A discrete event simulation procedure for validating programs of requirements
The case of hospital space planning

Çubukçuoglu, Cemre; Nourian, Pirouz; Sariyildiz, Sevil; Tasgetiren, Mehmet Fatih

DOI
10.1016/j.softx.2020.100539

Publication date
2020

Document Version
Final published version

Published in
SoftwareX

SoftwareX 12 (2020) 100539

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

A discrete event simulation procedure for validating programs of requirements: The case of hospital space planning

Cemre Cubukcuoglu [a,b,*], Pirouz Nourian [a], I. Sevil Sariyildiz [a], M. Fatih Tasgetiren [c]

[a] Faculty of Architecture and the Built Environment, Chair of Design Informatics, Delft University of Technology, Delft, The Netherlands
[b] Department of Interior Architecture and Environmental Design, Faculty of Architecture, Yasar University, Izmir, Turkey
[c] Department of International Logistics Management, Faculty of Business, Yasar University, Izmir, Turkey

ARTICLE INFO

Article history:
Received 2 December 2019
Received in revised form 15 March 2020
Accepted 8 June 2020

Keywords:
Discrete event simulation
Program of requirements
Hospital space planning

ABSTRACT

This paper introduces a Discrete-Event Simulation (DES) tool developed as a parametric CAD program for validating a program of requirements (PoR) for hospital space planning. The DES model simulates the procedures of processing of patients treated by doctors, calculating patient throughput and patient waiting times, based on the number of doctors, patient arrivals, and treatment times. In addition, the tool is capable of defining space requirements by taking hospital design standards into account. Using this tool, what-if scenarios and assumptions on the PoR about space planning can be tested and/or validated. The tool is ultimately meant for reducing patient waiting times and/or increasing patient throughput by checking the match of the layout of a hospital with respect to its procedural operations. This tool is envisaged to grow into a toolkit providing a methodological framework for bringing Operations Research into Architectural Space Planning. The tool is implemented in Python for Grasshopper (GH), a plugin of Rhinoceros CAD software using the SimPy library.

Code metadata

Current code version	*Beta*
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX_2019_367
Legal Code License	*MIT*
Code versioning system used	*git*
Software code languages, tools, and services used	*Python, IronPython*
Compilation requirements, operating environments & dependencies	*SimPy library,* https://anaconda.org/mutirri/simpy *Numpy library,* https://numpy.org/
If available Link to developer documentation/manual	*none*
Support email for questions	cemre.cubukcuoglu@yasar.edu.tr

Current software version	*Beta*
Permanent link to executables of this version	https://github.com/CemreTUDelft/DES_PoR_Tool
Legal Software License	*MIT*
Computing platforms/Operating Systems	*Rhino3D & Grasshopper3D or Jupyter on Windows or Mac*
Installation requirements & dependencies	*SimPy library,* https://anaconda.org/mutirri/simpy *Numpy library,* https://numpy.org/
If available, link to user manual – if formally published include a reference to the publication in the reference list	https://simpy.readthedocs.io/en/latest/
Support email for questions	cemre.cubukcuoglu@yasar.edu.tr

* Corresponding author at: Department of Interior Architecture and Environmental Design, Faculty of Architecture, Yasar University, Izmir, Turkey.
E-mail addresses: c.cubukcuoglu@tudelft.nl, cemre.cubukcuoglu@yasar.edu.tr (C. Cubukcuoglu), p.nourian@tudelft.nl (P. Nourian), i.s.sariyildiz@tudelft.nl (I.S. Sariyildiz), fatih.tasgetiren@yasar.edu.tr (M.F. Tasgetiren).

