Infectious Hospital Agents: an individual-based simulation framework

Róbert Pethes*, Tamás Ferenci*, Levente Kovács*

*Research and Innovation Center of Óbuda University, Physiological Controls Group,

Abstract—In this paper we present the plan, motivation, background, and the design of an agent-based simulation framework describing the spread of Hospital-Associated Infections (HAIs). We are developing a general simulation environment that is able to model wide range of pathogen transmission scenarios in hospital environment. The elements of the simulation include among others: admission and discharge patients, pathogen transmission via healthcare workers, colonization and infection, modelling hospital events, scheduling treatments, the interventions against HAI spreading. The evolution of the model is tracked in discrete time, and the simulation is driven by stochastic events sampled from predefined distributions. Our aim is to build a general, customisable and extensible simulation environment for the domain of HAIs, therefore the presented design is in Object-Oriented fashion. We implement the system in R using S4 classes, although the design is general. The results of the simulations are time series and transmission networks.

I. Introduction

Healthcare-associated infections are infections that patients get while receiving treatment in healthcare settings. HAIs cause each year around 150,000 deaths in Europe and 100,000 in the USA [1], and drastically increase the length and the cost of the treatment [2]. Furthermore, these infections are often spread by antibiotic-resistance bacteria such as Methicillin-Resistant Staphylococcus Aureus (MRSA), Clostridium difficile, Drug-resistant Streptococcus pneumonia, therefore the treatment of these illnesses are very complicated. This is the reason why prevention gets great emphasis. Infection control is the discipline concerned with preventing HAIs. In infection control, it is essential to understand the dynamics of infection spreading, and predicting the effects of the interventions against HAIs, furthermore clinical trials are very costly and its scope is strongly limited, therefore mathematical modelling and simulations have an increasing importance in decision making.

A. Infection control measures

Infection control measures are all the interventions against HAIs what a hospital can use to prevent the infections. In this section we briefly overview the most common infection control measures:

- Hand hygiene: Studies have shown that hand hygiene is one of the most important factors of HAI spreading [3], [4]. Pittet at al [3] identified the 5 main steps of pathogen transmission via healthcare workers' (HCW) hands and the evidences supporting each steps:

  1) Pathogens are present on the patient's skin or in the patient's immediate environment
  2) Transfer of pathogens to HCW's hands
  3) Pathogens must be survive on HCW's hands for at least several minutes
  4) Hand decontamination (hand washing/rubbing or hand antisepsis) by the health-care worker must be inadequate or omitted entirely
  5) The HCWs contaminated hand(s) must come into direct contact with another patient or with a fomite in direct contact with the patient

  In the prevention of HAI spreading, our aim is to break this sequence. There are two main factors when we are talking about hand hygiene: compliance and the quality of hand decontamination. In brief, the first one refers to the frequency of hand washing/rubbing, the second refers to the quality of it.

- Staff cohorting: This is a method to restrict the transmission network. If we assign a caregiver to a subgroup of patients, we can eliminate the transmission of the pathogen (via the HCW's hand) between patient groups. In other word, we can decrease the role of the 2. and 5. steps of the transmission sequence discussed before.

- Patient isolation: This is an another way to restrict the transmission network. If a patient is found to be colonized or infected, then isolation is justifiable. In this case, there are special hygiene and precaution rules for the HCW who enters or leaves the room of the isolated patient. However, patient isolation is very costly and often practically impossible as the number of colonized/infected patients are increasing

- Surveillance: a collection of methods for collecting every information regarding to the epidemic process. This can contain a lot of data collection methods: swabbing (microbiological sampling) at admission, periodically repeated swabbing of the patients, tracking the patients' temperature chart, illness records, compute the risk factors, etc. The result of the surveillance can be used to ordain special interventions, such as isolation.

