A BPMN extension to support discrete-event simulation for healthcare applications

DOI:
10.1057/s41274-017-0267-7

A BPMN Extension to Support Discrete-Event Simulation for Healthcare Applications: An Explicit Representation of Queues, Attributes and Data-Driven Decision Points

Abstract

Stakeholder engagement in simulation projects is important, especially in healthcare where there is a plurality of stakeholder opinions, objectives and power. One promising approach for increasing engagement is facilitated modelling. Currently, the complexity of producing a simulation model means that the 'model coding' stage is performed without the involvement of stakeholders, interrupting the possibility of a fully-facilitated project. Early work demonstrated that with currently-available software tools we can represent a simple healthcare process using Business Process Model and Notation (BPMN) and generate a simulation model automatically. However, for more complex processes, BPMN currently has a number of limitations, namely the ability to represent queues and data-driven decision points. To address these limitations, we propose a conceptual design for an extension to BPMN (BPMN4SIM) using Model Driven Architecture. Application to an elderly emergency care pathway in a UK hospital shows that BPMN4SIM is able to represent a more-complex business process.

Keywords: discrete-event simulation; facilitated modelling; healthcare; BPMN; model-driven architecture

1. Introduction

Stakeholder engagement is important for the success of a simulation project (Robinson *et al*, 2014; Tako and Kotiadis, 2015). This is especially true in healthcare, where stakeholders often have a plurality of opinions and conflicting objectives (Proudlove *et al*, 2007; Franco and Montibeller, 2010; Robinson *et al*, 2012; Pitt *et al*, 2016). Thus, it is difficult to both engage multiple stakeholders during a simulation study and manage any conflicting interests (Brailsford, 2005; Taylor *et al*, 2009; Jahangirian *et al*, 2015). The apparent lack of success in implementing discrete event simulation (DES) studies in healthcare has prompted researchers to reflect on the specific barriers in this domain, particularly in managing to obtain and maintain stakeholder engagement (Brailsford, 2005; Brailsford *et al*, 2009b; Taylor *et al*, 2009; Jahangirian *et al*, 2015). One promising approach to improve the involvement of stakeholders in a simulation project is through facilitated modelling. If we define 'fully-facilitated' DES modelling to mean that *all* simulation lifecycle stages are conducted live with stakeholders in one or more workshop sessions then it is clear that existing studies (e.g. Robinson *et al*, 2012; Robinson *et al*, 2014; Kotiadis *et al*, 2014; Tako and Kotiadis, 2015) have not yet achieved this because the model coding stage is performed outside workshops in an 'expert' rather than 'facilitated' mode. Robinson *et al*, 2014 suggest that one approach would be the development of a seamless software environment suitable to support facilitated process mapping and generation of a simulation model (the model coding) with stakeholder groups, a capability thought not to be available.

We have successfully carried out a fully-facilitated DES modelling project in a hospital in Italy, maintaining stakeholder engagement throughout a healthcare improvement project (Bisogno *et al*, 2016b). This was made possible by two standards: BPMN (Business Process Model and Notation) (OMG, 2011) and BPSim (Business Process Simulation interchange standard) (WfMC, 2013, 2016). These two standards enabled the automatic generation of the DES model in BPSim from the BPMN process map, without the need for a modeller to perform the 'coding'. Software tools such as the one we used, which incorporate both standards, can make this a seamless process, possible to do live with the stakeholders. We applied the same approach to the simulation of more-complex patient flow cases in an NHS acute hospital trust. The stakeholders found BPMN a very useful and accessible tool for mapping patient flows, and have continued to use it themselves. However, in these simulation projects we experienced a number of technical complexity barriers when it came to generating the DES. These technical barriers are caused by the currently limited capability of the BPMN 2.0 and BPSim (1.0 and 2.0 beta) standards to represent important features of the more complex flows, particularly queues, attributes and data that can affect and be affected by an element in a business process.

In this paper, we propose the extension of BPMN to represent these features. This extended BPMN (BPMN4SIM) will be capable of representing more-complex healthcare processes which require an explicit representation of queues, attributes and data-driven decision points. BPMN4SIM is needed before we can implement a software tool that can translate a process model of a system containing such features, represented in a BPMN diagram, to a simulation model. This software would enable us to conduct fully-facilitated simulation modelling for more-complex business processes. Choices about how these features should be represented were informed by discussions with stakeholders in the NHS projects. The extension of BPMN to do this has not previously been described in the literature. Therefore, the research questions addressed in this paper are:

RQ1. *How can BPMN be extended to represent queues and data?*

RQ2. *Can the extended BPMN (BPMN4SIM) represent a more-complex real-world business process in healthcare?*

It should be noted that business process modelling and simulation have been discussed in the literature for over 20 years. In the early years, research in this area focused on the development of methods and tools to simulate business processes (Melao and Pidd, 2000; van der Aalst *et al*, 2010). As a result, a wide range of proprietary process modelling languages have been specially developed by simulation vendors for their own simulation software. Only a few simulation software tools support the simulation of a process model written in a standard process modelling language such as BPMN or UML (e.g. Bizagi and Simul8 support BPMN). This means that when a process model which has been created using a standard is available, we often cannot simulate it directly. Instead, we need to transform it to the proprietary language used by a particular simulation software tool (in many cases, we need to re-draw the structure using the simulation tool). This creates a few problems. The manual transformation from a process model to a simulation model requires expertise, can be a lengthy process and is prone to human error. In addition, modellers and stakeholders need to be familiar with multiple languages (a business process modelling language and a simulation modelling language). Alternatively, they could focus on just one environment and the proprietary modelling language provided by a particular simulation software vendor, but this ‘locks them in’ to that particular vendor.

Our work contributes to existing research that aims to enable the direct simulation of a business process model written in a standard modelling language, in this case BPMN. Although BPMN has advantages over other standards (Section 2) it has a number of limitations for use in simulation projects, as mentioned earlier. These, in turn, limit its usefulness in projects aiming to simulate more-complex systems, without resorting to additional proprietary features of a vendor's simulation software package. Specifically, the contribution of this paper is the design of an artefact (i.e. a BPMN extension) to incorporate queues and data. This artefact will bring us closer to the possibility of simulating a more-complex business process model represented as a BPMN diagram. Subsequently, this will bring us closer to the possibility to conduct fully-facilitated DES projects for more-complex real-world systems through automatic translation ('coding') of a BPMN diagram, created in a stakeholder workshop, to a working DES model.

The remainder of this paper is organised as follows. In Section 2, we present an overview of the methods previously used to extend BPMN to represent concepts that are relevant to simulation such as resources and performance indicators. This section also summarises research that aims to make the simulation of a model written in BPMN a reality. In Section 3, we explain the research method that we used in designing the BPMN extension. This is followed in Section 4 by discussion of the design of BPMN4SIM and its justification. Next, we evaluate the design using a real-world example (the elderly emergency care patient pathway) from an NHS acute hospital trust. Finally, we present our conclusions and suggestions for future work.

2. Related works

A recent analysis of the literature on simulation and modelling in healthcare shows that since 2000, there has been an increase in the amount of research on process mapping tools such as UML and IDEF to enable communication between healthcare process modelling and simulation modelling (Brailsford *et al*, 2009a). UML (OMG, 2015) is a standard managed by the Object Management Group (OMG). In healthcare simulation and modelling, researchers have demonstrated how UML can be used to represent a conceptual model before it is coded using simulation software (e.g. Roux *et al*, 2006; Vasilakis *et al*, 2009; Reynolds *et al*, 2011). A software tool can be written to convert the process map into a simulation model automatically (e.g. Fanti *et al*, 2012; Augusto and Xie, 2013) or semi-automatically (e.g. Roux *et al*, 2006). Since UML is not designed with simulation modelling in mind, an automatic translation from a process map into a simulation model requires researchers to extend the existing UML meta-model (Roux *et al*, 2006; Augusto and Xie, 2013). However, UML is designed with system architects, software engineers and software developers in mind and is complex for process stakeholders (such as healthcare and service improvement professionals) to understand (Onggo 2013). The biennial survey conducted by Harmon (2016) supports this view, as it shows a decrease in the proportion of respondents interested in using UML as their choice of business process modelling language (from 33% in 2005 to 17% in 2015).

