What-If Scenarios for the BedreFlyt Digital Twin

Åsmund A. A. Kløvstad [1], Paul Kobialka [1], Riccardo Sieve [1], Andrea Pferscher [1], Laura Slaughter [2], Silvia Lizeth Tapia Tarifa [1], and Einar Broch Johnsen [1] ✉

[1] Dept. of Informatics, University of Oslo, Norway
`{aaklovst,paulkob,riccasi,andreapf,sltarifa,einarj}@uio.no`
[2] dScience Center, University of Oslo, Norway
`l.a.slaughter@dscience.uio.no`

Abstract. Digital twin technology is emerging as a valuable tool for both short-term decision-making and long-term strategic planning across domains such as process industry, energy, space, transport, and healthcare. This paper reports on ongoing work in designing a digital twin to enhance resource planning in hospitals, e.g., for in-patient needs. Our focus here is on a novel technique to express what-if scenarios in digital twins to improve strategic planning processes, spanning, e.g., average-case and worst-case resource needs, expected patient treatments, and ranging over variations in available resources such as bed bays in the hospital ward. Due to the modularity of our digital twin architecture, different what-if scenarios can be explored simply by configuring the digital twin's orchestrator, which triggers a formal methods analysis pipeline that combines executable formal models for simulation, optimization over constraints and a knowledge base that formalizes domain knowledge. We illustrate what-if scenario analysis in our digital twin architecture by considering the problem of bed bay allocation in a hospital ward.

1 Introduction

Predicting the future is easy: most likely, tomorrow will be exactly like today. However, sometimes we may wonder if events could have played out differently. Although people tend to blame destiny when things go wrong, Casanova claims that for the numerous bad turns in his life the blame was his alone (and that, if he could live again, he would do exactly the same) [8]. Thus, he lived his life according to a locally optimal strategy. In contrast to *predictive* analysis [22], which is concerned with what we expect to happen in the near future, *prescriptive* analysis [48] is concerned with so-called *what-if* scenarios, i.e., exploring and comparing the outcomes of strategies that decide between alternative choices. Humans have proven quite good at such prescriptive analysis; i.e., we routinely reason about and compare possible scenarios to derive appropriate strategies. However, many of us struggle with our strategies when the problems get sufficiently complex. Here, computer-aided analyses can help to overcome human limitations and help us find and evaluate strategies. This problem of finding and exploring strategies touches on several aspects of Christel Baier's inspirational work, from the analysis of Markov decision processes (e.g., [4, 5]) to learning strategies (e.g., [3, 53]).[3]

Digital Twins (DTs) are virtual information constructs that capture the structure, context, and behavior of the "real" system they are twinning, are dynamically updated with data from the twinned system, have predictive capability, and inform decisions that realize value, according to a recent definition by the National Academies of Science, Engineering and Medicine (NASEM) [41]. Historically, DTs have been developed in engineering disciplines, where increasingly sophisticated "virtual replicas" have been used to simulate the behavior of a cyber-physical system and a closed feedback loop feeds control decisions back to the twinned system (e.g., [17]), however, the recent definition by NASEM is broader. Today, DTs can be found in many domains outside of cyber-physical systems, such as healthcare [51], manufacturing [6], and transportation [9].

In this paper, we are concerned with the use of DTs for the model-driven exploration of so-called what-if scenarios, moving from the predictive analysis of near-future events to the prescriptive analysis of hypothetical scenarios. We believe that DTs have a strong potential for applications in both short-term decision-making and long-term strategic planning in various domains. Seen from a formal methods perspective, DTs go beyond standard model-driven techniques by supporting the dynamic update of the model, leveraging a live data feed from the twinned system (known as the "physical twin"). Thus, the DT becomes an infrastructure for data-driven formal methods (e.g., [34]), in which the live data from the twinned system is used to configure a formal model. Similarly, the what-if scenario to be explored need not be fixed in advance, but may be requested on-the-fly by the user of the DT. This dynamically requested scenario may also determine aspects of the model's configuration, as well as the properties to be analyzed. In short, we may think of DT infrastructure as a self-adaptive system [52] for advanced model management, generating the different models and determining the analyses to be performed over these models.

Our focus here is on prescriptive analysis in BedreFlyt (/ˈbeːdrə flyːt/, Norwegian for "Better Flow") [47]. BedreFlyt is a DT for resource management in healthcare. The proper handling of resources at a hospital is crucial to efficient operations [40], e.g., to determine how trained staff, bed availability in the hospital ward, and necessary rooms and equipment match the needs of different activities at the hospital, such as the treatment of patients. The dynamic allocation of these resources is necessary to efficiently manage the workflow and adjust it to avoid bottlenecks in operations, and to improve the prioritization and utilization of available resources [56]. Simulations have been successfully used to improve resource allocation in a hospital [46]. By connecting simulation models to live data, the DT can ensure that the simulations more accurately reflect the actual resource allocation problems of the hospital. This way, a DT becomes a meeting point between static planning and dynamic optimization, allowing a bet-

[3] In particular, the last author of this paper had the pleasure of collaborating with Christel in the EU project *CREDO* (including an unforgettable incident in Bonn, the further details of which shall not be unveiled).

ter and more dynamic management of the workflow and its associated resources. By configuring the models to explore different scenarios, the DT further supports the comparison of resource management strategies under different assumptions concerning the resources as well as the incoming patients to the hospital.

The main contributions of this paper are (1) a technique to express what-if scenarios in DTs for prescriptive analysis that is parametric in risk tolerance, and (2) a simulation interface for such predictive analyses for human-in-the-loop decision making. We explore *worst-case scenarios* for the bed bay allocation problem at a hospital ward, as well as *sample* statistical information when assigning treatments to incoming patients, by enriching the domain knowledge of the BedreFlyt DT [47] with statistical distributions for patient treatments. We further seamlessly combine such sampling with worst-case scenarios to capture *risk tolerance* in long-term planning. The result is a wide variety of strategies for the long-term planning of bed allocation for patients, that minimizes the number of reallocations. We evaluate the design on a realistic patient diagnosis stream, based on a historical dataset for a hospital ward at the Norwegian hospital *Rikshospitalet*.[4]

2 The BedreFlyt Digital Twin

The BedreFlyt DT [47] aims to aid hospital staff with resource planning in a ward by solving the problem of room allocations for an incoming stream of patients. The complexity of this problem arises from the unknown, new patients arriving at every time step, creating a dynamic scheduling problem. Patients arrive at the hospital with a diagnosis and are assigned a treatment. Then, depending on the needs of the treatment, they will have different requirements in terms of monitoring and time over their stay. Additionally, the hospital wishes to separate patients by gender and to keep contagious patients isolated.

The DT takes a stream of patients with their diagnoses, genders, and contagiousness status as input and then outputs bed bay allocations for all patients so that all the requirements are met. The following describes the architecture and components of the BedreFlyt DT and introduces a simple running example. We start out with a high-level view and then discuss each component individually.