- Patient decolonization: We suppose that patients are persistent carriers after colonization, and this fact has an

important role in the pathogen transmission sequence. For example, patients can carry MRSA on their skin, nose or injured skin, therefore they act as a constant source for MRSA transmission [5]. MRSA eradication can be effectively done by using mupirocin and chlorhexidline for decolonization [6]

- Antibiotic usage protocols: antibiotic usage in hospitals has a key role in emergence and spread of HAIs [7], and the applied protocol has a great impact [8]

- Cleaning and sterilization: contaminated environment may also a source of pathogens in hospitals. For example strains of MRSA can survive and remain viable on dust particles or skin scales for many weeks and months [9], and it is also proved that low densities of MRSA can initiate infections [10]

B. Mathematical modelling and simulations in infection control

The most important model types in HAI modelling and simulation are compartment-based and agent-based models [11]–[13]. In a compartment-based model, the population is divided to groups (compartments), and the number of agents of each compartments are tracked in the model. Each compartment represents a stage of the infection history. The most common compartments are Susceptible (S), Exposed (E), Infectious (I) and Recovered or removed (R). Different combinations of these compartments lead to different model structures, and the usual model structures, depending on the aims and the level of details are: S-I, S-I-S, S-I-R and S-I-R-S. Inside a compartment, we suppose homogeneous mixing of the agents. After the compartments are decided, one can define the governing equations of the model, therefore the compartment models are given by closed mathematical equations. These equations can be differential or stochastic equations, and since the nature of the system is highly stochastic and the population size in a hospital is relatively low, therefore the later one is more common.

In an agent-based or individual-based model, we separately represent each actor/agent/individual of the domain, and simulate the interactions between the agents. In case of modelling HAIs, the agents are patients and HCWs, and the interactions are the treatments. The model is driven by discrete (usually stochastic) events. A model like this can be use to predict the effect of the interventions, and therefore, it can support decision making. In an agent-based model we can investigate models where the population is inhomogeneous, and we can define any interaction what we can represent with a computer program, therefore we have greater flexibility in the modelling compared to the compartment-based models, but this flexibility has a price [14]: the simulation time can be very long due to the complexity, the validation of the model is much more difficult, and in addition, it is very hard to plug an agent-based model into an estimation method to estimate the model parameters from recorded time series.

In this paper our focus is on agent-based models. Ferrer at al. [15] built a model that combines the operational and the epi-demiological perspectives to size-up the effect of understaffing and overcrowding in a intensive care unit. In their model they have taken into account the work schedule, sick leaves, workload, fatigue and occupation state of HCWs. Milazzo at al. [14] tested the effect of spatial and personnel cohorting. In [16] the spread of influenza like illness was simulated. The model contains the immunity of the patients and the spatiality of the ward (emergency ward), and they tested the effect of infection control policies. Meng at al. [17] built a transmission model based patient to patient transmission routes, and tested the effect of admission and repeat screening tests, shorter test turnaround time, isolation, and decolonisation. Lee at al. [18] investigated the effect of an MRSA outbreak in a region (Orange Country, California) containing multiple hospitals, they modelled the patient movement and the MRSA spreading between the institutions. Hernbeck at al. [19] tracked the motion of HCWs and the patient-HCW, HCW-HCW interactions using sensor network. They have built an agent-based simulation on the result hospital society network, and investigated the effect of peripatetic HCWs (having large and diverse set of contacts) on the spread of HAIs.
Safdar at al [20] have created a roadmap for future research in HAIs, and argued for the need of more complex models. Our aim is to build an extensible simulation framework that is able to simulate wide range of scenarios in the domain of hospital infection control.

II. SYSTEM DESIGN

The main motivation of this framework is building an extensible simulation environment for simulating the spreading of HAIs. At designing the system we have two main contradictory goals: build an extensible and customisable system while maintaining the complexity. To achieve these aims, after identifying the main building blocks of the system, we represent each component of the system in a general way, and parallel, we provide one or more (commonly used) implementation of the basic building blocks. We implement the system in the R [21] programming environment using S4 classes and general methods (The S programming language has two implementations: S-plus is commercial, R is free. S4 is the fourth version of S, and it supports object-oriented programming. As extension, R also contains the object-oriented elements of S4 [22]), however the design is general, one could implement it in any object oriented language with small modifications.