In contrast, BPMN has been designed with business users in mind, and is easier for such stakeholders to understand (Onggo, 2013). Harmon's latest survey (2016) shows that the proportion of respondents who are interested in BPMN has increased significantly (from 22% in 2005 to 64% in 2015). BPMN is a widely-used standard for modelling business processes, particularly when software systems are being designed to control, monitor and audit business processes. However, like UML, BPMN is not specifically designed for simulation modelling; the BPMN 2.0 specification (OMG, 2011, p. 22) states that operational simulation (of the business processes) is out of its scope. Hence, a BPMN diagram does not provide a visual representation of some elements commonly used in a simulation model such as queues and resources. Despite the better fit of BPMN to use with stakeholders, to the best of our knowledge, there has not been any work to extend BPMN as has been done with UML. This is unfortunate since researchers and vendors have recognised the potential of a stronger link between BPMN and simulation (Waller *et al*, 2006; Vasilecas *et al*, 2103; Wagner, 2014). This is because BPMN aims to bridge business process design and its implementation. It is desirable that thorough analysis is carried out for a new business process before its implementation. Simulation modelling is one of the best tools for such analysis of a new system (including a new business process). Given that most DES is process-oriented and supports a good animation that is useful in engaging with business users, it can be argued that simulation modelling (especially DES) is the best tool for testing and demonstrating models developed with BPMN. The main goal of this research area is to make it easier for users to convert a business process model represented using a BPMN diagram to a simulation model of the business process.

From the literature on BPMN extensions, we have identified three modelling elements that are relevant to closing the gap between a BPMN diagram and simulation (see Table 1). They are resources, queues, and key performance indicators (KPIs).

Table 1: Summary of modelling elements needed to convert a BPMN diagram into a simulation model from literature

Resources	Queues	KPIs	Paper
		Explicit	Bocciarelli and D'Ambrogio (2011)
		Explicit	Friedenstab *et al* (2012)
		Explicit	Lodhi *et al* (2011)
		Explicit	Salles *et al* (2013)
Explicit			Stroppi *et al* (2011a)
Explicit	Implicit		Waller *et al* (2006)
Explicit			Zor *et al* (2011)

2.1 Resources

One of the modelling elements that does not have any visual representation in BPMN is resources. Resources are a key modelling element in simulation. Hence, simulation software vendors (e.g. Waller *et al*, 2006) and researchers (Stroppi *et al*, 2011a; Zor *et al*, 2011) have recognised this as a key issue. Stroppi *et al* (2011a) extend the BPMN meta-model to provide support for the visualisation of resources. They introduce resource structure (characterisation and classification of resources), work distribution (how work is distributed to resources) and authorisation (defining a set of task privileges granted to a resource). Although their main objective is to provide interoperability between resource in BPMN and WS-BPEL (Web Services - Business Process Execution Language), their proposed extension is also relevant to bridging the gap between BPMN and simulation. Zor *et al* (2011) extend BPMN to make it more suitable to model manufacturing processes. Their extension includes resources and resource flows (because in manufacturing it is often necessary to move resources such as small machines). They propose an extension similar to the resource structure in Stroppi *et al* (2011a). These BPMN extensions have not addressed the more-complex resource behaviours such as time-dependent resource availability (such as staffing levels), resource pre-emption and prioritisation of resources.

2.2 Queues

From the perspective of simulation modelling, a queue is an important modelling element. In fact, DES modellers view flow systems as networks of queues and activities. Furthermore, organisations often set their KPIs based on length-of-time-in-system measures (e.g. the percentage of patients discharged from the A&E [Accident and Emergency] department within four hours in the English NHS), within which queuing times can be substantial and should be identified in an improvement project. A focus on queue lengths can also be useful in identifying the bottleneck in a business process. To the best of our knowledge, there is nothing in the research literature that proposes a BPMN extension to support an explicit representation of queues. A recent extensive survey of BPMN extensions (Braun and Esswein, 2014) makes no mention of how queues could be represented. It is not surprising that vendors of simulation tools that support BPMN model building or import often have to automatically add an implicit queue in front of every activity translated from a BPMN diagram (e.g. Waller, 2006).

2.3 Key Performance Indicators

Another important modelling element is key performance indicators (KPIs). KPIs are needed at the analysis, design, execution, and evaluation stages of a business process lifecycle. At the analysis stage, KPIs are needed to indicate any possible issue with the current system. If there is an issue, we may need to compare alternative designs. KPIs are often used to select the best design. When a new design is executed (i.e. implemented in the real world) KPIs can be used for business process monitoring. This may enable corrective actions to be taken when the performance falls below a certain threshold. KPIs are also needed in the evaluation of a business process that is being executed (for a non-terminating system) or has been executed (for a terminating system). From the perspective of simulation, KPIs (or 'outputs' in Robinson, 2008) are an essential element in a simulation model because they provide measures that indicate the expected outcomes from a system that is being modelled. Hence, KPIs are important for both BPMN and simulation. Among the business process modelling community, several extensions have been proposed. Lodhi *et al* (2011) propose an extension to BPMN to support the evaluation of a business process at post-implementation stage, i.e. based on KPIs collected from the real world. For business process monitoring, KPIs are often specified in relation to a service level agreement (SLA), for example, the four-hour discharge from A&E target in the NHS. Examples of work to extend BPMN to represent SLAs include Bocciarelli and D'Ambrogio (2011), Friedenstab *et al* (2012), and Salles *et al* (2013).

It should be noted that many other extensions have been proposed for BPMN (see a review by Braun and Esswein, 2014). However, few of them have direct relevance to simulation modelling.

2.4 Transforming a BPMN diagram into a simulation model

BPMN specifies the structure of a business process from its library of element types. To be more useful for DES we have identified that an extension to this library is necessary. A further component of this extension is a specification language that assigns additional information to the elements of the BPMN diagram (such as specifying a service-time distribution function to an activity). The top box in Table 1 represents in the grey section the standard elements available for BPMN diagrams (e.g. rounded rectangle for activity, circle for event, diamond for gateway, arrow for sequence flow) and proposed extended elements in the white section (e.g. triangle for queue). The bottom box denotes a specification language which adds information to relevant BPMN elements (e.g. service-time to activity, time between event occurrences to start event, capacity to queue). This can be written in a specification language such as BPEL (Business Process Execution Language) or a standard such as BPSim to provide information on parameters and logic needed by the simulation model.

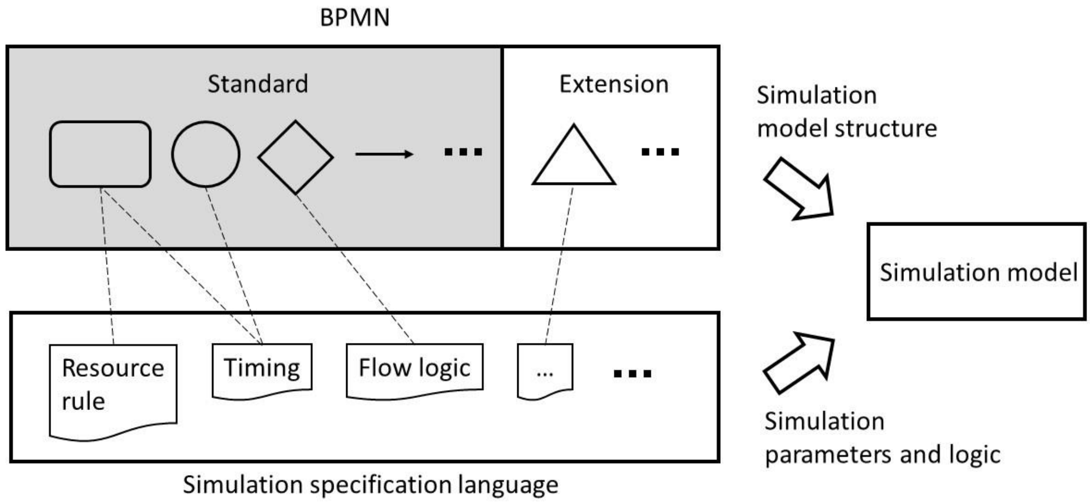


Figure 1: Converting a BPMN diagram to a simulation model

Currently, BPSim is the only standard specifically designed to support the simulation of a BPMN diagram. BPSim 1.0 (WfMC 2013) groups the information needed to simulate a BPMN model into six categories (time, control, resource, cost, property and priority). BPSim has promising capabilities (Bisogno *et al*, 2016a) but it also has some major limitations. The implicit queues assumed in BPSim are not suitable to represent some important queues in healthcare in which patients are still consuming resources while waiting in the queues (for example a patient in an A&E cubicle waiting for an inpatient bed). In addition, BPSim 1.0 does not support a number of commonly used simulation parameters such as time-dependent behaviours and a warm-up period (the latter is supported in BPSim 2.0 beta released in July 2016 (WfMC, 2016)). BPSim does not make use of any extensions to BPMN, but translates the structure of a standard BPMN (2.0) diagram into a DES and then allows a modeller to add, at simulation stage, necessary parameters and logic.

