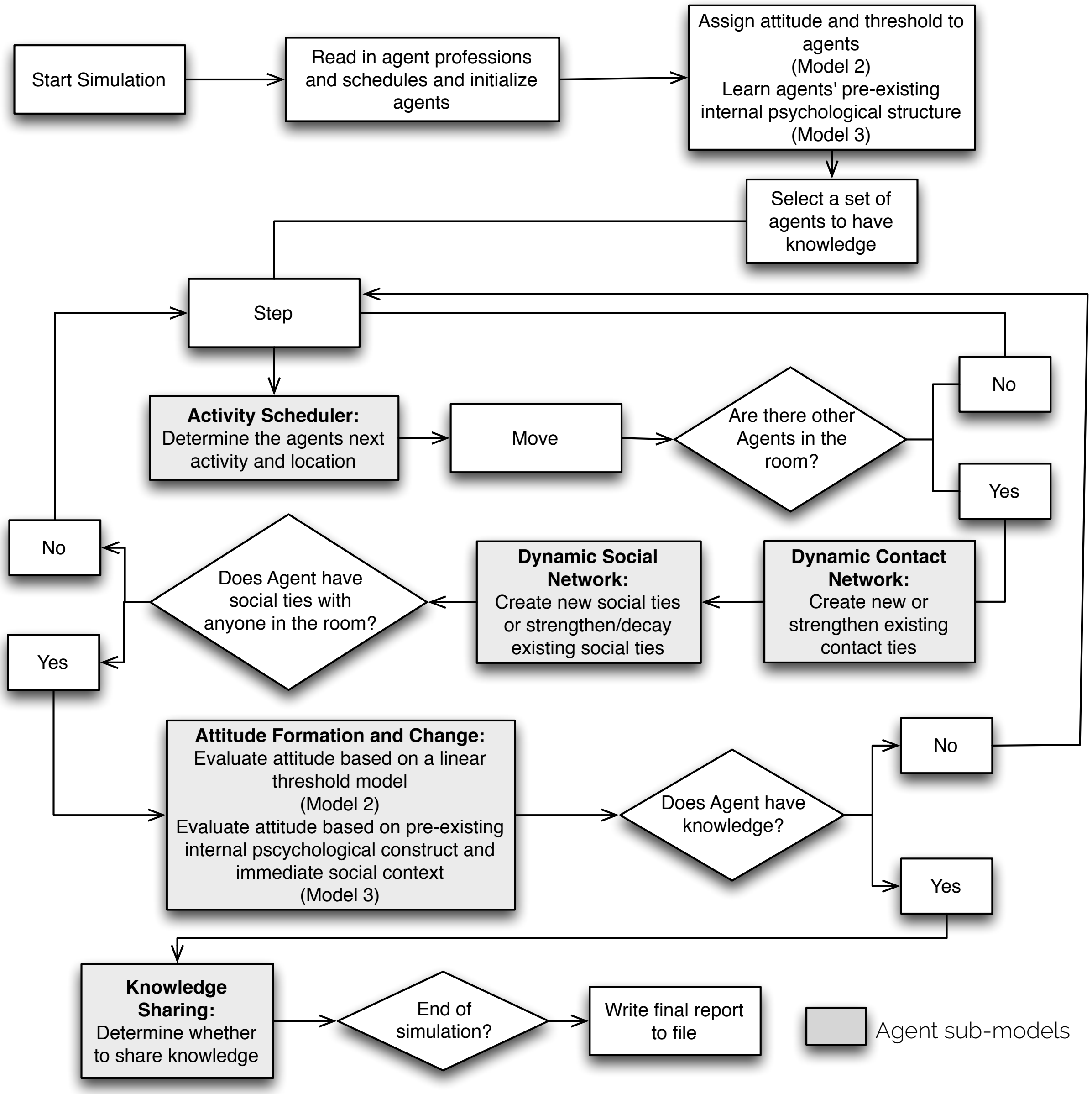


Figure 4. Process flow diagram of the model's key processes.

2 DESIGN CONCEPTS

2.1 Theoretical and empirical background

The main empirical components of the model are the input data used to develop the contact and social networks. Theory, on the other hand, drives the development of the agent's cognitive model of attitude formation in Model 3.

The contact and social networks. We used data collected from an earlier empirical study on disease outbreak within a hospital in southwest Virginia (as discussed in Section 1.2). This data provided the synthetic information model of the hospital, including information on the profession of synthetic healthcare workers and their daily activity schedules. The data was collected from 431 healthcare workers representing 30 different healthcare disciplines (e.g., physicians, nurses, nurse assistants, social workers, physical therapists) by directly observing and shadowing the employees over time spans of 4 to 8 hours during normal hospital operations. Figure 5 provides an example schedule of the activities associated with a day shift Intensive Care Unit (ICU) nurse. As seen in the figure, an ICU nurse will perform a series of different activities throughout the day, including participating in shift meetings, assessing patients, and going to lunch.

