

Northumbria Research Link

Citation: Mileva Boshkoska, Blijana, Damij, Talib, Jelenc, Franc and Damij, Nadja (2015) Abdominal surgery process modeling framework for simulation using spreadsheets. *Computer Methods and Programs in Biomedicine*, 121 (1). pp. 1-13. ISSN 0169-2607

Published by: Elsevier

URL: <https://doi.org/10.1016/j.cmpb.2015.05.001>
<<https://doi.org/10.1016/j.cmpb.2015.05.001>>

This version was downloaded from Northumbria Research Link:
<http://nrl.northumbria.ac.uk/id/eprint/29272/>

Northumbria University has developed Northumbria Research Link (NRL) to enable users to access the University's research output. Copyright © and moral rights for items on NRL are retained by the individual author(s) and/or other copyright owners. Single copies of full items can be reproduced, displayed or performed, and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided the authors, title and full bibliographic details are given, as well as a hyperlink and/or URL to the original metadata page. The content must not be changed in any way. Full items must not be sold commercially in any format or medium without formal permission of the copyright holder. The full policy is available online: <http://nrl.northumbria.ac.uk/policies.html>

This document may differ from the final, published version of the research and has been made available online in accordance with publisher policies. To read and/or cite from the published version of the research, please visit the publisher's website (a subscription may be required.)



ELSEVIER

journal homepage: www.intl.elsevierhealth.com/journals/cmpb

Abdominal surgery process modeling framework for simulation using spreadsheets

Biljana Mileva Boshkoska^{a,*}, Talib Damij^b, Franc Jelenc^c, Nadja Damij^a

^a Faculty of Information Studies, Ulica talcev 3, Novo mesto, Slovenia

^b Faculty of Economics, University of Ljubljana, Kardeljeva ploščad 17, Slovenia

^c University Medical Centre, University of Ljubljana, Ljubljana, Slovenia

ARTICLE INFO

Article history:

Received 6 January 2015

Received in revised form 6 May 2015

Accepted 8 May 2015

Keywords:

Business process management

Spreadsheets

Activity table modeling technique

Simulations

ABSTRACT

We provide a continuation of the existing Activity Table Modeling methodology with a modular spreadsheets simulation. The simulation model developed is comprised of 28 modeling elements for the abdominal surgery cycle process. The simulation of a two-week patient flow in an abdominal clinic with 75 beds demonstrates the applicability of the methodology. The simulation does not include macros, thus programming experience is not essential for replication or upgrading the model. Unlike the existing methods, the proposed solution employs a modular approach for modeling the activities that ensures better readability, the possibility of easily upgrading the model with other activities, and its easy extension and connectives with other similar models. We propose a first-in-first-served approach for simulation of servicing multiple patients. The uncertain time duration of the activities is modeled using the function “rand()”. The patients movements from one activity to the next one is tracked with nested “if()” functions, thus allowing easy re-creation of the process without the need of complex programming.

© 2015 The Authors. Published by Elsevier Ireland Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Healthcare process simulation is an interesting technique for research and practice. Many different approaches exist for modeling a healthcare system [1], such as system dynamics, agent-based modeling, optimization techniques, electronic medical records information system [2], disease management-process modeling approach [3] and discrete event simulation (DES) [4]. The paper focuses on the usage of DES. The model considers the limitations of the Slovenian hospital for abdominal surgery in terms of the number of available beds, the mean time of activity durations and the number of available doctors.

This paper proposes a two-step general modeling framework for healthcare process simulation. Firstly, the Activity table modeling technique is applied. Secondly, spreadsheets are used for simulation of the patient’s flow. The applicability of the proposed modeling framework is demonstrated on a real-life case study of the abdominal surgery process. The modeling starts with defining the process workflow using the activity table technique [24]. Next, the spreadsheet is used as a tool to simulate the healthcare process.

The motivation for using the spreadsheet is threefold. Firstly, a spreadsheet provides a complete overview of the process at any point of time (the location of the patient, the availability of beds, and the availability of resources). Secondly,

* Corresponding author. Tel.: +386(0)59 087 902; fax: +386 (0)59 08 79 03.

E-mail address: biljana.mileva@fis.unm.si (B.M. Boshkoska).

<http://dx.doi.org/10.1016/j.cmpb.2015.05.001>

0169-2607/© 2015 The Authors. Published by Elsevier Ireland Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

the simplicity of a spreadsheet allows upgrading the model by Excel users, without the need of performing complex programming tasks. The combination of the simplicity of spreadsheets and their supportive presentation of data has proven to be of a higher importance than complex and detailed model representation for decision support on operational-level models [5], and for policy-level models [6]. Furthermore, a spreadsheet allows easy upgrade of the model to connecting different hospital departments, thus leading to a step forward toward modeling the hospital as a whole system.

Thirdly, the issue has a particular practical merit. Many of the discrete event simulation models described in the literature were built for a specific hospital, and are never reused due to economic reasons or the complexity of the models [1]. In order to overcome this obstacle, this paper provides enough details of the simulation approach so that it can be easily recreated by an Excel user, for different hospital departments. The aim of this paper is therefore to test the proposed two-step modeling framework on a specific healthcare process, not focusing on the comparison to other commercial products. Suggested framework points to an optimal management of resources and relative time, as well as supports a review of the presented structure of the activities.

The approach applied in this work starts by modeling the process discussed using the Activity table modeling technique, and continues by transmitting the process model developed into a spreadsheet in order to simulate it.

The starting point for simulation of the healthcare process was based on the simulations performed by [7] who use one spreadsheet to simulate an emergency department for patient flow. We changed and improved their approach in the following manners:

1. The simulation approach is performed without macros, thus no Visual Basic programming experience is required in order to replicate the model, or to upgrade it; however, a proficient knowledge of using formulas in Excel is required.
2. The solution employs a modular approach for modeling of the activities. It means that each activity is described in a separate worksheet instead of using only one. This way we ensure: better readability; the possibility of easily upgrading the model with other activities; and its easy extension and connectives with other Clinic's models;
3. Three different modeling elements are introduced depending on their functionality: activity, transaction, and gate; in addition we provide two models for activities with multiple inputs;
4. The serving priority of the activities with multiple inputs is solved with the principal of first-in-first-served.

2. Activity table modeling technique

There are more than fifty tools [8] or methods that are mentioned in the literature that are available for business process modeling, which makes the selection of the proper tool difficult. An extensive comparison can be found in [9] where an empirical review of existing methodologies, tools and techniques for business process change are given, including a reference framework to assist the positioning of tools

and techniques that improve re-engineering strategy, people, management, structure and the technology dimensions of business processes.

The criteria that is of greatest interest when comparing the techniques for business process modeling are: the relevance, the completeness, and the simplicity of the built models [10,11]. Additionally, the process modeling tools must be capable of showing interconnections between the activities and conducting a decomposition of the processes. A common way to represent a consecutive flow of activities in health-care which includes all encounter criteria is through workflows [12,22,23] or mapping [13]. Here we chose Activity table modeling technique as a tool for representing the activities and the existing relations among them.