https://doi.org/10.1016/j.softx.2020.100539

1. Introduction

1.1. Motivation and problem statement

Discrete event simulation (DES) is a method that mimics the operations of real and/or proposed systems as an ordered series of events. During the simulation, each event shows a specific change in the system's state at separate points in time. DES allows decision-makers to build complex models of operations, to quickly test what-if scenarios on their operations, and to explore alternative ways of implementing new strategies [1]. Space is almost always constrained and that indeed provides a sense of maximum efficiency/throughput. However, in addition to space, hospitals are the kind of buildings whose operations might be much more costly than their building/space in their lifetime. The tool provides the means to simulate the volumes of operations in terms of flows of people and waiting times etc. such that space planners and/or architects can 'engineer' the program of requirements before realizing it; i.e., to get a sense of scale and time-wise implications of decisions to provide the right amount of spaces for certain uses. DES has been widely used in hospital planning [2,3] for modeling patient flow processes, estimating patient satisfaction, optimizing the healthcare human/physical resources and reducing healthcare costs [4,5]. In patient flow models, the most commonly considered inputs are patient admission schedules, admission rules, patient routing, flow schemes, facility, and staff resources; the most common outputs are patient throughput, patient waiting times, physician utilization, staff and facility utilization [6–8]. In particular, DES has been mostly utilized to model outpatient areas in the literature. Zhao and Lie [9] proposed the DES to model patient flow in the emergency department for resource utilization and reducing department crowding. Oddoye et al. [10] modeled medical assessment unit (MAU) using the DES for evaluating the length of stay and bed utilization. Reynolds et al. [11] modeled the outpatient dispensing process by using the DES and evaluated the staffing levels and workload. Haji and Darabi [12] modeled Ear, Nose, Throat clinic and the appointment system using the DES for reducing outpatient waiting times. Al-Araidah et al. [13] utilized the DES method for reducing delays in the ophthalmology outpatient department. Rau et al. [14] focused on the strategic capacity planning of an outpatient physical therapy service using the DES to reduce waiting times and length of stay. Weerawat et al. [15] built a DES model for an orthopedic outpatient area for the assessment of wait times to see doctors. Baril et al. [16] used the DES to model outpatient flows and appointment scheduling in an orthopedic clinic to reduce patient lead times, maximize the number of patients seen by the orthopedist doctor. Best et al. [17] used the DES to improve patient flow, improve patient throughput and reduce the length of stay in emergency and acute care. Pan et al. [18] proposed a DES model to represent the patient and information flow in an ophthalmic outpatient department and aims at reducing patient waiting times. Baril et al. [19] utilized the DES for improving patient trajectories in a hematology–oncology department by reducing patient delays. Dan et al. [20] applied the DES to outpatient pharmacy queuing problem where the factors to evaluate queue system are average waiting time, the average length of the queue, average utilization of servers and length of busy time. Babashov et al. [21] developed a DES model of the patient journey for reducing the patient waiting times to consult or treatment in radiation oncology department. Shin et al. [22] implemented a DES model to characterize the patient flow in an emergency department with the aim of minimizing patient length of stay (LoS), number of handoffs, staff utilization levels, and cost. Recently, Moretto et al. [23] used the DES to improve service planning in orthopedic and neurosurgical outpatient department. Baril et al. [24] built the DES model to analyze ambulatory patient length of stay in an emergency department with a resource and staff planning. In Cho et al. [25], authors proposed a decision support framework for a clinician's schedule using the DES for an outpatient area of a hospital by assessing the patient waiting times for consultation.

The system is not meant to make decisions but to help the planners engineer their space plans. Hospital operations consist of a critically important sequence of medical activities and procedures where any delay in patient care may prove fatal. However, often it is unavoidable. Therefore, the fitness of the building for the planned operations of the hospital plays an important role in the layout design [26]. DES provides an explicit mechanism for testing the degree to which the building matches with its operations [6], i.e. by simulating the patient flow patterns and thus hinting towards how estimated patient waiting times can be reduced by a functional/logical layout during the conceptual design phase of a hospital. In other words, DES provides an explicit way of modeling and understanding the functionality of the building, which can be used to inform the process of shaping the building accordingly.

Space requirements pertain to hospital design standards, such as minimum space area constraints and light requirements. Therefore, such requirements are among important factors that should be fed to the layout design process (a.k.a. space planning). The area requirements generally differ based on several medical staff. For example, according to the hospital design standards of Turkey, outpatient waiting areas must be minimum 12 m^2 for 1 doctor, and a minimum of 24 m^2 for 2 doctors and an additional 5 m^2 should be considered for each additional doctor [27]. On the other hand, space requirements must also handle people's expectations e.g. waiting areas must be large enough for the [estimated] number of visitors waiting in a queue.

The motivation for this paper can be summarized as follows:

- Hospital standards affect their design considerations (space requirements).
- Patient flow and waiting times can be estimated through DES.
- Space requirements differ from one context to another (country-specific regulations, the size of the program of requirements, etc.).
- Space requirements can be adjusted to reduce the waiting times (or wider waiting area can be needed if patient waiting times are too high) using what-if scenarios in DES.
- DES provides a clear understanding of critical procedures within a building and thus can inform the spatial layout process.

1.2. State of the art

There are some DES tools, namely Arena [28], ExtendSim [29] that are mostly used in industrial engineering for system simulation [30]. Other recent DES tools are FlexSim [31,32], SimEvents [33,34], SIMUL8 [35,36], MedModel [37,38] and Manpy [39–41]. Specifically, MedModel focuses on healthcare simulations. Similar to the DES_PoR tool, Manpy is another Python-based open-source tool, which is developed on top of SimPy. However, they do not present straightforward ways of interaction with CAD software applications commonly used in architectural practice.

Utilizing DES in architectural design was suggested by Wurzer et al. [42], especially for early-stage conceptual design. Their approach integrates DES simulation into a hospital space planning tool. In addition, Vos et al. [26] presented an evaluation method using Discrete Event Simulation for the assessment of hospital layout design from the viewpoint of operations management to test if the building design provides for the efficient operation of patient care.

