

Formal and computational modeling of anticipation mechanisms of resilience in the complex sociotechnical air transport system

Blok, Anne Nynke; Sharpanskykh, Alexei; Vert, Matthieu

DOI

[10.1186/s40294-018-0058-2](https://doi.org/10.1186/s40294-018-0058-2)

Publication date

2018

Document Version

Final published version

Published in

Complex Adaptive Systems Modeling

Citation (APA)

Blok, A. N., Sharpanskykh, A., & Vert, M. (2018). Formal and computational modeling of anticipation mechanisms of resilience in the complex sociotechnical air transport system. *Complex Adaptive Systems Modeling*, 6(1), Article 7. <https://doi.org/10.1186/s40294-018-0058-2>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

RESEARCH

Open Access



Formal and computational modeling of anticipation mechanisms of resilience in the complex sociotechnical air transport system

Anne-Nynke Blok^{1,2}, Alexei Sharpanskykh^{1*}  and Matthieu Vert¹

*Correspondence:

o.a.sharpanskykh@tudelft.nl

¹ Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, Delft, The Netherlands
Full list of author information is available at the end of the article

Abstract

With ever-growing numbers of passengers and complexity of the air transport system, it becomes more and more of a challenge to manage the system in an effective, safe, and resilient manner. This is especially evident when disruptions occur. Understanding and improving resilience of the air transport system and its adaptive capacity to disruptions is essential for the system's uninterrupted successful performance. Using theoretical findings from behavioral sciences, this paper makes the first steps towards formalization of the adaptive capacity of resilience of the air transport system with a particular focus on its ability to anticipate. To this end, an expressive logic-based language called Temporal Trace Language is used. The proposed approach is illustrated by a case study, in which anticipatory mechanisms are implemented in an agent-based airport terminal operations model, to deal with a disruptive scenario of unplanned and challenging passenger demand at the security checkpoint. Results showed that the timing of an adaptive action could have a significant influence on reducing the risk of saturation of the system, where saturation implies performance loss. Additionally, trade-off relations were obtained between cost, corresponding to the extra resources mobilized, and the benefits, such as a decrease in risk of saturation of the passenger queue.

Keywords: Resilience, Anticipation, Agent-based modeling and simulation, Air transport system

Introduction

The modern air transport system is a complex, highly dynamic, sociotechnical system with many diverse actors actively interacting with each other. With ever-growing numbers of passengers, as well as the increasing complexity and diversity of air transport operations, it becomes more and more of a challenge to manage the air transport system in an effective, safe, and resilient manner. This is especially evident when disruptions occur. Unexpected disruptions of air transport operations are often handled inefficiently, their effects persist for days, if not weeks (Delta Malfunction 2016; Boos 2017; Amsterdam Schiphol Airport 2017). Stranded passengers frequently complain about the lack of information and overall coordination of recovery processes.

Aviation stakeholders often put a high priority on efficiency goals, whereas resilience of operations is addressed to a lesser extent. One of the reasons for this is that resilience of air transport operations is still a largely unexplored performance area, in contrast, for example, to efficiency and safety. It is often not clear how to balance resilience with other performance areas of the air transport system, such as efficiency and safety. Current research in air transport resilience focuses largely on empirical case studies and qualitative descriptive theories (Hollnagel 2013). There is a clear need for conceptual frameworks and based on them formal, computational methods and tools for resilience modeling, analysis, improvement, and design, both in academic and industrial communities (Righi et al. 2015; Ouyang 2014).

Resilience is defined in this work following the definition by Hollnagel (2011): *“resilience is the intrinsic ability of a system to adjust its functioning prior to, during, or following changes and disturbances, so that it can sustain required operations under both expected and unexpected conditions.”* According to a known framework (Francis and Bekera 2014), resilience is characterized by (1) adaptive capacity—the ability of a system to adjust to existing or anticipated undesirable situations by undergoing some changes; (2) absorptive capacity—the degree to which a system can absorb the impacts of system perturbations; (3) restorative capacity characterized by rapidity of return to normal or improved operations. Adaptive capacity underlies both disruption absorption and recovery from performance degradation, and thus plays a central role in resilience. Adaptive capacity includes abilities to anticipate and recognize disruptive events, and re-organize to handle these events. Adaptive capacity is the least understood capacity of resilience. Theory of graceful extensibility is proposed by Woods (2018), which is a comprehensive conceptual theory defining foundational concepts of resilience and its adaptive capacity.