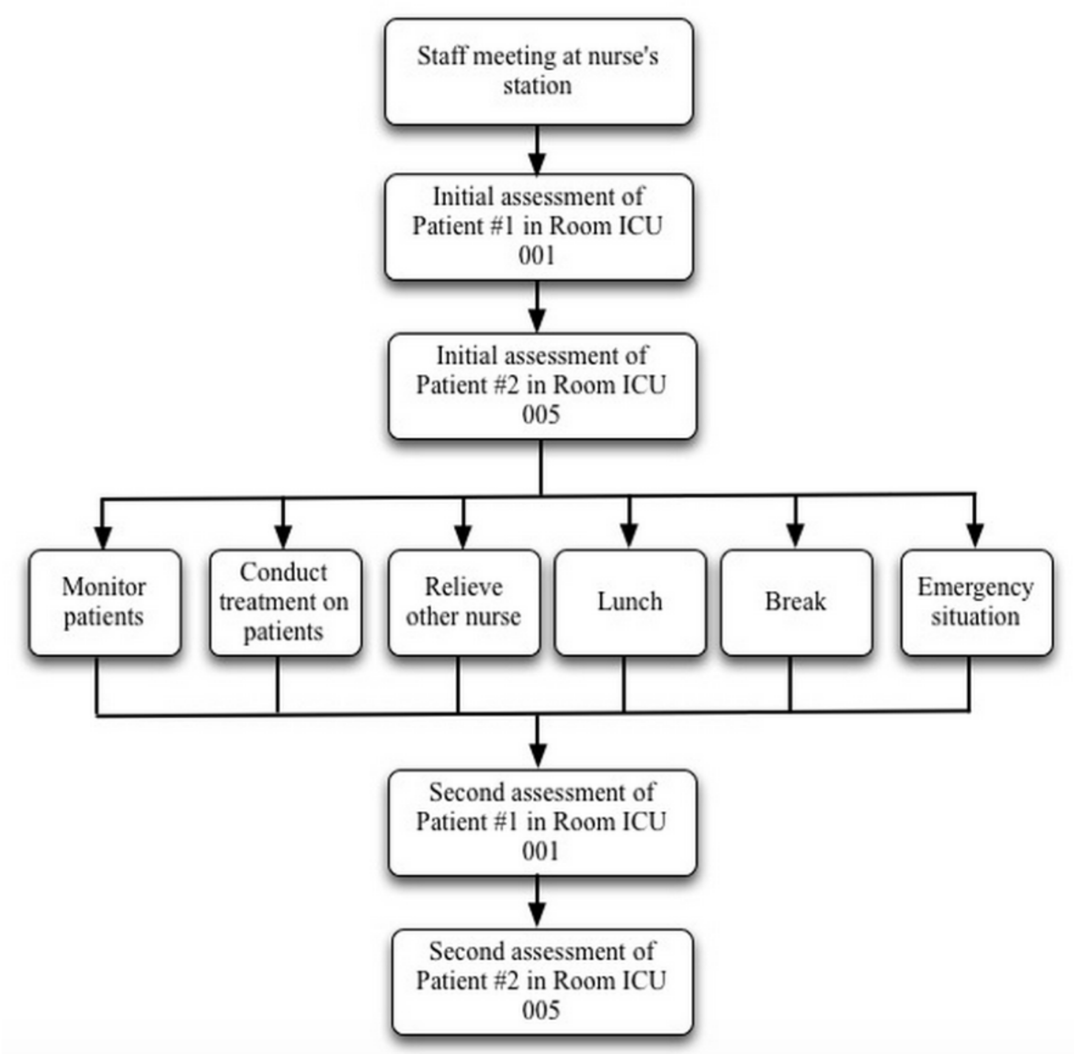


Figure 5. An example of an activity schedule for an ICU nurse (adapted from Jiménez et al., 2013). The ICU nurse here begins the day by attending a staff meeting at the nurse's station and then performs two patient assessments in different ICU rooms. From there, the nurse may perform one of several activities, including monitoring the patients, conducting treatment, or going to lunch. Finally, the nurse will end the day by following-up on the patients.

Through a population builder program in Python, this data was used to develop the synthetic information model of the entire hospital at the individual level. Furthermore, the program stochastically created multiple schedules for healthcare workers in the same profession, such that the same activities will not be performed at the same location at the same time (Jiménez et al., 2013). The synthetic information model includes the hospital's healthcare workers and their movements and contacts over the course of 200 days. This data directly drives development of the contact network. Contact ties in this network are a function of the duration of contacts as determined by their activity schedules. Furthermore, the use of empirical data provides an appropriate level of empirical grounding to the ABM.

The development of social networks is driven by factors such as physical proximity, homophily, and social influence. The contact network is a network of physical proximity, in that contact ties are created or strengthened only when agents are geographically near. Agent similarity (homophily) is measured in terms of profession (e.g., two nurses are more likely to form a stronger social tie than a nurse and doctor) and attitudes towards knowledge sharing. Moreover, there is a feedback effect between the social network and attitudes—agents will form stronger ties with those with which they share a similar attitude (homophily) and they are more likely to be influenced by those with which they have a stronger social tie (social influence).

Attitude formation and knowledge sharing. The agents' cognitive model (Model 3) is grounded in the Theory of Reasoned Action (TRA). We utilized a previously developed computational formalization of TRA using artificial neural networks (ANN) developed to study the dynamics of attitude formation and change (Orr & Plaut, 2014; Orr et al., 2013). This computational model is a reconceptualization of TRA that puts dynamic, in-the-moment attitude formation at the forefront. We use this reconceptualization of TRA to study attitude formation in the context of knowledge sharing. Beliefs on knowledge sharing include, for example, "My knowledge sharing with other organizational members will be an enjoyable experience," and "My knowledge sharing with other organizational members will make me feel valued" (Bock, Zmud, Kim, & Lee, 2005). The attitudinal state of an agent at any point in time is dependent on beliefs, valence units, and a constraint satisfaction process.