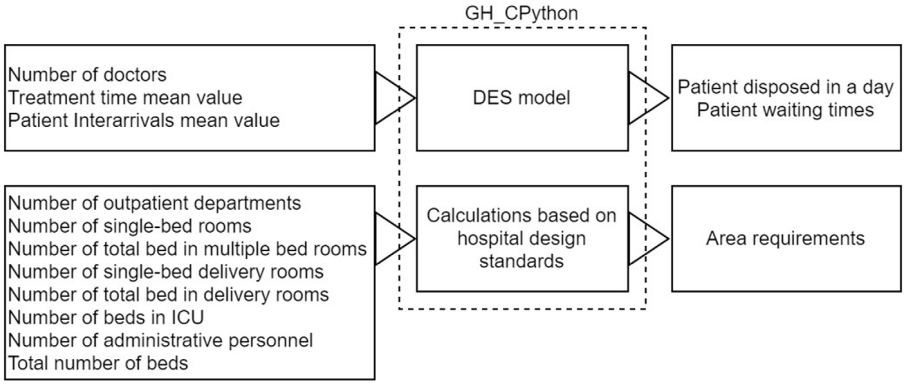

Fig. 1. Components of the tool.

1.3. Contributions

The primary contribution of the paper is that users of this tool will be able to utilize the abovementioned functionalities in a Parametric Modeling environment, called Grasshopper (a plug-in of Rhinoceros CAD software). Using this tool, architects/decision-makers/designers can practically define space requirements to design a hospital considering hospital design standards. Rapid integration of meter square information to the model plays an essential role to define the program of requirements during the conceptual design phase in this tool. Providing real-time Discrete-Event Simulation at the same time, the tool can give feedback on the performance indicators (patient waiting times, patient throughput), which is translatable into the likely functional/logical performance of the hospital plan layout patterns. Furthermore, the outputs of this tool can be used in architectural design optimization models as an input to be minimized or maximized.

We applied the tool to a hospital in Izmir, Turkey (to be a newly built hospital in Seljuk). Therefore, minimum space requirements for each hospital department are taken from Turkey's hospital standard using an assumed building program. The DES part of the tool was applied to outpatient departments. In this part, patient throughput and patient waiting times are taken as outputs, which can be also defined as performance indicators of our tool. The number of staff is considered as an input, which also affects the spatial configuration and space area of the outpatient departments. The mutual interaction between the operational planning of the hospital and its spatial planning is central to our approach. Throughout this work, we show how DES models can be used as design-decision-support tools to create a bridge between architecture and hospital management. Specifically, the tool can help answer these questions:

(1) How to test the match of the layout with the operational planning of the hospital to reduce the patient waiting times and increase the patient throughput in a day?
(2) How to scale and adjust the program of requirements (PoR) of hospitals through DES and hospital standards that are affected by hospital operational planning?

2. Software description

The core functionality of this tool is to validate the program of requirements of a hospital referring to the patient-flow model by discrete-event simulation and hospital design standards and make this tool compatible with parametric design models created in GH algorithmic modeling for Rhinoceros CAD software. The first functionality is to simulate patient flows in the outpatient area. An outpatient department consists of many subsections with distinct specialties, and thus it has some special properties distinct from other departments in a hospital in its operations; and so it is a commonly modeled area by DES in the literature for capacity planning [2]. The outpatient department is typically the most active area in terms of human flow with many types of uncertainties in patient arrivals, treatment patterns and service time. Waiting times can be defined as the time that patients arrive in the clinic and wait until doctors call them for treatment. And they are depending on both operational and space planning of the hospital. For the patient-flow modeling with DES, Simpy library [43] is used in GH_CPython [44] component that implements CPython (the standard implementation of Python3.7) codes inside Grasshopper (GH). Therefore, GH_CPython scripting is used in this tool. Fig. 2 shows a screenshot from the GH interface. The component named as DES_PoR tool in this figure is run a python code, which is available in Table 1. The number sliders at the left part are defined as input values of this component and the panels at the right side of the component are the outputs of the simulation and PoR values. Users can change these number sliders and simultaneously see the change in output values during the decision-making and planning processes for hospital space planning.

Besides, both waiting areas and treatment rooms have strict hospital design standards as described above. Therefore, we formulate another desired functionality as an automatic generation of area requirements based on hospital design standards. It is straightforward to adjust input parameters (e.g. staff planning) in the proposed tool and to quickly see the change in performance indicators e.g. area requirements. Simply, this tool is easy to use by architects/designers since it is compatible with GH. Users of this tool can write scripts inside the GH_CPython component and update the model according to the needs of the focused hospital. For example in our case study, the hospital is planned as 16 outpatient departments with a discrete waiting area for each of them. Components are also adjustable that means if one more input is needed, users can add it easily e.g. this situation can happen if the number of doctors varies in each outpatient area.

To sum up, this tool facilitates the conceptual [digital] design phase of hospital space planning through computational tools. The intent here is not to present a fully working space planning tool, but rather to supports architects/designers as a decision-support tool in the early design-decision making process and to highlight the use of hospital management tools in spatial design. The components of this tool are illustrated in Fig. 1.

