**PRESENTING A HYBRID PROCESSING MINING FRAMEWORK FOR AUTOMATED
SIMULATION MODEL GENERATION**

Mohammed Mesabbah
Susan McKeever

Centre for Applied Data Analytics Research (CeADAR)
School of Computing
Dublin Institute of Technology (DIT)
Kevin Street, Dublin 8, IRELAND

ABSTRACT

Recent advances in information technology systems have enabled organizations to store tremendous amounts of business process data. Process mining offers a range of algorithms and methods to analyze and extract metadata for these processes. This paper presents a novel approach to the hybridization of process mining techniques with business process modelling and simulation methods. We present a generic automated end-to-end simulation framework that produces unbiased simulation models using system event logs. A conceptual model and various meta-data are derived from the logs and used to generate the simulation model. We demonstrate the efficacy of our framework using a business process event log, achieving reduction in waiting times using resource reallocation. The intrinsic idea behind our framework is to enable managers to develop simulation models for their business in a simple way using actual business process event logs and to support the investigation of possible scenarios to improve their business performance.

1   INTRODUCTION

Improvement of business processes allows organizations to optimize the allocation of finite resources in order to stream-line activities. A key enabler of business process improvement is process simulation (Liu et al. 2012). For instance, in the healthcare domain there is a global pressure to optimize patient outcomes and treatment whilst managing the spiraling costs of providing healthcare services to aging or growing populations. This domain is a rich application area for simulation modelling and the provision of statistical estimations of 'what if' scenarios. Complex business processes, uncertain parameters, time critical activities and both budget and resource constraints typify an environment suitable for scenario testing. Rather than incurring the cost of "learning by doing", simulation modelling is one of the most powerful and cost-effective methods for system performance improvement (Hlupic and Robinson 1998).

Simulated processes rely on solid information for derivation of an accurate conceptual model. In the typical top down approach, this information is gathered manually from interviews of knowledge holders, direct observation and procedural documentation. However, this approach is time-consuming and liable to subjective or optimistic observations (Lynch et al. 2014). Process models can be derived using a bottom-up approach through the distillation of information from actual system event logs (Măruşter and van Beest 2009). This exercise of process mining can be used to establish an unbiased picture of the actual executed processes.

Application areas such as healthcare processes are increasingly underpinned with system logs of significant events – from the patient being checked in at reception, through each of the clinical checks. Recent approaches have used process mining of system event logs for conceptual modelling, as an input

to the for simulation modelling (Abohamad et al. 2017). This allows for the development of simulation models based on realistic representations of processes as logged by the underlying system logs.

The aim of this research is to present a generic and automated end-to-end simulation framework that produces an accurate unbiased simulation model. The model is based on a conceptual model and various metadata, both of which are derived from actual system recordings of process events. Various components are developed to parse system events log files and extract data for different perspectives of the business process, including agents' pathways, processing times, resources, rate of agents' arrival to system. This information is used as the input for generating a functional simulation model using a generic Discrete Event Simulation (DES) engine that is developed to serve as model builder. This model supports the execution of "what-if" scenarios on variations in resources, agents arrival patterns and/or process - in order to optimize the underlying processes through reduction of waiting times, queues lengths and higher system throughout. The framework is not restricted to a specific domain and could be applied in any domain that is suitable for DES simulation, with available system event logs.

This paper is structured as follows: Section 2 provides the background of the research topic through a literature review. Section 3 highlights the main components of the Auto-Simulation Model Builder (ASMB) framework with a higher level of details in its sub-sections. Finally, Section 4 provides an end-to-end demonstration of the simulation model builder using a business process example based on event logs. We also execute "what-if" scenarios that reflect the framework capabilities. The conclusions of our work are presented in Section 5.

2   LITERATURE REVIEW

A variety of techniques are used in the domain of simulation modelling. In their review of simulation modelling in the healthcare domain, Katsaliaki and Mustafee (2011) identified Monte Carlo Simulation (MCS), Discrete-Event Simulation (DES), System Dynamics (SD) and Agent-Based Simulation (ABS) as the most widely used techniques. Each technique has a particular focus of simulation: MCS enables the testing of probability distributions for uncertain factors in the modelled system; DES models pre-defined systems with predictable interactions; ABS models the behavior of particular individual agents within a system where such agents have autonomous behavior; SD examines how the fundamental structure might influence the progressive behavior of a system including tangible elements and intangible elements (Taylor and Lane 1998). The success of these models is linked to the use of an accurate conceptual model upon which the simulation will be based.

It is suggested that hybridization across different methods and approaches will reduce the limitations of individual methods and increase their capabilities (Balaban et al 2014). This can be done by combining more than one paradigm from one category of techniques. For instance, multiple simulation paradigms are integrated into many simulation studies to leverage the models' capabilities for addressing complex problems (Mesabbah et al. 2016). Hybridization can be also used by combining paradigms from different categories of techniques. For example, researchers have investigated the integration between simulation techniques and optimization approaches to facilitate complex evaluation functions for the systems that are optimized (Figueira and Almada-Lobo 2014). Recently, as organisational information systems advance to store and generate increasingly large volumes of data, scholars have been encouraged to explore the integration between data and process mining and modelling and simulation techniques (Mărușter and van Beest 2009). Such hybridization can help to shorten the simulation model development life cycle and overcome the problems that are usually encountered during the data collection and analysis phases.