The Activity Table consists of two parts. The first part (first few columns) provides information about each activity listed by defining a number of parameters that describe the activity. The second part is a tabular-graphical representation of the process. To develop the activity table, information about process functioning should be gained during interviews that are organized with knowledgeable employees.

2.1. Part 1

In the columns of Part 1 of the Activity Table a number of the following parameters are defined for each activity(*i*), where *i* ranges from 1 to the number of activities.

- *Description*. A short and precise description of what exactly is the work carried out by the activity defined in row(*i*)
- *Time*. The expected duration needed for activity(*i*) to be processed and accomplished
- *Rule*. One or more constraints or rules that must be satisfied in order for activity(*i*) to be performed
- *Input/Output*. Input(s) and output(s) of activity(*i*)

2.2. Part 2

Process: For each process, a new Activity Table is created. This table represents a model called the "as-is" process model, which describes the behavior of the process as it exists in reality. Thus, the name of the process selected is written in the first column of the Activity Table (see Fig. 1).

If the process is large and complex, then it should be partitioned into a set of sub-processes, which are in this case listed in the second column of the Activity Table. Each process consists of a number of work processes.

Work Processes: A work process is a process that is performed within a department or organization unit. In this step all work processes of the process discussed are listed in the Work Process column of the Activity Table. For each work process listed, the name of the department in which the work process is performed, is written in the first row of the table (see Fig. 1). A work process is a process that consists of a set of activities.

Activities: This step deals with identifying the activities performed within each of the work processes defined in the previous step. An activity is a simple micro-process that consists of one or more tasks that represent well-defined work performed by one resource. An activity may consist of a

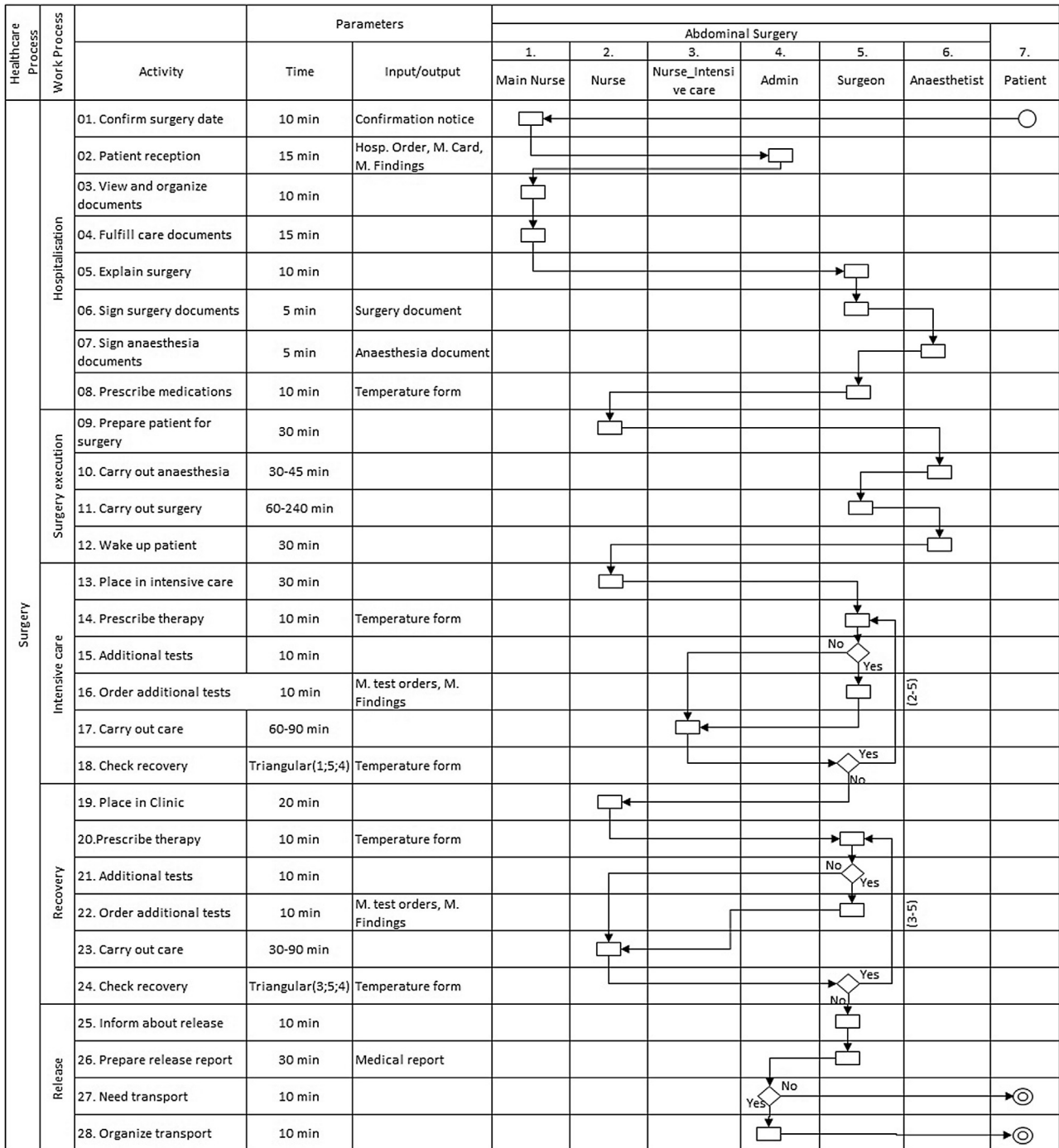


Fig. 1 – Activity diagram for abdominal surgery process simulation.

number of tasks [14]. A task could be understood as an elementary work within an activity [14]. To identify the activities of each work process, further interviews are organized with the employees involved in performing them. The Activity Table technique uses a small set of flowchart symbols to model a process, such as: ○, ⊙, □, ◇, →, ←, ↓, ↑.

To demonstrate the functionality of the proposed simulation approach the paper introduces a real case study of the abdominal surgery process taken from the abdominal clinic in Slovenia. Firstly, the Activity table modeling technique is

used to model the abdominal surgery process, resulting in the model presented in Fig. 1. This table divides the surgery process into five work processes, each one consisting of several activities. Several entities are presented in the columns of the Activity table and linked to specific activities, where they carry out particular tasks. No other entities are participating in this process, however, some of the entities are also included in carrying out tasks outside the selected process. The proposed model represents the whole process by modeling each of its 28 activities in a separate worksheet. These are interconnected

as in reality, so that a patient is directed from one activity to the next one automatically.

In our example, the process simulation is carried out over a period of four weeks, a week consists of five working days and eight working hours per day. Consequently, we defined 9600 rows in each spreadsheet (1 working day consists of 8 h; $8 \text{ h} * 60 \text{ min} * 20 \text{ working days} = 9600 \text{ min}$).

3. Model description and case study implementation

The process model that we describe is a discrete-event model implemented in an Excel workbook.