A. The basic building blocks

This framework provides a programming environment for simulating HAI spreading in hospital settings. The programming environment means a predefined (but customisable and extensible) structure of the simulation, which is implemented using (abstract) classes, and predefined methods. The simulation is agent-based (individual-based), and the agents are patients and HCWs. The basic organization unit of the hospital is the ward, the patients are assigned to a specific ward, the HCWs can be assigned to multiple wards. The colonization

state of the patients can be clear, colonized or infected, and the colonization state of HCWs can be clear or colonized. The system is designed to be able to handle multiple pathogen spreading, and we suppose that patients are reservoirs, therefore if a patient become colonized, she/he remains colonized (unless we do not do a complete decolonization), but HCWs are not reservoirs, so for HCWs we define maximum colonization time. The most important processes of the simulated model are the following:

1) The admission process refers to the process of the arrival of new patients to a hospital ward. New patients can arrive from outside (other hospital, community) or from an another ward of the hospital, and their state can be clear, colonized or infected. At simulation, we refer to the process which decides if a new patient is colonized or not as admission colonization. Since the wards "own" the patients, and the characteristic of each ward can be very different, therefore every ward has it's own admission process (but these processes can communicate with each other in some implementations).

2) The discharge of the patients means the removal of the patient from the ward. In this framework, the discharge process is simple: determine (sample from a predefined distribution) the Length-of-Stay (LOS) of each patient at admission, and if the LOS is elapsed, the patient will be removed from the ward. After admission only one thing can modify the value of LOS: if a patent become infected this increases the LOS value of the patient.

3) Treatment scheduling: Every HCW has a list about the treatments that she/he can perform, and at admission, a list of treatments is assigned to each patients. Every treatment has an urgency value between 0 and 1 (which can increase in time). The treatment scheduling process assigns an available HCW to the patients according to the necessary treatments urgency, and also sample the length of the treatment duration from a predefined distribution. The default implementation is a randomized greedy scheduler algorithm, but one can implement an (in some sense) optimal stochastic scheduler.

4) Treatment processing: The treatment scheduler generates HCW-patient pairs, and assigns a treatment and a treatment duration for each pair. In this step the HCWs perform the treatment. The treatment processing step calls the transmission and infection processes as a sub step, and records the temporal evaluation of the transmission network.

5) Transmission process refers to the pathogen transmission from contaminated HCWs to not contaminated patients or from contaminated patients to not contaminated HCWs. Pathogen transmission or colonization does not means infection, it means only that the pathogen moves from an agent to the other.

6) If a patient become colonized it does not means that she/he will be infected and become sick. Infection process is the process that a patient become infected

and has symptoms.

7) The infection control measures are sub-processes or modifiers of the previously defined processes. For example hand washing/rubbing is performed in the treatment processing part, staff cohorting strategy modifies the treatment scheduler, surveillance can be part almost all of the main processes.

B. Selected parts of the design

In this section we review some highlighted parts of the design, omitting the details.