Hence, it is clear from the literature that more development is needed to allow a seamless conversion from a BPMN diagram to a simulation model, particularly where more-complex flow features must be represented. Wagner (2014, p. 111) acknowledges that the best way to adapt the syntax and semantics of BPMN for simulation modelling is still an open question. However, he argues that despite the limitations of BPMN, it still provides the best choice in comparison to alternatives such as flow-charts, Petri nets and UML. His argument is supported by Onggo (2013) who compares the use of flow-charts, Petri nets, DEVS (Discrete Event System Specification) and UML for simulation modelling. Similarly, although the link between BPMN and simulation still has gaps, a number of software vendors have started to provide some functionality to run a simulation from a BPMN model, for example ADONIS (BOC Group, 2016), Bizagi (Bizagi, 2016), L-SIM (Lanner, 2016), Signavio (Signavio, 2016) and Simul8 (Simul8 Corporation, 2016). The compromise is that they have to use proprietary formats to support the simulation of a BPMN model. It should be noted that Bizagi, L-SIM and Simul8 use BPSim. However, it is clear that they also have to use proprietary formats to deal with elements that are not supported by BPMN and BPSim such as queues and data/attributes. This paper proposes a design for a BPMN extension to incorporate queues and data.

3. Method

The overarching objective of our research is to make fully-facilitated DES modelling for complex healthcare processes a reality. The main technical complexity that hinders us in achieving this objective is the lack of a tool that allows us to translate a standard business process diagram such as BPMN into a simulation model for anything other than simple processes. The work presented in this paper deals with the design of a BPMN extension that makes it suitable for simulation modelling in healthcare.

We conducted an action research programme to investigate the use of BPMN to map patient pathways during facilitated simulation modelling projects at two hospitals in the UK and Italy. The main output from the research (Bisogno *et al*, 2016b) is features that BPMN should have to enable fully-facilitated simulation modelling. The features are:

1. Flows and Hierarchies:

- There are situations where it can be useful to distinguish types of flow (e.g. patients vs. information). Likewise, it can be useful to distinguish between push flows (e.g. an activity transfering a patient downstream to an unbounded queue as soon as it is finished) and pull flows (e.g. do not transfer a patient until a pull signal is received from elsewhere in the system).
- Patient pathways can be complex, so a hierarchical approach is very useful (i.e. the ability to represent sub-models in a model).

2. Interactions between data and flow: Activities along a clinical pathway may generate a lot of data. This information may affect the flow of patients (e.g. evidence-based routing). Hence, there is a need to capture how information (from data) affects a BPMN flow element (such as an activity and event). The data can be stored in a physical database and/or attached to a token (in BPMN terminology, or ‘entity’ in DES) as an attribute of the token.
3. Queues: Managing queues is essential in healthcare operations. A number of key performance indicators and improvement objectives are related to queues. Hence, there is a need for explicit representation of queues. Stakeholders in our study have expressed the need to differentiate between a queue in which entities are holding resources while waiting (i.e. a queue with resource-holding entities) and a queue in which entities do not hold any resources while they are waiting. Figure 2 shows an example in which entities arrive and join Queue 1 for the Activity. The Activity requires a Resource. When an entity has completed the Activity, it will wait in Queue 2 until the downstream activity is ready. However, while waiting in Queue 2, the entity does not release the Resource. The Resource is released only when the entity leaves Queue 2. This is a typical situation in healthcare. For example, whilst being treated, elderly inpatients occupy beds in wards. Once their treatment is finished and they are medically fit for discharge, patients can only leave the beds when they can be safely transferred to their homes or community care. Thus the patients continue to occupy the beds while waiting for the next destination to become ready (i.e. Queue 1 is the queue for treatment, the Activity is treatment, the Resource is beds, and Queue 2 is the queue for home or community care). The need to differentiate between queues with non-resource-holding entities (Queue 1 in Figure 2) and queues with resource-holding entities (Queue 2 in Figure 2) arises partly from the different level of control that the management has. The local management may have little or no control over Queue 2 in comparison to Queue 1 – it may be outside the span of their

control. Hence, they want to be able to explain whether the performance of their unit is due to their decision or someone else's decision. Another reason why they need to differentiate the two queues is because they want to know whether resource is utilised for an activity or is held by an entity who has completed the activity as this will lead to different managerial decisions.

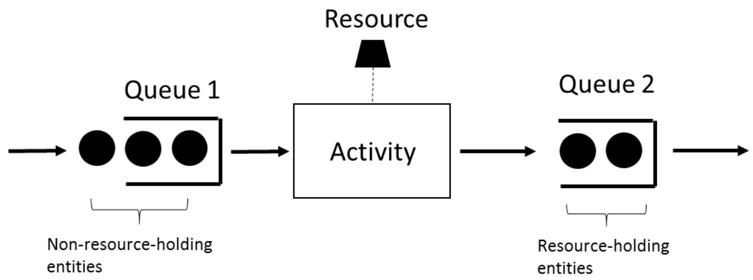


Figure 2: Queues with resource-consuming entities and non-resource-consuming entities

In addition to the above features, stakeholders in our projects highlighted the importance of icons or notation that is prominent and distinct. This is because queue identification and management are particularly important in healthcare improvement and the prevalence of queuing states may not otherwise be very apparent to frontline staff, for example the patient still occupying a bed despite the activity of treatment having been completed.

To help us design an artefact (i.e. BPMN extension) that is suitable for simulation modelling in healthcare based on the above features we use an engineering design approach introduced by OMG called Model Driven Architecture (MDA) (OMG, 2014). MDA provides a set of guidelines for applying model-driven engineering principles (Schmidt, 2006) to the development of software systems. Metamodeling techniques are key enabling principles introduced in the field of model-driven engineering. A metamodel is a model used to describe a family of models, in other words it is a model that defines the primitives of a modelling language, which is used to specify models at user level. As an example, the BPMN metamodel is the model defining the BPMN primitives (i.e., task, gateway, event, etc.) that are instantiated in standard BPMN models. The proposed BPMN extension has been developed as a metamodel extension, according to principles and standards set in MDA.

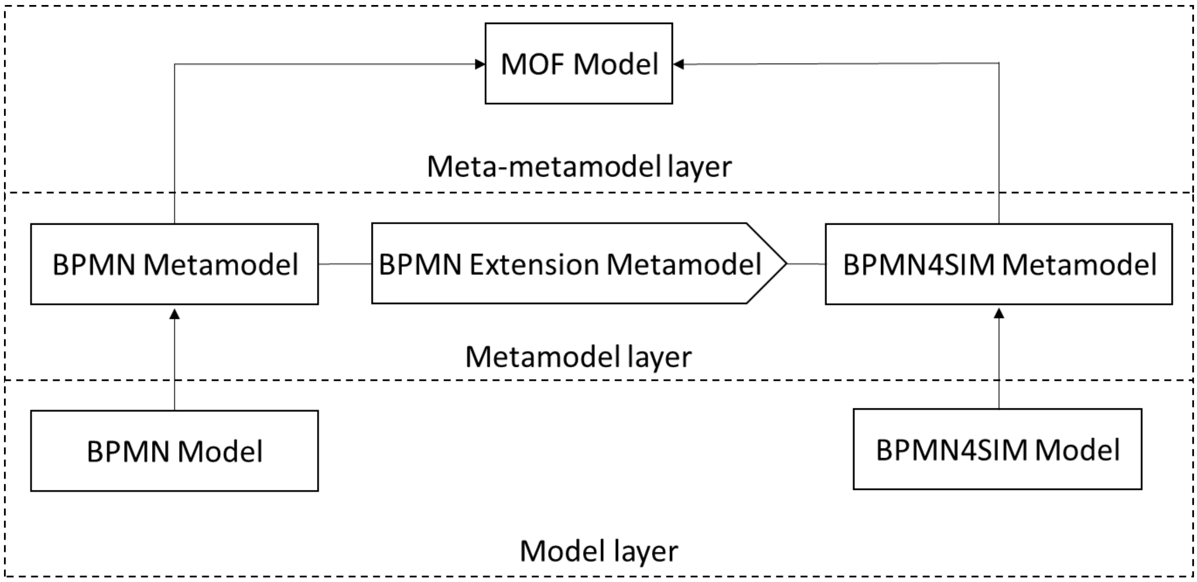


Figure 3: BPMN4SIM model-driven extension method

Figure 3 illustrates the metamodel-based method used to implement the BPMN4SIM. A BPMN model is an instance of the BPMN metamodel, which has been extended and transformed into the BPMN4SIM metamodel, which is in turn used to create BPMN4SIM models, as detailed in the next section. Both the BPMN metamodel and the BPMN4SIM metamodel are instances of the MOF (Meta Object Facility) model, i.e. the key MDA standard that provides an abstract language and a framework for specifying, constructing and managing technology-neutral metamodels (OMG, 2016). The proposed approach is 'lightweight', because it maintains BPMN-compliance by simply extending the BPMN metamodel, as required by the BPMN extension mechanism (OMG, 2011, pp. 57-61), and can be easily automated, in order to effectively support its adoption.