Performing simulations in Excel requires defining three modeling elements: an activity, a transaction, and a gate. An activity is a description of a task with certain duration and which requires resources to be performed [15]. Transactions are flow units that move through the process model from one activity to the next ones [16]. In the proposed model, patients have the role of transactions. A Gate is a modeling element that is activated at a particular time of the day. It means that the transactions that arrive at the gate have to wait until the gate's door opens at the time defined. A gate does not have time duration nor resources, therefore it is not considered as an activity. Consequently, it is not included in the activity diagram presented in Fig. 1.

Each worksheet implements one modeling element. A worksheet called *Input data worksheet* keeps the input data of the model.

3.1. Input data worksheet

The *Input data* worksheet specifies the configuration parameters of the modeling elements and the constraints of the model. It organizes the input data into three tables.

The first table defines the constraints of the model presented with the maximum number of available beds. The second table provides information about parameters of the modeled activities, described with different distributions. The last table shows for each activity its properties, such as ID, number of resources, and duration.

3.1.1. Example of input data worksheet

Table 1 introduces an example of the input data worksheet for the model of abdominal surgery process. It consists of three tables: Table 1a, 1b, and 1c. Table 1a shows that the number of available beds in the Clinic is 75.

Table 1b gives the parameters of the triangular distributions that are required in order to determine the duration of some of the activities, such as activities 18, 24, and others. For example, a triangular distribution determines the duration of the patient's stay in intensive care (activity 18 in Table 1c). It has the following parameters: minimum stay of the patient in the intensive care is 1 day, maximum stay is 5 days, and the most probable stay is 4 days. The actual duration is estimated with the function "rand()" that generates a pseudo-random number uniformly distributed in the interval (0,1), which is scaled to the desired distribution according

to the given parameters in Table 1b. This transformation is known as Probability Integral Transformation [12].

Table 1c provides names and time durations of the activities. These parameters are read from the third and fourth columns in the Activity Table (Fig. 1) and inserted into Table 1c (the spreadsheet).

The activity durations (last column in Table 1c) are defined using a number of time units, an interval denoted by *Between(a;b)*, see activities 10, 11, 17, 23, or a triangular distribution denoted by *TriangDist(a;c;b)*, where *a* is the minimum time duration, *b* is the maximum time duration and *c* is the most frequent time duration, see activities 18 and 24.

Our approach in this paper assumes a LogNormal prior distribution for the activities 10, 11, 17 and 23 [17,18]. For example, the calculation of the time duration of activity 11 ("Carry out surgery") is obtained with the formula: $\text{LOGINV}(\text{RAND}(), 5, 0.45)$. Consequently, the bulk of the probability mass is located in the interval [60,240] minutes. The LogNormal prior was selected purely for demonstration purposes. In many cases, more informative prior is needed. In Excel, drawing random numbers from an arbitrary prior can be achieved in two ways:

1. If the distribution is known, and the increase cumulative distribution (icdf) exists in closed form, than the Inverse transform sampling theorem can be used to generate the samples from the original distribution.
2. If the icdf does not exist in closed form, one can generate a VB script that performs Gibbs sampling, thus solving this issue.

The lognormal priors of activities 10, 11, 17 and 23 are calculated using the following parameters for the mean and standard deviation:

- Activity 10 $\sim \text{LN}(\text{mean}=3.58, \text{std}=0.27)$
- Activity 11 $\sim \text{LN}(\text{mean}=5, \text{std}=0.45)$
- Activity 17 $\sim \text{LN}(\text{mean}=4.84, \text{std}=0.37)$
- Activity 23 $\sim \text{LN}(\text{mean}=5.02, \text{std}=0.41)$

The sampling was performed using the inverse transform sampling theorem i.e., in Excel $\text{LOGINV}(\text{RAND}(), \text{mean}, \text{std})$.

More details regarding definition on distributions of the durations of the activities can be found in [17,19–21].

3.2. Types of modeling elements

Each of the three modeling elements – gate, activity, and transaction is modeled with a separate spreadsheet. Each row in the spreadsheet represents the state of the modeling element at a particular moment of time.

3.2.1. Gate

The model of a gate in a worksheet comprises the following four columns:

- (A) states the time when the gate is activated;
- (B) gives the number of transactions that are currently waiting at the gate;
- (C) lists all transactions that wait at the gate; and

Table 1 – Excel simulation input data worksheet.

Table 1a			
Number of beds	75		
Table 1b			
Parameters of triangular distribution	Activity 18	Activity 24	
a	1	3	
b	5	5	
c	4	4	
Table 1c			
Activity ID	Activity Name	Resource	Duration (Minutes)
01	Confirm surgery date	1	10
02	Patient reception	1	15
03	View and organize documents	1	10
04	Fulfill care documents	1	15
05	Explain surgery	1	10
06	Sign surgery documents	1	5
07	Sign anesthesia documents	1	5
08	Prescribe medications	1	10
Gate1			Every morning
09	Prepare patient for surgery	1	30
10	Carry out anesthesia	1	Between(30;45)
11	Carry out surgery	1	Between(60;240)
12	Wake up patient	1	30
13	Place in intensive care	1	30
14	Prescribe therapy	1	10
15	Addition-al tests	1	10
16	Order additional tests	3	10
17	Carry out care	1	Between(90;180)
18	Check recovery	2	TringDist(1;5;4)
IntG			Every morning
19	Place in Clinic	1	20
20	Prescribe therapy	2	10
21	Addition-al tests	1	10
22	Order additional tests	2	10
23	Carry out care	1	Between(90;240)
CliG			Every morning
24	Check recovery	2	TringDist(3;5;4)
Gate2			Every morning
25	Inform about release	1	10
26	Prepare release report	1	30
27	Organize transport	1	10
28	Need transport	1	10

(D) lists all transactions which are ready to be propagated to the next activity.

Table 1c presents four gates for the example of abdominal surgery Clinic simulation: Gate1, IntG, CliG, and Gate2.

3.2.2. Gate 1 as an example of a gate modeling element

Fig. 2 provides details of the Gate1 as an example of the modeling element gate.

The first column shows the activation time moment of Gate1 measured in minutes. In the example at hand, Gate1 is activated at 8:00 AM, hence the value of 480 min is obtained as a multiplication of $8\text{ h} \times 60\text{ min}$. In this study, the clinic works 8 h per day. Therefore, in the simulation each working day consists of 480 min and consequently the next working day starts at 481 min and lasts until 960 min (this is 2 working days $\times 8\text{ h} \times 60\text{ min}$), and so on.

Column B gives the number of transactions that arrived at Gate1 during the previous day and are waiting for the next activity the following morning. The transactions in the example are patients that completed the activity “08 Prescribe medications” (Fig. 3) and are waiting for the next activity “09 Prepare patient for surgery” (see Table 1c). The value in the cell B2 is 25, and it is obtained using the information stored in the worksheet “08 Prescribe medications.”