Using theoretical findings of Woods (2018) and Hollnagel (2011, 2013), as well as several theories from behavioral sciences, this paper makes the first steps towards formalization of the adaptive capacity of resilience of the air transport system with a particular focus on its ability to anticipate. Anticipation is a future-oriented action, decision, or behavior based on a prediction (Pezzulo et al. 2008). Anticipation was selected as focus for this research, because in air transportation the abilities to anticipate and prepare for disruptive events were identified as critical for minimizing the negative effects of disruptions (Day et al. 2010; Baylis et al. 2015; Delta Malfunction 2016). In the air transport system, handling of disruptions in general and anticipation in particular are performed by human operators (such as airport and airline managers, air traffic controllers, pilots) supported by technical tools (e.g., automated planning and scheduling systems, radars, security scanners). Thus, understanding mechanisms of human anticipation in the context of a sociotechnical system is important for modeling of the adaptive capacity of resilience of the air transport system.

To formally model the adaptive capacity of sociotechnical systems, approaches based on Complex Adaptive Systems paradigm and agent-based modeling and simulation as its prominent tool were previously identified as a promising direction (Righi et al. 2015; Ouyang 2014; Cook and Rivas 2016). This methodological direction is also taken in this work. An agent is an autonomous entity able to interact with the environment and to make decisions. For example, in an airport environment, both airport operators and passengers can be represented by agents. To formalize adaptive abilities of agents

an expressive formal language is needed, able to represent temporal, spatial, qualitative (logical), quantitative, as well as stochastic properties. Temporal Trace Language (TTL) (Sharpanskykh and Treur 2010) satisfies these requirements. In this paper it is demonstrated how to formalize a multiagent system with anticipatory abilities using TTL, and to apply it in the context of airport terminal operations.

System resilience is also often modeled and analyzed by using the complex network paradigm (Convertino et al. 2014; Hosseini et al. 2016; Ouyang 2014), which is another popular modeling tool of Complex Systems Theory.

In an airport terminal, the security checkpoint is regarded as the most significant bottleneck, especially under non-nominal conditions. In May 2017, for 2 weeks, line capacity at the security check at Schiphol Airport was exceeded due to the unexpectedly high number of Spring break passengers, which was not well anticipated. As a result, many passengers missed or delayed their flight (KLM Boos 2017), which caused high costs on the side of airlines. This research presents an illustrative case study, which implements anticipatory abilities in an agent-based airport terminal operations model, to deal with a disruptive scenario of unplanned and challenging passenger demand at the security checkpoint.

A simulation study was conducted to analyze the effects of anticipation and the corresponding adaptive actions. Results showed that the timing of an adaptive action could have a significant influence on reducing the risk of saturation of the system, where saturation implies performance loss. This indicates that the ability to act proactively may be essential to be resilient. Additionally, trade-off relations were obtained between cost, corresponding to the extra resources mobilized, and the benefits, such as a decrease in risk of saturation of the passenger queue.

The paper is organized as follows. In the next section a theoretical background used for modeling of mechanisms of anticipation is described. Then formal modeling of anticipation mechanisms in a multiagent system is considered. After that a case study on anticipation in airport security operations is discussed. The paper ends with conclusions and an outlook.

Anticipation in sociotechnical systems

Hollnagel identified four abilities necessary for resilient performance of sociotechnical systems (Hollnagel 2011). These are the ability to:

- Monitor: knowing what to look for, or being able to recognize what is or could seriously affect the system's performance in the near future. This includes monitoring own system performance as well as the environment.
- Anticipate: being able to make predictions to know what to expect, and being able to act on developments further into the future.
- Respond: being able to respond to regular and irregular changes, disturbances, and opportunities by activating prepared actions or by adjusting current mode of functioning.
- Learn: knowing what has happened and use this experience to be better prepared for disruptions in the future.

Note that to be able to anticipate, an actor also needs the abilities to monitor and to respond. Therefore, the four abilities identified above are not mutually exclusive.

A general definition of an anticipatory system is provided by Minch (1986): “*An anticipatory system is a system containing a predictive model of itself and/or its environment, which allows it to change state at an instant in accordance with the model’s predictions pertaining to a latter instant.*”

Hence, anticipation can be understood as a future-oriented action, decision, or behavior based on a prediction (Pezzulo et al. 2008). Predictions are usually triggered by cues or stimuli, which an anticipatory actor obtains from the environment. These cues are interpreted by the actor as precursors of possibly significant disruptions. In the context of resilience we are particularly interested in predicted disruptive states, which actors actively try to avoid (Damasio 1994). To identify such cues the actor needs to pay close attention to weak signals of malfunctioning, which often manifest themselves as discrepancies between actor’s expectations and actual observations or growing risks of system saturation (Sutcliffe 2001; Woods 2018).