4. BPMN4SIM

In this section, we discuss our BPMN extension design to meet the requirements of modelling complex healthcare patient-flow systems. We aim to make use of concepts and constructs from standard BPMN 2.0 as much as possible in order to minimise the number of extension elements. Hence, we analyse the features mentioned earlier to check whether any of them can be implemented using existing BPMN concepts; Braun *et al* (2014) refer to this analysis as the equivalence check.

4.1. Flows and Hierarchies

Hospitals sometimes use pull flows, for example by pulling patients from a waiting list when capacity has been released (for example, patients may be asked if they are willing to be contacted at short notice if an operating theatre slot has become available). For such reasons the stakeholders in our study expressed the need to differentiate push from pull flows in a process map. Figure 4 shows how existing BPMN concepts can be used to represent a pull flow. In this example, we use a conditional event to control the movement of an entity from the Inpatient activity to Administration for discharge. Hence, an existing concept such as a conditional event can be used to implement a pull flow. In a first iteration, we attempted to address this by introducing a new flow representation (denoted by a new arrow style), so that when an activity (e.g. Administration for discharge) pulls a token from another activity (e.g. Inpatient) we used this new arrow style to connect the two activities. Although this design addressed the gap, it is simpler to reuse existing elements from the BPMN standard (i.e. applying Braun *et al*'s equivalence check principle).

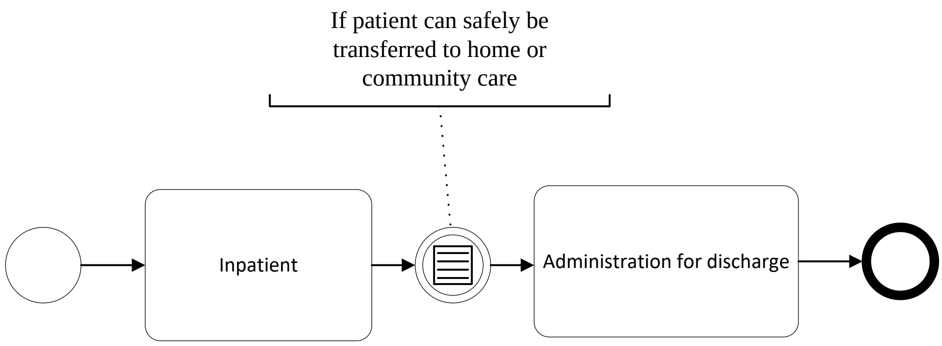


Figure 4: Representing a pull flow using BPMN

Stakeholders in our study noted that sometimes there is the need to differentiate physical and information flows. Although BPMN is not designed to support a data flow diagram, it has a concept called *ItemDefinition* that has a property called *itemKind*. This property is used to differentiate physical and information items. BPMN also provides different visualisation for information flow (using data association) and physical flow (using sequence flow) as shown in Figure 5. Hence, this requirement can be implemented using existing BPMN concepts. Likewise, BPMN supports the concept, and provides a notation for, collapsible sub-processes. This concept can be used to support a hierarchy of nested process maps. Hence, there is not any need to introduce a new concept for this.

4.2. Interaction between data and flow

Healthcare processes generate a lot of data. Some data guide healthcare professionals to make better decisions for their patients. Data generated by activities such as triage or X-ray will determine the flow of patients later along a clinical pathway. Hence, there is a need to capture the interaction between data and flows of entities in a healthcare process in BPMN4SIM. In should be noted that for a simulation model, modellers also need to specify other types of data, such as input data (e.g. service-time distributions) and output data (for results analysis, experimentation and scenario analysis). These are better implemented in a separate artefact such as a specification language or a standard (e.g. BPSim) because it is not part of model structure (see Figure 1).

BPMN concepts can be used to implement how data is generated and accessed by an activity (i.e. an extension is not needed). BPMN has a concept called *DataObject* that provides a visualisation for data that exist within a process. If the data persist beyond the process, a BPMN concept called *DataStore* can be used. To represent how an activity interacts with data, BPMN uses data association. Figure 5 shows an example of a simplified process (clinical pathway) in which a patient with a fracture arrives at A&E and receives an X-ray. The X-ray result (e.g. location and severity of fracture) is added to the patient's health record (e.g. paper or electronic record). Based on the severity, the patient may need treatment at the fracture clinic (in which the patient's health record will be used to guide the treatment) or a nurse will fit a polysling. The patient will leave after receiving the appropriate treatment.

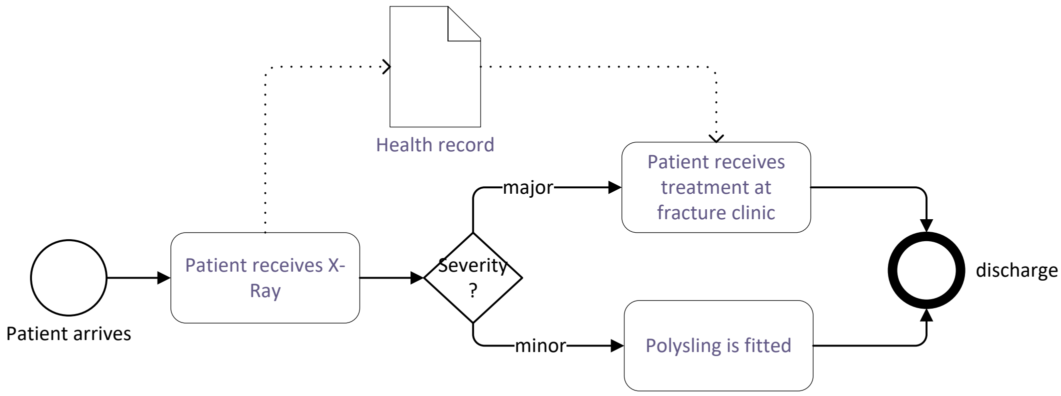


Figure 5: Interaction between activities and data in BPMN

Although BPMN concepts are able to represent the interaction between activities and data, it was clear from our discussion with stakeholders that they desire an explicit and structured representation of data. This data representation would prompt stakeholders, facilitators and modellers to look for particular types of data when considering particular types of activity. The importance of data representation in healthcare is also highlighted in Braun *et al* (2014). For this reason, a simple extension to the existing BPMN *DataObject* can be justified. The meta-model is shown in Figure 6.