Figure 6 represents a conceptual model of an agent's attitude. In this example, attitude is comprised of three beliefs as indicated by the numbers in each circle. A single belief is split between positive and negative valence units. Each valence unit can have a numeric value between 0 and 1 that represents the activation of the valence for that belief. There is an inhibitory connection between the valence units where each valence of a belief constrains the other valence of the same belief to be less active. Constraint satisfaction is the process of satisfying the constraints in an iterative, dynamic way (Read & Miller, 1998).

Constraint satisfaction in this model refers to the connection between different beliefs and the expectation that when certain beliefs are activated, it leads to the activation or inhibition of other beliefs. Each belief has excitatory, inhibitory, or no constraints with every other belief, and these constraints are reflected in connections between the beliefs. These connections are not pre-specified, but learned by the system from past experience through modification of the strength and sign (e.g., inhibitory) of the connections. For instance, given a prior social context in which knowledge sharing was seen as valued and enjoyable, we would expect an excitatory constraint between the beliefs "knowledge sharing will make me feel valued" and "knowledge sharing will be enjoyable." The immediate social context are the agent dyads based in the social network (see discussion above on "The contact and social networks"). An agent's attitude is updated based on the input from the other agent's last attitude and the weight of the social tie between the two agents. As such, the development of dynamic social networks is critical to the process as it directly influences attitude formation. The agent's attitude towards knowledge sharing is therefore a function of the agent's pre-existing attitudinal state and its immediate social context.

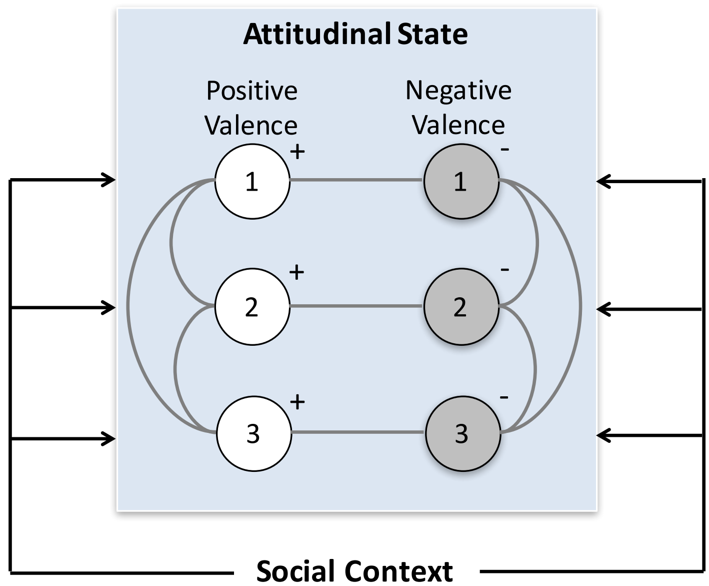


Figure 6. A conceptual model of an agent's attitudinal state. The attitude is comprised of three beliefs numbered 1 to 3 in the figure and each belief is split among a positive (white circles) and negative (grey circles) valence unit. The connections between valence units within a belief is always inhibitory, in that the activation of one valence unit causes the other unit to be less active. The connections between different beliefs can be inhibitory or excitatory. Thus, the activation of one belief can lead to the activation or inhibition of other beliefs. The immediate social context is quantified through the value of the valence units of another agent and weight of the social tie between the two agents.

2.2 Individual decision-making

Decision-making in the model is made at the individual agent level. If an agent has knowledge, the agent must make the decision to share (or not) that knowledge with another agent with whom it is currently interacting with. In Model 1, The decision to share knowledge is based solely on a simple success probability. In Models 2 and 3, the decision to share knowledge is a function of the agent's attitude towards knowledge sharing. This is discussed in detail in Section 3.4.

2.3 Learning

In order to simulate past experiences and create the agents pre-existing attitudinal state in Model 3 (see Figure 6), we provide each agent with a set of training examples during model initialization. This process allows the model to use past experiences to learn the weights between valence units and the biases of the individual units. We discuss this process in detail in Section 3.2. In future implementations, we could additionally turn learning "on" during the simulation. This would allow weights and biases within the agents' attitudinal state to dynamically update during simulation runs as a consequence of agent interactions.

2.4 Individual sensing

The agents are aware of their interactions and their social networks. They are aware of who in their social network currently does not have knowledge, as they will only share with those whom currently do not have the knowledge. In Model 2, they are further aware of the attitudes (+1 or -1) of other agents in its social network. In Model 3, agents know the other agent's belief valence units during an interaction.

2.5 Individual prediction

Prediction is not modeled as the purpose is to explore the dynamics of attitude diffusion and knowledge sharing and the underlying network structures that impact these organizational processes.

2.6 Interaction

As agents go about their routine activities, they interact with other agents. With each interaction, a tie in the contact network is either created or strengthened. This provides an important input into the social network. Each interaction can result in an update to the individual agent's attitude towards knowledge sharing, which subsequently effects the decision on whether to share knowledge (if the agent has knowledge).

2.7 Collectives

The collectives in the model are a result of the structure of the social network. Collectives would include subgraphs, ego networks, network clusters, and the hospital's entire social network which are dynamically developed as agents interact.

2.8 Heterogeneity

Agents are heterogeneous in terms of their profession, attitudinal states, and pre-determined activity schedules, which vary based on profession, shift, and department.

2.9 Stochasticity

Stochasticity is introduced at several places in the model, including at model initialization, during agent interactions, and prior to knowledge sharing. The details can vary depending on whether the simulation is of Model 1, 2, or 3 (see Table 2). Below we discuss areas of stochasticity within different components of the model.