In Excel, transferring such an information can be achieved by using the function “indirect” which returns the value of the specified reference. For example, the formula `indirect(“8!”&address(2;6))` returns the value from cells in the worksheet “8” positioned on the `address(row;column)=address(2;6)`, which is cell F2 in the worksheet “08 Prescribe medications.” Sometimes we don’t know the exact row into which a certain value resides in the previous worksheets. In this case, we can use the function “match” which searches the given item in a specified range of cells returning the

	A	B	C	D
1	Time in minutes	Patients waiting	All transactions that arrived	Transactions waiting at 8:00 AM
2	481	25	1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24:25	1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24:25
3	961	0	1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24:25:26	26:27:28:29:30:31:32:33:34:35:36:37:38:39:40:41:42:43:44:45:46:47:48:49:50:51:52:53:54:55
4	1441	0	1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24:25:26	56:57:58:59:60:61:62:63:64:65:66:67:68:69:70:71:72:73:74:75

Fig. 2 – Example of a gate modeling element: Gate1.

relative position of the required given item. For example, the formula $match(A3,'8!'\$A\$2:\$A\$18000;0)$ searches for a match of the value given in the cell A3 in the worksheet “8” in the cell range A2 – A18000. Finally, the following formula is used in cell B3:

$B3=sum(indirect(“8!”&address(2;6);indirect(“8!”&address(match(A3;'8!'\$A\$2:\$A\$18000;0)+1;6)).$

The formula sums the values starting from cell F2 (address(2;6)) in the worksheet “08 Prescribe medications” until the row that matches the value of A3 from the worksheet “Gate 1,” in the A column of the worksheet “08 Prescribe medications.”

Column C gives the list of all transactions that arrived at Gate1, separated with colon “:”. The name of the transaction is given according to the minute in which it commenced. For example, 24 means that the transaction commenced at the 24th minute. The following formula obtains the list in cell C3:

$C3=indirect(“8!”&address(match(A3;'8!'\$A\$2:\$A\$18000;0)+1;9)).$

The formula returns the list of transactions from the worksheet “08 Prescribe medications” which have finished by the 961st minute specified in cell A3 in the worksheet Gate1.

The last column D gives only those transactions that arrived in the last interval, such as between 1 and 480th minute, 481st and 960th minute, and so on. For example, the list in column D3 comprises the activities obtained by applying the following formula:

$D3=if(or(C3=0;C3=“”),“”,if(len(C2)=0;mid(C3;1;len(C3)-len(C2));mid(C3;len(C2)+2;len(C3)-len(C2))))).$

The formula firstly examines if the list of activities is empty (with the conditional checkup $if(or(C3=0;C3=“”))$). Otherwise, the formula checks if the list in the previous interval was empty ($if(len(C2)=0)$), by introducing the formula $len(text)$, which checks the length of characters of the variable $text$. If $text$ is empty then the result is adding the whole list of activities, otherwise the formula subtracts the activities of the previous interval from the current one. In Excel this is achieved using the function $mid(text, start, length)$, which from the $text$ returns a number of characters defined in $length$ and starting from the position $start$ in the $text$.

3.2.3. Activity

Fig. 4 presents an example of an activity.

The worksheet that models an activity is comprised of the following eight columns (A-H):

- (A) States the simulation minute of the time which applies to the whole row in the worksheet;
- (B) Stores whether a transaction has arrived in the indicated minute; “1” denotes arrival of a transaction, and “0” denotes that no transaction has arrived;
- (C) Shows until when a resource is busy; meaning until which minute;
- (D) States the availability of the resource; “1” denotes that the resource is busy, and “0” denotes that the resource is available;
- (E) Gives a cumulative number of transactions that are waiting in a queue for processing;
- (F) Indicates whether the transaction defined in row I is finished or not. If it is finished then 1 is inserted in cell(I, F), which means that the current transaction continues in the next activity. Otherwise, “0” is written in cell(I,F);
- (G) Lists all transactions that are waiting in a queue for processing at the current activity. The transaction that will be the first processed is first in the queue.

row number	A	B	C	D	E	F	G	H	I
	Time in minutes	Arrival of transaction	Resource busy until	Resource Busy (1 = busy; 0 = free)	Transactions waiting for execution	Finished transaction	List of transactions that started	Transaction ID ended	All transaction that finished with this activity
401	476	0	482	1	0	0	26		1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24:25
402	477	0	482	1	0	0	26		1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24:25
403	478	0	482	1	0	0	26		1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24:25
404	479	0	482	1	0	0	26		1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24:25
405	480	0	482	1	0	0	26		1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24:25
406	481	0	482	1	0	0	26		1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24:25
407	482	0	482	0	0	1		26	1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24:25:26
408	483	0	482	0	0	0			1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24:25:26
409	484	0	482	0	0	0			1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24:25:26
410	485	0	482	0	0	0			1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24:25:26
411	486	0	482	0	0	0			1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24:25:26
412	487	1	497	0	0	0	27		1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24:25:26
413	488	0	497	1	0	0	27		1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24:25:26

Fig. 3 – A sample of the activity “08 Prescribe medications”.

	A	B	C	D	E	F	G	H
	Time in minutes	Arrival of transaction	Resource busy until	Resource Busy (1 = busy; 0 = free)	Transactions waiting for execution	Finished transaction	List of transactions that started	Transaction ID ended
1	1	1	11	1	0	0	1	0
2	2	1	11	1	1	0	1:2	0
3	3	1	11	1	2	0	1:2:3	0
4	4	1	11	1	3	0	1:2:3:4	0
5	5	1	11	1	4	0	1:2:3:4:5	0
6	6	1	11	1	5	0	1:2:3:4:5:6	0
7	7	1	11	1	6	0	1:2:3:4:5:6:7	0
8	8	1	11	1	7	0	1:2:3:4:5:6:7:8	0
9	9	1	11	1	8	0	1:2:3:4:5:6:7:8:9	0
10	10	1	11	1	9	0	1:2:3:4:5:6:7:8:9:10	0
11	11	1	11	0	10	1	2:3:4:5:6:7:8:9:10:11	1

Fig. 4 – Spreadsheet of “Confirm surgery date” activity.

- (H) The last column lists the transactions that were processed by the current activity. Consequently, these activities are propagated to the next worksheet (activity). The values are given as integer numbers, representing the transactions IDs.
- (I) When a gate follows an activity we need to list all transactions that have finished with the activity. Such an example is the activity “08 Prescribe medications” presented in Fig. 3.

Generally, an activity input is a transaction that has finished its execution from the preceding activity. The model defines activities, which may have either one or two sources of incoming transaction.

3.2.4. “Confirm surgery date” activity as a typical activity example

The model of the activity “Confirm surgery date” is presented in Fig. 4. Column A defines the simulation time that applies to the whole row. It measures the time in minutes. The row itself defines the state of the activity in the minute defined in column A. Cell A1 is the starting simulation time, which in this example is minute 1. Each consecutive cell in column A increases the simulation time for one minute which is achieved with the formula: $A_{i+1} = A_i + 1$, where $i = 2, 3, \dots$. Column B states the arrivals of transactions. The formula that calculates the values in column B considers the maximum number of transactions, which for the given example is 75. This number coincides with the number of available beds and is defined in Table 1a. For example, the formula that calculates the value in the cell B4 is:

$$B4=IF(SUM(\$B\$2:B3)-END!K3<info!\$K\$7;1;0).$$

The formula checks if the difference between all commenced transactions ($SUM(\$B\$2:B3)$) and all ended transactions ($END!K3$) is smaller than the number of beds ($info!\$K\7). In such a case, a new transaction may commence and the value of the cell is set to 1, otherwise the value is set to 0.