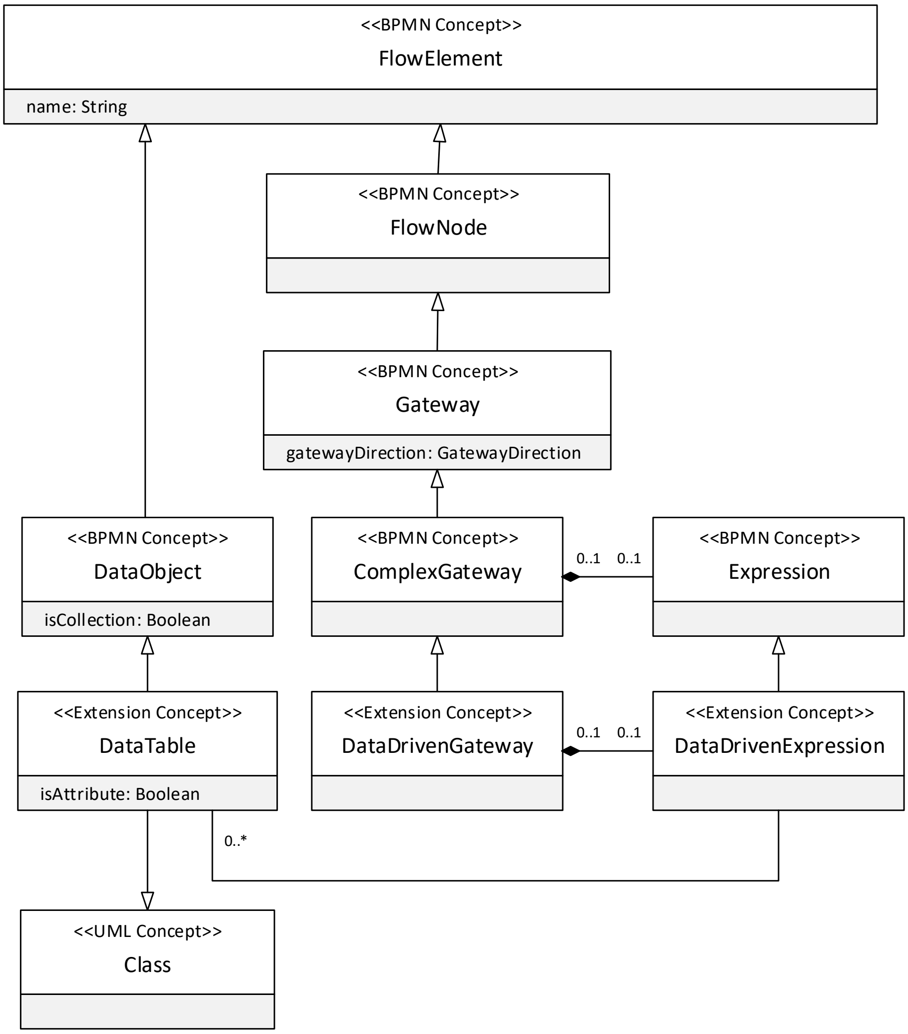


Figure 6: BPMN4SIM meta-model – interaction between data and flow

Braun *et al* (2014) introduce a concept called *Document* which is also applicable to our purpose. Hence, we incorporate the relevant part of their design into BPMN4SIM. For our purpose, we rename their concept (i.e. *Document*) to *DataTable*. This concept extends *DataObject* to represent a table. Instead of specifying a fixed structure for the table, we use the UML class to define the structure of the table, which provides more flexibility. With this extension, we can represent an activity that generates and stores data in a standard database table which can be read by another activity. The *DataTable* can be used to represent attributes attached to an entity or a variable accessible by a process in which the *DataTable* is defined.

Another interaction between data and flow that is needed in healthcare is a data-driven gateway that controls the flow of entities based on the available data. The importance of this data-driven gateway in healthcare is also highlighted in Braun *et al* (2014). BPMN does not have this gateway type. Hence, we need to add this in BPMN4SIM. Although not proposed specifically with simulation modelling in mind, Braun *et al* (2014) has suggested a useful extension design for the data-driven gateway that is suitable for simulation modelling (they call it an *Evidence-based Gateway*). This extension (*DataDrivenGateway* in Figure 6) allows us to represent a data-driven decision point which controls the flow in the business process. The gateway will use an expression to access data in a *DataTable* (this is enabled by *DataDrivenExpression*, which is an extension of BPMN *Expression*).

4.3. Queues

There was a strong view from stakeholders that queues need to be represented in a process map for healthcare. This is because some of the key indicators that are used to measure and understand process performance are related to queues (e.g. the four-hour A&E target). The importance of queues has also been highlighted in previous research (Waller *et al*, 2006). We identify two distinct and important types of queues in healthcare: queues in which entities are holding resources while waiting (e.g. A&E patients in a treatment cubicle waiting to be transferred to a hospital ward and so blocking treatment of a further patient) and queues in which entities are not holding any resource while waiting (e.g. patients with ‘minor’ injuries waiting in a large seated reception area in A&E to be seen by a nurse or a doctor).

BPMN does not have a concept that represents queues. Hence, we need to implement queues as a BPMN extension concept. First, we need to understand why a queue is not represented in BPMN. A BPMN process contains a sequence of flow nodes such as activities, events and gateways. Semantically, a process is instantiated when one of its start events occurs. Each start event creates a token that moves around the process. An activity is instantiated when there is a token moving into it. (There are a few exceptions such as an activity without any incoming flow and compensation activities.) This semantic does not support a queuing behaviour, in which tokens from multiple process instances want to move into a shared activity with a limited capacity. Hence, to allow queuing behaviour, we need to introduce a new Task called *SharedTask* (see Figure 7).

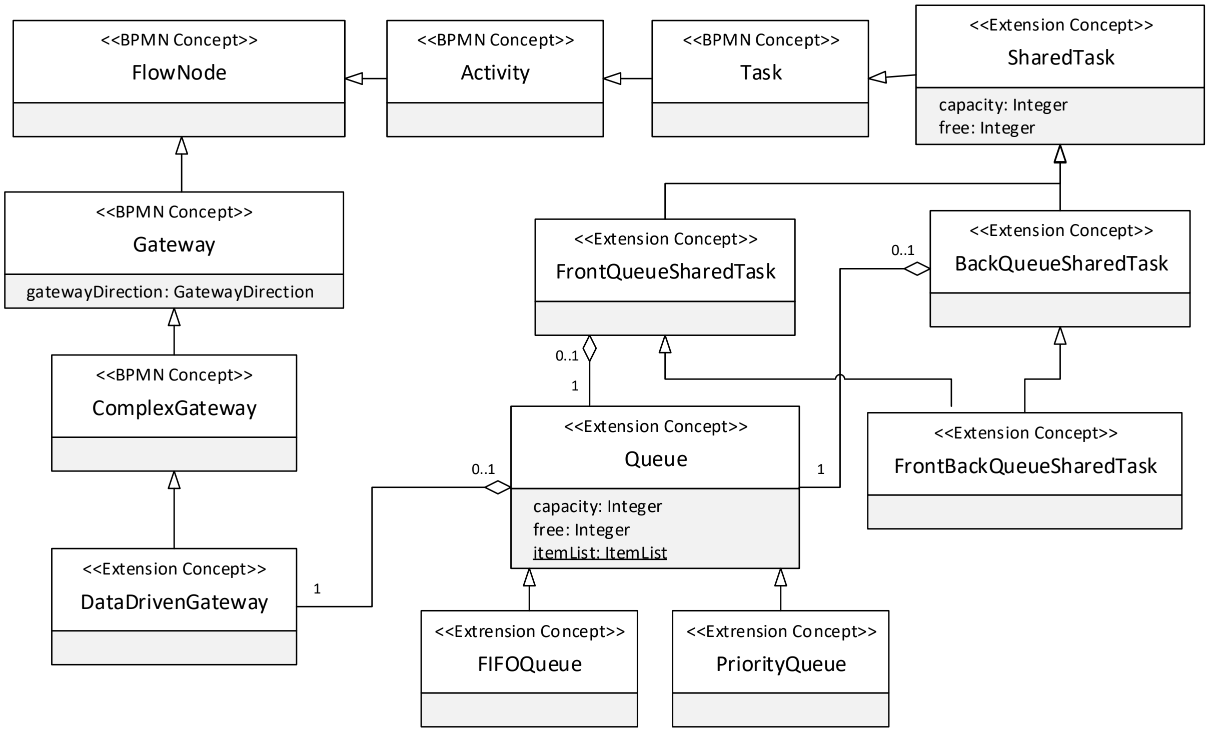


Figure 7: BPMN4SIM meta-model – Queues

Semantically, when a token in a process instance moves into a shared task, it will instantiate the shared task if the shared task has not already been instantiated by another process instance. The token will move into the shared task and reduce its free capacity. When the last token in the shared task leaves, the instance of the shared task will be destroyed. This mechanism is represented using a class with two static properties called *capacity* (the maximum number of tokens that can be inside the shared task) and *free* (the number of available spaces for tokens) in the meta-model. The process map can use these properties to trigger events such as signal or escalation, which can be used to block the upstream activities.

A shared task can have a queue associated with it. This is implemented as a new extension concept called *Queue* in the meta-model. A queue has a static property called *itemList*. This list provides a record of process instances (or in simulation terms, entities) queuing for a shared task. A queue contains a data-driven gateway that we have introduced earlier. A data-driven gateway can be used to determine whether a token can leave or must remain in the queue. A queue can be placed in front of a shared task to represent a standard queue where entities do not hold any resource while waiting. In other words, process instances in the queue will wait until the shared task can accept them (i.e. until free > 0). A shared task with this type of queue is implemented as *FrontQueueSharedTask* which is an extension to *SharedTask*. A queue can also be placed at the end of a shared task to represent a queue in which entities are holding resources while waiting in the queue. In other words, process instances that cannot leave the shared task (for example being blocked by the downstream activity being full) will be placed in this queue and keep consuming the resources in the shared task (i.e. the number of free spaces is not reduced). This shared task is implemented as *BackQueueSharedTask,* which is an extension to *SharedTask*. A shared task can also have both types of queue. This shared task is implemented as *FrontBackQueueSharedTask,* which is an extension to *SharedTask*. The queue can also be extended to represent queues with specific behaviours such as first-in-first-out queue (*FIFOQueue*) and priority queue (*PriorityQueue*).