Model initialization. The simulation allows for the option to either randomly select a given number of agents to have the knowledge at initialization or to select specific agents using agent IDs. Models 2 and 3 include attitude formation and change. In Model 2, where attitude is simply a binary variable, agents are randomly assigned either a positive (+1) or negative (-1) attitude towards knowledge sharing. Moreover, attitude thresholds are heterogeneous across agents and is pulled from a normal distribution. In Model 3, stochasticity enters primarily in the process of training the neural network (this is discussed further in Section 3.4). The agents are randomly assigned a prototype (i.e., the neural networks desired output) representing either a positive or negative attitude towards knowledge sharing. Variations of this prototype (i.e., training examples) are stochastically reproduced. After training is complete, a subset of agents are randomly selected to have an initial activation that represents a positive attitude, while the remaining agents' activation is neutral.

Interactions. In order to introduce noise into development of the contact network, which is otherwise pre-defined from agent schedules (see Section 2.1), there is a small probability that an agent will interact with a random agent after an activity. Updates to ties in the contact and social networks occurs between all pairs of agents located in the same room, at the same time. For instance, if three agents are in contact, network ties are updated for three agent dyads. Attitude change and knowledge sharing behavior, however, occurs between only one pair of agents. Thus, after an activity, an agent will randomly select another agent in the room for which to further interact.

Knowledge sharing. Once an agent has selected another agent for which to share knowledge, there is a given probability of success. In the case of Models 1 and 2, the probability of success is set at model initialization. While attitudes are not simulated in Model 1, in Model 2 an agent must also have a positive attitude ($A = +1$) towards knowledge sharing in order to share knowledge. In Model 3, the probability of sharing knowledge is a function of the agent's attitude ($-1 \geq A \geq 1$) and follows a logistic curve (this is discussed further in Section 3.4).

2.10 Observation

At the global level, we monitor several statistics: the direction of knowledge flows (i.e., from/to agent), the individual agents attitude towards knowledge sharing, the interactions (contacts) of the agents, and the social network. Agent attitudes and knowledge flows are collected by time step, while the contact and social networks are collected at the end of each simulation day. The model exports a series of files with these statistics. These output files include the interaction times between all agent dyads, attitude values of each agent, the contact network, the social network, and the agent dyads participating in knowledge sharing. This allows us to assess any trends or changes in behavior. With respect to emergence, structures of the social network (e.g., clusters, cliques, central agents), the diffusion of attitudes, and the patterns of knowledge flows are emergent phenomena.

3 DETAILS

3.1 Implementation details

The model was developed in Mesa, a Python framework for agent-based modeling (Masad & Kazil, 2015), and uses PostgreSQL for storage and retrieval of the input data. Moreover, the cognitive model implemented in Model 3 is a modified version of lens (the light, efficient network simulator), a neural network simulator written primarily in C, developed by Rohde (2002) and used by Orr et al. (2013) to explore attitude formation and change. Given the importance of the neural network approach to modeling agent cognition in the simulation, we discuss here in more detail the technical aspects of the computational re-conceptualization of the Theory of Reasoned Action.

As discussed in Section 2.1, the agents' cognitive model is a computational formalization of TRA using artificial neural networks (ANN). We implemented two versions of this computational formalization: a multilayer feed-forward neural network and a fully recurrent neural network (RNN) that allows for modeling the dynamic process of constraint satisfaction. In a multilayer feed-forward neural network, there is an input layer, one or more hidden layers, and an output layer. This model is static in that inputs from one layer determine the activation (i.e., value) of the unit in the next layer. We can re-draw the notional attitudinal state in Figure 6 to conceptualize it as a fully connected feed-forward neural network with ten sets of beliefs. Figure 7 illustrates this re-conceptualization. The drawing within the blue box depicts the neural network, the processes outside of this box are related to the learning process, which we discuss in the next section (Section 3.2). Note that this is a different implementation of Figure 6 in that while it captures non-linear correlations among beliefs, the connections between beliefs (i.e., the inhibitory connections within each belief and the inhibitory/excitatory connections across beliefs) are not modeled.