2.1 BedreFlyt DT Architecture

The BedreFlyt DT, depicted in Fig. 1, integrates several formal techniques into a tool chain for prescriptive analysis. Our DT combines formalized domain knowledge about patient treatments and hospital wards, an actor-based executable formal model to explore strategies for streams of incoming patients with associated treatments, and an optimizer to perform the actual bed bay allocation. Patient flow is expressed in the *abstract behavioral specification* language ABS [28, 29], which specifies object-oriented control flow and flexible communication between actors with a timed semantics. The resulting model is compiled into Java and

[4] https://www.oslo-universitetssykehus.no/steder/rikshospitalet/

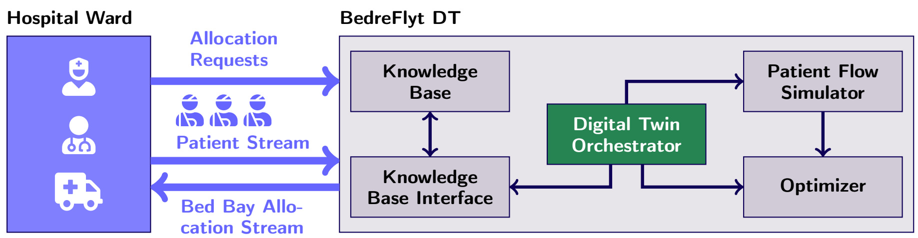


Fig. 1: BedreFlyt DT information flow.

used as a patient flow simulator. We define the optimization problem for bed bay allocation as a constraint satisfaction problem. For this purpose, DT integrates the theorem prover Z3 [12], which is an established SMT solver implementation. The orchestration language SMOL [30, 31] is used to connect the knowledge graph to the ABS and Z3 models. SMOL supports querying a knowledge base, which includes the reflection of the runtime state of the SMOL program itself, via SPARQL and SHACL queries (e.g., [24]). The ABS model transforms the stream of patient data into a stream of constraint problems that capture the bed bay allocation problem at different points in time. Together with a description of the ward, these are turned into optimization problems. The four key components of the architecture are: (1) a digital twin orchestrator, (2) a knowledge base and its interface, (3) a patient flow simulator, and (4) an optimizer.

Static domain-specific information is kept in the knowledge base, while dynamic information arrives in two input types. The first type of dynamic information is a stream of incoming patients on a daily basis. The second type of information is *allocation requests* that are received via a simulation interface. The user of the interface, presumed to be hospital staff, requests a bed bay allocation given the current patients, in response to which the interface proposes a possible allocation. Alternatively, the user may ask for a simulation of a stream of patients under different strategy assumptions. In the following, we describe the functionality of the individual components to output such an allocation. We provide details on the different strategies in Sect. 3. We note that this architecture is generally applicable when considering similar problems.

Communication between the orchestrator, patient flow simulator and optimizer components is via *discrete timed streams*. For a set X, a timed stream over X is a sequence of elements of X tagged with monotonically increasing timestamps in $\mathbb{N}$. We write $t_0{:}x_0$, $t_0{:}x_1$, $t_1{:}x_2$ to denote a timed stream where x_0 and x_1 occur at time t_0 and x_2 occurs at time t_1.

2.2 The Digital Twin Orchestrator

The digital twin orchestrator is the interface to the twin and coordinates information flow to the twin's other components. It receives a stream of patients and allocation requests, and creates a timed stream of so-called *packages* detailing patient and treatment information, that serves as input for the patient flow simulator.

Table 1: Example of timed patient input stream.

Arrival Time	Patient	Diagnosis	Gender	Contagious
1	Alice	D1	♀	True
	Bob	D1	♂	False
2	Charlie	D2	♀	False

Let P be the set of patients. A patient $p \in P$ is a tuple $\langle id, g, q, d\rangle$, where id is a unique identifier, $g \in \{♀, ♂\}$ their gender as distinguished by the hospital, $q \in \mathbb{B}$ a Boolean value indicating if the patient is contagious, and d their diagnosis.

The digital twin orchestrator receives at a time $t \in \mathbb{N}$ a set of patients $\{p_1, \ldots, p_n\}$, which all arrive at t. It then constructs a timed stream of *packages* for the patient flow simulator by selecting treatments for each patient's diagnosis based on a strategy. A package is a triple $\langle t, p, \phi\rangle$ consisting of the time of arrival t, patient tuple $p \in P$, and a sequence of tasks ϕ associated with a treatment $tr \in Tr$. The strategy for selecting a treatment tr for a given diagnosis can be based on cost functions or probabilities, we detail the strategies currently supported in the BedreFlyt DT in Sect. 3. After providing the packages, the twin orchestrator receives the stream of bed bay allocations from the optimizer, formats the data and returns it to the user.

Example 1. Table 1 depicts a stream of three patients arriving over two time steps. In the first step, two patients with diagnosis D1 arrive, and in the second step a single patient with diagnosis D2. Note that the patient identified by Alice is contagious and should therefore be isolated. From this patient stream, the digital twin orchestrator may generate the package stream

$$\langle 1, p_{\text{Alice}}, \langle 1,3\rangle\langle 2,2\rangle\rangle\,, \langle 1, p_{\text{Bob}}, \langle 1,3\rangle\langle 2,2\rangle\rangle\,, \langle 2, p_{\text{Charlie}}, \langle 1,3\rangle\langle 1,2\rangle\rangle\,,$$

by picking the tasks the most frequent treatment (see Table 2); here p_{Alice} is the patient tuple for Alice, etc.

2.3 The Knowledge Base and its Interface

The knowledge database contains static information about the rooms in the hospital ward and about the considered diagnoses and their associated treatments. For a well-structured representation of this knowledge, we use ontologies [45]. The BedreFlyt DT ontology was modeled based on existing available healthcare ontologies and standards, e.g., [11, 42, 54].

To interface the knowledge base, we use SMOL [30], a small imperative object-oriented programming language that leverages ontologies to develop DTs through semantic reflection. SMOL allows an easy integration of ontologies in DT architectures and provides access to the knowledge modeled in the ontology for DT orchestration [32]. Currently, SMOL is only used to query the knowledge base. In the future, SMOL could orchestrate components of the DT represented

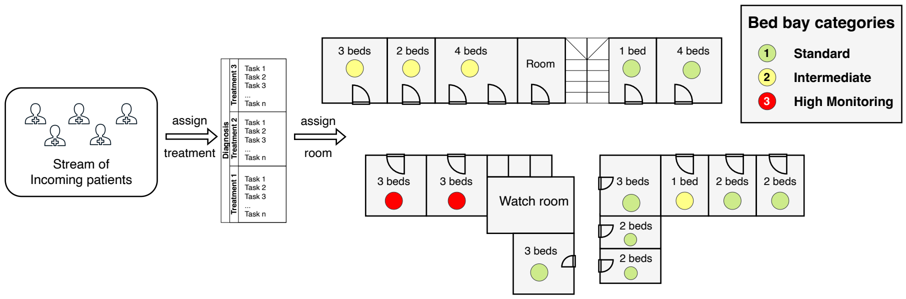


Fig. 2: Example of a hospital ward in BedreFlyt DT.

in a knowledge base, where queries to the knowledge base consider the semantic reflection of the runtime state [30].

One part of the ontology covers the modeling of assets, i.e. rooms, in the hospital ward. Our example hospital ward of *Rikshospitalet* is depicted in Fig. 2. We describe a room from the set of rooms R as a triplet $\langle r, b, c\rangle$, where r is a room identifier, $b \in \mathbb{N}$ denotes the number of beds in the room, and $c \in \mathcal{C}$ defines the monitoring category. In the following, we refer to a room in R also by its identifier r. We distinguish between three levels of monitoring categories $\mathcal{C} = \{1, 2, 3\}$, which describe the amount of monitoring capabilities a room provides. Categories are arranged in ascending order, where category 1 maps to standard monitoring efforts, 2 to intermediate, and 3 to high monitoring efforts.

The second part of the ontology models diagnoses and their subsequent treatments, where each treatment consists of a sequence of tasks. Let Ta be the set of task, where a task $ta \in Ta$ is a pair $\langle d, c\rangle \in \mathbb{N} \times \mathcal{C}$, with d the task duration, and c the minimal required monitoring category.

We describe a treatment as a pair $tr = \langle \phi, \omega\rangle$, where $\phi = \langle d_1, c_1\rangle, \ldots, \langle d_n, c_n\rangle$ is a sequence of n tasks and $\omega \in \mathbb{R}_{\geq 0}$ is a real-valued weight. The weight allows the assignment of a real-valued effort to a treatment, where the higher the weight the higher the effort for the treatment. The assignment of the weight is domain-specific, and relies on expert knowledge from the hospital staff. For our DT setup, we used for a treatment $\langle \phi, \omega\rangle$ a simple approximation for ω based on the sequence of tasks ϕ using

$$\omega = \sum_{\langle d_i, c_i\rangle \in \phi}^{n} d_i \cdot c_i. \tag{1}$$

A *distribution* over a finite set X is a function $\mu\colon X \to [0, 1]$, where $\sum_{x \in X} \mu(x) = 1$. We write $Dist(X)$ for the set of distributions over X. Let $\mathcal{D}$ be the set of identifiers of possible diagnoses; in the data from *Rikshospitalet* they are alphanumeric codes like "D320", "I60" and "C713". We define a probabilistic treatment function $R\colon \mathcal{D} \to Dist(Tr)$ that maps a diagnosis to a distribution of treatments.