4.4. Graphical symbols

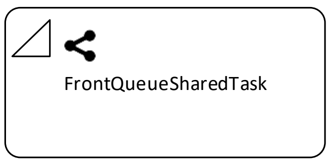
Table 2 provides the summary of concepts (BPMN and our extension) that we propose for BPMN4SIM. All extensions are extended from standard BPMN elements (*Task*, *Gateway* and *DataObject*), so we can use the standard BPMN shape containers. We introduce two new icons for BPMN *Task* to represent a combination of shared task (using the shared symbol commonly used in social network or mobile applications) and queue (using right-angled triangles) and a new icon for BPMN *Gateway* and *DataObject* to represent data (a table). The proposed graphical notations are shown in Table 3.

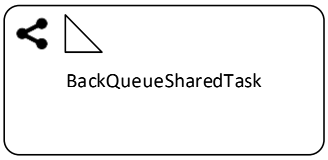
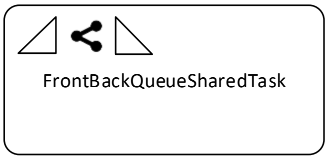
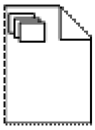
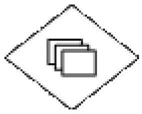
Table 2: Equivalence check of the required BPMN4SIM features

Feature	**Requirement**	**Equivalence check**	**Concept**
Hierarchies	Processes and collapsible sub-processes	**Equivalence**: Processes and collapsible sub-processes	BPMN concept
Flow	Ability to differentiate physical and information flow; Ability to differentiate push and pull flows	**Equivalence**: flow of information can be done using directed data association **Equivalence**: BPMN assumes push flow only; however, a pull flow can be represented by using a conditional event	BPMN concept BPMN concept

Queues	'Front', 'back', ability to define a finite capacity	**No equivalence**: BPMN does not define the concept of queue	Extension concepts (add *SharedTask*, *Queue*, *FrontQueueSharedTask*, *BackQueueSharedTask*, *FrontBackQueueSharedTask*)
Interaction between information and flow	Representation of how an activity affects data; representation how data affects flow of entities	**Equivalence**: Data object provides a visualisation for information that exists within a process. If the information persists beyond the process, Data Store can be used. Data association can be used to represent how an activity affects information **No equivalence**: There is no concept of a gateway that controls the flow of entities based on a set of data (i.e. data-driven gateway)	BPMN Concept Extension concepts (add *DataDrivenGateway*, *DataTable*)

Table 3: Graphical notations

Concept	**Notation**
SharedTask	
FrontQueueSharedTask	

BackQueueSharedTask	
FrontBackQueueSharedTask	
DataTable	
DataDrivenGateway	

4.5. BPMN4SIM and BPSim

The BPMN specification defines three conformance sub-classes: descriptive, analytic and common executable (OMG, 2011, p. 2). The main users of a descriptive conformance sub-class are business analysts who are comfortable with high-level modelling using a flowcharting tool. The analytic conformance sub-class contains all constructs in the descriptive conformance sub-class plus a number of constructs from the BPMN process modelling conformance class (analytic conformance is a superset of descriptive conformance). The common executable conformance sub-class requires more-detailed constructs that are needed to make the model executable (common executable conformance is a superset of analytic conformance). BPMN4SIM in this paper extends the analytic conformance sub-class (see Figure 8) and makes this easily available at the descriptive level through the new notation in Table 3. This approach enables the link between the queues in a model to the corresponding real world queues (and the collection of measurements from both the simulation run and the execution in the real world).

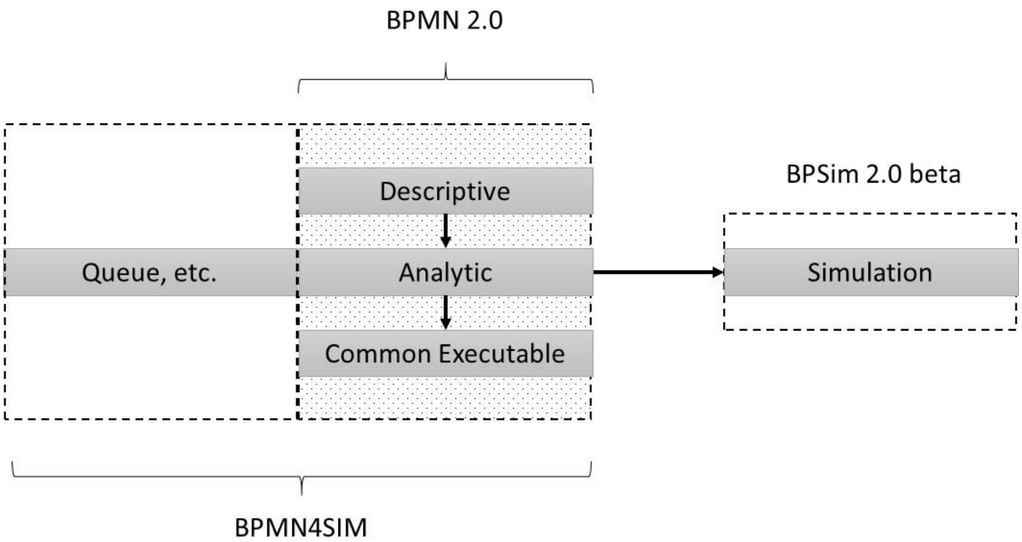


Figure 8: BPMN4SIM and BPSim

A number of organisations have worked together to develop a standard called BPSim that reads a model written in BPMN and adds the extra information that is needed to simulate the BPMN model. The main advantage in this approach is that it leaves BPMN as it is. The main disadvantage is that the capability of BPSim is limited by the limitations of BPMN (e.g. the lack of explicit queues). BPMN4SIM will enable us to extend the capability of BPSim to simulate both front queues (entities do not consume resource) and back queues (entities continue to consume the resource required by the activity), to use attributes and to use data-driven decision points. The latest version of BPSim (2.0 beta) was released in July 2016. However, it has not addressed the issues related to the lack of support for queues, attributes and data-driven decision points.

5. Evaluation using an inpatient pathway case from an NHS Trust

In this section, we demonstrate how the proposed BPMN4SIM could be used to map the elderly emergency care pathway in a UK hospital. The stakeholders in this case study were interested in exploring the impact of varying the provision in order to reduce the patient length of stay and so to increase capacity. This case is one of the two NHS cases that showed us that BPMN could not (yet) support fully-facilitated simulation modelling of complex flows. Hence, this case can be used to evaluate whether the proposed BPMN4SIM is capable of representing the real process. Furthermore, this case study requires all the new elements proposed in this paper. Hence, this case study is a good case to evaluate BPMN4SIM.

Elderly emergency patients arrive via the A&E or directly to the Emergency Assessment Unit (EAU). From A&E, patients are either discharged or sent to the EAU. From the EAU, patients are either discharged or transferred to an elderly care inpatient ward (or to other specialisms). Because the pathway is for elderly patients, many of the patients are discharged to residential care facilities. When the appropriate residential care facility is not able to receive a patient, they remain waiting in the acute hospital (in an inpatient bed) even though they are considered to be medically fit for discharge.

Figure 9 shows the BPMN4SIM model of the pathway. There are two start events, representing two patient arrival points. Those who arrive at A&E will queue for registration (the patients do not hold any resource while waiting). The registration process is represented as a shared task as it is shared among patients. Once a patient has been registered, the patient may need to wait for treatment depending on A&E cubicle availability. Some patients may decide to leave the hospital for various reasons such as waiting for too long or being transferred to another hospital. This behaviour is represented in BPMN as a non-interrupting boundary event attached to the registration task.