An instrumental mechanism for generating predictions of future states based on observations is described by the theory of simulated behavior and perception chains proposed by Germund (2002). The simulation hypothesis underlying this theory postulates the presence of an associative mechanism that enables elicitation of sensory activity which resembles the activity that would normally be caused as a consequence of completed overt behavior. Hence, the association or prediction of the consequences of certain behavior or action is solemnly based on the preparatory stages of an action. Essentially, such perceptual simulation can be executed in multiple loops forming simulated chains of behavior containing information of possible responses and the corresponding expected consequences:

$$s1 \rightarrow r1 \rightarrow s2 \rightarrow r2 \rightarrow \dots$$

Here some situation elicits activation of $s1$ in the sensory cortex that leads to preparation for response $r1$. Then associations are used such that $r1$ will generate $s2$, which is the most connected sensory consequence of the response for which $r1$ was generated. This sensory state may serve as a stimulus for a new response, and so on, until, for example, an undesired state is reached.

To avoid undesired states, humans identify applicable action options or plans using learned action-effect relations (Hoffmann 2008). In the context of resilience these action options often involve mobilization of additional resources. To evaluate the consequences of actions options, humans use simulated behavioral chains without the need to actually execute them. Based on this valuation a decision can be made on what action or plan to execute to achieve some goal, such as avoiding an undesired state.

In psychological research also other types of anticipatory mechanisms are considered (Butz et al. 2003). In the following, three different types will be reviewed: implicit anticipation, payoff anticipation, and sensory anticipation. The first type—implicit anticipation—does not involve any form of prediction. Observations and internal states of actors are directly mapped to actions. Usually these mappings are realized by natural evolutionary processes, e.g., in animals. This form of anticipation is not appropriate for the air transport system, in which predictions play an important role. In the second type of

anticipation, predictions concern only future payoffs, and not future states. This form of anticipation is too limited for our purposes, as decisions in the air transport system are often based on prediction of future states (e.g., the accumulation of passengers in a queue over time). The third type is sensory anticipation, in which future predictions influence sensory processing of information. Through sensory anticipation the actor becomes more susceptible to particular types of sensory input, in such a way expectations influence actor's attention and sensory information processing. However, sensory anticipation does not directly influence the actor's behavior. Therefore, this form of anticipation is less suitable for the air transport system.

Implementations of anticipatory intelligent agents for controlling technical systems are often based on simplified principles of anticipation, not taking into account the theoretical basis from Behavioral Sciences. However, these principles are usually in line with behavioral theories of anticipation. For example, Abras et al. (2008) proposed an anticipation mechanism for power management in a smart home based on prediction of future demand and planning based on Tabu search.

In the following section it is demonstrated how the identified principles and theories related to anticipation can be formalized for a multiagent system using the agent-based modeling paradigm.

Formal modeling of anticipation in a multiagent system

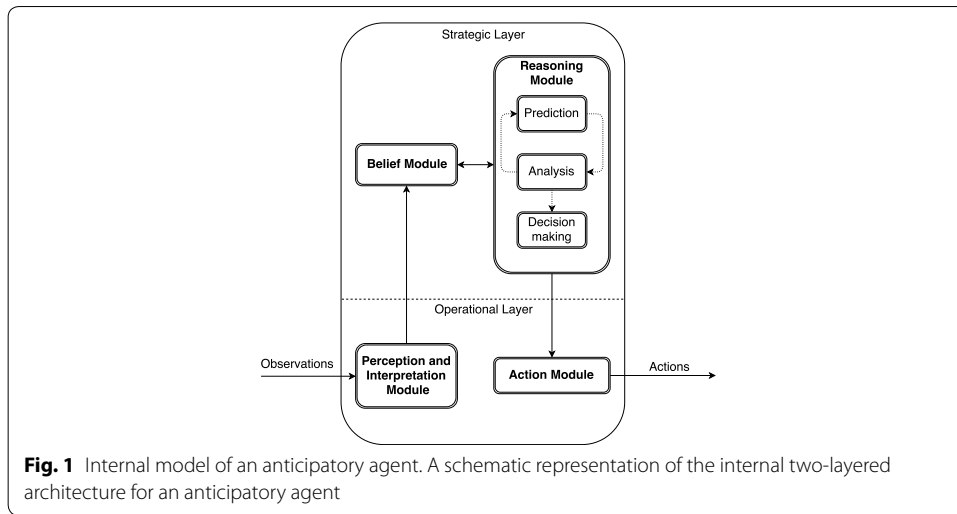
In this section first a cognitive architecture for an anticipatory agent is introduced. After that the modeling language chosen for formalization is described. Finally, a formal model of anticipation is provided.