1) Preparations: *Object* is the base class of all classes what we want to store in the *ObjectDB* storage class. *ObjectDB* is a simple class to store objects, practically we use it to store the representations of patients and HCWs. It is often necessary to draw a random number from a distribution. To be able to handle the distributions flexible, all the distribution samplers has a common base class: *DistrSampler*. The specific distribution sampling methods are wrapped into its subclasses such as *GaussDistrSampler*, *UniformDistrSampler*, etc.
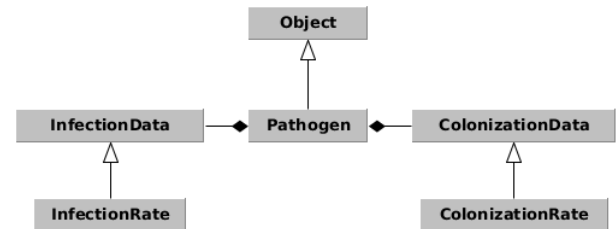

Fig. 1. Pathogen representation

2) Pathogen representation: The class *Pathogen* represents a pathogen in the system (Figure 1). Each pathogen has a name, colonization and infection properties. Colonization properties contain the pathogen specific informations regarding to the colonization process, and infection properties regarding to the infection process.
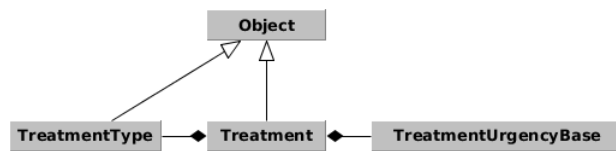

Fig. 2. Treatment representation

3) Treatment representation: The class *TreatmentType* represents a possible treatment (Figure 2). One can define particular treatments, or just an "any treatment" with averaged properties. The *Treatment* class refers to a treatment assigned to a patient. Every instance of a Treatment has an assigned treatment type, and has an *urgency* property, which is implemented via the *TreatmentUrgencyBase* class and it's descendants. The urgency is a number between 0 and 1, and increases as the time elapsed from the last treatment time. The

urgency property has role at the scheduling of the treatments. There are some predefined urgency class, such as: *TreatmentUrgencyStep*, *TreatmentUrgencyExp*, etc. For example the *TreatmentUrgencyExp* type changes as $u(\Delta t) = 1 - e^{\lambda \Delta t}$, where u is the urgency value, Δt is the elapsed time since the last treatment, and λ is a predefined parameter.
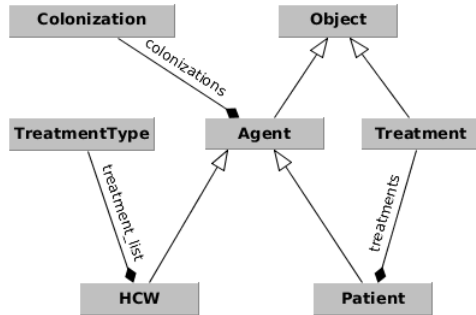

Fig. 3. Agents

4) Patients and HCWs: The structure of agents is sketched on Figure 3. The base class of all agents is the *Agent* class. Every agent contains it's colonization state represented by the *Colonization* class. The *Colonization* class contains the id and the source of the pathogen. An agent can be colonized by multiple pathogens. The *Agent* class has two subclasses: *Patient* and *HCW*, representing a patient and an HCW. An HCW can perform multiple treatments, and a patient can have multiple treatment needs. In the processes of the system, we do not create agents directly, factory classes are the abstraction of instance creation, in this specific case, these are *PatientFactory* and *HCWFactory* classes. One can extend this structure with inheritance from the agent and the factory classes, and plug it in to the other parts of the system.
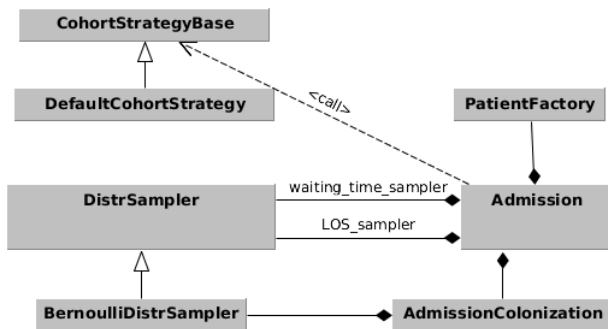

Fig. 4. Admission and disharge

5) Admission and discharge: The admission process is governed by the *Admission* class (Figure 4). The process is driven by random events – arrival of new patients –, where the waiting times are sampled using a predefined *DistrSampler* instance. The admission process (depending on the current implementation) can affect the cohorting process, because one can assign the new patient to a HCW at admission time.