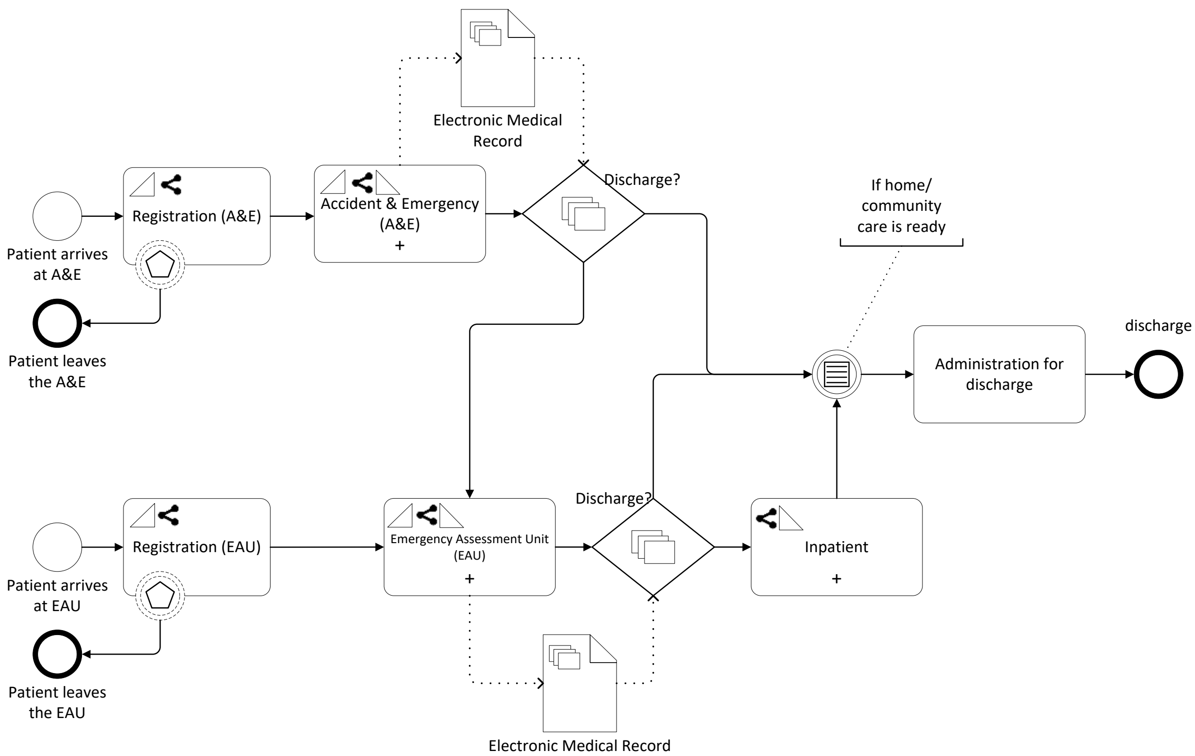


Figure 9: Pathways of elderly emergency care patients, represented with BPMN4SIM

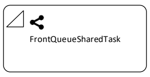
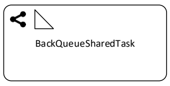
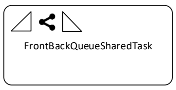
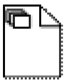
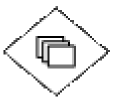
A&E is shown as a sub-model (denoted by the plus sign) which can be expanded to see a more-detailed process inside it. A&E is represented as a shared task with a back queue. The back queue is needed because patients who are ready for discharge from A&E may have to wait in cubicles (hence, holding resources), for example, when a residential care facility that is supposed to receive the patients is not ready, or when the EAU does not have any free capacity for patients who need to be transferred there. The decision whether a patient is discharged from A&E or transferred to EAU depends on the patient's health status, which is updated by A&E. Hence, this decision point is represented as a data-driven gateway associated with a data table (e.g. information held in the patient's record).

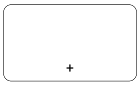
The process from the arrivals of patients referred by a GP (General Practitioner, i.e. local family doctor) until they receive treatment in the EAU is similar to that at A&E. At the end of the EAU activity, a patient is either discharged or transferred to an Inpatient ward (bed) depending on their health status. Hence, the decision point is represented as a data-driven gateway. Patients who no longer need EAU services may still have to wait in the EAU for the same reasons as for A&E.

Finally, the Inpatient ward is represented as a shared task with a back queue. A patient who is ready to be discharged from the Inpatient ward may still have to occupy a bed if the patient cannot be sent home or the required residential care facility is not ready. This "pull" mechanism is represented using a conditional event that separates Inpatient and Administration for discharge. When home or residential care is ready, the condition will be met and a patient will go through the Administration for discharge. A&E, EAU and Inpatients shown in the model represent a high-level view of the pathway (denoted by the plus signs). Hence, they can be represented as possessing sub-processes (or sub-models). Each sub-process could be expanded to see the more detailed lower-level activities. This is a facility in BPMN software.

This case study shows that the extensions proposed in this paper can be used to represent a relatively complex healthcare process. Table 4 shows that the features of the case can now be represented in BPMN4SIM. Figure 9 also shows that the graphical notation is not so cumbersome as to make the diagram untidy and confusing, whilst the diagram is expressive enough to explicitly represent the queues and the interactions between information and flows. There is one BPMN4SIM concept that is not directly used in the case study. This concept, *SharedTask*, is introduced as a parent class of *FrontQueueTask* and *BackQueueTask*. This is good design practice because the parent class encapsulates common elements that can be extended by the child classes. Hence, the parent class is reusable and extensible for future need.

Table 4: BPMN4SIM used in ACM

Concept	Registration (A&E)	A&E	Registration (EAU)	EAU	Inpatient	Patient Medical Record	Discharge	If A&E has free capacity	If EAU has free capacity
SharedTask 									
FrontQueueSharedTask 	X		X						
BackQueueSharedTask 					X				
FrontBackQueueSharedTask 		X		X					
DataTable 						X			
DataDrivenGateway 							X		
Pull flow 								X	X

Hierarchy 		X		X	X				

6. Conclusions and future work

This paper set out to answer two research questions: “how can BPMN be extended to represent queues and data?” and “can the extended BPMN (BPMN4SIM) represent a more-complex real-world business process in healthcare?” To answer the first question, we demonstrate how this can be done through our proposed BPMN extension (BPMN4SIM). This is the key contribution of this paper. The main advantage of implementing the concept of queues explicitly in BPMN is twofold. Firstly, during process mapping, it will make a business analyst more aware of queues in a business process. This is important because some business process KPIs are related to queues such as queue size and waiting time. Secondly, in a project proceeding to DES modelling this would facilitate the model coding stage by enabling a seamless transformation between a process model and a simulation model in which modellers can specify different types of queues at the right place in the process model. The need to represent data arises from situations in which the paths taken by entities (or tokens) are dependent on data attached to entities (i.e. attributes or labels) . An explicit representation of these data will also help users to locate relevant data more easily from a process map. The extension consists of a meta-model, which formally describes BPMN4SIM, and the suggested graphical notations of the new concepts.

Both these features are often important in representing healthcare systems: queues are important components of KPIs and improvement projects involving patient waiting times, and data representing patient characteristics and outcomes from the processes they flow through can determine their subsequent routing along patient pathways. This is why the second research question conceptually tests the ability of our extension (BPMN4SIM) to represent such a complex real-world business process in healthcare using a case study of an emergency inpatient pathway in an NHS acute hospital trust. We demonstrate that our BPMN4SIM extension can indeed represent these important features, BPMN4SIM has been designed using the Model Driven Architecture approach to enable a smoother conversion from a BPMN diagram to a simulation model. The ability to perform this conversion would enable us to carry out fully-facilitated simulation modelling involving key stakeholders at all simulation modelling stages. Since BPMN is a standard, a model specified using BPMN4SIM, which is BPMN-compliant, can, in theory, be run with any BPMN4SIM-compliant simulation software. It is our intention to develop a software tool that enables us to use BPMN4SIM. This software will allow us to collect empirical data to test the suitability of BPMN4SIM for fully-facilitated simulation modelling projects. Although the proposed extension has been designed with healthcare applications in mind, BPMN is a powerful and increasingly widely-used general-purpose process-modelling technique, and features like 'back' queues (in which entities continue to consume the resource required by an activity) and data-driven flow routing can be found in other domains such as manufacturing and service providers. Hence, the idea behind BPMN4SIM should also be applicable in those domains.

References

Augusto V and Xie X (2013). A modeling and simulation framework for health care systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. **44**(1): 30-46.

Bisogno S, Calabrese A, Gastaldi M and Levialdi Ghiron N (2016a). Combining modelling and simulation approaches: How to measure performance of business processes. *Business Process Management Journal*. **22**(1): 56-74.

Bisogno S, Proudlove NC, Calabrese A, Levialdi Ghiron N and Onggo BSS (2016b). Towards fully-facilitated DES modelling: A successful project. *Management Science Working Paper Series*. Lancaster University, Department of Management Science. Available from http://www.research.lancs.ac.uk/portal/en/publications/-(24154c12-cac0-4b28-97fe-836e7da4c147).html. { accessed 22/06/2016 }

Bizagi (2016). Bizagi Modeler. [online] available from http://www.bizagi.com/en/products/bpm-suite/modeler {accessed: 3th November 2016}

BOC Group (2016). ADONIS:CE Free BPM Tool. [online] available from http://en.adonis-community.com/ {accessed: 3th November 2016}

Bocciarelli P and D'Ambrogio A (2011). A BPMN extension for modeling non functional properties of business processes. *Proceedings of the 2011 Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, Society for Computer Simulation International, Boston, MA, USA, April 03-07, 2011, pp 160–168.