The next column C defines the simulation time until which the resource is occupied. For example, when a transaction arrives in B1, the value of C1 changes according to the formula:

$$C1=if(B1>0;A1+info!E3;A1).$$

The formula sums the current time given in cell A1 to the duration of the activity as specified in Table 1c. It is extended for the consequent cells in column C. For example the formula in C5 is:

$$C5=if(B5=0; if(E4=0; C4; if(or(C4>A5;C4=A5);C4;A5+info!\$E\$3)); if(or(C4>A5;C4=A5);C4;A5+info!\$E\$3)).$$

The formula checks if a new transaction has not arrived ($B5=0$) and if no transaction is waiting to be executed ($E4=0$), then it copies the time from the cell C4. It also checks if the resource is busy (OR ($C4>A5;C4=A5$)), then it copies the above cell C4. Otherwise, if it is available it sets the new time until it is busy ($A5+info!\$E\3) by reading the time duration of the activity that is set in the input spreadsheet cell ($info!\$E\3). If a new transaction has arrived ($B5 \neq 0$), then the formula checks the availability of the resource and sets the time accordingly.

Column D keeps a record of the availability of the resource. The values are set with the formula $D_{i+1} = if(C_i > A_{i+1}; 1; 0)$, which checks if the simulation time is higher than the current time ($C_i > A_{i+1}$), then the value is set to one, meaning that the resource is busy; otherwise, it is set to zero, meaning that the resource is available. The initialization of column D is obtained with $D1 = if(B1 > 0; 1; 0)$, which checks if the transaction has arrived ($B1 > 0$) and sets the value in D1 accordingly.

The number of transactions waiting for execution is stored in column E. For example, the formula for E5 is:

$$E5=if(B5=0; if(C5>C4; if(E4>0; E4-1; 0); E4; if(or(C4>A5; C4=A5); E4+1; E4))$$

This formula checks if the transaction has not arrived in the current minute ($B5=0$) and the availability of the resource has changed in the current minute ($C5 > C4$), then the value of E5 is set to the value of the previous cell (E4) decreased by one ($E4-1$). Otherwise, the value of E5 stays the same as in E4. On the other

hand, if the transaction has arrived ($B5 \neq 0$) and the resource is busy, the simulation time is higher than equal to the current time (or($C4 > A5; C4 = A5$)), then the value of E5 is incremented by one ($E4 + 1$), meaning that one more transaction is waiting for execution, otherwise, the value of E5 stays the same as E4.

Column F provides information whether a transaction has ended in the current minute. For example, F5 is calculated as $F5 = \text{if}(C5 = A5; 1; 0)$. It means that if the current time is equal to the simulation time until which the resource is busy, then the resource is released and the transaction has finished.

All transactions that commenced enter the queue in column G. The first value in column G is obtained with $G1 = \text{IF}(B1 = 1; A1; "")$. A more complex formula calculates the values in the following rows. For example, the formula that calculates the value in G5 is:

```
G5=if(len(G4)=0; if(B5=1;A4;"");
    if(and(B5=1;F5=1); concatenate(mid(G4;search(":";G4)
+1;len(G4)-search(":";G4));":";A5);
    if(and(B5=1;F5=0);concatenate(G4;":";A5);
    if(and(B5=0;F5=1); mid(G4;search(":";G4)
+1;len(G4)-search(":";G4));G4)))
```

This is the most complex formula in the simulation. It comprises of several scenarios. Supposing an empty queue ($\text{len}(G4) = 0$), the newly arrived transaction ($B5 = 1$) is added to the queue. This is denoted by $\text{if}(B5 = 1; A4; "")$. Two scenarios are possible upon arrival of a transaction in a non-empty queue.

Scenario 1, simultaneously another transaction finishes ($\text{if}(\text{and}(B5 = 1; F5 = 1))$). In such a case, the newly arrived transaction is added to the queue, and the finished transaction is removed from it. This is achieved by using formula:

```
concatenate(mid(G4; term_1; term_2); ":"; A5).
```

In the last formula, $\text{term}_1 - \text{term}_2$ are introduced for better readability. The $\text{term}_1 - \text{term}_2$ are:

```
term_1: search(":";G4)+1
term_2: len(G4)-search(":";G4).
```

The function $\text{concatenate}(\text{text1}; \text{text2}; \dots)$ is used to join text1 with text2 and so on, while the function $\text{search}(\text{text1}; \text{text2})$ looks for the text1 into text2 and returns its position.

Scenario 2, there is no transaction which finishes simultaneously ($\text{if}(\text{and}(B5 = 1; F5 = 0))$), so only the newly arrived transaction is concatenated to the queue ($\text{concatenate}(G4; ":"; A5)$).

The last scenario occurs in a case when a transaction ends while no other transaction arrives simultaneously ($\text{if}(\text{and}(B5 = 0; F5 = 1))$). The ended transaction is removed from the queue using function $\text{mid}(G4; \text{search}(":"; G4) + 1; \text{len}(G4) - \text{search}(":"; G4))$. If none of the described scenarios happens, the formula copies the queue from the previous cell G4.

Finally, column H keeps the IDs of the finished transactions in the current minute. For example, the value of H5 is calculated as $H5 = \text{IF}(F5 = 1; \text{left}(G4; \text{search}(":"; G4) - 1); 0)$. When a transaction finishes ($F5 = 1$) the formula reads the ID of the ended transaction from the queue in G4. In Excel this is achieved with function $\text{left}(\text{text}; \text{length})$ which returns the left-most part of the text with number of characters defined with length .

At first glance, the activity structure might seem too complex. However, each row in Fig. 4 can be interpreted as follows. Row 1 states that in the first minute (cell A1), a transaction has arrived ($B1 = 1$), and it will be executed until minute 11 ($C1 = 11$). Consequently, the resource is busy ($D1 = 1$) starting from the current minute until minute 11, when the activity finishes with the execution of the commenced transaction ($F11 = 1$). There are no other transactions that are waiting in the queue ($E1 = 0$) for execution. Only one transaction has commenced, denoted with $ID = 1$ ($G1 = "1"$). Transactions that ended with execution are removed from the queue in column G, such as the case in G11. The ID of the ended transaction is given in column H, as in the case of H11. Such a status report can be determined for each row of the activity.

3.2.5. Different types of activities

The model distinguishes two main modifications of an activity: a decision activity and activity with multiple sources of incoming transactions. In the latter case, the model defines two modifications of the activity depending on the type of the modeling element that generates the incoming transaction to the activity:

1. one transaction is generated from preceding activity, and another is generated from a preceding decision activity, or
2. one transaction is generated from preceding activity, and another is generated from a preceding gate.