Some portion of the new patients can be colonized, it is also a random process, the default implementation is the *AdmissionColonization* class, which uses Bernoulli distribution sampler to decide the colonization state. The discharge time is also determined at admission time, sampling from the *LOS_sampler* member, which is again a distribution sampler. The LOS value of the patient can be changed only if the patient becomes infected.
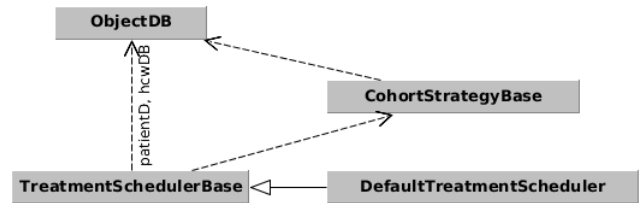

Fig. 5. Treatment scheduler

6) Treatment scheduling: The treatment scheduler (Figure 5) creates (patient, HCW) pairs from the patients who have treatment needs and from the available HCWs who can perform the necessary treatment. The cohorting process naturally modifies the working of the scheduler. The default scheduler is simple and greedy: it shorts the patients according their most urgent treatment needs, and assigns a HCW randomly from the available HCWs who can do the treatment, if the urgency is greater than a predefined threshold.
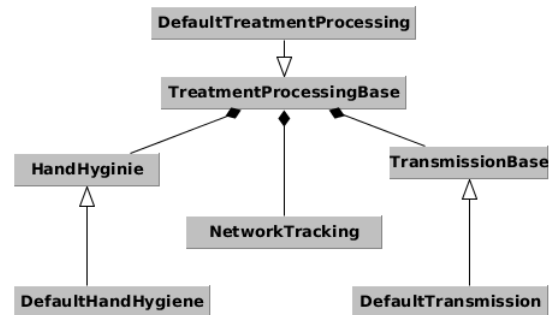

Fig. 6. Treatment processing

7) Treatment processing: After a list of *treatment(patient, HCW)* created, the treatment processor (Figure 6) samples the length of the treatments, and performs them. The hand hygiene infection control measure (one of the descendants of the *HandHyginie* class), and the colonization process (one of the descendants of the *TransmissionBase* class) are sub-processes of the treatment processes. After a pathogen transmission occurred, the system records this transmission as a new edge of the transmission graph using the instance member of the *NetworkTracking* class. At the end of the simulation running one can get the transmission graph and it's temporal evolution from the *NetworkTracking* instance.

8) Simulation: The basic organization unit of the hospital in this model is the ward (Figure 7). The admission, the
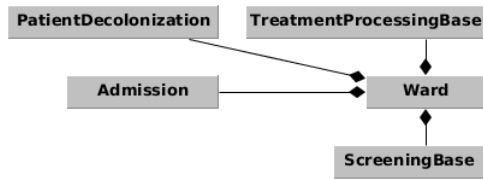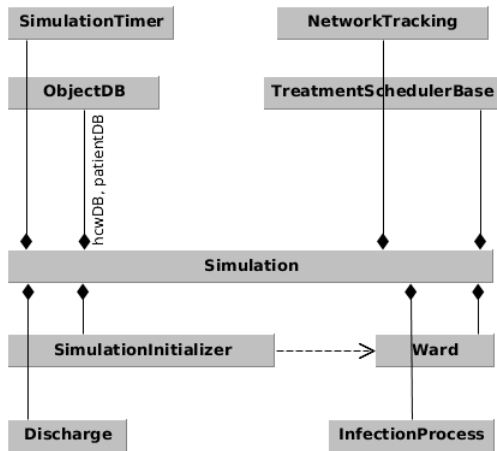
Fig. 7. One ward

Fig. 8. Simulation

treatment processing, the surveillance (screening), and the patient decolonization belong to the ward.