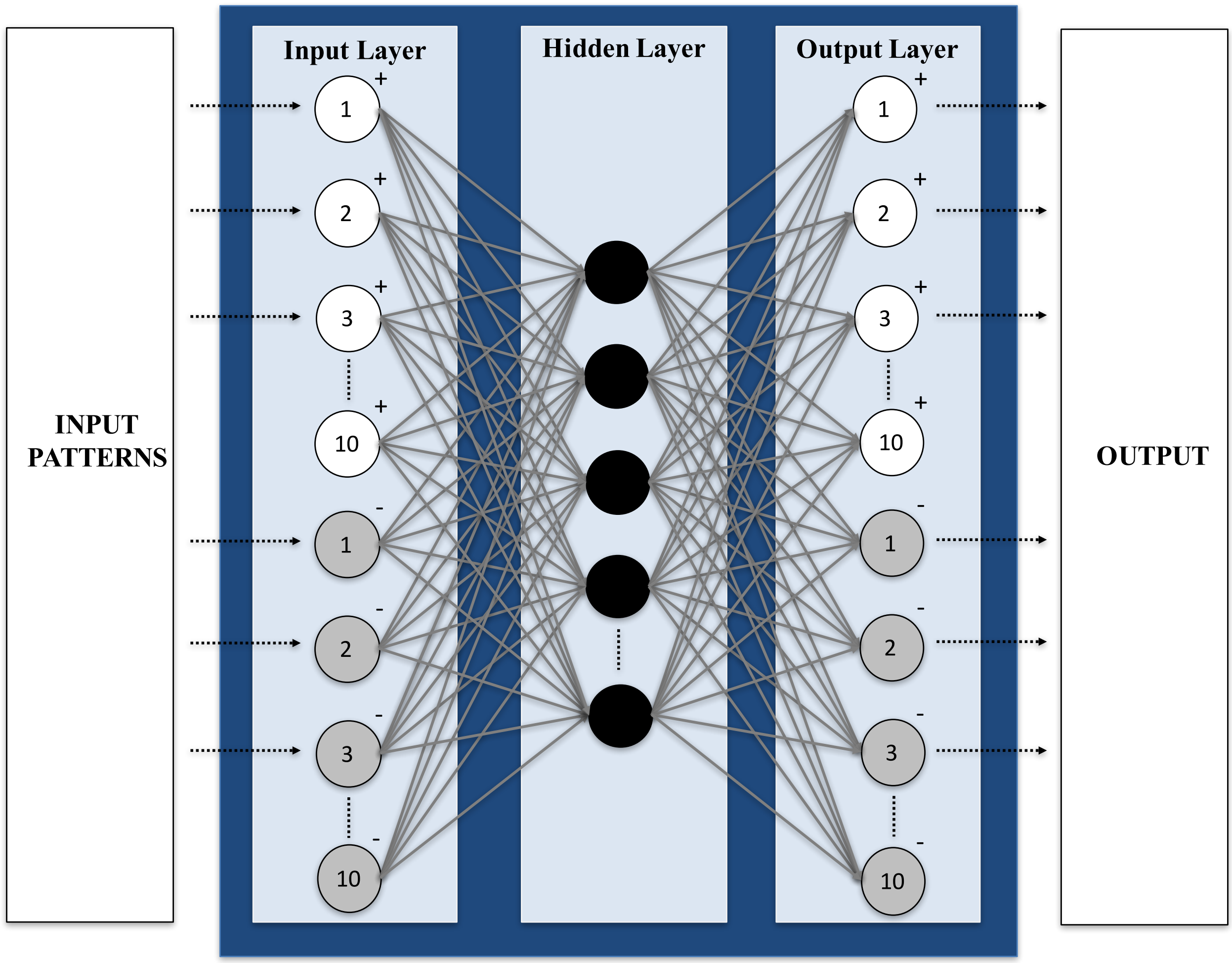


Figure 7. The re-conceptualization of the agent's attitudinal state as a multi-layer feed-forward neural network.

In our simulation, neural network s has three layers – an input layer l_1, a hidden layer l_2, and an output layer l_3. The input and output layers have 20 units $a^l \in \{1, \ldots, 20\}$ representing the positive and negative valence units of ten beliefs. The connection/weight between the k^{th} unit in layer $(l-1)^{th}$ layer to the j^{th} unit in the l^{th} layer is denoted as w_{jk}^l. The bias of the j^{th} unit is denoted as b_j^l. The values of each of these components is the state of neural network s_i, which represents the agent's i current attitudinal state. The value of the bias effects how much new inputs $a^1 \in \{1, \ldots, 20\}$ will influence the activation a_j^l of unit j. We use the logistic activation function shown below to compute the activation of units. The weights w_{jk}^l and biases b_j^l are determined for each agent i at model initialization using a supervised learning algorithm and remain static throughout the course of the simulation.

$$a_j^l = \frac{1}{1 + e^{(-\sum_k w_{jk}^l a_k^{l-1} - b_j^l)}}$$

Valence unit activation ranges between 0 and 1, where 0 is considered not active and 1 is considered highly active. We can think of activation as analogous to the strength of a belief within a person's memory. The stronger the activation, the stronger the belief is activated in memory (Orr & Plaut, 2014).

In the second implementation of the cognitive model, we incorporate a fully recurrent neural network that introduces the constraint satisfaction process (Read & Miller, 1998). Unlike the feed-forward network, co-located units are connected as shown in Figure 8. This allows us to model a theoretical commitment between belief units that influences units to be on or off (i.e., inhibitory and excitatory connections between and within beliefs). This implementation was inspires by Shultz and Lepper (1996) and most closely resembles Orr et al. (2013) computational model of health behaviors. In this model, each unit is an input and an output and error correction occurs within the layer.

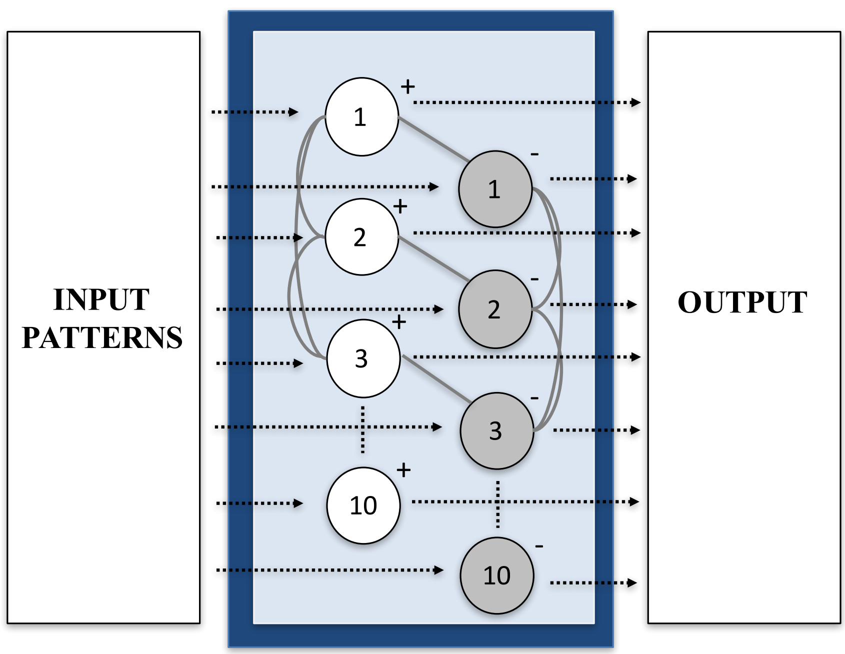


Figure 8. The re-conceptualization a constraint satisfaction model of the agent's attitudinal state as a recurrent neural network.