A cognitive architecture for an anticipatory agent

To model a sociotechnical system, an agent-based modeling and simulation (ABMS) approach based on Complex Adaptive Systems paradigm is often used (Van Dam et al. 2012; Mitleton-Kelly 2013). An agent is an autonomous entity able to interact with the environment and to make decisions. Both technical systems and human actors can be represented by agents. Global systemic properties of a multiagent system, e.g., systemic performance indicators or safety risks, emerge in a bottom-up way, from many distributed interactions between agents and their local properties. Furthermore, an agent-based modeling approach allows representing cognitive processes, which shape intelligent behavior of agents. In particular, processes of reasoning, prediction, simulated behavior and perception chains, and decision making considered in the previous section as components of anticipation, can be represented by a cognitive agent model.

A cognitive model for an anticipatory agent was developed following a two-layered architecture based on the work of Blumberg and Galyean (1995) and Hoogendoorn and Bovy (2004) (see Fig. 1). The architecture comprises the modules that belong to the *operational layer* and *strategic layer*, which are described in the following:

- Perception and interpretation module: This module ensures that the agent is able to observe the environment. Based on these observations the belief module is updated, this way the agent is aware of the environment, potential changes in the environment, and signals of malfunctioning that trigger anticipation.



- **Belief module:** The belief module contains the knowledge of an agent. This can be in the form of facts, rules and information about (causal) relations. This knowledge is constantly updated by input from observations, resulting in awareness of the environment. Consequently, an update of the belief module can result in activation of the reasoning module. The belief module is also updated based on output from the reasoning module such as, future state predictions, outcomes of analysis and outcomes of decision-making.
- **Reasoning module:** The reasoning module, which is fed by the belief module can be explained as a machinery including the processes of prediction, analysis, and decision making. The interaction between the belief module and the reasoning module represents the prediction model of the anticipatory agent. Following Hesslow’s simulation theory (Germund 2002), the prediction model generates internally simulated (chains of) interactions with the environment and evaluates these simulated interactions. Additionally, the results of action option valuation, performed by the analysis module, are fed to the decision module to come to a decision on what action to execute.
- **Action module:** The action module is responsible for the preparation of the action to be executed and the actual execution. Execution of an action is modeled as an interaction with the environment.

To illustrate the different steps of the cognitive anticipation model as presented in Fig. 1, consider the example of a passenger agent navigating its way through an airport. The main goal of the passenger is to go to the gate corresponding to its flight and arrive there on time. When the passenger enters the airport, it perceives the environment and other passengers around it. The passenger observes long queues at the security check. This observation is interpreted and the passenger’s belief base is updated, hereby the agent becomes aware of the situation, which activates the reasoning module. As part of reasoning the agent predicts what the impact of the long queues might be on its planning to arrive at the gate on time. The agent makes two predictions: one about the impact on its planning considering the option of first going to the restroom and shop before

entering the queue, and the second prediction concerns the option where the agent would immediately enter the security queue. The prediction outcomes are analyzed and based on valuation the passenger agent decides on whether it will first execute the other activities before entering the queue, or not. Based on this decision the corresponding actions are executed.

In the following section a language is introduced used to formalize an anticipatory agent-based model.

Formal modeling language

To formally model the internal dynamics of an anticipatory agent and its interaction with other agents and the stochastic environment, a suitable formal modeling language is required. The desiderata for such a language include the following abilities: (1) to represent and reason about discrete and continuous time; (2) to specify quantitative and qualitative aspects and relations; (3) to specify stochastic processes; (4) to represent cognitive and behavioral properties of agents and interaction between agents; (5) to represent the system at different aggregation levels and to establish relations between the levels.

TTL has proven to be a suitable formal language for the specification of agent-based models of complex sociotechnical systems (Bosse et al. 2009; Sharpanskykh and Stroeve 2011; Sharpanskykh 2011). TTL is a variant of an order-sorted predicate logic (Manzano and Manzano 1996). Whereas standard multi-sorted predicate logic is meant to represent static properties, TTL is an extension of such language with explicit facilities to represent dynamic properties of systems. TTL has been applied in diverse domains, including air transport.