As a test-case for the tool, we consider hospital standards pertained to its spatial planning as follows [27]:

- Outpatient rooms must be min 16 m^2.
- Outpatient waiting areas must be

Table 1
GH_CPython code snippet.

```
import random
import numpy as np
import simpy
TotalBeds_Num=SingleBedRoom_Num + MultipleBedRoom_BedNum +
SingleBed_DeliveryRoom_Num + MultipleBed_DeliveryRoom_BedNum + ICU_BedNum
#Area calculation of treatment rooms & waiting areas in each outpatient
department
if NUM_DOCTORS == 1:
    Outpatient_Dept_Area = (16 * NUM_DOCTORS) + 12 #treatment rooms +
waiting areas
elif NUM_DOCTORS == 2:
    Outpatient_Dept_Area = (16 * NUM_DOCTORS) + 24 #treatment rooms +
waiting areas
elif NUM_DOCTORS > 2:
    Outpatient_Dept_Area = (16 * NUM_DOCTORS) + 24 + (5 * (NUM_DOCTORS -
2)) #treatment rooms + waiting areas
RANDOM_SEED = 50 #seed number of simulation
SIM_TIME = 480 #simulation time in minutes: 8 hours
data_wait = []
data_patientdisposed=[]
#create outpatient components
class Outpatient(object):
    def __init__(self, env, num_doctors, treatmenttime):
        self.env = env
        self.doctor = simpy.Resource(env, num_doctors)
        self.treatmenttime = treatmenttime
    def treat(self, patient):
        yield self.env.timeout(random.expovariate(1/TREATMENTTIME))
#create patient arrivals
def patient(env, name, pr):
    print('%s arrives at the outpatient department at %.2f.' % (name,
env.now))
    arrivetime=env.now
    with pr.doctor.request() as request:
        yield request
        print('%s enters the outpatient department at %.2f.' % (name,
env.now))
        entertime=env.now
        yield env.process(pr.treat(name))
        print('%s leaves the outpatient department at %.2f.' % (name,
env.now))
        print('%s waiting time %.2f.' % (name, entertime-arrivetime))
        data_wait.append(entertime-arrivetime)
#create treatment process
def setup(env, num_doctors, treatmenttime, t_inter): #t_inter means that
one patient arrives in each t_inter minutes
    outpatient_department = Outpatient(env, num_doctors, treatmenttime)
    #create 4 initial patients
    for i in range(4):
        env.process(patient(env, 'patient %d' % i,
outpatient_department))
    #create more patients
    while True:
        yield env.timeout(np.random.poisson(t_inter)) #simulate patient
arrivals in a poisson process with a lambda value of 7
        i += 1
```

(*continued on next page*)

  - min 12 m^2 for 1 doctor
  - min 24 m^2 for 2 doctors
  - additional 5 m^2 for each additional number of doctors.

- There must be min 6 elevators in 60–200 bed hospitals.
- There must be min 9 elevators in 201–350 bed hospitals.
- One-bed patient rooms must be min 9 m^2.
- Patient wards must be min 7 m^2 per each bed.
- One-bed delivery patient rooms must be min 12 m^2.
- Delivery patient wards must be min 10 m^2 per each bed.
- ICU units must be min 12 m^2 per each bed.
- Neonatal ICU units must be min 6 m^2 per each bed.

Table 1 (*continued*).

```
        env.process(patient(env, 'patient %d' % i,
outpatient_department))
        data_patientdisposed.append(i)
#run the simulation
random.seed(RANDOM_SEED)
env = simpy.Environment()
env.process(setup(env, NUM_DOCTORS, TREATMENTTIME, T_INTER))
env.run(until=SIM_TIME)
#get the outputs of DES from a,b
a=data_patientdisposed #number of patients disposed (treated) from the
system
b=data_wait #number of waiting times
#calculate space requirements of hospital units
c=Outpatient_Dept_Area*OutpatientDept_Num #total area of all outpatient
dept.s
d=(SingleBedRoom_Num*9)+(MultipleBedRoom_BedNum*7) #One-bed patient rooms
must be min 9 m2.Patient wards must be min 7 m2 per each bed.
e=(SingleBed_DeliveryRoom_Num*12)+(MultipleBed_DeliveryRoom_BedNum*10)
#One-bed delivery patient rooms must be min 12 m2.Delivery patient wards
must be min 10 m2 per each bed.
f=ICU_BedNum*12 #ICU units must be min 12 m2 per each bed.
g=Administrative_Num*random.randint(8,12) #Administrative offices must be
8-12 m2 for each personnel.
h=TotalBeds_Num+(TotalBeds_Num*0.2) #Bunker area = (number of beds) +
(number of beds*20%)
#There must be min 6 elevators in 60-200 bed hospitals.
#There must be min 9 elevators in 201-350 bed hospitals.
if TotalBeds_Num >= 60 and TotalBeds_Num <= 200:
    MIN_ELEV = 6
elif TotalBeds_Num > 200 and TotalBeds_Num <= 350:
    MIN_ELEV = 9
i=MIN_ELEV
#get the outputs of PoR from c,d,e,f,g,h,i
```

Table 2
The procedure implemented in the DESPoR tool.

```
Begin
      Put the simulation settings
      Create DES model
            Create patient arrivals
            Create a treatment process using GH inputs
            Run the simulation
      Get the outputs of DES

      Calculate space requirements for each unit
            Formulations using GH inputs and hospital design standards
      Get the outputs of PoR
End
```

- Administrative offices must be 8–12 m^2 for each personnel.
- Bunker area = (number of beds) + (number of beds*20%)

Snippets of GH_CPython code for the DES model and calculation of area requirements are shown in Table 1 and the pseudo-code of this tool is given in Table 2.