Process mining has emerged since the early 2000s for the development of accurate unbiased conceptual models. The practice of process mining attempts to reconstruct complete process models from data logs containing real process execution data (Tiwari et al 2008). Through the mining of events logs, accurate portrayals of activities and events can be derived, thus providing a solid basis for overlaying simulation techniques. Mărușter and van Beest (2009) proposed a methodology to use the outputs of process mining as an input to simulation. Their approach provided detailed work of process flow

derivations from mined event logs, and comparative performance of the "as-is" and "to-be" business processes. Liu et al. (2012) derived event graphs from process mining as a conceptual input to simulation models. They illustrated their method in a credit card scenario. Aguirre et al (2012) produced a higher-level methodology for combining process mining with simulation modelling. Their evaluation focused on the domain of procurement processes in a university. Similarly, Abohamad et al (2017) used process mining of system event logs in a health care domain to derive a conceptual model, which was in turn used as the basis for setting up a simulation model.

The work in this paper elaborates on the previous efforts of integrating process mining algorithms with modelling and simulation techniques, in order to enhance and automate the simulation model building process. The proposed framework in this research provides an end-to-end automated simulation model builder. The framework supports high levels of flexibility and adaptability for the developed models because any changes in the process design or its parameters will be reflected in the system event logs and, hence, captured in the simulation model.

3   METHODOLOGY

The Auto-Simulation Model Builder (ASMB) framework (Figure 1) contains two key components: 1) The process miner and 2) The data analyzer. The process miner reads the system events log data which are provided to the system in a pre-defined format. From these logs, the miner derives the underlying process map meta data, consisting of an activities preceding matrix and the associated resources. Using this meta data, the agent pathways are identified along with the existing set of resources and their specializations. On the left-hand side of the diagram, the second component - the data analyzer - uses the system events log data to extract the processing times for each activity, fitting them to the appropriate statistical distribution. In addition, it analyses agents' inter-arrival times in order to draw a statistical distribution for the agents arrival patterns. The ASMB uses the information provided by the two components in order to build the DES model for the system under investigation. This model can be used by the system stakeholders to run different scenarios and investigate their impact on the system KPIs. The following subsections explain each component of the ASMB in more detail.

For the purpose of this paper, the process mining algorithm developed by Janssenswillen and Depaire (2017) will be used for the process miner component. A synthetic events log dataset will be used to demonstrate the various components of the ASMB framework and to evaluate it. The dataset represents a business process of a system that consists of 8 activities, which are handled by 9 different resources with different assignment rules (Figure 3). The time length of the log data is 25 days with more than 19,000 executed events for 1438 unique objects (i.e., agents).

3.1   System Event Logs Data

The events log data is the input for the process miner component and is required in order for the ASMB to work properly. It is a log file that will typically be generated by the IT systems of the organization. It consists of a set of transactions that record a series of events for specific objects (i.e., agents) within the system processes, and may include: 1) a reference for an activity within a process; 2) a reference to an object that was processed or served during this activity; 3) a reference to the resources used to perform this activity; 4) the associated event timestamp; 5) the activity instance state that may be either *start* or *complete* (van Dongen et al. 2005). Since each IT system has a unique format for storing its events log files, a generic format for event logs data tables is developed for the proposed ASMB. This format requires that the system events log data tables must have the set of fields as presented in Table 1. The events log files of any system can be extracted from its database and converted to comply with the required format of the ASMB. Table 2 shows an example of an events log involving 14 events, 5 activities and 4 resources for processing two objects.
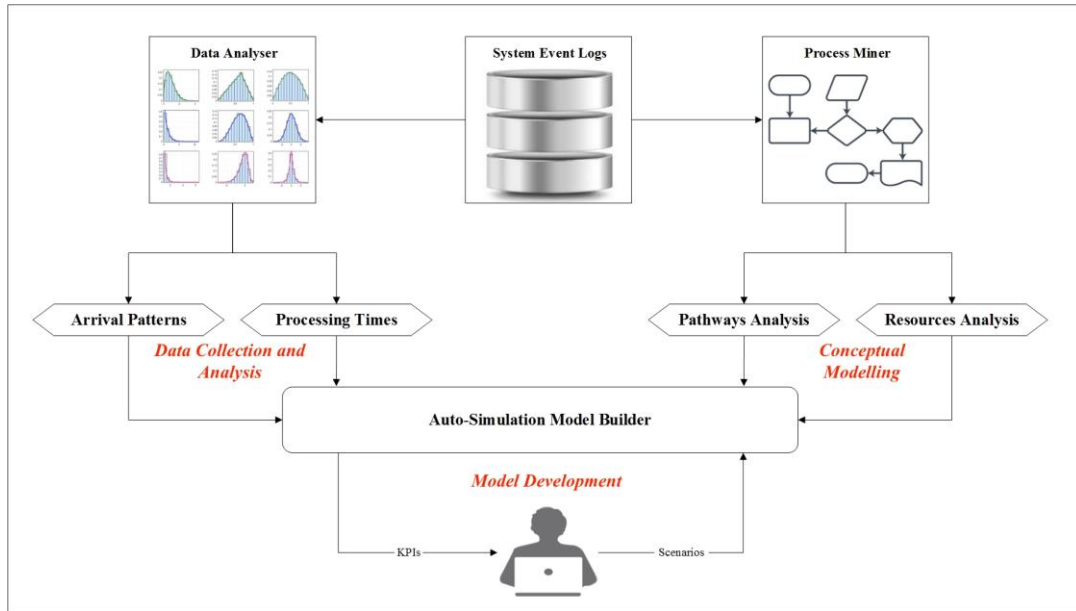
Figure 1: The ASMB Framework.