A possible structure of a hospital-level simulation is sketched on Figure 8. *Simulation* class encapsulates all the building blocks:

- Database of HCWs and patients
- *SimulationInitializer* class initialize the simulation: creates the HCWs and the list of initial patients of the wards, etc.
- The *SimulationTimer* class is responsible for the simulation timing
- The scheduling of treatments is done in hospital level, therefore the simulation has a *TreatmentScheduler* object.
- The *NetworkTracker* object
- A *Discarge* object. The simplest discharge methods just simply removes the patient if it's LOS time is elapsed. A sophisticated version can model the patient movements between the wards.
- Representation of wards
- *InfectionProcess* governs the process if a colonized patient become infected or not, and the consequences: increased LOS, additional treatment needs, etc.

9) Simulation scheme: In this section we overview the main loop of the simulation (see Algorithm 1.): Lines 3-5. perform the admission process for each ward in the hospital. In line 6-7 the simulation calls the scheduler method and creates the list of *treatment(patient,hcw)*. The lines 9, 11, 12 calls respec-

tively the treatment processing, screening and decolonization methods for each ward. Line 15 calls the infection process to decide if a colonized patient become infected. Line 16 calls the discharge method, and line 17 remove the colonized status of HCWs if their maximum colonized time is elapsed. In line 18 the simulation time is increased.

Algorithm 1 The main loop of the simulation

1: **procedure** THE MAIN LOOP
2:     **while** timer.run() **do**
3:         **for** ward in wardList **do**
4:             ward.admission(timer.currentTime, patientDB)
5:         **end for**
6:         treatmentList = scheduler.schedule(patientDB,
7:         hcwDB, timer.currentTime)
8:         **for** ward in wardList **do**
9:             ward.treatmentProcessing.process(treatmentList,
10:         patientDB, hcwDB)
11:             ward.screening(patientDB, timer.currentTime)
12:             ward.decolonization(patientDB,
13:         timer.currentTime)
14:         **end for**
15:         infectionProcess.run(patietDB, timer.currentTime)
16:         dischargeProcess.run(patietDB, timer.currentTime)
17:         hcwDecolonization(hcwDB, timer.currentTime)
18:         timer.tick()
19:     **end while**
20: **end procedure**

C. Parametrization

The processes of the simulation is driven by discrete stochastic events, which means they are sampled by some predefined distributions. This subsection contains some results about how to choose these distributions and rates. The values can hardly depends on the ward type and the pathogen. The most discussed case in the literature is the intensive care unit (ICU) and the MRSA pathogen.

- Admission of patients: we can suppose that the probability of more than one patient arriving to the ward in the same time is low, and the $p(\Delta t)$ probability that at least one patient arrives in a Δt time interval is $p(\Delta t) = a\Delta t + o(\Delta t)$ (where a is a positive constant and o is the common asymptotic notion: $f(x) \in o(g(x))$ means $\lim_{x \to \infty} \frac{f(x)}{g(x)} = 0$) The consequence of these assumptions is that arrival of new patients as a stochastic process is a Poission-process [23]. The Poission-process is a counting process, where the waiting times are independent random variables. If the waiting times between the arrival of two patients are sampled independently from a common exponential distribution, then the result is a Poission-process [24]
- Length-of Stay: Statistical parameter fitting methods shows that Lognormal, Weibull and Gamma distributions fit best to LOS empirical data [25]. Additionally, one can

find country-specific average LOS values on the OECD web page: [26]