The recurrent neural network is also dynamic in that state changes occur continuously until equilibrium is reached. The benefit of this is that we can model the internal processes of constraint satisfaction, which is a more realistic representation of attitude formation and change. The activation of a valence unit is also a function of the constraints and activation of other units. Constraint satisfaction therefore captures the idea that the activation of one belief is dependent on the activation of other beliefs, either in an inhibitory or excitatory manner. If the relationship between two units is inhibitory, then the activation of one unit decreases the activation of the other unit. If the relationship is excitatory, then the relation is direct and the increase in the activation of one unit results in an increase in the unit belief.

Up to this point, we have not discussed the role of social context in the model. Social context, or exposure to another's set of beliefs, is one of several inputs into the system. In both versions of the cognitive model, we capture aspects of the agent's immediate social context. This exposure is strictly excitatory in that there is a direct relationship between the activation of units in another's belief and the activation of the analogous unit in the model. This process along with an explanation of the other inputs and the outputs of the system is further discussed in Section 3.4.

3.2 Initialization

Upon model initialization, for Models 1, 2, and 3, agent professions and schedules are pulled from the database and a pre-determined number of agents are selected to have the piece of knowledge. For Model 2, we additionally assign agents a binary attitude $A \in \{-1, +1\}$ towards knowledge sharing. The initialization process for Model 3 is more complex. Earlier we mentioned that the weights and biases of the neural network are determined at model initialization using a supervised learning algorithm. The learning process requires that we input a set of initial conditions as shown in Table 5, including the desired output (prototype), the set of input patterns (training examples), a criterion (minimum error rate), and the maximum number of epochs (training cycles). Prototypes are binary vectors of length 20 representing either a positive attitude or negative attitude. For instance, in a positive prototype the first 10 items in the vector are 1 and the second ten items are 0. The input patterns are 50 training examples of the prototype used for learning. The criterion is the minimum error allowed, where error is calculated using a cost function. One epoch represents one cycle through the 50 training examples. We set the maximum number of epochs at 1,000. Learning will stop once the network is either below the criterion or has reached the maximum number of epochs. Initial input parameter settings were selected based on earlier research by Orr, Ziemer, and Chen (2017).

Table 5. Input parameters used in training the neural network.

Parameter	**Description**
Prototype	The agent's desired output is its prototype. Prototypes are binary vectors of length 20 representing either a positive attitude or negative attitude. Each agent is randomly assigned a positive or negative prototype.
Training examples	Binary vectors of length 20 representing variations/training examples of the prototype.
Criterion	Minimum error rate allowed before learning to stop if the maximum number of epochs has not been reached.
Maximum epochs	Epoch is one training cycle through the input patterns. Learning will stop once the network is below the criterion or has reached the maximum number of epochs

We construct two different training sets (i.e., input patterns) representing the positive and negative prototypes. All things being equal, these training sets lead to an internal bias that is captured by changes in the weights towards positive or negative attitudes of knowledge sharing, respectively. After each training cycle, we calculate the error using the cost function C.

$$C = \frac{1}{n}\sum_x \|y(x) - a(x)\|^2,$$

where n is the total number of training examples, the sum is over the individual training example x, y is the input pattern (i.e., training example), and $a(x)$ is the vector of output activations when x is input. If the error is greater than the criterion and the maximum number of epochs has not been reached, then the weights and bias are adjusted given the error rate. If we are running the feed-forward neural network, we use standard backpropagation, which is a commonly used method for finding weights and biases that minimize the cost function (Nielsen, 2015). If we are running the recurrent neural network, on the other hand, we use a simple localized error correction that accounts for the constraint satisfaction process.

Once learning is complete, we have the weights and biases of each agent's cognitive model that will be used for the course of the simulation. This structure represents the agents pre-existing attitudinal state. Note that this training process can be viewed as a balancing act – train too well and agents will have a very strong bias towards intending or not intending to share knowledge and may not be influenced by their interactions with other agents; train too poorly and agents may be too easily influenced by their interactions. We sought to select initial training conditions that would create biases that fall somewhere in between these two extremes.

While we can think of the agents' learned attitudinal state as representing a tendency towards positive or negative attitudes on knowledge sharing, in order to initialize the model, we still need to provide each agent with an initial input activation. For this purpose, a pre-determined number of agents are seeded with a positive attitude towards knowledge sharing (a vector of length 20 with the first 10 items being 1s and the second ten items being 0s), while the remaining agents have a neutral attitude (a vector of 0s of length 20).

3.3 Input data

The input data to create the agents and assign their schedules was provided by a previous study of the hospital (see Sections 1.2 and 2.1). Table 6 and Table 7 show the structure of the input data used to create the hospital's population and drive the movement of agents across the hospital. This input data is stored in a secure PostgreSQL database. Schedules for individual agents are pulled directly from this database during simulation runs.

Table 6: The agents and their healthcare profession.

Field	Description
Agent ID	The unique identifier of the agent.
Profession	The profession associated with the agent.

Table 7: The agents' activity schedules.