Table 1: Events Log Data Format.

Field Name	Field Description
Object ID	A unique ID for each object or agent that is being handled during an event. The agent differs according to the system context, e.g., patients in healthcare system, cars in an auto maintenance center, etc.
Activity Name	Unique name for the activity that takes place during an event.
Activity Instance Id	A unique id for each activity instance.
Activity State	A field that specifies the state of the activity instance of an event. The value for this field could be either *'start'* or *'complete'* state.
Resource ID	The unique ID for the resource associated with the activity instance of an event.
Time Stamp	The date and time corresponding to the event occurrence time

3.2 Process Miner Component

The purpose of this component is to generate the process map. The process miner parses the system events log data and applies the process mining algorithm in order to identify the underlying process map along with the associated metadata (van der Aalst et al. 2007). This metadata includes information about the pathways that the agents follow throughout the system process map and the resources assignment to the different activities within the system.

3.2.1 Pathways Analysis

The discovered process map represents the work flow within the system based on all the events that occurred and were recorded in the log file. It is a graph that consists of a set of nodes and a set of arcs, where the nodes represent the process activities and the arcs represent the preceding orders of the activities. The process map can be presented by either graph or matrix views as illustrated in Figure 2. The third column of the activity preceding matrix view in Figure 2.b is the count of objects (i.e., agents)

that moved between any pair of antecedent and consequent activities. That number is also presented on the arcs of the graph view in Figure 2.a. Also, each activity node of the process map graph is labelled with the count of objects that passed through it.

Table 2: Events Log Data Sample.

| Object Id	Activity Name	Activity Instance Id	Activity State	Resource ID	Time Stamp
obj 1	Activity 1	501	start	Res_1	02/01/2017 12:40
obj 2	Activity 1	502	start	Res_2	02/01/2017 12:50
obj 2	Activity 1	502	complete	Res_2	02/01/2017 13:24
obj 2	Activity 2	1474	start	Res_2	02/01/2017 13:31
obj 1	Activity 1	501	complete	Res_1	02/01/2017 13:32
obj 1	Activity 2	1001	start	Res_1	02/01/2017 13:33
obj 2	Activity 2	1474	complete	Res_2	02/01/2017 13:57
obj 2	Activity 5	2231	start	Res_4	02/01/2017 14:02
obj 1	Activity 2	1001	complete	Res_1	02/01/2017 14:14
obj 1	Activity 3	1238	start	Res_3	02/01/2017 14:18
obj 2	Activity 5	2231	complete	Res_4	02/01/2017 14:37
obj 1	Activity 3	1238	complete	Res_3	02/01/2017 14:42
obj 1	Activity 4	1735	start	Res_1	02/01/2017 14:45
obj 1	Activity 4	1735	complete	Res_1	02/01/2017 15:18
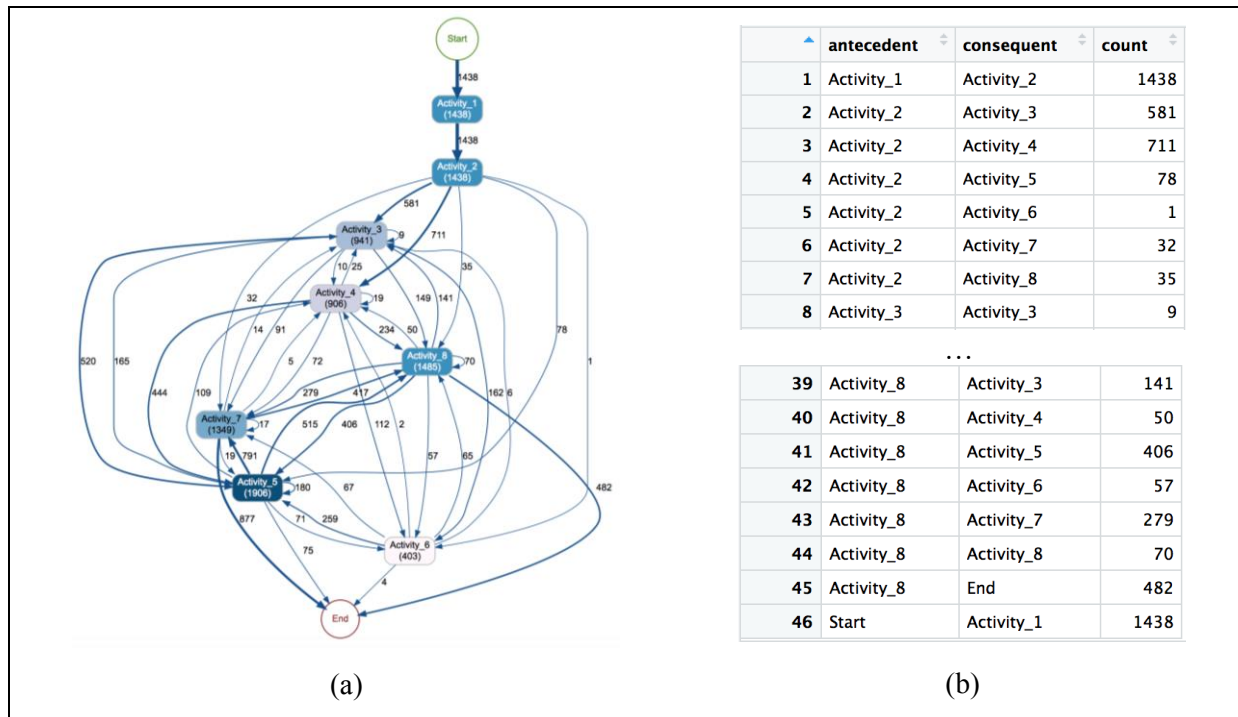
Figure 2: Process Map Graph (a) and Matrix (b) Views.