3. Illustrative examples

The presented method has been implemented as a computational tool and is currently being tested with a sample hospital. Since we were considering Turkey's hospital standard, we selected a hospital in Izmir as a case study, which is going to be newly built in Seljuk. There are 16 specialties in the outpatient department and 2 inpatient departments as surgery and medicine.

We construct a DES model based on the procedures for patients to be treated by some doctors in outpatient departments in a hospital. In the DES model shown in Fig. 2, input values are taken the same for all outpatient departments. The number of doctors is input for every outpatient department as part of the resource planning process in this paper. In the literature [9,14,45,46], the patient inter-arrivals to an outpatient department have typically been modeled as Poisson arrivals and the treatment time durations, which refers to the duration of consultation with the physician for this case, have been mostly modeled as exponential.

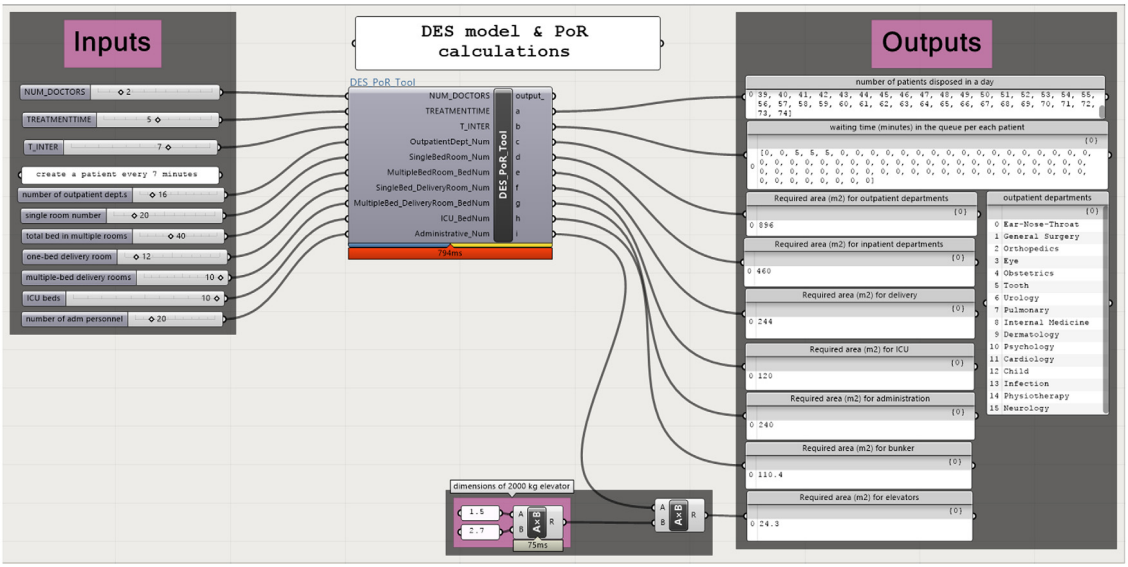

Fig. 2. A screenshot of the DES tool as an editable Python script inside the Grasshopper interface. The tool can be combined with procedural workflows in this environment to give or take variabale outputs and/or inputs.