The main difficulty in the first modification is determining the list of arrived transactions. In order to resolve this issue, the proposed solution includes two additional columns. The first added column contains the information whether a transaction has arrived from the second preceding activity. The second added column summarizes the arrival of a transaction from any of the inputs. Its main functionality is to facilitate the writing of the formulas in columns dependent on the inputs of the activity. The second modification of the activity model additionally introduces a single new column in which it holds the information of whether it is time to withdraw transactions from the gate. The principal first-in-first-served solves the priority of the transaction that should be served first in cases when two transactions arrive simultaneously.

3.2.6. Decision activity

A decision activity is used to check whether a certain condition is fulfilled and on the basis of the outcome decides for further actions. The decision activity usually has two paths. One path is indicated by "YES" and means that the condition is fulfilled. "NO" denotes the other path.

Fig. 5 is an example of a model of a decision activity.

In addition to the columns defined for an activity, it comprises the following columns:

- (I) generates random number using the "RAND()" function;
- (J) gives a decision YES or NO based on the obtained number in (A);
- (K) auxiliary column that keeps record of the transactions that ended with a decision "YES", and
- (L) auxiliary column that keeps record of the transactions that ended with a decision "NO".

3.2.7. Decision activity for checking the recovery of the patient

Activities 15, 18, 21, and 24 are the decision activities. Fig. 5 presents a selection of states (listed in the first column) of activity 18, which decides positively in 20% and negatively in 80% of cases. Columns H and I provide such a distribution of the outcomes. In column I a random number in the interval (0–1) is generated using the command `rand()`. Statement `if(I720<(0,8);“NO”;“YES”)` checks the number in column I is higher or lower than some threshold and saves the result in J. In the given example, the threshold is 0.8 representing 80% of the cases. When a transaction ends, value “1” is denoted in column H and the number in I is higher than 0.8, then the decision in J is “YES.” In such a case, the transaction is concatenated to the list of transactions in column L. This list is required because the next modeling element is gate *Int*, which is activated once per day. Otherwise, if the number in I is smaller or equal to 0.8 at the time when a transaction ends, then the transaction is only written to the column M, which keeps record of the transactions that are directly propagated the next activity “19.” The rest of the columns have the same functionality as described above.

3.2.8. Activity with incoming transactions from two preceding activities

Fig. 6 represents the activity 17, which is a successor of two activities; the decision activity 15; and the activity 16. The arrival of transactions is simulated with three columns: columns B and C show arrival of the transaction from activities 15 and 16, respectively, and column D shows if the transaction has arrived either from activity 15 or 16.

An example of the formula that examines arrival of the transaction from a previous decision activity is represented for the case of cell B2 with:

```
B2=IF (AND
(indirect(“15!”&address(match(A2;’15!’$A$2:A1441;0)+1;7)));
NOT(indirect
(“15!”&address(match(A2;’15!’$A$2:A1441;0)+1;10))))=
TRUE; 1;0).
```

The formula checks two conditions: if a transaction has completed the activity 15 (`indirect(“15!”&address(match(A2;’15!’A2:A1441;0)+1;7))`) and the decision was “YES,” then the completed transaction follows the current activity 17 (`indirect(“15!”&address(match(A2;’15!’A2:A1441;0)+1;10))`).

The time for execution of this activity is uncertain, in the sense that it may take any value between 90 and 180 min (see activity 17 in Table 1c). Consequently, to simulate the time execution, a random number is generated in the interval (0–1) in each minute, which is scaled to the required interval. Because it is an activity with uniform distribution of duration, column J calculates the time in minutes required for execution of the transaction with the formula $J=MROUND(90+(180-90)*rand();1)$. The function `MROUND` is used to trunk the integer part of the number obtained with the formula.

As mentioned before, if there are several activities in the queue that are waiting for execution, the priority of the transaction that should be served first is solved by the

principal first-in-first-served. The major difficulty in applying this principal lies in defining column M, which keeps a list of transactions that arrived to a certain point. The following example of nested “if”-statement that calculates M3, solves the problem:

```
M3=if(len(M2)=0;if(D3=1;if(B3=1;term_1;term_2);“”);
if(and(D3=1;I3=1); concatenate(term_3;“.”;term_4);
if(and(D3=1;I3=0); concatenate (M2; “.”;
term_4));
if(and(D3=0;I3=1);term_5; M2))))).
```

In the last nested if-statement some of the long expressions are replaced with expressions *term.1* - *term.7*, only for easier readability in this paper. These have to be incorporated into the if-statement, when applied for modeling. The *term.1*–*term.7* are:

```
term_1: indirect(“15!”&address(match
(A3;’15!’$A$2:$A$10000;0)+1;17)) – imports the
transaction ID from activity 15
term_2: indirect (“16!”& address (match
(A3;’16!’$A$2:$A$10000;0)+1;10)) – imports the
transaction ID from activity 16
term_3: mid(M2; if(iserror(search (“.”;M2));
len(M2); search(“.”;M2)+1);
len (M2)-
if(iserror(search
(“.”;M2)); len(M2);
search (“.”;M2)))
term_4: if(and(B3=1; C3=1);
concatenate(term_4.1;“.”;
term_4.2);if(B3=1;term_4.1;term_4.2))
term_4.1: indirect (“15!”& address
(match
(A3;’15!’$A$2:$A$10000;0)+1;17));
term_4.2: indirect (“16!”& address
(match
(A3;’16!’$A$2:$A$10000;0)+1;10))
term_5 mid(M2;search(“.”;M2)+1;len(M2)- search
(“.”;M2)).
```

This formula handles the following cases:

1. If the list of transactions (M) is empty and a transaction has arrived, then add the ID of the transaction from activity 15 (*term.1*), or from activity 16 (*term.2*).
2. If the list of transactions is not empty, then the following cases are possible:
 - a. A new transaction arrives simultaneously when a current one finishes (condition $D3=1$ and $I3=1$); in this case, the list of transactions has to be updated twice: firstly, to remove the transaction that finished its execution (*term.3*), and secondly, to add the newly arrived transaction (*term.4*). In *term.3* the function `iserror(expression)` is used which returns value of `True` if it finds errors in *expression*. *Term.4* is divided in two cases: arrival of transactions simultaneously from activities 15 and 16 and arrival of a transaction either from activity 15 (*term.4.1*) or from 16 (*term.4.2*).

Row No.	A	B	C	D	E	F	G	H	I	J	L	M
	Time in minutes	Arrival of transaction	Resource busy until	Resource Busy (1 = busy; 0 = free)	Transactions waiting for execution	Finished transaction	List of transactions that started	TransactionID ended	Decision	20% YES; 80% NO	Auxiliary column YES; follow the gate INT	Auxiliary column NO; follow the activity 19
163	1316	1	1326	0	0	0	49	0	0,56087	NO		
314	1448	1	1458	0	0	0	74	0	0,88199	YES	49	
446	1552	1	1562	0	0	0	99	0	0,010479	NO	49	
550	1722	1	1732	0	0	0	124	0	0,605636	NO	49	
720	1878	1	1888	0	0	0	49	0	0,661489	NO	49	
876	2020	1	2030	0	0	0	149	0	0,233557	NO	49	
1176	2178	1	2188	0	0	0	174	0	0,005266	NO	49	
1305	2307	1	2317	0	0	0	199	0	0,796961	NO	49	
1422	2424	1	2434	0	0	0	224	0	0,51937	NO	49	
1577	2579	1	2589	0	0	0	249	0	0,974492	YES	49	
1748	2750	1	2760	0	0	0	274	0	0,023592	NO	49	
1926	2928	1	2938	0	0	0	299	0	0,518122	NO	49:274	

Fig. 5 – Example of a decision activity 18.