The preceding matrix (i.e., Figure 2.b) along with the numerical information about the objects transactions within the process map will be used to construct: 1) activities adjacency matrix (**A**) and 2) the transition probability matrix (**P**). Both matrices will be helpful for determining the agent pathways during the simulation of the ultimate model. The adjacency matrix (equation 1) enables the ASMB to identify the set of activities that agent can move to from the current one, while the transition probability matrix (equation 2) is used to reflect the likelihood of moving to any of these activities. The matrices are constructed as follows:

$$A_{ij} = \begin{cases} 1, & if\ activity\ i\ is\ a\ predecessor\ for\ activity\ j \\ 0, & otherwise \end{cases} \tag{1}$$

Where both i and $j = 1, 2, \ldots, m$, and m is the number of the nodes in the process map graph. As mentioned earlier these nodes represent the different activities within the system. Since we have virtual Start and End nodes, the number of activities is (**m-2**).

$$P_{ij} = X_{ij}/C_i \tag{2}$$

Where X_{ij} is the count of agents that moved from activity i to activity j, and C_i is the number of agents that passed through activity i.

The ASMB will use the two matrices to build the DES model structure by creating a layout of 1) the basic blocks for all the activities and their queues; 2) the branching blocks and 3) the relevant connections between the blocks.

3.2.2 Resources Analysis

In addition to the process mining algorithm's ability to capture process pathways data, it is also able to derive data regarding the resources behavior within a business process from the system events log data (Swennen et al. 2016). The behavior of the resources within the business process can be analysed on many perspectives, including: 1) resources-activities involvement and 2) resources-agents involvement (Janssenswillen and Depaire 2017). For the scope of the current version of ASMB in this paper, we will analyze for resources-activities involvement. An activity-resource matrix (**R**) is defined in equation 3.

$$R_{ij} = \begin{cases} 1, & if\ resource\ i\ can\ handle\ activity\ j \\ 0, & otherwise \end{cases} \tag{3}$$

Where $i = 2, 3, \ldots, m\text{-}1$ (No resources are assigned to the start node ($i = 1$) nor the end node ($i = m$)), and $j = 1, 2, \ldots, k$, and k is the number of resources available. Figure 3 shows an example of an activity- resource matrix.

The ASMB relies on the information provided by the extracted activity-resource matrix to create the various resource pools and attach them to the relevant activity blocks within the auto-developed DES model for the underlying system.

3.3 Data Analyzer Component

This component is responsible for extracting all the information regarding the processing times for the various activities, along with the inter-arrival times for the agents from the system events log data. In the

classical approaches for developing DES models, this information is usually driven through direct observations via statistical sampling and measuring cycles techniques (Skoogh and Johansson 2008).