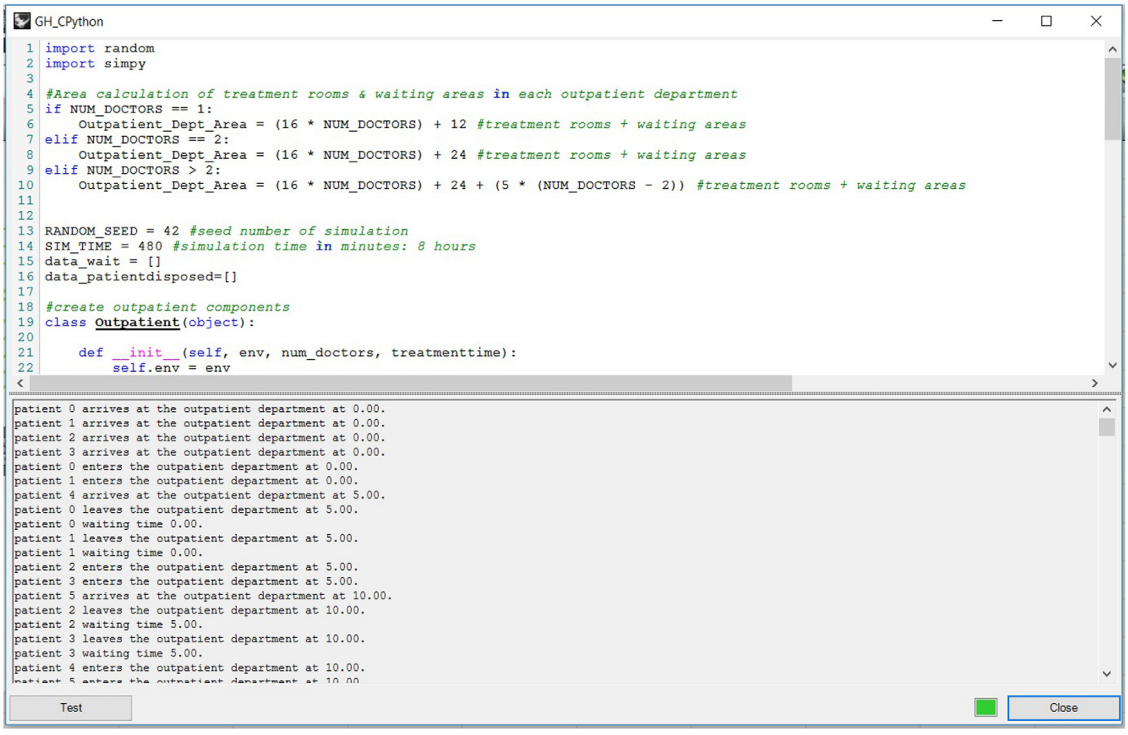

Fig. 3. The script and some exemplary results.

Therefore, in this paper, we assumed that the interarrival times are followed a Poisson process with 5 min mean value and the treatment times are exponentially distributed with 7 min mean value. We took 5 replications with 5 different seed numbers for the simulation. Minimum, maximum, average and standard deviation of the waiting time results in each replication have been recorded in Table 3.

According to the simulation results, the minimum waiting time is obtained as 11.87 min in replication-5 with a seed number of 50. In this case, there are 72 patients treated in a day in each department. When the number of doctors is increased to 3, patient waiting time is reduced to 1.57 min. Then, the area required for the outpatient area is changed from 896 to 1232 m^2. On the other hand, when the number of single-bed rooms in the inpatient department is changed from 20 to 30, the area required for the inpatient department is increased from 460 to 550 m^2.

When the mean value of treatment time is taken as 12 min and the number of doctors is taken as 3 in each outpatient department, the total waiting time of patients is 89.5 min so that 1232 m^2 is needed for all outpatient departments in the hospital. After simulating with 4 doctors, waiting time is reduced to 3.5 min and the required area for outpatient departments is increased to 1568 m^2. The latter case requires 336 m^2 more area for the outpatient department with 86 min less waiting time. Furthermore, when the number of doctors is taken as 5, the maximum estimated waiting time becomes zero. This case

Table 3
Simulation results for each replication.

	Rep.1	Rep.2	Rep.3	Rep.4	Rep.5	Min	Max	Avg	Std. Dev.
Total waiting time in a day (minutes)	13.07	29.38	49.56	24.57	11.87	11.87	49.56	25.69	15.29

has great advantages from the point of waiting time, however, the outpatient area is very large (1904 m^2), which may not be handled with space available constraint, which is 1800 m^2 in this case. Therefore, decision-makers could be recommended to plan 4 doctors in each outpatient department. As an example of sizing an inpatient department, when the number of single rooms is 20 and the number of beds in multiple rooms is 40, the required area is 460 m^2 for patient wards. If the number of single rooms is increased to 30 and the number of beds in multiple rooms is reduced to 30, then the area requirement amounts to 480 m^2. In the second scenario, the area requirement is slightly larger than the first one. It could be concluded that it would be more advantageous to select the second scenario for inpatient departments as it has 10 more single inpatient rooms; because single patient rooms reportedly have a better effect on the patient healing process then the multiple-bed rooms. This is how the tool allows for checking different scenarios during the decision-making process of hospital space planning (see Fig. 3).

4. Impact

This tool is designed to handle two main research questions introduced in Section 1. Since this tool is created in a procedural modeling environment, it can be utilized to test the match of the building layout with the operational planning of the hospital in computational design workflows. Specifically, what-if scenarios on hospital planning can be tested through an interactive input–output connection over this tool. During such tests, changes in (patient-focused) performance indicators can be interactively observed by a decision-maker/architect by changing the decisions regarding input parameters. Our tool enables both identifying and validating building programs based on performance indicators obtained by the DES model and hospital design standards. In particular, it helps to answer the existing research question introduced by Vos et al. [26]. This tool represents a step towards closing the gap between these two worlds by representing a more practical way from the point of architectural design practitioners. The tool is a plug-in of Rhino CAD software, i.e. the de facto standard environment of choice for computational design in architecture, and at the same time portable to open environments because it is a Python script also implemented in a Jupyter notebook. This tool can be used in space planning of hospital designs in practice (as a Rhino/Grasshopper plugin) as well as in research studies (as a Python script). It is envisioned to be an extensible and open-source tool for hospital planning. As it is freely available on a public domain (https://github.com/CemreTUDelft/DES_PoR_Tool), users can extend the tool to their liking or integrate it in their works, such as focusing on different case studies (types of hospitals) and system models (e.g. queue model) or different users (e.g. administrative people flow). In addition, users of this tool can estimate the flowrates between each space with the help of the DES. More flows between spaces require more closeness between their locations. Therefore, these flowrates are significant parameters to identify closeness ratings between each space, which also helps to create relationship charts (REL-charts) to be used during the space planning process. To the authors' knowledge, currently, there does not exist any publicly available DES model and building PoR test tool for hospitals in GH, especially for computational design and space planning.