Example 2. Table 2 depicts a treatment knowledge base consisting of two diagnoses, each with two treatments. The knowledge base includes the calculated

Table 2: Example ontology of diagnoses.

Diagnosis	Treatment	Frequency	Weight	Tasks
D1	T1	0.8	7	$\langle 1,3\rangle$, $\langle 2,2\rangle$
	T2	0.2	8	$\langle 2,3\rangle$, $\langle 1,2\rangle$
D2	T3	0.6	5	$\langle 1,3\rangle$, $\langle 1,2\rangle$
	T4	0.4	3	$\langle 1,2\rangle$, $\langle 1,1\rangle$

weight and a distribution over the treatments for each diagnosis. For example, the diagnosis D2 has two possible treatments (T3 and T4) and the treatment T3 consists of two length 1 tasks with respective monitoring categories 3 and 2. The probabilistic treatment function R maps diagnosis D2 to the distribution defined by the frequencies of treatments T3 and T4, thus, $R(D2)(\mathrm{T3}) = 0.6$ and $R(D2)(\mathrm{T4}) = 0.4$. Additionally, the knowledge base describes the hospital ward depicted in Fig. 2, i.e., a hospital ward with 14 rooms, two of which have the highest monitoring category.

2.4 The Patient Flow Simulator

We simulate the workflow by connecting the static structure in the knowledge base, see Sect. 2.3, with the dynamic patient/treatment stream. The workflow simulator takes a timed stream of packages, see Sect. 2.2, including patient information with associated treatments, as input and produces a timed stream of bed bay requirements as output, each of which captures the bed bay allocation problem to be solved at a particular point in time.

The simulator, implemented in ABS [28], receives[5] an input stream of data from the digital twin orchestrator. The simulator retrieves new data from the digital twin orchestrator at different points in time and reuses the notion of packages internally to capture the active patient treatments; a package consists of patient information and the remaining tasks in the patient's treatment at a certain point of time.

We define a bed bay requirement as a tuple $\beta = \langle id, g, q, c\rangle$ where id is a patient identifier, $g \in \{♀, ♂\}$ is a gender, $q \in \mathbb{B}$ indicates contagiousness, and $c \in \mathcal{C}$ is the minimum monitoring category. The bed bay requirements are calculated from the sequence of tasks ϕ in the treatment tr. The ABS model keeps track of active packages and their remaining tasks. At time t, the simulator

1. checks for new packages (i.e., the incoming packages for time t);
2. for each active package $\langle t', p, \langle d_1, c_1\rangle \ldots \langle d_n, c_n\rangle\rangle$, with $t' \leq t$:
   (a) output the bed bay requirement $\langle id, g, q, c_1\rangle$ for time t where id, g, q are the identifier, gender and contagiousness status of patient p and c_1 is the minimum monitoring category of the current task, and

[5] Technically, the digital twin orchestrator stores the data locally in an embedded SQLite database (see https://sqlite.org), that is queried from the simulator.

(b) decrement d_1 and remove the associated task if it reaches 0; and
3. remove any packages that have no more tasks.

The ABS simulator runs as long as there are active packages, generating a stream of bed bay requirements for different points in time. This output is used to generate a stream of optimization problems for the optimizer component. Remark that the simulation model in ABS is more general than our current case study, because the architecture of the simulator can handle tasks that occur at the same time and have different resource needs; e.g., a laboratory test can occur while the patient occupies a bed bay during recovery. Furthermore, dynamic, unforeseen variations in task duration can be simulated by exploiting the timed semantics of ABS [29].

2.5 The Optimizer

We use the theorem prover Z3 [12] to compute a bed bay allocation for a given stream of packages (patients with treatments); i.e., for each simulated step $t \in T$, we compute an assignment from patients to beds such that all constraints on gender, monitoring categories, contagiousness status, and room capacities are satisfied. We then introduce a target function to minimize the number of required bed bay reallocations, i.e. the number of bed bay changes of single patients over their stay, by minimizing the number of required bed bay reallocations, i.e. the number of bed bay changes of single patients over their stay.

Recall that room $\langle r, b, c\rangle \in R$ is a tuple over the number of beds b and the monitoring category c, and is referenced by r. Further, a bed bay requirement for a patient p is a tuple $\beta = \langle id, g, q, c\rangle$. The input to the optimizer is a timed stream of bed bay requirements $S = t_0{:}\beta^0_1, \ldots, t_0{:}\beta^0_{m_{t_0}}, \ldots, t_n{:}\beta^n_1, \ldots, t_n{:}\beta^n_{m_{t_n}}$ where m_t denotes the number of patients arriving at time t. To shorten notation, let $P^t = \{id \mid t : \langle id, g, q, c\rangle \in S\}$ be the set of patient ids for a time t, and g^t_{id}, q^t_{id}, and c^t_{id} the gender, contagiousness, and minimum room requirement of a patient with identifier id at time t. Note that gender and contagiousness are constant over time. To compute a valid bed bay allocation, we reformulate the entire problem into a quantifier-free linear real arithmetic formula.

To encode the constraint problem, we introduce two types of variables: (1) variable $a^t_{id,r} \in \{0, 1\}$ encodes that at time t, a patient with identifier id is assigned to room r, and (2) variable g^t_r specifies the gender of the room r at that time step. For each time t, the assignment problem is decomposed into the following sub-formulas:

- $\varphi^t_{patient}$ assigns each patient to exactly one room,
- φ^t_{room} limits the number of patients in a room by the room's capacity,
- φ^t_{gender} ensures that patients sharing a room have the same gender by enforcing that all patients in a room have the gender assigned to that room,
- $\varphi^t_{contagious}$ ensures that contagious patients are alone in their room, and
- $\varphi^t_{category}$ restricts the bed bay assignable to a patient based on the monitoring category.

$$
\begin{aligned}
\varphi^t_{patient} &\coloneqq \bigwedge_{id\in P^t} \sum_{r\in R} a^t_{id,r} = 1,\\
\varphi^t_{room} &\coloneqq \bigwedge_{r\in R} \sum_{id\in P^t} a^t_{id,r} \leq b,\\
\varphi^t_{gender} &\coloneqq \bigwedge_{id\in P^t, r\in R} a^t_{id,r} \Longrightarrow g^t_{id} = g^t_r,\\
\varphi^t_{contagious} &\coloneqq \bigwedge_{id\in P^t, r\in R} a^t_{id,r} \wedge q^t_{id} \Longrightarrow \bigwedge_{id'\in P^t\setminus\{id\}} \neg a^t_{id',r},\\
\varphi^t_{category} &\coloneqq \bigwedge_{id\in P^t, r\in R} a^t_{id,r} \Longrightarrow c^t_{id} \leq c
\end{aligned}
$$

Fig. 3: The sub-formulas of the bed bay allocation constraint problem.

The formulas are detailed in Fig. 3. Then, the formula $\varphi^t \coloneqq \varphi^t_{patient} \wedge \varphi^t_{room} \wedge \varphi^t_{gender} \wedge \varphi^t_{contagious} \wedge \varphi^t_{category}$ ensures that there exists an assignment of patients to bed bays at time t if and only if that assignment is sound.