Data mining techniques are applied to the system events log data to retrieve two arrays for time stamps of the *start* and *complete* events for each activity of the business process. The absolute time difference between the two arrays will be the array of the activity processing times. Afterwards, statistical distribution fitting techniques can be used to estimate a statistical distribution for the processing time of each activity (Law 2008). The ASMB relies on these distributions for generating random processing times for the various activity blocks in the developed model. For simplicity, the proposed ASMB described in this paper uses empirical distributions derived from activities processing time arrays for that purpose. But, ASMB can be easily adjusted to use parametric distributions estimated from the events log data.
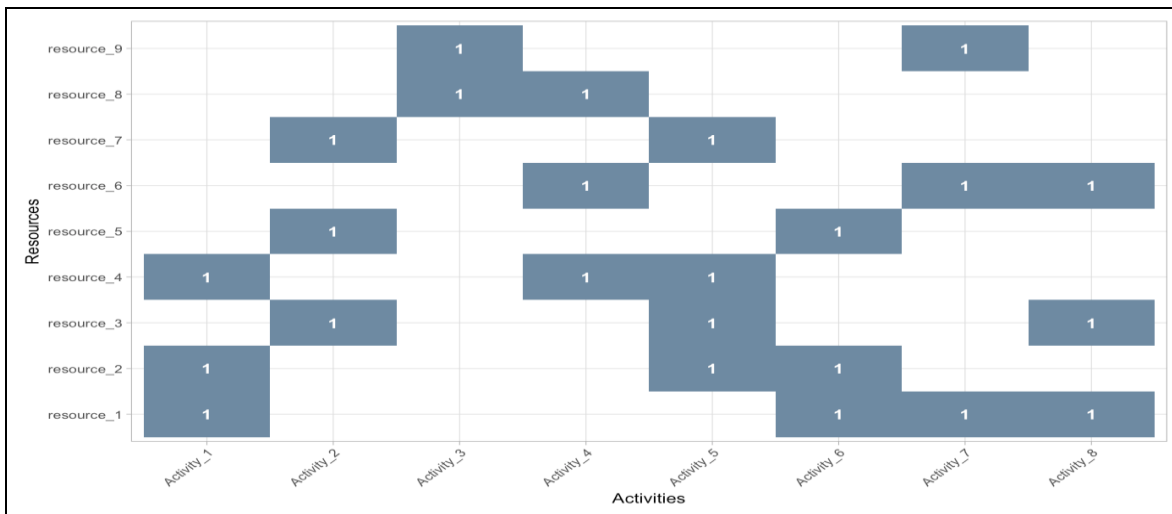

Figure 3: Activity-Resource Matrix.

Similar to activities processing time analysis, the time stamps for the earliest start event for each agent will be retrieved from the system events log data. Then an array of the inter-arrival times between every two successive agents will be calculated. The values of this array will be fitted to relevant statistical distributions to be used for simulating agents arrival in the developed model using the ASMB.

3.4   Auto-Simulation Model Builder

This component is responsible for building a DES model for the business process identified from the system events log files. To do this, we have developed a generic DES engine to construct a DES model based on specific inputs, which are listed in Table 3. These inputs are basically the outputs of both the process miner and data analyzer components for a given system events log data file.

Based on the inputs of Table 3, the DES engine creates a number of activity entities with a linked queue entity for each of them. It also creates a resources pool entity where the resources will be seized by and released to the various activities, subject to agents' interactions during the simulation run time. The work flow of the engine for simulating the DES model is quite simple. It is triggered by the first arrival event, where a sequence of activities (i.e., path of activities) for the arrived agent will be generated according to the AdjMtrx and TProbMtrx matrices (Figure 4). The agent then follows the simulated path of activities: it may be processed directly at any activity or queued up for a time before processing depending upon the resources available. As the simulation time advances, a series of events of different types (i.e., activity start or complete events and new agents arrival events) are created and occur sequentially until the simulation stopping condition is reached. Meanwhile, the engine collects the

relevant statistics regarding a set of performance measures, such as waiting times, queues lengths, resources utilization and agents cycle time.

Table 3: DES Engine Inputs.

| Input Name	Description	Source	Equation No.
ActvArray	Array of 'activity' classes, each class contains: 1) activity name; 2) activity id and 3) processing time distribution.	Data Analyzer	
ResArray	Array of 'resource' classes, each class contains: 1) resource name and 2) resource id.	Process Miner	
AdjMtrx	Binary square matrix for connections between activities	Process Miner	1
TProbMtrx	Transition probability matrix	Process Miner	2
ActvResMtrx	Binary matrix for resources assignment to activities	Process Miner	3
InterArrDist	Distribution for agents inter-arrival times	Data Analyzer	
SimTime	The required simulation time	The User	
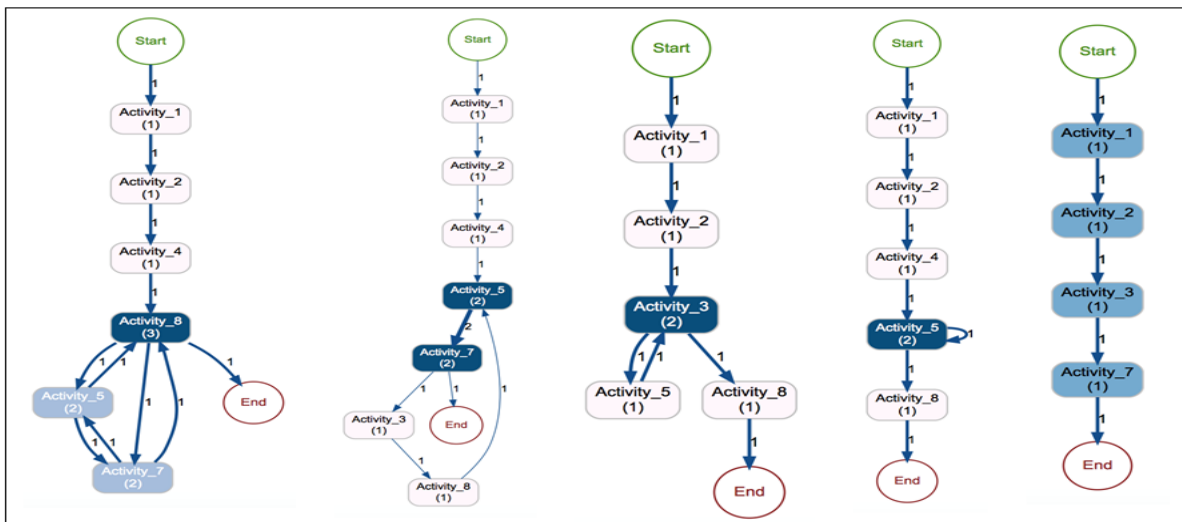
Figure 4: Sample of various simulated paths for arrived agents.