Brailsford S (2005). Overcoming the barriers to implementation of operations research simulation models in healthcare. *Clinical and Investigative Medicine*. **28**(6): 312-315.

Brailsford S, Harper PR, Patel B and Pitt M (2009a). An analysis of the academic literature on simulation and modelling in health care. *Journal of Simulation*. **3**: 130-140.

Brailsford S, Bolt T, Connell C, Klein JH and Patel B (2009b). Stakeholder engagement in health care simulation. *2009 Winter Simulation Conference*. Austin, TX.

Braun R, Schlieter H, Burwitz M and Esswein W (2014). BPMN4CP: Design and implementation of a BPMN extension for clinical pathways. *IEEE International Conference on Bioinformatics and Biomedicine,* pp 9–16.

Braun R, and Esswein W (2014). Classification of domain-specific BPMN extensions. In Frank U, Pastor O, Loucopoulos P, Petrounias I (eds.) *Proceedings of the 7th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling2014*, Manchester, UK, November 12–13, 2014, pp 42-57.

Fanti MP, Mangini AM, Dotoli M and Ukovich W (2012). A three-level strategy for the design and performance evaluation of hospital departments. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. **43**(4):742-756.

Franco LA and Montibeller G (2010). Facilitated modelling in operational research. *European Journal of Operational Research*. **205**(3): 489-500.

Friedenstab J, Janiesch C, Matzner M and Muller O (2012). Extending BPMN for business activity monitoring. *45th Hawaii International Conference on System Science (HICSS)*, pp. 4158–4167. IEEE

Harmon P (2016). The state of business process management 2016. http://www.bptrends.com/bpt/wp-content/uploads/2015-BPT-Survey-Report.pdf

Jahangirian M, Taylor SJE, Eatock J, Stergioulas LK and Taylor PM (2015). Causal study of low stakeholder engagement in healthcare simulation projects. *Journal of the Operational Research Society*, **66**(3): 369-379.

Kotiadis K, Tako AA and Vasilakis C (2014). A participative and facilitative conceptual modelling framework for discrete event simulation studies in healthcare. *Journal of the Operational Research Society*, **65**(2): 197-213.

Lanner (2016). L-SIM. [online] available from http://www.lanner.com/en/l-sim.cfm {accessed: 3th November 2016}

Lodhi A, Kuppen V and Saake G (2011). An extension of BPMN meta-model for evaluation of business processes. *Scientific Journal of Riga Technical University. Computer Sciences* **43**(1): 27–34.

Melao N and Pidd M (2000). A conceptual framework for understanding business processes and business process modelling. *Information System Journal*, **10**:105–129.

OMG (2011). *Business Process Modeling and Notation (BPMN) version 2.0*. [online] available from www.omg.org/spec/BPMN/2.0/ {accessed: 3th November 2016}

OMG (2014). Model Driven Architecture Guide (MDA), rev. 2.0. [online] available from http://www.omg.org/mda/specs.htm {accessed: 3th November 2016}

OMG (2015) Unified Modeling Language 2.5. [online] available from http://www.omg.org/spec/UML/ {accessed: 3th November 2016}

OMG (2016). Meta Object Facility, version 2.5.1. [online] available from http://www.omg.org/spec/MOF/2.5.1/ {accessed: 3th November 2016}

Onggo BSS and Karpat O (2011). Agent-based conceptual model representation using BPMN. In Jain S, Creasey RR, Himmelspach J, White KP and Fu M (eds) *Proceedings of the 2011 Winter Simulation Conference*, 11-14 December 2011, Phoenix, Arizona (WSC11), IEEE Computer Society Press, Los Alamitos, California, pp 671-682.

Onggo BSS (2013). Agent-based simulation model representation using BPMN. In Fonseca P (ed) *Formal Languages for Computer Simulation: Transdisciplinary Models and Applications*. Information Science Reference, Hershey, PA, pp 378-399.

Pitt M, Monks T, Crowe S and Vasilakis C (2016). Systems modelling and simulation in health service design, delivery and decision making. *BMJ Quality & Safety*. **25**(1): 38-45.

Proudlove NC, Black S and Fletcher A (2007). OR and the challenge to improve the NHS: Modelling for insight and improvement in in-patient flows. *Journal of the Operational Research Society*. **58**(2): 145-158.
Reynolds M, Vasilakis C, McLeod M, Barber N, Mounsey A, Newton S, Jacklin A and Franklin BD (2011). Using discrete event simulation to design a more efficient hospital pharmacy for outpatients. *Health Care Management Science*. **14**: 223–236.
Robinson S (2008). Conceptual Modelling for Simulation Part I: Definition and Requirements. *Journal of the Operational Research Society*. **59**(3): 278-290.
Robinson S, Radnor ZJ, Burgess N and Worthington C (2012). SimLean: Utilising simulation in the implementation of lean in healthcare. *European Journal of Operational Research*. **219**(1): 188-197.
Robinson S, Worthington C, Burgess N and Radnor ZJ (2014). Facilitated modelling with discrete-event simulation: Reality or myth? *European Journal of Operational Research*. **234**(1): 231-240.
Roux O, Combes C and Duvivier D (2006). A modeling methodology dedicated to simulation and based on generic metamodels using MDA-UML: An application to a chronic renal dialysis unit. *Proceedings of the 2006 International Conference on Service Systems and Service Management*, pp 692–697.
Salles G, Fantinato M, Nishijima M and de Albuquerque JP (2013). A contribution to organizational and operational strategic alignment: Incorporating business level agreements into business process modeling. *IEEE 10th International Conference on Services Computing*, pp 17–24.
Schmidt DC (2006). Model-Driven Engineering. *IEEE Computer*. **39**(2): 25–31.
Signavio (2016). Signavio Process Editor. [online] available from http://www.signavio.com/ {accessed: 3th November 2016}
Simul8 Corporation (2016) Simulation with Business Process Model and Notation. [online] available from http://www.simul8.com/bpm/bpmn.htm {accessed: 3th November 2016}
Stroppi LJR, Chiotti O and Villarreal PD (2011a). A BPMN 2.0 extension to define the resource perspective of business process models. *XIV Congreso Iberoamericano en Software Engineering*.
Stroppi LJR, Chiotti O and Villarreal PD (2011b). Extending BPMN 2.0: Method and tool support. *Business Process Model and Notation*, Springer-Verlag, Berlin Heidelberg, pp 59-73.
Tako AA and Kotiadis K (2015). PartiSim: A multi-methodology framework to support facilitated simulation modelling in healthcare. *European Journal of Operational Research*. **244**(2): 555-564.
Taylor SJE, Eldabi T, Riley G, Paul RJ and Pidd M (2009). Simulation modelling is 50! Do we need a reality check? *Journal of the Operational Research Society*. **60**(S1): S69-S82.
van der Aalst WMP, Nakatumba J, Rozinat A, and Russell N (2010). Business process simulation. In *Handbook on Business Process Management 1, International Handbooks on Information Systems*, Springer-Verlag, Berlin Heidelberg, pp 313–338.
Vasilakis C, Lecznarowicz D and Lee C (2009). Developing model requirements for patient flow simulation studies using the Unified Modelling Language (UML). *Journal of Simulation*. **3**(3): 141-149.
Vasilecas O, Smaizys A and Rima A (2013). Business process modelling and simulation: Hybrid method for concurrency aspect modelling. *Baltic Journal of Modern Computing*. **1**(3-4): 228–243.

Wagner G, Nicolae O and Werner J (2009). Extending discrete event simulation by adding an activity concept for business process modeling and simulation. *Proceedings of the 2009 Winter Simulation Conference*, pp 2951–2962.
Wagner G (2014). Tutorial: Information and process modeling for simulation. *Proceedings of the 2014 Winter Simulation Conference*, pp 103–117.
Waller A, Clarke M and Enstone L (2006). L-SIM: Simulating BPMN diagrams with a purpose built engine. *Proceedings of the 2006 Winter Simulation Conference*, 591–597.
WfMC (2013). BPSim: Business Process Simulation Specification 1.0. [online] available from http://bpsim.org/specifications/1.0/WFMC-BPSWG-2012-01.pdf {accessed: 3th November 2016}
WfMC (2016). BPSim: Business Process Simulation Specification 2.0. [online] available from http://bpsim.org/specifications/2.0.beta/WFMC-BPSWG-2012-01.pdf {accessed: 3th November 2016}
Zor S, Leymann F and Schumm D (2011). A proposal of BPMN extensions for the manufacturing domain. *Proceedings of 44th CIRP International Conference on Manufacturing Systems*.