We further constrain φ^t to respect the previous bed bay allocation by minimizing the number of required reallocations, avoiding patients being moved around when they stay at the hospital. To this aim, we introduce variable δ^t_{id} indicating whether the patient with identifier $id \in P^t \cap P^{t-1}$ had to move beds between time $t-1$ and t. Minimizing $\sum_{t\in T} \sum_{id\in P^t} \delta^t_{id}$ under $\bigwedge_{t\in T} \varphi^t \wedge \varphi^t_{changes}$ constructs bed bay allocations minimizing the aggregated patient moves for all time steps, where

$$
\varphi^t_{changes} \coloneqq \begin{cases} \bigwedge_{id\in P^0} a^0_{id,r_{id}}, & \text{if } t = 0,\\ \bigwedge_{id\in P^t\cap P^{t-1}} (a^{t-1}_{id,r} \Longrightarrow a^t_{id,r}) \vee \delta^t_{id}\ , & \text{otherwise.} \end{cases}
$$

Let $a^0_{id,r_{id}}$ denote the initial bed bay allocation for patients $p \in P^0$. If no such allocation is given, the first case in $\varphi^0_{changes}$ defaults to $\top$.

Further, we note that by encoding all time steps, patients and rooms into a single problem, a significant number of variables is introduced. However, our experiments in Sect. 4 reveal that the problem remains computationally feasible in the context of *optimization modulo theories* as implemented in Z3 [7]. If the constructed optimization problem is satisfiable, an optimal allocation of rooms is returned along with the patients that need to be moved to a different room.

3 What-If Scenarios

To reason about potential futures, the BedreFlyt DT employs what-if scenarios. In particular, there might be several treatments for the same diagnosis, depending on the availability of equipment, the patient's preference and underlying health conditions; and a choice of treatment changes the scenario.

Adopting the terminology of *Scenic* [19], a *scenario* is a distribution over configurations, while a *scene* is one such configuration. In our setting, scenes are timed streams of patients with assigned treatments — the inputs to the patient flow simulator. Scenarios are distributions over scenes, given by a patient stream and a (potentially stochastic) *strategy* for selecting treatments — the inputs to the DT orchestrator. Since a strategy determines a scenario if the patient stream is fixed, we conflate the two if the patient stream is clear from the context or does not matter. For example, “worst-case scenario” means the worst-case strategy applied to an understood patient stream.

The BedreFlyt DT implements one deterministic and one stochastic strategy, where the stochastic strategy can be used in simulations to compute expected outcomes. We first explain strategies considered for the hospital ward planning problem in the BedreFlyt DT and how they are used to explore what-if scenarios, before a brief discussion of implementation.

3.1 Different Strategies for the BedreFlyt DT

The existing BedreFlyt DT framework [47] for bed bay allocation is useful for understanding the current state of the hospital ward in relation to the incoming patients by solving the bed bay allocation of incoming patients with given treatments. To analyze the hospital’s ability to accommodate patients under different treatments, we now develop a what-if analysis by considering different *strategies* for assigning treatments to patients. We implement (1) worst-case and (2) sampled-case strategies for assigning treatments to patients. When performing simulations, a so-called *risk tolerance* parameter additionally determines the probability of defaulting to the worst-case strategy. By sampling and varying this risk tolerance, we can construct a wide range of what-if scenarios and simulate their expected outcomes.

1. **Worst-case strategy.** The worst-case strategy always picks the treatment with the highest weight. Since the choice of treatment may depend on many factors outside our control (i.e., not-modeled resource requirements, patient health, patient preferences, etc.) this strategy allows us to simulate a scenario in which the hospital is maximally unlucky. This strategy is *deterministic* in that the same diagnosis always results in the same treatment. In the case of a tie we assume a total order on the treatments and pick the first.
2. **Sampled-case strategy.** The sampled-case strategy stochastically picks a treatment for a given diagnosis $\mathcal{D}$ based on the frequencies provided by the probabilistic treatment function $R(\mathcal{D})$. Thus, a treatment with a frequency of 0.2 is expected to be picked one in five times, for patients arriving with that diagnosis.

The treatment strategy is used by the DT to map each patient to a treatment in each *run*, where a run comprises the patient flow simulation and optimizer allocation for a scene. Each run consists of three steps, (1) the DT orchestrator receives a scenario and a strategy, and generates a scene using the given strategy, (2) then the patient flow simulator turns the scene into a stream of optimization problems, (3) finally the optimizer computes a stream of bed allocations.

The non-deterministic sampled-case strategy and the Monte Carlo method allow us to calculate expected values by aggregating a number of runs into a *simulation*. We compute the expected amount of time taken, the expected proportion of time steps without a feasible bed allocation, and the expected number of patients that need to be moved between bed bays in the hospital ward.

We parameterize each simulation by a so-called *risk tolerance* $\tau \in [0, 1]$ that indicates the degree to which we account for a worst-case assignment of treatments. The sampled-case strategy is used with a probability of τ in each run, otherwise the worst-case strategy is used. Thus, with a risk tolerance of 0, the twin will use the worst-case strategy for all patients in all runs. Note that a choice of risk tolerance and a patient stream constitute a scenario — it defines a distribution of treatment assignments.

Example 3. Consider again the setting from Examples 1 and 2. Using what-if scenarios, there are now multiple ways to assign the patients to their bed bays.

Under the *worst-case strategy*, Alice and Bob will both be assigned the treatment T2, and Charlie will be assigned T3. At time 1, all is well as Alice and Bob are assigned to the two high monitoring rooms. At time 2, there is a problem: Charlie needs to be kept in high monitoring, but she cannot be assigned to Bob's room because of their different genders, nor can she be assigned to Alice's room because Alice is contagious. At time 3, Alice and Bob are moved to standard rooms, and there is space for Charlie in an intermediate room (assuming she got the first step of her treatment elsewhere).

Under the *sampled-case strategy*, there are multiple solutions. If Alice is assigned T1, there will be a free room for Charlie because the female high monitoring room is no longer contagious, and if Bob is assigned T1 the similar situation applies since Charlie can move into the now unoccupied male room and turn it female. On the other hand, Charlie could be assigned T4, and not need a high monitoring room at all.

Performing a *simulation* with the simulation tolerance set to 0.8 over 1000 runs, we find that the scenario always takes 4 time steps, the expected number of unsatisfiable allocation problems is ≈ 0.24, and the expected number of room changes per satisfied time step is ≈ 0.11.

If we take the step size to be days, this means the hospital should expect to be unable to accommodate the patients on $\frac{24\%}{4} = 6\%$ of days, and have to move a patient on 11% of days in this scenario.

3.2 Implementation of Strategies in the BedreFlyt DT

Exploiting the modular nature of the BedreFlyt DT, strategies are implemented entirely in the DT orchestrator — directly reusing the patient flow simulator and optimizer components of the existing digital twin architecture [47].

Allocation requests may contain a strategy flag indicating the strategy to be used, and the choice of strategy determines how the DT orchestrator constructs scenes for the patient flow simulator.

Alternatively, a user may send a simulation request to estimate expectations for a given stream of patients. This request is parameterized by a simulation tolerance as described above, a number of runs to perform, and the stream of patients. The DT orchestrator performs the requested number of runs and reports the expected amount of time treatments for all patients will take, the expected number of unsatisfiable time steps (that is, the number of steps where there are not enough bed bays of the right categories), and the expected number of room changes. Since the worst-case strategy is deterministic, its results are cached and reused.

Compared to the previous work [47], we have decoupled the patients diagnosis from their treatments in the BedreFlyt DT. By enriching the knowledge base with multiple treatments for a diagnosis — as well as the additional information concerning their respective weights and frequencies — we can use this information to implement the strategies described above.