In contrast to many other logic-based languages for multiagent systems, for example based on modal logic (Dastani et al. 2010), TTL allows expressing quantitative relations between variables. In particular, it allows specifying arithmetic operations, differential and difference equations. This ability is essential for specifying quantitative cognitive and social models, for example based on theories from Neuroscience (Treur 2016), which were also used for defining the concept of anticipation in this paper. Furthermore, quantitative relations are needed to describe quantitative prediction models, as also demonstrated in this paper.

At the same time, quantitative mathematical languages based on differential equations such as Dynamical Systems Theory (Michel et al. 2001) are not capable of expressing the dynamics of processes with a qualitative character, such as agent reasoning, some forms of agent decision making, normative and organizational processes. Modeling of human anticipation involves reasoning, and thus, the language should be able to express qualitative (logical) aspects. For analysis of multiagent systems, the ability to express logical relations between parts of the system considered at different aggregation levels is important. Mathematical languages based on Dynamical Systems Theory do not provide this possibility. TTL is a hybrid language, which allows expressing both quantitative and qualitative aspects in an integrated formal agent-based model specification.

Furthermore, the air transport system is characterized by many diverse uncertainties. A specification language for the air transport system models is required to capture these non-deterministic aspects. There are few logic-based languages that allow

expressing uncertainties quantitatively (e.g., Richardson and Domingos 2006; Jamroga 2008). In TTL one can express random state variables, probabilistic transitions between states, and random delays between states. This expressivity allows to capture all types of uncertainties in the air transport system required for modeling of anticipation as considered in this paper.

An other advantage of TTL is that agent-based model specifications in TTL can be analyzed by dedicated automated tools, in particular for simulation and verification. These tools allow studying emergent, system-wide properties, relations between aggregation levels in the model specification, checking desirable and undesirable (safety) properties, analyzing by exploration ‘what-if’ scenarios. These types of analysis are highly suitable for studying anticipation in complex sociotechnical systems.

To specify state properties for system components in TTL, ontologies are used which are specified by a number of sorts, sorted constants, variables, functions and predicates. State properties are specified based on such ontology using a standard multi-sorted first-order predicate language. For every agent a number of ontologies can be distinguished used to specify state properties of different types. That is, the ontologies $IntOnt(A)$, $InOnt(A)$, $OutOnt(A)$, and $ExtOnt(A)$ are used to express respectively internal, input, output, and external state properties of agent A . Input ontologies contain elements for describing perceptions of an agent from the external world (e.g., $observed(a)$ means that a component has an observation of state property a), whereas output ontologies describe actions and communications of agents (e.g., $act(b)$ represents action b performed by an agent in its environment).

To express dynamic properties, TTL includes special sorts: $TIME$ (a set of linearly ordered time points), $STATE$ (a set of all state names of a system), $TRACE$ (a set of all trace names; a trace or a trajectory can be thought of as a timeline with a state for each time point), $STATPROP$ (a set of all state property names), $ASPECT$ (a set of the aspects (input, output, internal) of the agents), and $VALUE$ (an ordered set of numbers).

A state is described by a function symbol $state : TRACE \times TIME \times ASPECT \rightarrow STATE$. A trace is a temporally ordered sequence of states. A time frame is assumed to be fixed, linearly ordered, for example, the natural or real numbers.

States are related to state properties via the satisfaction relation denoted by the prefix predicate $holds$ (or by the infix predicate \models): $holds(state(\gamma, t, a), p)$ (or $state(\gamma, t, a) \models p$), which denotes that state property p holds in trace γ at time point t at agent’s aspect a .

Transition relations between states are described by dynamic properties, which are expressed by TTL-formulae. The set of *atomic TTL-formulae* is defined as:

1. If v_1 is a term of sort $STATE$, and u_1 is a term of the sort $STATPROP$, then $holds(v_1, u_1)$ is an atomic TTL formula.
2. If τ_1, τ_2 are terms of any TTL sort, then $\tau_1 = \tau_2$ is a TTL-atom.
3. If t_1, t_2 are terms of sort $TIME$, then $t_1 < t_2$ is a TTL-atom.
4. If v_1, v_2 are terms of sort $VALUE$, then $v_1 < v_2$ is a TTL-atom.

The set of well-formed TTL-formulae is defined inductively in a standard way using Boolean connectives and quantifiers over variables of TTL sorts. An example of the TTL formula, which describes observational belief creation of an agent, is given below:

In any trace, if at any point in time $t1$ agent A observes a queue at the security checkpoint, then there exists a point in time $t2$ after $t1$ such that at $t2$ in the trace the agent A believes that there is a queue at the security checkpoint.

$$\forall \gamma \forall t1 [\text{holds}(\text{state}(\gamma, t1, \text{input}(a)), \text{observed}(\text{queue_at_checkpoint})) \Rightarrow \\ \exists t2 t2 > t1 \text{ holds}(\text{state}(\gamma, t2, \text{int}(a)), \text{belief}(\text{queue_at_checkpoint}))]$$

The possibility to specify arithmetical operations in TTL allows modeling of continuous systems, which behavior is usually described by differential equations. Such systems can be expressed in TTL either using discrete or dense time frames.