5. Conclusions

This paper introduced a new plug-in tool for grasshopper algorithmic modeling (GH) in the Rhinoceros CAD program. This tool is envisioned as a part of a larger suite of tools to bridge the gap between Operations Research and Architectural Design, specifically for computational space planning of hospitals. A discrete event simulation (DES) is implemented in this tool for patient-flow modeling. Outputs of the DES model facilitate the validation of the match between space planning elements such as PoR and REL charts and the operational logic of the building. For future work, hospital standards of different countries will be added to the model. Different users such as nurses can be added to the model. DES model can be improved considering different types of scenarios, such as patient-flows in an emergency. Eventually, this tool can be generalized for other types of complex buildings such as airports.

6. Limitations

We focus on the outpatient department in this paper. In intensive care units or inpatient departments, patient flow logic is different. Treatment processes are different. Also, outpatient departments are working 8 h a day. Simulation runs for 8 h. Other departments' operations run for 24 h in a day. Therefore, the DES is specialized for outpatient, where the waiting times are the most important in these places.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Shakoor M. Using discrete event simulation approach to reduce waiting times in computed tomography radiology department 9(1);177–81. 2015, http://waset.org/publications/10000287/using-discrete-event-simulation-approach-to-reduce-waiting-times-in-computed-tomography-radiology-department.

[2] Günal MM, Pidd M. Discrete event simulation for performance modelling in health care: A review of the literature. J Simul 2010;4(1):42–51. http://dx.doi.org/10.1057/jos.2009.25.

[3] Jun JB, Jacobson SH, Swisher JR. Application of discrete-event simulation in health care clinics: A survey. J Oper Res Soc 1999;50(2):109–23.

[4] Tiwari V, Sandberg WS. Perioperative bed capacity planning guided by theory of constraints. In: 2016 Winter simulation conference. 2016. p. 1894–903.

[5] Zuo X, Li B, Huang X, Zhou M, Cheng C, Zhao X, et al. Optimizing hospital emergency department layout via multiobjective tabu search. IEEE Trans Autom Sci Eng 2019;16(3):1137–47.

[6] Gibson IW. An approach to hospital planning and design using Discrete Event Simulation. In: Proceedings - winter simulation conference. 2007. p. 1501–09. https://doi.org/10.1109/WSC.2007.4419763.

[7] Kougias P, Tiwari V, Berger DH. Use of simulation to assess a statistically driven surgical scheduling system. J Surg Res 2016;201(2):306–12.

[8] Zhou J, Wang J, Wang J. A simulation engine for stochastic timed petri nets and application to emergency healthcare systems. IEEE/CAA J Autom Sin 2019;6(4):969–80.

[9] Zhao L, Lie B. Modeling and simulation of patient flow in hospitals for resource utilization. Norway: Telemark University College N-3901; 2008.

[10] Oddoye JP, Jones DF, Tamiz M, Schmidt P. Combining simulation and goal programming for healthcare planning in a medical assessment unit. European J Oper Res 2009;193(1):250–61.

[11] Reynolds M, Vasilakis C, McLeod M, Barber N, Mounsey A, Newton S, et al. Using discrete event simulation to design a more efficient hospital pharmacy for outpatients. Health Care Manage Sci 2011;14(3):223–36.

[12] Haji M, Darabi H. A simulation case study: Reducing outpatient waiting time of otolaryngology care services using VBA. In: 2011 IEEE International conference on automation science and engineering. 2011. p. 525–30.