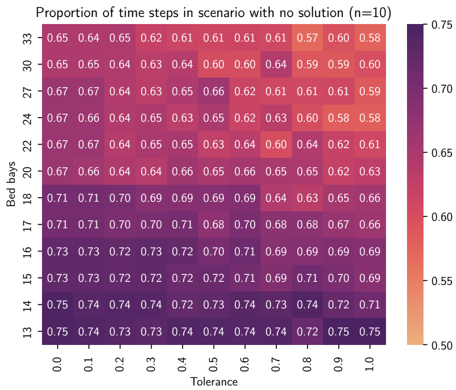
4 Evaluation

We evaluate the resulting extension of the BedreFlyt DT along two axes: (1) the use of simulation to analyze what-if scenarios, and (2) the optimality of the solution with respect to the number of bed bay changes to which patients are subjected. Using the simulation requests described in Sect. 3, we investigate the expected number of satisfiable time steps and number of bed changes for a fixed stream of patients across different what-if scenarios. We note that the BedreFlyt DT implements an *online* approach, i.e., one that considers only one step at the time. Alternatively, if the full patient stream is known in advance, we may compute an optimal offline solution as described in Sect. 2.5, we evaluate the quality of the online solution in comparison to that optimal solution.

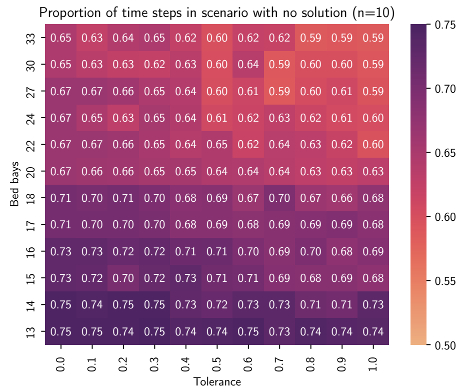
Comparing strategies. To investigate the impact of the strategy choice, we create different simulation experiments — sets of simulations with different tolerance levels and bed bay availability. We fix the diagnosis-treatment information in the knowledge base and the stream of patients for each experiment, and vary the tolerance and the number of available high-category bed bays. This answers the question: "Given a tolerance for risk, how many bed bays do we need to upgrade to a higher monitoring category for the incoming stream of patients?". Note that we do not add or remove rooms, but upgrade existing rooms by adjusting their monitoring category. Furthermore, the number of steps it takes to treat all the patients in a stream may vary in each run since the selection of treatments for the same diagnosis is stochastic in general, and so, the DT outputs the average *proportion* of unsatisfiable steps. For our experiments, we randomly generate incoming patient streams, using the anonymized patient identifiers and diagnoses from given historical hospital data. Stream A has 350 unique patients arriving over the course of 35 time steps and stream B has 75 patients over 7 steps.

Having fixed a stream of patients, we execute simulations with n runs with the sampled-case strategy and a tolerance $\tau \in [0, 1]$. If there exists a time step for which no feasible assignment was found, we find the room with the smallest number of bed bays and a less than maximal category. We upgrade this room to the maximal category, and run again n simulations. We continue this process until there are no more rooms to upgrade or no steps are unsatisfiable.

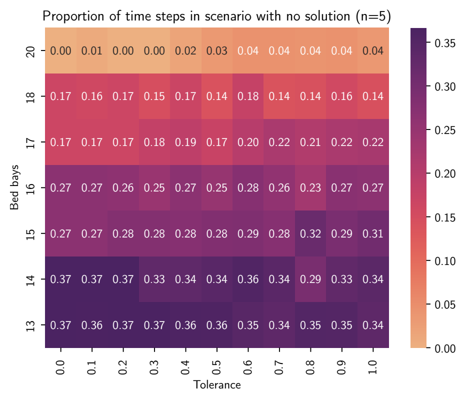
Figure 4 depicts our obtained results. The expectation before the experiments was to confirm that higher tolerance levels will *decrease* the number of



(a) Patient Stream A



(b) Patient Stream A



(c) Patient Stream B

Fig. 4: Unsatisfiable time steps by tolerance and available category-3 bed bays for two different patient streams using simulations with n runs. Plots (a) and (b) show different simulations of the what-if scenario for Patient Stream A.

unsatisfiable problems, since fewer patients receive the worst-case treatment. Consequently, the number of unsatisfiable problems should be inversely proportional to the number of available category-3 bed bays, as there are simply more available slots. Remember that a category-3 room can host a patient of any category, as discussed in Sect. 2.5. Our experiment results approximately align with our expected result — the proportion of unsatisfiable problems generally decreases going up and right, where high tolerance is captured. However, we also observe some noise, which is caused by the complex interactions of patients' requirements. For example, one patient being assigned a treatment with less weight may result in moving them to a higher category room sooner, thus conflicting with other patients already there. Note that the results are relatively

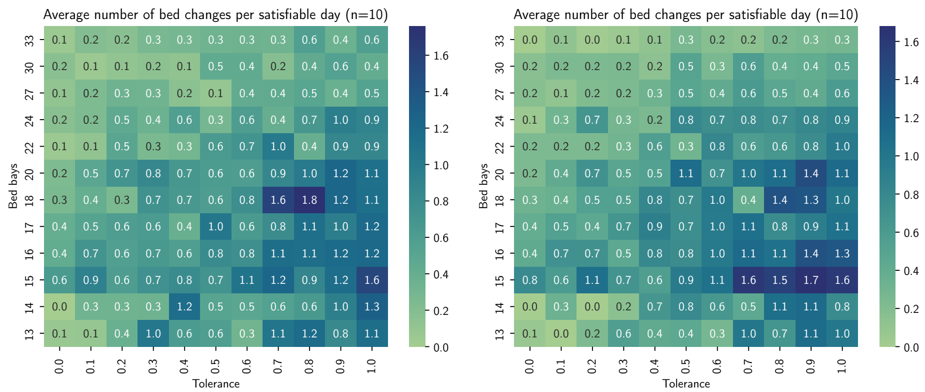


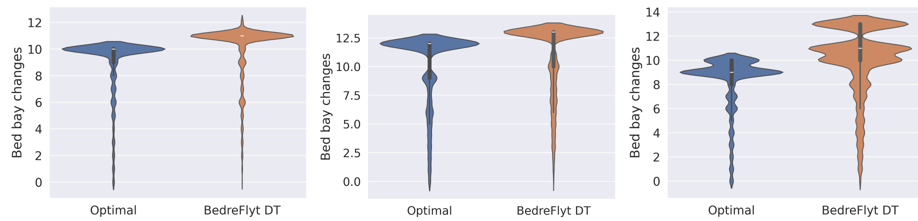
Fig. 5: Average number of bed bay changes per time step, for Patient Stream A; the plots show different simulations of the same what-if scenario.

stable across simulations. The plots (a) and (b) depict two different simulations of the same patient stream and are very similar — though not exactly the same.

If there exist multiple possible allocations of patients to bed bays such that all constraints are satisfied, the hospital would like to pick the one that requires the fewest number of patients to be moved. To this end, we provide the optimizer with the allocation for the previous time step and compute the number of room changes as explained in Sect. 2.5. The optimizer then computes an allocation that minimizes the number of changes. Figure 5 reports the number of changes per satisfiable time step in two simulations of the same patient stream. As before, we vary the tolerance and number of available category-3 bed bays. Note that more bed bays and lower tolerance levels lead to a larger number of satisfiable problems and, hence, a larger number of possible moves. For this reason, Fig. 5 shows the average number of bed bay changes per satisfiable time step.

Quality of the online solution. Due to the limited predictability of bed bay allocations in hospitals in real time, BedreFlyt DT assigns bed bays in an online fashion. Specifically, BedreFlyt DT implements a greedy algorithm that minimizes the number of bed bay changes for the current time step. Note that we can easily construct a situation where moving one patient now will prevent two forced moves in the next step, thus the greedy solution cannot be optimal. In fact, the greedy solution for the very similar k*-server problem* can be arbitrarily bad, but competitive algorithms exist [35].

To investigate the gap between online and optimal allocation solutions, we implemented a t-indexed version of the optimization problem, computing a optimal offline solution and comparing it with the greedy approach in the BedreFlyt DT. We employ a meta-heuristic search for synthetic package streams that maximize the number of bed bay changes of the optimal solution, and bypass the simulation component to directly generate problem instances for the optimizer.