TTL specifications can be verified automatically by using TTL checker tool Bosse et al. (2009).

For a more detailed description of the syntax and semantics of TTL please refer to Sharpanskykh and Treur (2010).

Formal modeling of anticipation

In accordance with the theoretical background described above, the process of anticipation can be formalized. Based on this formalization an anticipatory agent can be modeled. The formalization is structured in the following steps: (1) observation of cues that triggers prediction of reaching an undesired state, (2) generation of applicable action options to avoid the undesirable state, (3) generation of action options consequences, (4) valuation of the consequences of the action options, (5) making a decision on which action option to execute.

Step 1: Observation of cues that trigger the prediction on whether an undesirable state will be reached in the future. In case an undesirable state is predicted, the agent will generate the goal to avoid this state. The prediction is done by using Hesselow's simulated behavior and perception chains (Germund 2002) represented in TTL by functions $\text{leadsto}(s : \text{SENSORY_STATE}, r : \text{RESPONSE_STATE})$ and $\text{leadsto}(r : \text{RESPONSE_STATE}, s : \text{SENSORY_STATE})$:

$$\forall \gamma, t, q : \text{CUE} [\text{holds}(\text{state}(\gamma, t, \text{input}(a)), \text{observed}(q)) \Rightarrow \\ \text{holds}(\text{state}(\gamma, t + \delta, \text{int}(a)), \text{sensory_state}(q))]$$

$$\forall \gamma, t, s : \text{SENSORY_STATE}, r : \text{RESPONSE_STATE} \\ [\text{holds}(\text{state}(\gamma, t, \text{int}(a)), \text{sensory_state}(s) \wedge \text{leadsto}(s, r)) \Rightarrow \\ \text{holds}(\text{state}(\gamma, t + \delta, \text{int}(a)), \text{response_state}(r))]$$

$$\forall \gamma, t, s : \text{SENSORY_STATE}, r : \text{RESPONSE_STATE} \\ [\text{holds}(\text{state}(\gamma, t, \text{int}(a)), \text{response_state}(r) \wedge \text{leadsto}(r, s)) \Rightarrow \\ \text{holds}(\text{state}(\gamma, t + \delta, \text{int}(a)), \text{sensory_state}(s))]$$

$$\forall \gamma, t, r : \text{RESPONSE_STATE} [\text{holds}(\text{state}(\gamma, t, \text{int}(a)), \text{response_state}(r) \wedge \text{undesired}(r)) \Rightarrow \\ \text{holds}(\text{state}(\gamma, t + \delta, \text{int}(a)), \text{goal}(\text{avoid_state}(r)))]$$

Step 2: Generation of applicable action options to avoid the undesirable state:

$$\begin{aligned} & \forall \gamma, g : GOAL, o : ACTION_OPTION \\ & [holds(state(\gamma, t, int(a)), goal(g) \wedge action_option_for_goal(o, g)) \Rightarrow \\ & holds(state(\gamma, t + \delta, int(a)), action_option(o))] \end{aligned}$$

Note that if direct relations between the goal and applicable action options are not known in advance, these relations could be identified by backward chaining of action–effect relations, as discussed in Davidsson (2003), Hoffmann (2008).

Step 3: Generation of the action option consequences by forward chaining based on simulated behavior and perception chains, specified in the same way as in step 2 using functions $leadsto(s : SENSORY_STATE, r : RESPONSE_STATE)$ and $leadsto(r : RESPONSE_STATE, s : SENSORY_STATE)$.

Step 4: Valuation of the identified action options. Each simulated behavior and perception chain corresponding to each action option obtained in step 3 is valued based on the utilities or values of the states from this chain, which reflect the degree of desirability of the option. The precise form of the value integration function $aggr_values$ depends on the system and application under consideration.

$$\begin{aligned} & \forall \gamma, t, o : ACTION_OPTION \\ & [holds(state(\gamma, t, int(a)), sensory_state_for(s1, o) \wedge \dots \wedge sensory_state_for(sN, o) \wedge \\ & response_state_for(r1, o) \wedge \dots \wedge response_state_for(rN, o) \wedge \\ & val(s1, v1) \wedge \dots \wedge val(sN, vN) \wedge val(r1, w1) \wedge \dots \wedge val(rN, wN)) \Rightarrow \\ & holds(state(\gamma, t + \delta, int(a)), value_option(o, aggr_values(v1, \dots, vN, w1, \dots, wN)))] \end{aligned}$$

Step 5: Choice of the action option to be executed based on the values of the options. The choice function again depends on the application. Usually the option with the maximum value is chosen. However, when decision making is combined with learning, action options with lower values may be chosen to perform exploration (cf. reinforcement learning).