- b. A new transaction arrives, while no other transaction finishes (the case of $D3=1$ and $I3=0$); the list of transactions is updated by the newly arrived transaction(s), A transaction finishes its execution and there is no newly arrived transactions (the case of $D3=0$ and $I3=1$); in this case the finished transaction is removed from the list of transactions (*term.5*).
- c. Each row of such an activity can be easily read. For instance, in the 4809th minute (A7), a transaction arrived (D7) from activity 15 (B7). However, it must wait for processing because the resource is busy (F7) until minute 4888 (E7). There are two transactions (H7) in the queue with transaction IDs 24 and 21 (M6), to which the newly arrived transaction 18 from activity 26 is the last one indicated in the list (M7). The duration of this activity for the arrived transaction is 170 minutes (J7). The transaction that is currently processed is 24 (the first one in M7). No transaction completes this activity at the current simulation minute (N7).

3.2.9. Activity with incoming transactions from preceding activity and a gate

The second modification of an activity with multiple incoming transactions discusses the case when the incoming transactions come from: a preceding gate and a preceding activity. Fig. 7 shows Activity 14 as an example of this case.

Columns H and I are the newly introduced ones in the model of such an activity. Column H checks whether it is time to remove transactions from the Gate IntG. This is achieved with the formula: $\text{countif}(\text{IntG!}:\$A\$2:\$A\$24;\text{A}2)$, which may result in two values: 0 or 1. Value "1" denotes that transactions have to be removed from the gate IntG. These transactions are written into column (I) by using the formula:

$$I2=\text{if}(H2=1; \text{if}(\text{indirect}(\text{"IntG!"}\&\text{address}(\text{match}(\text{A}2;\text{IntG!}:\$A\$2:\$A\$24;0)+1;5))=\text{""; 0; indirect}(\text{"IntG!"}\&\text{address}(\text{match}(\text{A}2;\text{IntG!}:\$A\$2:\$A\$24;0)+1;5)))));0).$$

The formula checks if it is time ($H2=1$) to remove transactions from the gate IntG. Furthermore, it checks if the queue of transactions waiting at IntG is empty. If the queue is not empty, then the formula removes the transactions from IntG and writes them into column I. The formula in column M concatenates the removed transactions to the existing list of transactions in column M, which are waiting for processing by the current activity.

For example, column A shows that in minute 2880 three transactions (column E) are removed from IntG (columns H, I and J) and are added to the existing list of transactions (column M). No transactions completed the activity in the current simulation minute (column N).

	A	B	C	D	E	F	H	I	J	M	N
	Time in minutes	Arrival of transaction from 15	Arrival of transaction from 16	Arrival of transaction from 15 or 16	Resource busy until	Resource Busy (1 = busy; 0 = free)	Transactions waiting for execution	Finished transaction	Uniform distribution	List of transactions that started	Transaction ID ended
1											
2	4804	0	0	0	4888	1	1	0	180	24:21	0
3	4805	0	0	0	4888	1	1	0	109	24:21	0
4	4806	0	0	0	4888	1	1	0	96	24:21	0
5	4807	0	0	0	4888	1	1	0	169	24:21	0
6	4808	0	0	0	4888	1	1	0	95	24:21	0
7	4809	1	0	1	4888	1	2	0	170	24:21:18	0
8	4810	0	0	0	4888	1	2	0	155	24:21:18	0
9	4811	0	0	0	4888	1	2	0	138	24:21:18	0
10	4812	0	0	0	4888	1	2	0	159	24:21:18	0

Fig. 6 – Example of an activity 17 with two inputs: from activity 15 and 16.

A	B	C	D	E	F	H	I	J	M	N
Time in minutes	Arrival of transaction	Resource busy until	Resource Busy (1 = busy; 0 = free)	Transactions waiting for execution	Finished transaction	Check if time to withdraw transactions from IntG	Transaction has arrived from IntG	Transaction from 13 or from IntG	List of transactions that started	TransactionID ended
726	1	736	1	0	0	0	0	1	1	0
960	0	938	0	0	0	1	0	0		0
1440	0	1352	0	1	0	1	3	1	3	0
1920	0	1801	0	0	0	1	0	0		0
2400	0	2354	0	0	0	1	0	0		0
2880	0	2799	0	3	0	1	10:11:12	1	10:11:12	0
3360	0	3345	0	2	0	1	13:14	1	13:14	0
3840	0	3810	0	2	0	1	12:16	1	12:16	0
4320	0	4296	0	1	0	1	17	1	17	0

Fig. 7 – Example of an activity 14 with two inputs: from activity ID 13 and the gate activity ID IntG.

Table 2 – Simulated time executions in days of activities in days with Excel model.

Maximal number of beds	Number of served patients in 2 weeks	Average Cycle Time (days)	Average Work (days)	Average Wait (days)	Minimum Cycle Time (days)	Maximum Cycle Time (days)
75	44-47	20.80	4.26	16.52	7.11	32.23

3.3. Outputs of the model

The output of the simulation model provides calculations and graphical representation of distributions of the following parameters:

1. Minimum Cycle Time
2. Average Cycle Time
3. Maximum Cycle Time
4. Average Work Time
5. Average Wait Time

Work Time is the amount of time during which transactions have been actively processed. It is directly calculated as a sum of the activities duration times.

The amount of time work is actively being performed on transactions. Calculated from activity task duration times.

Wait Time is the simulated amount of time that transactions are waiting. The waiting may occur due to different reasons: waiting for a resource, or because of a gate.

Cycle Time is the simulated amount of time that a transaction spends within the process starting with the first activity and ending with the last one. A cycle time may differ from one activity to another due to the different paths in the process and to the different activity durations calculated from their distributions. The cycle time is a sum of work time and wait time.

The following equation connects the last three parameters:

$$\text{Cycle Time} = \text{Work time} + \text{Wait Time.}$$

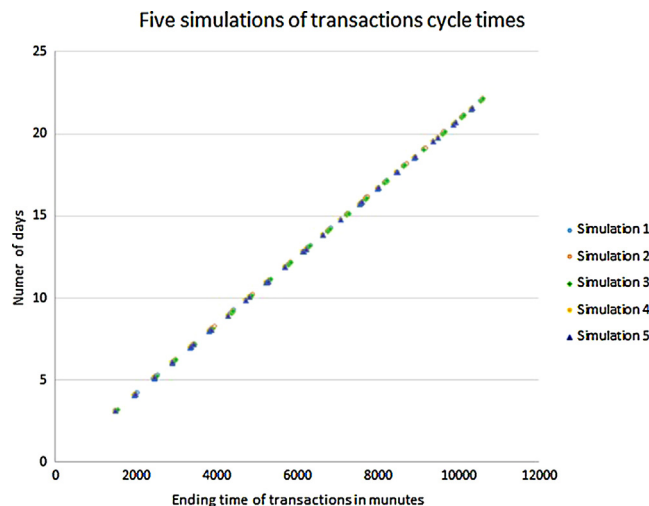


Fig. 8 – Five simulations of cycle times of 44-47 finished transactions ordered by time spent in the clinic.