(a) 40 Patients (b) 50 Patients (c) 60 Patients

Fig. 6: Comparison of optimal and BedreFlyt DT solutions.

Specifically, we search for length 5 streams for the same hospital ward as before. This hospital ward has a total of 37 beds — 13, 11, and 13 of respectively category 1, 2, and 3 — distributed across 17 rooms. We vary the total number of patients and then compare the optimal solution, i.e., the minimum possible number of bed changes when all arrivals are known, to the online solution (with partial information) employed by BedreFlyt DT. For each choice of patent number, we generate in total 30 000 instances, and report the number of changes for the optimal (left) and the online solution (right). The results are displayed in Fig. 6. For 40 and 50 patients, both approaches produce similar distributions of results — the median BedreFlyt DT solution makes one more move than optimal. For 60 patients, BedreFlyt DT skews further than the optimal solution, requiring up to 14 moves where optimal does not reach above 11 moves.

These results indicate that while theoretically, the greedy solution can be arbitrarily bad, it performs similar to the optimal solution on realistic scenes with a moderate number of patient moves required. The gap grows slightly larger with larger numbers of patients, but stays within reasonable bounds.

5 Related Work

Digital Twins in healthcare is an emerging topic [1,57], especially with the impact that COVID-19 had on our lives [23], where emergent DT technologies, e.g. [37], can be integrated with different devices to provide a more comprehensive view of the patients' health. Many solutions explore the application of AI to improve DTs for healthcare, as done in [33], which combined with context-sensitive applications, can lead to a comprehensive and scalable health system, e.g. [13]. Furthermore, existing work reports on the use of DTs in conjunction with AI to monitor the health of patients in real time [39], management of computational resources [26], etc. In contrast, our work explores digital twins for operational resource analysis in healthcare. Although the use of AI for resource allocation in hospitals has also been explored, e.g. [36], our work focuses on the use of formal methods for resource analysis.

Digital twins have proven to be effective in resource management and resource allocation, e.g. [43, 49], but they have not been extensively explored in the context of healthcare [14, 25, 51]. Resource analysis in hospitals is a critical issue that has been extensively addressed, especially in the Emergency Department (ED) [18], where crucial and trustworthy decisions must be made effectively under uncertainties; in this domain, having a tool to make informed decisions can help drastically. Simulation tools for resource management and decision making are a well-established technique in healthcare [2, 44], where they can be used for e.g., resource allocation [16], demographic trends forecasting [21], etc. However, such simulation models have not been integrated into digital twins, as done in this paper. As explored in this paper, the notion of a digital twin goes beyond simulation to include tighter integration between models, data, and decisions [55].

In the context of healthcare and resource management, a data driven performance measurement technique has been used to evaluate the efficiency of hospitals [10] to align the resource allocation needs, while meta-heuristic methods [50] have been assessed for patient allocation. Toward data-driven applications, the integration of different models for resource and capacity allocation in hospitals have been explored [58,59], in both cases, strategies were used to leverage the big volume of data incoming from the hospitals to create a more efficient system to improve operational performance. Closer to the analysis of live incoming streams, as done in digital twins, existing work reports on the use of an adaptive method from near real-time data to predict future bed occupancy levels during a pandemic wave [20], but not for concrete allocation of bed bays at the hospital ward.

In contrast to all the work reported in this section, our work explores digital twin architectures that take advantage of domain knowledge, captured in a knowledge base, for the online and offline analysis of incoming streams of data for decision making support. Furthermore, our work also explores the orchestration of formal models for digital twins, with analysis strategies that are parametric in risk tolerance and consider a human in the loop, touching on various open challenges for digital twins [55].

6 Conclusion

We report on the development of the BedreFlyt digital twin (DT), a tool for resource allocation planning in healthcare. The BedreFlyt DT combines a patient flow simulator and an optimizer to allocate bed bays in a hospital ward according to a stream of incoming patients and their treatments. The proposed DT architecture orchestrates a knowledge base, a patient flow simulator, and an optimizer. In this paper, we consider how the capabilities of the BedreFlyt DT can be extended to analyze hypothetical scenarios over patient streams and strategies for selecting treatments, so called what-if scenarios. We showed how the Monte Carlo method allows the DT to explore expected outcomes for a given knowledge base and patient stream, parameterized by a tolerance factor determining how many of the patient treatments we should expect to be worst case.

Finally, we evaluate the DT's proposed bed bay allocations along two axes: a *qualitative* look at what-if simulations of the same patient stream under different tolerance levels and bed bay availabilities, and a *comparison* of online (day to day) bed bay allocation results with the optimal allocation. The simulation results confirm our assumption that a lower number of bed bays and a lower risk tolerance lead to more unsatisfactory time steps. However, results can vary and be non-monotone due to complex interacting factors. In terms of optimizing the necessary bed bay moves during a patient's stay, we find that the online solution can be arbitrarily sub-optimal, but that for realistic problems the deficit is relatively small.

There are several interesting directions for future work. First, our analysis of what-if scenarios currently work over fixed patient streams, and we would like to sample traffic patterns in a similar way as we now sample treatments. Further, the greedy online bed bay allocation has proven reasonably effective, but does not utilize all the available data. By integrating more knowledge about future steps in treatment sequences, we may further reduce the number of necessary bed bay changes. In addition, the assumption of static information in the knowledge base could be too restrictive in practice. By using techniques to self-adapt the knowledge base, we may ensure that DT better reflects the hospital's reality. Furthermore, online learning could be used to make what-if scenarios more realistic by reinforcing observed behavior in the selected strategies. We note that our assumptions on treatment frequencies and weights are artificial; Using real data from the hospital instead would allow for more realistic modeling and prediction, where it will be interesting to explore advanced stochastic techniques for resource allocation to address, e.g., uncertainties, as explored in [15,38]. Such techniques may also include probabilistic sampling over distributions of patient streams to enable even greater variation in scenario modeling. Finally, we would like to explore the use of Markov decision processes (MDPs) in the healthcare setting. As a modeling and analysis tool for systems with both probabilistic and non-deterministic elements, MDPs are well suited to the world of hospitals where the flow of patients is unknown and staff makes non-deterministic decisions.

Acknowledgment. The work was partly funded by the South-Eastern Norway Regional Health Authority (Helse Sør-Øst) through the project *BedreFlyt*, and the Research Council of Norway through the *Smart Journey Mining* project (grant no. 312198). We thank the other project participants, especially Céline Cunen, Ingrid Konstanse Ledel Solem, Frode Strisland and Manuela Zucknik who helped collect and organize data about patient treatments and patient streams at Oslo University Hospital - Rikshospitalet. We thank the anonymous reviewers for their helpful feedback.

References

1. Alazab, M., Khan, L.U., Koppu, S., Ramu, S.P., Meenakshisundaram, I., Boobalan, M.P., Baker, T., Maddikunta, P.K.R., Gadekallu, T.R., Aljuhani, A.: Digital twins for Healthcare 4.0 - recent advances, architecture, and open challenges. IEEE Consumer Electron. Mag. **12**(6), 29–37 (2023). https://doi.org/10.1109/MCE.2022.3208986