Field	Description
Agent ID	The unique identifier of the agent.
Start	The start time (in seconds) of the activity.
End	The end time (in seconds) of the activity.
Location	A room number identifying the location of the activity

3.4 Submodels

There are four sub-models that together determine agent behavior in Models 1, 2, and 3 (see Table 2). The *Activity Scheduler* determines the agent's current activity and any interactions. The *Dynamic Contact Network* creates new and/or updates existing ties based on these interactions. The *Dynamic Social Network* creates new and/or updates existing social ties based on the contact network and other effects. Attitudes diffuse or update through the *Attitude Formation and Change* sub-model. The *Knowledge Sharing* sub-model determines whether or not agents will share knowledge with interacting agents. Note that first three sub-models are identical, while the *Attitude Formation and Change* and the *Knowledge Sharing* sub-models are different depending on whether we are running Model 1, 2, or 3. We describe each of these sub-models in detail.

Activity Scheduler. Because schedules are pre-determined, running the scheduler consists of querying a database for the activity associated with the current simulation time. From the database, we get a start time, end time, and location of the activity. We then query the database for any other agents at the same location during the same time. If other agents are present, the time that the agent is in the same room as other agents is calculated. This is the interaction time of agent dyads and is used as input into development of the contact network. Given the static nature of these schedules across runs, we chose to introduce a small level of noise into these interactions. After each activity, there is a small probability that the agent will interact with an agent selected at random (as discussed in Section 2.9).

Dynamic Contact Network. As shown in Figure 3, the contact network X^c is a weighted two-mode affiliation network, where agents a_i ($i = 1$ to n number of agents) and a_j ($j = 1$ to $n - 1$) represent the first mode and the events e_{ij} that affiliate the agents represent the second mode (Wasserman, 1994). The weight of the tie w_{ij}^c and w_{ji}^c $\left(w_{ij}^c = w_{ji}^c\right)$ is a function of the duration of events e_{ij} and the total time T that has passed in the simulation as shown below.

$$w_{ij}^c(t) = \frac{\sum_0^t e_{ij}(t)}{T}$$

Dynamic Social Network. The social network X^S in the model is a weighted one-mode network between agents a_i ($i = 1$ to n) and a_j ($j = 1$ to $n - 1$) (see Figure 3). The weight of the tie between agents w_{ij}^s and w_{ji}^s ($w_{ij}^s = w_{ji}^s$, $0 \leq w_{ij}^s, w_{ji}^s \leq 1$) is a function of w_{ij}^c, profession homophily P_{ij}, and attitude homophily A_{ij}. The effect size of each of these parameters on the social tie weight is determined by β_1 and β_2.

$$w_{ij}^s(t) = \beta_1\left[w_{ij}^c(t)\right] + \beta_2\left[e^{|A_{ij}(t-1)|} + P_{ij}\right],$$

$$\text{where } A_{ij}(t-1) = A_i(t-1) - A_j(t-1),$$

$$1 \leq A_i, A_j \leq 1,$$

$$P_{ij} = \begin{cases} 1, \text{if } P_i = P_j \\ 0, \text{otherwise} \end{cases}$$

As agents move and interact across the hospital, the contact and social networks are dynamically updated in parallel.

Attitude Formation and Change and Knowledge Sharing. We implemented three distinct models of *Attitude Formation and Change* and *Knowledge Sharing Behavior* (see Table 2). The agents' behavior is the decision to share (or not) knowledge. Agents will randomly select an agent with whom they are currently interacting with (based on the *Activity Scheduler*). If the selected agent already has the piece of knowledge, nothing happens. Otherwise, the agent will evaluate whether to share the knowledge. We discuss each model is detail here.

Model 1: Cascade Model of Knowledge Sharing. An active node in this model is an agent that has the knowledge. Knowledge spread can occur between two interacting agents. If an agent has the knowledge, it will share that knowledge with an interacting agent that does not have the knowledge with a given success probability. Attitudes are not simulated.

Model 2: Threshold Model of Attitude Diffusion. Attitude diffusion in Model 2 is based on a linear threshold model. Attitude is a binary variable $A \in \{-1, +1\}$, representing whether an agent has a positive or negative attitude towards knowledge sharing. An agents attitude can "flip" if agents are influenced enough by those in their social network with the opposite attitude. In other words, the sum of the strength of the ties of direct connections (i.e., one's ego network) must exceed the agent's Attitude Threshold. Heterogeneous thresholds are randomly assigned from a normal distribution to agents at the beginning of the simulation (see Table 4). Knowledge sharing behavior is similar to Model 1 but with the additional constraint that an agent must have a positive attitude towards knowledge sharing in order to share knowledge. If this occurs, the agent will share the knowledge with a given success probability.

Model 3: Cognitive Model of Attitude Formation. During interactions, agents have the potential to influence another agent's attitude towards knowledge sharing. This interaction between agent dyads captures the immediate social context to be used as input into the cognitive model. Social context is quantified as the strength of the social tie between two agents. The stronger the social relationship between the agents, the more likely they are to influence one another's attitudes towards knowledge sharing. Specifically, we compute the input vector $a_i^1(t)$ as a function of agent $i's$ previous output activation $a_i^L(t-1)$, agent $j's$ output activation $a_j^L(t-1)$, and the weight of the social tie $w_{ij}^s(t)$ between agents i and j, such that:

$$a_i^1(t) = \frac{a_j^L(t-1) - a_i^L(t-1)}{2} w_{ij}^s(t) + a_i^L(t-1)$$

This provides the dynamic input into the cognitive model. The output activation $a_i^L(t)$ of agent i after an interaction with agent j is therefore function of $a_i^1(t)$ and the agents pre-existing attitudinal state s_i.