Table 3 – Mean and standard deviation of the five simulation experiments.

	Simulation 1	Simulation 2	Simulation 3	Simulation 4	Simulation 5
Mean	12,72	13,08	12,57	12,36	12,54
Standard Deviation	5,62	5,49	5,63	5,43	5,16

Maximum (minimum) Cycle Time refers to the highest (lowest) accumulated time for the transactions at hand, at any point in time.

The term *average* refers to the total time that all transactions spent in each activity divided by the number of completed transactions for each activity.

4. Simulation results

In this section, we provide an example of a simulation obtained with the presented model.

Firstly, we set the number of available beds in the Clinic to 75. After that, we run the simulation five times, and we obtained the number of served patients, the average cycle time per patient, the average work and wait times, and the minimum and maximum cycle times. Table 2 presents the obtained results. The model also provides a graphical representation of the distributions of measured parameters. The transactions cycle times for the five simulation experiments (Simulation 1–5) are given in Fig. 8. The mean and the standard deviations for each of the simulations experiments 1–5 are given in Table 3. The experts in the abdominal clinic confirmed the results as logical.

5. Conclusion

This paper introduces a general two-step modeling framework for healthcare process simulation by using the Activity table modeling technique and spreadsheets. The starting point was a spreadsheet model used for simulation of emergency department for patient flow, for which we propose an upgrade and improvement in several manners. Firstly, the newly developed modeling framework does not require usage of macros. Hence, it can be reproduced in a fairly easy way as it does not require a particular programming experience, and it can be used by users with proficient knowledge of spreadsheets. Secondly, the solution employs a modular approach for modeling of the activities thus ensuring better readability; easy integration of the model with other activities; and its easy extension and connectives with other Clinic's models. Finally, it shows the state of the activity at any point of time. The proposed modeling framework is applied on a real-life case study for modeling and simulation of the abdominal surgery process.

Acknowledgements

The work is supported by Creative Core FISNM-3330-13-500033 'Simulations' project funded by the European Union, The European Regional Development Fund.

REFERENCES

- [1] M. Gunal, M. Pidd, Discrete event simulation for performance modelling in health care: a review of the literature, *J. Simul.* 4 (2010) 42–51.
- [2] A. Vishwanath, S. Rajan Singh, P. Winkelstein, The impact of electronic medical record systems on outpatient workflows: a longitudinal evaluation of its workflow effects, *Int. J. Med. Inf.* 79 (11) (2010) 778–791.
- [3] M. Askari, R. Westerhof, S. Eslami, S. Medlock, S.E. Rooij, A. Abu-Hanna, A combined disease management and process modeling approach for assessing and improving care processes: a fall management case-study, *Int. J. Med. Inf.* 82 (10) (2013) 1022–1033.
- [4] L. Holml, H. Lurås, F.A. Dahl, Improving hospital bed utilisation through simulation and optimisation, *Int. J. Med. Inf.* 82 (2) (2012) 80–89.
- [5] A. Bagust, M. Place, J. Posnett, Dynamics of bed use in accommodating emergency admissions: Stochastic simulation model, *BMJ Clin. Res.* 319 (1999) 155–158.
- [6] N. Proudlove, S. Black, A. Fletcher, OR and the challenge to improve the NHS: modelling for insight and improvement in in-patient flows, *J. Oper. Res. Soc.* 58 (2007) 145–158.
- [7] M.G. Klein, G. Reinhardt, Emergency department patient flow simulations using spreadsheets, *Simul. Healthc.* 7 (1) (2012) 40–47.
- [8] R.S. Aguilar-Save, Business process modelling: review and framework, *Int. J. Prod. Econ.* 90 (2004) 129–149.
- [9] W.J. Kettinger, T.C. Teng, S. Guha, Business process change: a study of methodologies, techniques, and tools, *MIS Q.* 21 (1997) 55–88.
- [10] N. Damij, Business process modelling using diagrammatic and tabular techniques, *Bus. Process Manag. J.* 13 (1) (2007) 70–90.
- [11] T.H. Davenport, L. Prusak, Working Knowledge: How Organizations Manage What They Know, Harvard Business School Press, Cambridge, MA, 1998.
- [12] L. Leemis, S. Park, Discrete-Event Simulation: A First Course, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2005.
- [13] L. Crystal Jiang, Z. Wang, T. Peng, J. Zhu, The divided communities of shared concerns: mapping the intellectual structure of e-Health research in social science journals, *Int. J. Med. Inf.* 84 (1) (2014) 24–35.
- [14] N. Damij, T. Damij, Process Management. A Multi-Disciplinary Guide to Theory, Modelling, and Methodology, Springer, 2014.
- [15] D. Hartvigsen, SimQuick Process Simulation with Excel, Mendoza College of Business Administration, University of Notre Dame, Prentice Hall, Upper Saddle River, NJ, 2001, pp. 07458.
- [16] "iGrafx Help system," iGrafx LLC, 2013.
- [17] F. Dexter, R. Epstein, E. Bayman, J. Ledolter, Estimating surgical case durations and making comparisons among facilities: identifying facilities with lower anesthesia professional fees, *Anesth. Analg.* 116 (2013) 1103–1115.
- [18] F. Dexter, J. Ledolter, Bayesian prediction bounds and comparisons of operating room times even for procedures with few or no historic data, *Anesthesiology* 103 (2005) 1259–1267.

-
- [19] F. Dexter, J. Ledolter, V. Tiwari, R. Epstein, Value of a scheduled duration quantified in terms of equivalent numbers of historical cases, *Anesth. Analg.* 117 (2013) 204–209.
- [20] E. Marcon, F. Dexter, An observational study of surgeons' sequencing of cases and its impact on postanesthesia care unit and holding area staffing requirements at hospitals, *Anesth. Analg.* 105 (2007) 119–126.
- [21] E. Marcon, F. Dexter, Impact of surgical sequencing on post anesthesia care unit staffing, *Health Care Manag. Sci.* 9 (2006) 87–98.
- [22] P. Gooch, A. Roudsari, Computerization of workflows, guidelines, and care pathways: a review of implementation challenges for process-oriented health information systems, *J. Am. Med. Inform. Assoc.* 18 (2011) 738–748.
- [23] K. Unertl, K.B. Johnson, N.M. Lorenz, Health information exchange technology on the front lines of healthcare: workflow factors and patterns of use, *J. Am. Med. Inform. Assoc.* 19 (2012) 392–400.
- [24] T. Damij, Development of a hospital information system using the TAD method, *J. Am. Med. Inform. Assoc.* 5 (1998) 184–193.