[13] Al-Araidah O, Boran A, Wahsheh A. Reducing delay in healthcare delivery at outpatients clinics using discrete event simulation. Int J Simul Model 2012;11(4):185–95.
[14] Rau C-L, Tsai P-FJ, Liang S-FM, Tan J-C, Syu H-C, Jheng Y-L, et al. Using discrete-event simulation in strategic capacity planning for an outpatient physical therapy service. Health Care Manage Sci 2013;16(4):352–65.
[15] Weerawat W, Pichitlamken J, Subsombat P. A generic discrete-event simulation model for outpatient clinics in a large public hospital. J Healthc Eng 2013;4(2):285–305.
[16] Baril C, Gascon V, Cartier S. Design and analysis of an outpatient orthopedic clinic performance with discrete event simulation and design of experiments. Comput Ind Eng 2014;78:285–98.
[17] Best AM, Dixon CA, Kelton WD, Lindsell CJ, Ward MJ. Using discrete event computer simulation to improve patient flow in a Ghanaian acute care hospital. Am J Emerg Med 2014;32(8):917–22.
[18] Pan C, Zhang D, Kon AWM, Wai CSL, Ang WB. Patient flow improvement for an ophthalmic specialist outpatient clinic with aid of discrete event simulation and design of experiment. Health Care Manage Sci 2015;18(2):137–55.
[19] Baril C, Gascon V, Miller J, Côté N. Use of a discrete-event simulation in a Kaizen event: A case study in healthcare. European J Oper Res 2016;249(1):327–39.
[20] Dan Z, Xiaoli H, Weiru D, Li W, Yue H. Outpatient pharmacy optimization using system simulation. Procedia Comput Sci 2016;91:27–36.
[21] Babashov V, Aivas I, Begen MA, Cao JQ, Rodrigues G, D'Souza D, Lock M, Zaric GS. Reducing patient waiting times for radiation therapy and improving the treatment planning process: a discrete-event simulation model (radiation treatment planning). Clin Oncol 2017;29(6):385–91.
[22] Shin SY, Brun Y, Balasubramanian H, Henneman PL, Osterweil LJ. Discrete-event simulation and integer linear programming for constraint-aware resource scheduling. IEEE Trans Syst Man Cybern 2017;48(9):1578–93, https://www.mathworks.com/products/simevents.html.
[23] Moretto N, Comans TA, Chang AT, O'Leary SP, Osborne S, Carter HE, et al. Implementation of simulation modelling to improve service planning in specialist orthopaedic and neurosurgical outpatient services. Implement Sci 2019;14(1):78.
[24] Baril C, Gascon V, Vadeboncoeur D. Discrete-event simulation and design of experiments to study ambulatory patient waiting time in an emergency department. J Oper Res Soc 2019;70(12):2019–38.
[25] Cho M, Song M, Yoo S, Reijers HA. An evidence-based decision support framework for clinician medical scheduling. IEEE Access 2019;7:15239–49.
[26] Vos L, Groothuis S, Van Merode GG. Evaluating hospital design from an operations management perspective. Health Care Manage Sci 2007;10(4):357–64. http://dx.doi.org/10.1007/s10729-007-9034-7.
[27] Republic of Turkey Ministry of Health. Türkiye Sağlık Yapıları Asgari Tasarım Standartları 2010 Yılı Klavuzu. 2010.
[28] Automation R. Arena user's guide. Milwaukee: Rockwell Software; 2004.
[29] Krahl D. ExtendSim 7. In: 2008 Winter simulation conference. 2008. p. 215–21.
[30] Jadrić M, Ćukušić M, Bralić A. Comparison of discrete event simulation tools in an academic environment. Croat Oper Res Rev 2014;5(2):203–19.
[31] FlexSim. 2020, https://www.flexsim.com/, [Retrieved 12 March 2020].
[32] Nordgren WB. Flexible simulation (Flexsim) software: Flexsim simulation environment. In: Proceedings of the 35th conference on winter simulation: Driving innovation. 2003. p. 197–200.
[33] Gray MA. Discrete event simulation: A review of simEvents. Comput Sci Eng 2007;9(6):62–6.
[34] SimEvents. 2020, https://www.mathworks.com/products/simevents.html, [Retrieved 12 March 2020].
[35] Concannon K. Simulation modeling with SIMUL8. Visual8 Corporation; 2003.
[36] SIMUL8. 2020, https://www.simul8.com/, [Retrieved 12 March 2020].
[37] Keller LF. MedModel-specialized software for the healthcare industry. In: Proceedings of winter simulation conference. 1994. p. 533–37.
[38] MedModel. 2020, https://www.promodel.com/products/medmodel/, [Retrieved 12 March 2020].
[39] Dagkakis G, Papagiannopoulos I, Heavey C. Manpy: an open-source software tool for building discrete event simulation models of manufacturing systems. Softw - Pract Exp 2016;46(7):955–81.
[40] Dagkakis G, Heavey C, Robin S, Perrin J. ManPy: An open-source layer of DES manufacturing objects implemented in SimPy. In: 2013 8th EUROSIM Congress on modelling and simulation. 2013. p. 357–63.
[41] ManPy. 2020, http://www.manpy-simulation.org/, [Retrieved 12 March 2020].
[42] Wurzer G, Lorenz WE, Röler M, Hafner I, Glock B, Bruckner M, Popper N. MODYPLAN: Early-stage hospital simulation based on treatment chains. IFAC-PapersOnLine 2015;28(1):868–73. http://dx.doi.org/10.1016/j.ifacol.2015.05.144.
[43] Simpy. Discrete event simulation for Python. 2002, https://Simpy.Readthedocs.Io/En/Latest/Index.Html. https://simpy.readthedocs.io/en/latest/index.html.
[44] AbdelRahman M. Gh_cpython. 2017, https://Www.Food4rhino.Com/App/Ghcpython. https://www.food4rhino.com/app/ghcpython.
[45] Bhattacharjee P, Ray PK. Patient flow modelling and performance analysis of healthcare delivery processes in hospitals: A review and reflections. Comput Ind Eng 2014;78:299–312.
[46] Hamrock E, Paige K, Parks J, Scheulen J, Levin S. Discrete event simulation for healthcare organizations: a tool for decision making. J Healthc Manage 2013;58(2):110–24.