3.5 Running Scenarios

The intrinsic idea behind the ASMB is to enable the managers to develop simulation models for their business in a simple way, using their information systems event logs to investigate possible scenarios to improve their business performance. Therefore, the DES engine is designed to provide flexibility to run different scenarios against the Business As Usual (BAU) scenario. The exploring scenarios may include changes for one or more of the following:

1. **Available Resources** – This can be done by applying changes to the *ActvResMtrx* matrix by either: i) reducing or increasing the activities to be processed by certain resource(s); ii) deleting a certain resource(s); or iii) adding more resource(s). Relevant changes need to take place in case of adding or deleting resource to the resources array (i.e., *ResArray*).

2. **Activities Processing Times** – This can be done by either: i) changing the parameters of the estimated distributions for some activities or ii) using a different distribution for one or more activities.

3. **Business Process Design** – Changes to both *AdjMtrx* and *TProbMtrx* matrices can result in changes to the process map of the business process. The user may eliminate or add new activities by removing or adding columns and rows from/to the two matrices. Relevant changes also need to be applied to the *ActvResMtrx* matrix and the activities and resources arrays (i.e., *ResArray* and *ActvArray*)

4. **Agent Arrival Rate** – This can be done by either: i) changing the parameters of the estimated distribution for inter-arrival times between agents or ii) using a different distribution to simulate agents' arrivals to the system.

In the following section the simulation results of the auto-developed DES model are presented for the BAU scenario. Afterwards, a set of scenarios are investigated using this model for the purpose of improving business process performance.

4 SIMULATION RESULTS

4.1 Model Validation and Verification

A starting assumption in our model validation is that the original system events log files contain the tractions of the agents and the events that occurred. Both the process miner and data analyzer are considered as trusted unbiased bodies for building the conceptual model and collecting data simultaneously based on the internal databases of the actual system.

As mentioned earlier, the dataset used in this paper is a synthetic events log dataset for a business process of 8 activities, 9 resources (Figures 2 and 3) covering a logging time period of 25 days. The process miner and data analyzer process the log file for this business process and calculate all the required inputs for the generic DES engine as illustrated in Table 3. The DES engine is used to simulate the *BAU* scenario for the same time horizon as the original log file (i.e., for 25 days). Table 4 shows the simulation results for 20 runs of the model against the actual system data for a set of performance measures. The results reflect how the performance simulation model is close to the reality which reflects the verification of the developed model.

4.2 Model Scenarios

The generic DES engine developed for ASMB facilitates different ways to explore scenarios for improving business process performance. We use two scenarios to demonstrate the efficacy of ASMB and its simulation engine. For simplicity, the two scenarios are developed to serve the same objective which is: improving the waiting times for the three activities with the highest waiting time as indicated in Table 4 (i.e., activities 1, 4 and 5). The first scenario *(S1)* is developed to improve the performance of the three activities by assigning more resources to them from the available set of resources. Under *S1*, the least three utilized resources, resources 5, 9 and 8 are assigned to activities 1, 4 and 5 respectively (within the constraints of the activity-resource matrix).

The second scenario *(S2)* adds one extra resource (i.e., resource 10) to the current set of resources, and it will be assigned to the three activities. Similar to the *BAU* scenario, and for the sake of consistency, both scenarios will be simulated for 20 runs. The outcomes for the two scenarios are reported in Table 5 along with calculations for the percentage deviations from the corresponding measures of performance under the *BAU* outcomes in Table 4. The same random generator seed is used for the two scenarios and the *BAU* one, also to ensure the consistency when comparing them against each other.

Table 4: Simulation results for 20 runs versus the real system data.