- Length-of-treatment or the total duration of HCW-patient contact. In the study [19] sensor network was used to track the interaction between the agents, and among others, the duration of the total contact was recorded. They found that the observed distribution of the contact durations was a heavy tailed distribution with median values in the 1-minute range, but the nighttime HCW-patient contacts tend to be longer in duration.
- Bed Occupancy (daily patients per bed) in a ICU can be chosen around 0.83 according the [27] study.
- The probability of admission colonization can be chosen between 0.1 and 0.2 according to [28].
- Pathogen transmission probability in a ICU, during a 30 min patient-nurse visit is about 0.25, and during a 15 min patient-physician visit as about 0.03 [15]

III. Summary

In this study we presented the plan, the motivation, and the design of an agent-based simulation system to simulate HAIs. We briefly reviewed the background of HAIs and their simulation, described the main processes of the system, and sketched some highlighted elements of the object-oriented (OO) design, illustrated with class diagrams. We have implemented the system in the R statistical programming environment using S4 classes, but the presented design is general, one can implement it in any OO programming language with small modifications.

Acknowledgment

References

[1] *European Centre of Disease Prevention and Control (ECDC): Annual Report, Solna, Sweden.* ECDC, 2010.

[2] E. Zimlichman, D. Henderson, O. Tamir, C. Franz, P. Song, C. K. Yamin, C. Keohane, C. R. Denham, and D. W. Bates, "Health care–associated infections: a meta-analysis of costs and financial impact on the us health care system," *JAMA internal medicine*, vol. 173, no. 22, pp. 2039–2046, 2013.

[3] D. Pittet, B. Allegranzi, H. Sax, S. Dharan, C. L. Pessoa-Silva, L. Donaldson, J. M. Boyce *et al.*, "Evidence-based model for hand transmission during patient care and the role of improved practices," *The Lancet infectious diseases*, vol. 6, no. 10, pp. 641–652, 2006.

[4] B. Allegranzi and D. Pittet, "Role of hand hygiene in healthcare-associated infection prevention," *Journal of Hospital Infection*, vol. 73, no. 4, pp. 305–315, 2009.

[5] T. V. Gurieva, M. C. Bootsma, and M. J. Bonten, "Decolonization of patients and health care workers to control nosocomial spread of methicillin-resistant staphylococcus aureus: a simulation study," *BMC infectious diseases*, vol. 12, no. 1, p. 302, 2012.

[6] H. S. Ammerlaan, J. A. Kluytmans, H. F. Wertheim, J. L. Nouwen, and M. J. Bonten, "Eradication of methicillin-resistant staphylococcus aureus carriage: a systematic review," *Clinical Infectious Diseases*, vol. 48, no. 7, pp. 922–930, 2009.

[7] E. M. DAgata, P. Magal, D. Olivier, S. Ruan, and G. F. Webb, "Modeling antibiotic resistance in hospitals: the impact of minimizing treatment duration," *Journal of theoretical biology*, vol. 249, no. 3, pp. 487–499, 2007.

[8] H.-R. Sun, X. Lu, and S. Ruan, "Qualitative analysis of models with different treatment protocols to prevent antibiotic resistance," *Mathematical biosciences*, vol. 227, no. 1, pp. 56–67, 2010.

[9] A. Rampling, S. Wiseman, L. Davis, A. Hyett, A. Walbridge, G. Payne, and A. Cornaby, "Evidence that hospital hygiene is important in the control of methicillin-resistant staphylococcus aureus," *Journal of Hospital Infection*, vol. 49, no. 2, pp. 109–116, 2001.

[10] S. Dancer, "The role of environmental cleaning in the control of hospital-acquired infection," *Journal of hospital Infection*, vol. 73, no. 4, pp. 378–385, 2009.

[11] H. Grundmann and B. Hellriegel, "Mathematical modelling: a tool for hospital infection control," *The Lancet infectious diseases*, vol. 6, no. 1, pp. 39–45, 2006.

[12] E. van Kleef, J. V. Robotham, M. Jit, S. R. Deeny, and W. J. Edmunds, "Modelling the transmission of healthcare associated infections: a systematic review," *BMC infectious diseases*, vol. 13, no. 1, p. 1, 2013.