In the next section it is illustrated how the formalized generic mechanism of anticipation was applied in the case study in which operations at airport security checkpoint are modeled.

Case-study: Anticipation in airport security operations

In airport terminals, the security checkpoint is the most narrow performance bottleneck, especially when flight planning disruptions occur, such as cancellations or delays. The security checkpoint is a complex sociotechnical system, as it is composed of numerous and heterogeneous human and technical system components interacting with each other, among which are passengers, security operators, orchestration agent, X-ray detectors, walk through metal detectors.

Scope and objective

For this case study, real data from a regional airport were used to model operations between 5:00 A.M. and 10:00 A.M. on a regular day. For this time span, the resources at the security checkpoint were planned based on the flight departure schedule, the expected number of passengers corresponding to these flights, and the expected arrival times of these passengers. In the simulated disrupted scenario, two delayed flights were introduced which resulted in a changed departure schedule. In reaction to the disruption, passengers from these two flights adjusted their arrival time at the security checkpoint. Consequently, the arrival pattern of passengers originally forecasted was disrupted, and the planned resources at the security checkpoint did not match the changed demand. In this case study, it was investigated by agent-based modeling and simulation how anticipation mechanisms could contribute to the ability to maintain a stable performance of a security checkpoint, under the specified disrupted conditions.

The computational simulations of this case study were executed by using an agent-based airport terminal operations simulator called AATOM (Janssen et al. 2018). This simulator has been designed to specifically conduct research on airport terminal operations using the agent-based modeling paradigm. AATOM is specified in an executable sublanguage of TTL. AATOM introduces a specific ontology in TTL for specifying agents, their states, the environment, and interaction between the agents and the environment. Each dynamic property in AATOM is specified by a direct causal temporal relation between two states. Such a model specification can be implemented in most existing programming languages, which support state transition systems representations. Specifically, AATOM was implemented in Java.

Multiagent system model specification

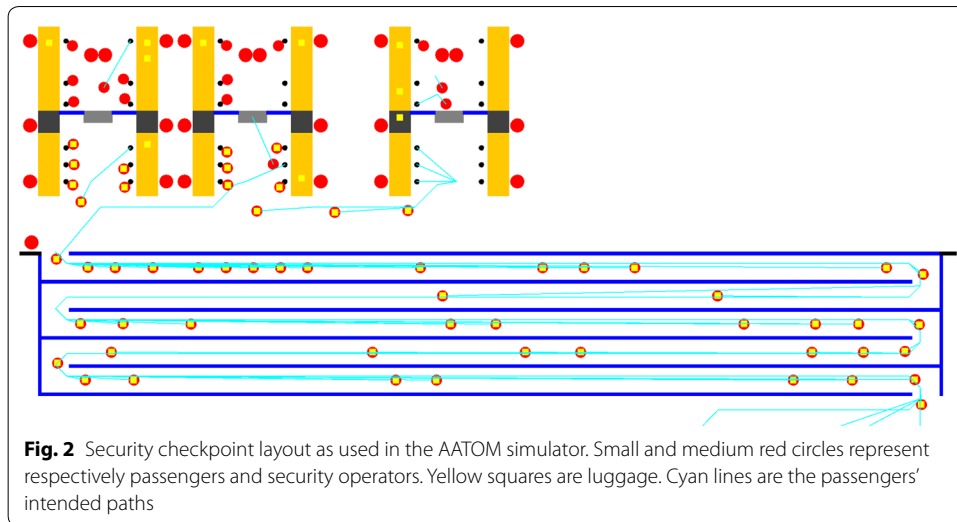
The multiagent system model specification consists of an environment specification, a specification of the passengers and operator agents and their interactions, and a specification of the anticipatory agent.

Environment specification

The areas modeled are an entrance area, a checkpoint area, and a queuing area of the security checkpoint. The queuing area consists of one queuing lane and provides access to the security checkpoint from the entrance area. The security checkpoint area is populated with operator agents and equipped with security sensors. It comprises multiple security lanes. Physical objects in the queuing area include walls, queue separators, belts, and security checkpoint sensors. Also luggage is modeled as a physical object. The lane setup implemented in the model is presented in Fig. 2.