Average Waiting Time (Hours)								
Activity No.	Activ_1	Activ_2	Activ_3	Activ_4	Activ_5	Activ_6	Activ_7	Activ_8
Simulation	0.29	0.15	0.28	0.30	0.31	0.13	0.19	0.25
Actual	0.34	0.15	0.31	0.35	0.33	0.12	0.23	0.29
Average Queue Length								
Activity No.	Activ_1	Activ_2	Activ_3	Activ_4	Activ_5	Activ_6	Activ_7	Activ_8
Simulation	1.43	0.93	1.10	1.12	1.67	0.59	1.04	1.31
Actual	1.61	0.97	1.15	1.22	1.74	0.58	1.13	1.42
Resources Utilization (%)								
Resource No.	Res_1	Res_2	Res_3	Res_4	Res_5	Res_6	Res_7	Res_8	Res_9
Simulation	86	92	91	95	64	86	88	85	77
Actual	83	88	87	91	61	82	84	81	73
Average Agent Cycle Time (Hours)								
Simulation	4.67							
Actual	5.03							

The results presented in Table 5, suggested that both *S1* and *S2* scenarios managed to achieve their objective to improve the performance of activities 1, 4 and 5. However, some undesired effects appeared under the two scenarios. Although *S1* is more efficient than *S2* since there are no incurred costs for new resources, some other activities' performance are negatively impacted (orange highlighted cells). In addition, the utilization for some resources has dropped as another side effect of this scenario. *S2* has a positive impact on all activities' performance despite the negative effect on the old resources utilization (yellow highlighted cells). We also note that both scenarios have a positive impact on the overall agent cycle time. However the magnitude of *S2* in this regard is higher than *S1*, but the increased costing of hiring extra resources should also be considered.

5   CONCLUSION

The main objective of this research is to present a generic and automated end to end simulation framework that produces an accurate unbiased simulation model, based on a conceptual model and various metadata derived from actual system recordings of process events. The different components of the ASMB framework are discussed in this paper along with evaluation of its functionality using a synthetic events log data for a business process. The simulation results suggested that the outcomes of the auto-generated DES model for the underlying business process are similar in comparison to the actual system data. To explore the efficacy of the ASMB, two scenarios are investigated to target the improvement of certain activities with long waiting times and queues lengths by 1) changing the specialization of low utilized resources and 2) adding extra resources to the system. The simulation results for the two scenarios demonstrate tangible improvements for the business process performance, in particular reduction in agents waiting times and a shortening of the waiting queues along with significant improvement in agents cycle time.

The hybridization of process mining and simulation techniques in this research has provided more capabilities beyond those suggested in previous attempts in this research area. In addition to the role of developing the conceptual model, the process mining algorithms are able to extract all the metadata needed to trigger a generic DES engine that can simulate both "as-is" and "to-be" business process scenarios. There are multiple benefits of the ASMB including: 1) Significant saving to the time and costs

of using the conventional simulation modelling steps; 2) Reducing the modeler role and their involvement in the model development process 3) Generation of an unbiased conceptual model as an auto-input to the simulation model and 4) High level of flexibility as the model is updated automatically subject to the changes in the system logs which reflects the changes of the real business process. We also note that the quality of the developed model is highly dependent on the quality of the system events log data. If the underlying data is deficient, the accuracy of the resultant model in this case will be compromised. Our work can be extended to enhance the developed DES engine and make it more sophisticated. For example, handling cases where there are limited capacities for the waiting queues of the various activities.

Table 5: Scenarios results for 20 runs and the percentage deviations of the BAU scenario performance.

Average Waiting Time (Hours)									
Activity No.	**Activ_1**	**Activ_2**	**Activ_3**	**Activ_4**	**Activ_5**	**Activ_6**	**Activ_7**	**Activ_8**	
S1	0.13	0.17	0.46	0.24	0.19	0.16	0.20	0.18	
(+/-)%	-55%	13%	64%	-20%	-39%	23%	5%	-28%	
S2	0.11	0.09	0.19	0.12	0.11	0.06	0.10	0.12	
(+/-)%	-62%	-40%	-32%	-60%	-65%	-54%	-47%	-52%	
Average Queue Length									
Activity No.	**Activ_1**	**Activ_2**	**Activ_3**	**Activ_4**	**Activ_5**	**Activ_6**	**Activ_7**	**Activ_8**	
S1	0.83	1.00	1.55	1.02	1.14	0.69	1.11	1.05	
(+/-)%	-42%	8%	41%	-9%	-32%	17%	7%	-20%	
S2	0.76	0.66	0.88	0.64	0.81	0.41	0.71	0.82	
(+/-)%	-47%	-29%	-20%	-43%	-51%	-31%	-32%	-37%	
Resources Utilization (%)									
Resource No.	**Res_1**	**Res_2**	**Res_3**	**Res_4**	**Res_5**	**Res_6**	**Res_7**	**Res_8**	**Res_9**	**Res_10**
S1	80	85	85	90	75	83	83	93	89	
(+/-)%	-7%	-8%	-7%	-5%	17%	-3%	-6%	9%	16%	
S2	75	81	80	85	55	76	78	78	70	85
(+/-)%	-13%	-12%	-12%	-11%	-14%	-12%	-11%	-8%	-9%	
Average Agent Cycle Time (Hours)									
S1	4.44								
(+/-)%	-5%								
S2	3.84								
(+/-)%	-18%								

REFERENCES

Abohamad, W., A. Ramy, and A. Arisha. 2017. "A Hybrid Process-Mining Approach for Simulation Modeling." In *Proceedings of the 2017 Winter Simulation Conference ,*edited by W. K. V. Chan et al., 1527-1538, Piscataway, New Jersey: IEEE.

Aguirre, S., C. Parra, and J. Alvarado. 2012. "Combination of Process Mining and Simulation Techniques for Business Process Redesign: a methodological approach." International Symposium on Data-Driven Process Discovery and Analysis.