2. Almagooshi, S.: Simulation modelling in healthcare: Challenges and trends. Procedia Manufacturing **3**, 301–307 (2015). https://doi.org/10.1016/j.promfg.2015.07.155, 6th Int. Conf. on Applied Human Factors and Ergonomics and the Affiliated Conferences, AHFE 2015
3. Baier, C., Dubslaff, C., Wienhöft, P., Kiebel, S.J.: Strategy synthesis in Markov decision processes under limited sampling access. In: Rozier, K.Y., Chaudhuri, S. (eds.) Proc. 15th International Symposium of NASA Formal Methods (NFM 2023). LNCS, vol. 13903, pp. 86–103. Springer (2023). https://doi.org/10.1007/978-3-031-33170-1_6
4. Baier, C., Katoen, J.: Principles of model checking. MIT Press (2008)
5. Baier, C., Piribauer, J., Ziemek, R.: Foundations of probability-raising causality in Markov decision processes. Log. Methods Comput. Sci. **20**(1) (2024). https://doi.org/10.46298/LMCS-20(1:4)2024
6. Billey, A., Wuest, T.: Energy digital twins in smart manufacturing systems: A case study. Robotics Comput. Integr. Manuf. **88**, 102729 (2024). https://doi.org/10.1016/j.rcim.2024.102729
7. Bjørner, N.S., Phan, A., Fleckenstein, L.: νz - an optimizing SMT solver. In: Baier, C., Tinelli, C. (eds.) Proc. TACAS 2015. LNCS, vol. 9035, pp. 194–199. Springer (2015). https://doi.org/10.1007/978-3-662-46681-0_14
8. Casanova, G.: Histoire de ma vie. Robert Laffont (1993), manuscript originally written 1789–1798.
9. Chang, X., Zhang, R., Mao, J., Fu, Y.: Digital twins in transportation infrastructure: An investigation of the key enabling technologies, applications, and challenges. IEEE Trans. Intell. Transp. Syst. **25**(7), 6449–6471 (2024). https://doi.org/10.1109/TITS.2024.3401716
10. Chu, J., Li, X., Yuan, Z.: Emergency medical resource allocation among hospitals with non-regressive production technology: A DEA-based approach. Comput. Ind. Eng. **171**, 108491 (2022). https://doi.org/10.1016/J.CIE.2022.108491
11. Dang, J., Hedayati, A., Hampel, K., Toklu, C.: An ontological knowledge framework for adaptive medical workflow. Journal of Biomedical Informatics **41**(5), 829–836 (Oct 2008). https://doi.org/10.1016/j.jbi.2008.05.012
12. De Moura, L., Bjørner, N.: Z3: An efficient SMT solver. In: Proc. 14th Intl. Conf. on Tools and Algorithms for the Construction and Analysis of Systems. LNCS, vol. 4963, pp. 337–340. Springer (2008). https://doi.org/10.1007/978-3-540-78800-3_24
13. Elayan, H., Aloqaily, M., Guizani, M.: Digital twin for intelligent context-aware IoT healthcare systems. IEEE Internet Things J. **8**(23), 16749–16757 (2021). https://doi.org/10.1109/JIOT.2021.3051158
14. Elkefi, S., Asan, O.: Digital twins for managing health care systems: Rapid literature review. J Med Internet Res **24**(8) (Aug 2022). https://doi.org/10.2196/37641
15. Fan, G., Huang, H.: Scenario-based stochastic resource allocation with uncertain probability parameters. J. Syst. Sci. Complex. **30**(2), 357–377 (2017). https://doi.org/10.1007/S11424-017-6178-5
16. Feng, Y.Y., Wu, I.C., Chen, T.L.: Stochastic resource allocation in emergency departments with a multi-objective simulation optimization algorithm. Health Care Manag Sci **20**(1), 55–75 (2015). https://doi.org/10.1007/s10729-015-9335-1
17. Fitzgerald, J., Gomes, C., Larsen, P.G. (eds.): The Engineering of Digital Twins. Springer (2024). https://doi.org/10.1007/978-3-031-66719-0
18. Florencia, J., Moyaux, T., Trilling, L., Bouleux, G., Cheutet, V.: Toward improving dynamic resource scheduling in the context of digital twin of emergency depart-

ment. IEEE Transactions on Automation Science and Engineering pp. 1–13 (2024). https://doi.org/10.1109/TASE.2024.3463489
19. Fremont, D.J., Dreossi, T., Ghosh, S., Yue, X., Sangiovanni-Vincentelli, A.L., Seshia, S.A.: Scenic: a language for scenario specification and scene generation. In: Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation. p. 63–78. PLDI '19, ACM (Jun 2019). https://doi.org/10.1145/3314221.3314633
20. Garcia-Vicuña, D., López-Cheda, A., Jácome, M.A., Mallor, F.: Estimation of patient flow in hospitals using up-to-date data. Application to bed demand prediction during pandemic waves. PLOS ONE **18**(2) (Feb 2023). https://doi.org/10.1371/journal.pone.0282331
21. Hajłasz, M., Mielczarek, B.: Simulation modeling for predicting hospital admissions and bed utilisation. Operations Research and Decisions **30**(3) (2020). https://doi.org/10.37190/ord200301
22. Hasan, A., Widyotriatmo, A., Fagerhaug, E., Osen, O.: Predictive digital twins for autonomous ships. In: Proc. Conference on Control Technology and Applications (CCTA 2023). pp. 1128–1133. IEEE (2023). https://doi.org/10.1109/CCTA54093.2023.10252433.
23. Hassani, H., Huang, X., MacFeely, S.: Impactful digital twin in the healthcare revolution. Big Data Cogn. Comput. **6**(3), 83 (2022). https://doi.org/10.3390/BDCC6030083
24. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman and Hall/CRC Press (2010). https://doi.org/10.1201/9781420090512
25. Hu, X., Cao, H., Shi, J., Dai, Y., Dai, W.: Study of hospital emergency resource scheduling based on digital twin technology. In: 2021 IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA). vol. 2, pp. 1059–1063 (2021). https://doi.org/10.1109/ICIBA52610.2021.9688239
26. Jameil, A.K., Al-Raweshidy, H.S.: AI-enabled healthcare and enhanced computational resource management with digital twins into task offloading strategies. IEEE Access **12**, 90353–90370 (2024). https://doi.org/10.1109/ACCESS.2024.3420741
27. Ji, S., Pan, S., Cambria, E., Marttinen, P., Yu, P.S.: A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Trans. Neural Networks Learn. Syst. **33**(2), 494–514 (2022). https://doi.org/10.1109/TNNLS.2021.3070843
28. Johnsen, E.B., Hähnle, R., Schäfer, J., Schlatte, R., Steffen, M.: ABS: A core language for abstract behavioral specification. In: 9th International Symposium on Formal Methods for Components and Objects (FMCO 2009). LNCS, vol. 6957, pp. 142–164. Springer (2010). https://doi.org/10.1007/978-3-642-25271-6_8
29. Johnsen, E.B., Schlatte, R., Tapia Tarifa, S.L.: Integrating deployment architectures and resource consumption in timed object-oriented models. J. Log. Algebraic Methods Program. **84**(1), 67–91 (2015). https://doi.org/10.1016/J.JLAMP.2014.07.001
30. Kamburjan, E., Klungre, V.N., Schlatte, R., Johnsen, E.B., Giese, M.: Programming and debugging with semantically lifted states. In: Verborgh, R., Hose, K., Paulheim, H., Champin, P., Maleshkova, M., Corcho, Ó., Ristoski, P., Alam, M. (eds.) Proc. 18th Intl. Conf. on the Semantic Web (ESWC 2021). LNCS, vol. 12731, pp. 126–142. Springer (2021). https://doi.org/10.1007/978-3-030-77385-4_8
31. Kamburjan, E., Klungre, V.N., Schlatte, R., Tapia Tarifa, S.L., Cameron, D., Johnsen, E.B.: Digital twin reconfiguration using asset models. In: Proc. 11th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation. Practice (ISoLA 2022). LNCS, vol. 13704, pp. 71–88. Springer (2022). https://doi.org/10.1007/978-3-031-19762-8_6