$$a_i^L(t) = f(s_i, a_i^L(t-1), a_j^L(t-1), w_{ij}^s(t))$$

The resulting output $a_i^L(t)$ is the agent's updated vector of valence unit values. Note that s_i is different depending on the version of the neural network being run. When running the fully recurrent neural network, this state includes constraint satisfaction processes between and within beliefs. We use $a_i^L(t)$ to determine an agent's attitude $A_i(t)$ $(-1 \leq A \leq 1)$ at time t, which is computed simply as the average of the difference between positive and negative valence units.

$$A_i(t) = \frac{\sum_{k=1}^{n/2} a_{ik}^L(t) - \sum_{k=\frac{n}{2}+1}^{n} a_{ik}^L(t)}{n/2},$$

where $a_{ik}^L(t)$ are the valence units at time t and n are the number of valence units. The probability that an agent will share knowledge is a function of A. This is operationalized through the use of a logistic curve.

$$p = \frac{1}{1 + e^{(-Ar)}},$$

where p is the probability that an agent shares knowledge, A is the agent's attitude, and r is the rate at which the curve rises or falls ($0 \leq r \leq 10$). Figure 9 shows the probability distribution under several values of r. When $r = 0$ the probability of sharing knowledge during an interaction is 50% regardless of the value of A and when $r = 10$ the probability of sharing knowledge is near 0% when A is negative and 100% when A is positive. The first case ($r = 0$) should yield behavior that is similar to Model 1, where attitude does not impact the decision to share knowledge and the second case ($r = 10$) is similar to Model 2, where agents will only share knowledge if they have a positive attitude.

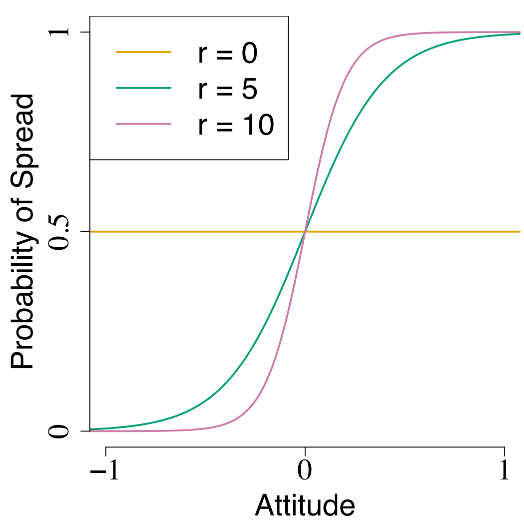


Figure 9. The probability of spread when attitude A is $-1 \leq A \leq 1$ and the rate of the logistic curve r is 0, 5, or 10.

REFERENCES

Adigaa, A., Agashe, A., Arifuzzamana, S., Barrett, C. L., Beckman, R., Bisset, K., & others. (2015). *Generating a synthetic population of the United States*. Retrieved from Blacksburg, VA:

Bock, G.-W., Zmud, R. W., Kim, Y.-G., & Lee, J.-N. (2005). Behavioral intention formation in knowledge sharing: Examining the roles of extrinsic motivators, social-psychological forces, and organizational climate. *MIS Quarterly*, 87-111.

Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., . . . Huth, A. (2006). A standard protocol for describing individual-based and agent-based models. *Ecological modelling, 198*(1), 115-126.

Jiménez, J. M. (2014). *The utilization of macroergonomics and highly-detailed simulation to reduce healthcare-acquired infections.* ((Doctoral Dissertation)

Ph.D. thesis), Retrieved from Virginia Tech Electronic Theses and Dissertations. (Accession Order No. 2014-02-08T09:00:26Z).

Jiménez, J. M., Lewis, B., & Eubank, S. (2013). Hospitals as Complex Social Systems: Agent-Based Simulations of Hospital-Acquired Infections *Complex Sciences* (Vol. 126, pp. 165-178). Santa Fe, NM: Springer.

Masad, D., & Kazil, J. (2015, July 6 - 12). *MESA: an agent-based modeling framework.* Paper presented at the Proceedings of the 14th Python in Science Conference (SCIPY 2015), Austin, TX.

Müller, B., Bohn, F., Dreßler, G., Groeneveld, J., Klassert, C., Martin, R., . . . Schwarz, N. (2013). Describing human decisions in agent-based models–ODD+ D, an extension of the ODD protocol. *Environmental Modelling & Software, 48*, 37-48.

Nielsen, M. A. (2015). *Neural Networks and Deep Learning*: Determination Press.

Orr, M. G., & Plaut, D. C. (2014). Complex systems and health behavior change: Insights from cognitive science. *American Journal of Health Behavior, 38*(3), 404-413.

Orr, M. G., Thrush, R., & Plaut, D. C. (2013). The theory of reasoned action as parallel constraint satisfaction: towards a dynamic computational model of health behavior. *PLoS ONE, 8*(5), e62490.

Orr, M. G., Ziemer, K., & Chen, D. (2017). Systems of Behavior and Population Health. In A. M. El-Sayed & S. Galea (Eds.), *Systems Science and Population Health*. New York, NY: Oxford University Press.

Read, S. J., & Miller, L. C. (1998). *Connectionist models of social reasoning and social behavior*. Mahwah, NJ: Lawrence Erlbaum Associates.

Shultz, T., & Lepper, M. R. (1996). Cognitive Dissonance Reduction as Constraint Satisfaction. *Psychological review, 103*(2), 219-240.

Wasserman, S. (1994). *Social network analysis: Methods and applications* (Vol. 8). Cambridge, MA: Cambridge University Press.