Balaban, M., P. Hester, and S. Diallo. 2014. "Towards a Theory of Multi-Method M&S Approach: part I." In *Proceedings of the 2014 Winter Simulation Conference,* edited by A. Tolk et al. , 1652-1663, Piscataway, New Jersey: IEEE.

Figueira, G., and B. Almada-Lobo. 2014. "Hybrid Simulation-Optimization Methods: A Taxonomy and Discussion." *Simulation Modelling Practice and Theory* 46:118-134. doi: 10.1016/j.simpat.2014.03.007.

Hlupic, V., and S. Robinson. 1998. "Business Process Modelling and Analysis using Discrete-Event Simulation. In *Proceedings of the 1998 Winter Simulation Conference ,* edited by D.J. Medeiros et al., 1363-1369, Piscataway, New Jersey: IEEE.

Janssenswillen, G., and B. Depaire. 2017. "BupaR: Business Process Analysis in R."

Katsaliaki, K., and N. Mustafee. 2011. "Applications of Simulation within the Healthcare Context." *Journal of the Operational Research Society* 62 (8):1431-1451.

Law, A. M. 2008. "How to Build Valid and Credible Simulation Models." In *Proceedings of the 2008 Winter Simulation Conference,* edited by S. J. Mason et al. , 39-47, Piscataway, New Jersey: IEEE.

Liu, Y., H. Zhang, C. Li, and R. J. Jiao. 2012. "Workflow Simulation for Operational Decision Support using Event Graph through Process Mining." *Decision Support Systems* 52 (3):685-697.

Lynch, C., J. Padilla, S. Diallo, J. Sokolowski, and C. Banks. 2014. "A Multi-Paradigm Modeling Framework for Modeling and Simulating Problem Situations." *In proceedings of the 2014 Winter Simulations Conference ,* edited by A. Tolk et al., 1688-1699, Piscataway, New Jersey: IEEE.

Măruşter, L., and N. R. van Beest. 2009. "Redesigning Business Processes: A Methodology Based on Simulation and Process Mining Techniques." *Knowledge and Information Systems* 21 (3):267.

Mesabbah, M., A. Mahfouz, M. A. F. Ragab, and A. Arisha. 2016. "Hybrid Modeling for Vineyard Harvesting Operations." In *the proceedings of 2016 Winter Simulations Conference (WSC)*, edited by T. M. K. Roeder et al. , 1642-1653, Piscataway, New Jersey: IEEE.

Skoogh, A., and B. Johansson. 2008. "A Methodology for Input Data Management in Discrete Event Simulation Projects." *In proceedings of the 2008 Winter Simulations Conference (WSC):* edited by S. J. Masone et al. , 1727-1735, Piscataway, New Jersey: IEEE.

Swennen, M., N. Martin, G. Janssenswillen, M. J. Jans, B. Depaire, A. Caris, and K. Vanhoof. 2016. "Capturing Resource Behaviour from Event Logs." *RWTH Aachen University.*

Taylor, K., and D. Lane. 1998. "Simulation applied to Health Services: Opportunities for Applying the System Dynamics Approach." *Journal of health services research & policy* 3 (4):226-232.

Tiwari, A., C. J. Turner, and B. Majeed. 2008. "A Review of Business Process Mining: state-of-the-art and Future Trends." *Business Process Management Journal* 14 (1):5-22.

van der Aalst, W., H. Reijers, A. Weijters, B. van Dongen, A. k. A. de Medeiros, M. Song, and H. M. W. Verbeek. 2007. "Business Process Mining: An Industrial Application." *Information Systems* 32 (5):713-732. doi: 10.1016/j.is.2006.05.003.

van Dongen, B. F., A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, W. M. P. van der Aalst, G. Ciardo, and P. Darondeau. 2005. "The ProM Framework: A New Era in Process Mining Tool Support." *Applications and Theory of Petri Nets 2005, Proceedings* 3536:444-454.

AUTHOR BIOGRAPHIES

MOHAMMED MESABBAH, PhD. is a senior Researcher in the Centre for Applied Data Analytics group, School of Computing, Dublin Institute of Technology (DIT). Mohammed obtained his PhD in developing smart planning and decision support tools for complex systems from the DIT and pursued research in system modelling, data mining, machine learning and statistical analysis. He also holds a MSc in system dynamics from university of Bergen (UiB) and bachelor degree from Faculty of Computers and Information in Cairo University. His Research interests include Big Data Analytics, Simulation and Modelling, Machine Learning and Optimization Algorithms. His email address is mohamed.mesabbah@dit.ie.

SUSAN MCKEEVER, PhD. is a principal investigator in the Centre for Applied Data Analytics technology group and senior lecturer in the School of Computing at the Dublin Institute of Technology (DIT). Her research is in the domains of autonomous & sensor driven systems and machine learning for both structured and unstructured data analytics. She holds an Honors degree in Engineering from Trinity College Dublin, an MSc in IT for Strategic Management from DIT and PhD from University College Dublin. Her email address is susan.mckeever@dit.ie.