32. Kamburjan, E., Pferscher, A., Schlatte, R., Sieve, R., Tapia Tarifa, S.L., Johnsen, E.B.: Semantic reflection and digital twins: A comprehensive overview. In: Hinchey, M., Steffen, B. (eds.) The Combined Power of Research, Education, and Dissemination: Essays Dedicated to Tiziana Margaria on the Occasion of Her 60th Birthday, Lecture Notes in Computer Science, vol. 15240, pp. 129–145. Springer (2025). https://doi.org/10.1007/978-3-031-73887-6_11
33. Kaul, R., Ossai, C.I., Forkan, A.R.M., Jayaraman, P.P., Zelcer, J., Vaughan, S., Wickramasinghe, N.: The role of AI for developing digital twins in healthcare: The case of cancer care. WIREs Data Mining Knowl. Discov. **13**(1) (2023). https://doi.org/10.1002/widm.1480
34. Kobialka, P., Pferscher, A., Bergersen, G.R., Johnsen, E.B., Tapia Tarifa, S.L.: Stochastic games for user journeys. In: Proc. 26th Intl. Symp. on Formal Methods (FM 2024). LNCS, vol. 14934, pp. 167–186. Springer (2024). https://doi.org/10.1007/978-3-031-71177-0_12
35. Koutsoupias, E.: The k-server problem. Computer Science Review **3**(2), 105–118 (2009). https://doi.org/10.1016/j.cosrev.2009.04.002
36. Lazebnik, T.: Data-driven hospitals staff and resources allocation using agent-based simulation and deep reinforcement learning. Eng. Appl. Artif. Intell. **126**, 106783 (2023). https://doi.org/10.1016/J.ENGAPPAI.2023.106783
37. Liu, Y., Zhang, L., Yang, Y., Zhou, L., Ren, L., Wang, F., Liu, R., Pang, Z., Deen, M.J.: A novel cloud-based framework for the elderly healthcare services using digital twin. IEEE Access **7**, 49088–49101 (2019). https://doi.org/10.1109/ACCESS.2019.2909828
38. Melouk, S.H., Fontem, B.A., Waymire, E., Hall, S.N.: Stochastic resource allocation using a predictor-based heuristic for optimization via simulation. Comput. Oper. Res. **46**, 38–48 (2014). https://doi.org/10.1016/J.COR.2013.12.010
39. Mohamed, N., Al-Jaroodi, J., Jawhar, I., Kesserwan, N.: Leveraging digital twins for healthcare systems engineering. IEEE Access **11**, 69841–69853 (2023). https://doi.org/10.1109/ACCESS.2023.3292119
40. Mukamel, D., Zwanziger, J., Bamezai, A.: Hospital competition, resource allocation and quality of care. BMC health services research **2**, 10 (06 2002). https://doi.org/10.1186/1472-6963-2-10
41. National Academies of Sciences, Engineering, and Medicine (NASEM): Foundational Research Gaps and Future Directions for Digital Twins. The National Academies Press (2024). https://doi.org/10.17226/26894
42. Neumann, J., Uciteli, A., Meschke, T., Bieck, R., Franke, S., Herre, H., Neumuth, T.: Ontology-based surgical workflow recognition and prediction. Journal of Biomedical Informatics **136**, 104240 (Dec 2022). https://doi.org/10.1016/j.jbi.2022.104240
43. Peng, K., Huang, H., Bilal, M., Xu, X.: Distributed incentives for intelligent offloading and resource allocation in digital twin driven smart industry. IEEE Trans. Ind. Informatics **19**(3), 3133–3143 (2023). https://doi.org/10.1109/TII.2022.3184070
44. Pitt, M.: A generalised simulation system to support strategic resource planning in healthcare. In: Proc. 29th Conference on Winter Simulation (WSC'97). pp. 1155–1162. ACM (1997). https://doi.org/10.1109/WSC.1997.641004
45. Shadbolt, N., Berners-Lee, T., Hall, W.: The semantic web revisited. IEEE Intell. Syst. **21**(3), 96–101 (2006). https://doi.org/10.1109/MIS.2006.62
46. Sharma, V., Abel, J., Al-Hussein, M., Lennerts, K., Pfründer, U.: Simulation application for resource allocation in facility management processes in hospitals. Facilities **25**, 493–506 (10 2007). https://doi.org/10.1108/02632770710822599

47. Sieve, R., Kobialka, P., Slaughter, L., Schlatte, R., Johnsen, E.B., Tapia Tarifa, S.L.: BedreFlyt: Improving patient flows through hospital wards with digital twins. In: Chechik, M., Fedeli, A., Filippone, G., Formica, F., Frasheri, M., Hochgeschwender, N., Marsso, L. (eds.) Proc. 1st Intl. Workshop on Autonomous System Quality Assurance and Prediction with Digital Twins (ASQAP 2025). Electronic Proceedings in Theoretical Computer Science, vol. 418, pp. 1–15. Open Publishing Association (2025). https://doi.org/10.4204/EPTCS.418.1
48. Steward, J., Furlong, J., Stutz, R., Cox, R., Highley, J., Wang, H., Johnstone, C., McBride, P., Noto, J., Kelly, R., Nguyen, R., Wilson, J., Shaxted, M., Long, M., Torreria, A.V., Gary, S.: Unleashing the power of "what if": Cloud-enabled high performance computing workflows in digital twins for scenario exploration. In: Proc. Intl. Geoscience and Remote Sensing Symp. (IGARSS 2024). pp. 2315–2318. IEEE (2024). https://doi.org/10.1109/IGARSS53475.2024.10642587
49. Sun, W., Wang, P., Xu, N., Wang, G., Zhang, Y.: Dynamic digital twin and distributed incentives for resource allocation in aerial-assisted internet of vehicles. IEEE Internet Things J. **9**(8), 5839–5852 (2022). https://doi.org/10.1109/JIOT.2021.3058213
50. Taieb, C., Tlili, T., Nouaouri, I., Krichen, S.: Towards an efficient hospital allocation to patients with resource constraints. In: 10th International Conference on Control, Decision and Information Technologies (CoDIT 2024). pp. 2284–2289. IEEE (2024). https://doi.org/10.1109/CODIT62066.2024.10708224
51. Vallée, A.: Digital twin for healthcare systems. Frontiers in Digital Health **5**, 1253050 (Sep 2023). https://doi.org/10.3389/fdgth.2023.1253050
52. Weyns, D.: An Introduction to Self-Adaptive Systems: A Contemporary Software Engineering Perspective. Wiley-IEEE Computer Society Press (02 2021). https://doi.org/10.1002/9781119574910
53. Wienhöft, P., Suilen, M., Simão, T.D., Dubslaff, C., Baier, C., Jansen, N.: More for less: Safe policy improvement with stronger performance guarantees. In: Proc. 32nd International Joint Conference on Artificial Intelligence (IJCAI 2023). pp. 4406–4415. ijcai.org (2023). https://doi.org/10.24963/IJCAI.2023/490
54. Wilk, S., Kezadri-Hamiaz, M., Amyot, D., Michalowski, W., Kuziemsky, C., Catal, N., Rosu, D., Carrier, M., Giffen, R.: An ontology-driven framework to support the dynamic formation of an interdisciplinary healthcare team. Intl. Journal of Medical Informatics **136** (Apr 2020). https://doi.org/10.1016/j.ijmedinf.2020.104075
55. Willcox, K., Segundo, B.: The role of computational science in digital twins. Nat. Comput. Sci. **4**(3), 147–149 (2024). https://doi.org/10.1038/S43588-024-00609-4
56. Withanachchi, N., Uchida, Y., Nanayakkara, S., Samaranayake, D., Okitsu, A.: Resource allocation in public hospitals: Is it effective? Health policy **80**, 308–313 (2007). https://doi.org/10.1016/j.healthpol.2006.03.014
57. Xames, M.D., Topcu, T.G.: A systematic literature review of digital twin research for healthcare systems: Research trends, gaps, and realization challenges. IEEE Access **12**, 4099–4126 (2024). https://doi.org/10.1109/ACCESS.2023.3349379
58. Yu, W., Liu, Q., Zhao, G., Song, Y.: Exploring the effects of data-driven hospital operations on operational performance from the resource orchestration theory perspective. IEEE Trans. Engineering Management **70**(8), 2747–2759 (2023). https://doi.org/10.1109/TEM.2021.3098541
59. Zhu, T., Liao, P., Luo, L., Ye, H.Q.: Data-driven models for capacity allocation of inpatient beds in a chinese public hospital. Computational and Mathematical Methods in Medicine **2020**(1) (2020). https://doi.org/10.1155/2020/8740457