[13] T. N. Doan, D. C. Kong, C. M. Kirkpatrick, and E. S. McBryde, "Optimizing hospital infection control: the role of mathematical modeling," *Infection Control & Hospital Epidemiology*, vol. 35, no. 12, pp. 1521–1530, 2014.

[14] L. Milazzo, J. L. Bown, A. Eberst, G. Phillips, and J. Crawford, "Modelling of healthcare associated infections: a study on the dynamics of pathogen transmission by using an individual-based approach," *Computer methods and programs in biomedicine*, vol. 104, no. 2, pp. 260–265, 2011.

[15] J. Ferrer, M. Salmon, and L. Temime, "Nosolink: an agent-based approach to link patient flows and staff organization with the circulation of nosocomial pathogens in an intensive care unit," *Procedia Computer Science*, vol. 18, pp. 1485–1494, 2013.

[16] M. Laskowski, B. C. Demianyk, J. Witt, S. N. Mukhi, M. R. Friesen, and R. D. McLeod, "Agent-based modeling of the spread of influenza-like illness in an emergency department: a simulation study," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 6, pp. 877–889, 2011.

[17] Y. Meng, R. Davies, K. Hardy, and P. Hawkey, "An application of agent-based simulation to the management of hospital-acquired infection," *Journal of Simulation*, vol. 4, no. 1, pp. 60–67, 2010.

[18] B. Y. Lee, S. M. McGlone, K. F. Wong, S. L. Yilmaz, T. R. Avery, Y. Song, R. Christie, S. Eubank, S. T. Brown, J. M. Epstein *et al.*, "Modeling the spread of methicillin-resistant staphylococcus aureus (mrsa) outbreaks throughout the hospitals in orange county, california," *Infection Control & Hospital Epidemiology*, vol. 32, no. 06, pp. 562–572, 2011.

[19] T. Hornbeck, D. Naylor, A. M. Segre, G. Thomas, T. Herman, and P. M. Polgreen, "Using sensor networks to study the effect of peripatetic healthcare workers on the spread of hospital-associated infections," *Journal of Infectious Diseases*, p. jis542, 2012.

[20] N. Safdar, D. J. Anderson, B. I. Braun, P. Carling, S. Cohen, C. Donskey, M. Drees, A. Harris, D. K. Henderson, S. S. Huang *et al.*, "The evolving landscape of healthcare-associated infections: recent advances in prevention and a road map for research," *Infection Control & Hospital Epidemiology*, vol. 35, no. 05, pp. 480–493, 2014.

[21] "The R Project for Statistical Computing," www.r-project.org.

[22] C. Genolini, "A (not so) short introduction to s4," Technical report, The R-Project for Statistical Computing, Tech. Rep., 2008.

[23] S. Karlin and H. E. Taylor, *A first course in stochastic processes.* Academic press, 1975.

[24] L. Lakatos, L. Szeidl, and M. Telek, *Introduction to queueing systems with telecommunication applications.* Springer Science & Business Media, 2012.

[25] A. Marazzi, F. Paccaud, C. Ruffieux, and C. Beguin, "Fitting the distributions of length of stay by parametric models," *Medical care*, vol. 36, no. 6, pp. 915–927, 1998.

[26] "OECDiLibrary," www.oecd-ilibrary.org.

[27] F. Schwab, E. Meyer, C. Geffers, and P. Gastmeier, "Understaffing, overcrowding, inappropriate nurse: ventilated patient ratio and nosocomial infections: which parameter is the best reflection of deficits?" *Journal of Hospital Infection*, vol. 80, no. 2, pp. 133–139, 2012.

[28] W. C. Albrich and S. Harbarth, "Health-care workers: source, vector, or victim of mrsa?" *The Lancet infectious diseases*, vol. 8, no. 5, pp. 289–301, 2008.