To describe the state of the environment the following functions are used:

- *demandValue*(d) describes the demand posed on the security checkpoint. Here, $d \in \mathbb{N}$, the demand value expresses the number of passengers in the queue at the security checkpoint at a certain point in time.
- *flightSchedule*(f) describes the flight schedule. Here, $f \in F = \{F_1, F_2\}$ represents the flight schedule that holds at some time point. The case study scenario defines two



flight schedules, the original flights schedule F_1 and the disrupted flight schedule F_2 , with two delayed flights. The flight schedule is represented by a matrix including for each flight the flight number, the departure time, and the number of passengers corresponding to the flight.

$$F_i = \begin{bmatrix} flightNumber_1 & departureTime_1 & numberOfPax_1 \\ \vdots & \vdots & \vdots \\ flightNumber_j & departureTime_j & numberOfPax_j \end{bmatrix}$$

- Here, $i = 1, 2$ and $j = 7$ the total number of flights included in the flight schedule.
- $lanesOpen(y)$ describes the number of security lanes opened. Here, $y \in Y = \{1, 2, 3, 4, 5\}$ is the number of lanes opened at some time point. This number is constrained by the total number of security lanes available at the airport.

The initial state of the environment is specified by

holds(demandValue(0))
holds(flightSchedule(F_1))
holds(lanesOpen(1))

Specification of passenger and operator agents

After arriving in the entrance area, all passengers aim to pass through security, and reach their flight on time. Each passenger is modeled with one luggage item, which is to be checked at the security checkpoint. Both passengers and security operator agents perform several activities.

The activities of the passengers are:

- Queuing activity: it is executed when other activities cannot be executed due to capacity limits of the activity area. The activity is in general performed in a queue area, but is not limited to this area type. The activity is modeled as a waiting period

that ends when a passenger in front moves, or the planned next activity has available space again.

- Checkpoint activity: it is mandatory for passengers and is executed at the checkpoint area. The checkpoint activity consists of several actions, of which some are mandatory and some are optional. The mandatory actions are: travel document check, luggage drop, and luggage collect. The optional actions are: physical check and Explosives Trace Detector (ETD) check. The optional actions are executed depending on whether a passenger is identified as a potential threat by the walk through metal detector (WTMD) sensor, or if the passenger is selected for an ETD check. Each of these actions can be preceded by an optional queuing activity.

The security operator activities are:

- Travel document check activity: this activity is modeled as an interaction with a passenger, where the passenger is ordered to wait for a period of time. The activity starts when a passenger is at the corresponding activity area.
- Luggage drop activity: modeled in the same way as Travel Document Check Activity.
- X-ray activity: luggage is checked by the X-ray sensor. The X-ray activity starts when the security operator observes the X-ray sensor's binary output. If the output is true (i.e., a forbidden object is detected), the X-ray activity includes an interaction with the operator for luggage checking. If the output is false, the activity is completed without any further action. The luggage will proceed to the luggage collect area.
- Luggage check activity: if the security operator performing the luggage check activity observes the search communication from the operator performing the X-ray activity, the luggage check activity is performed.
- Physical check activity: the security operator performing the physical checking activity is responsible for detecting illegal items on the body of a passenger. It performs a check based on observations of the WTMD, the check is initiated when the WTMD sensor output value is 1, which indicates metal is detected on the passenger.
- ETD check activity: the security operator performing the ETD checking activity is responsible for detecting explosive traces on a passenger. The check is based on indications of the WTMD, and is initiated when the WTMD sensor output value is 2. This is a random output with probability $\frac{1}{10}$, and is used to ensure that one in every ten passengers is randomly checked for traces of explosives.

Specification of interactions of passengers and operator agents

Three types of interactions are defined in the model. The first type is the interaction between agents and the environment. In particular, interactions are specified between security operators and the security sensors that reset the sensor state to idle after observations were made. The second type are interactions between an operator agent and a passenger. For most processes, a waiting period is communicated by the operator agent to a passenger. For instance, when a security operator performs a luggage check activity or an ETD check activity, the passenger is ordered to wait for a specific period of time. Finally, also interactions exist between operators. For instance if the security operator

who performs the X-ray activity observes a suspicious item in the luggage, it is communicated to the security operator who performs the luggage check activity that the luggage has to be checked.

Specification of the anticipatory agent

The internal architecture of the anticipatory agent is presented in Fig. 1 and described in “A cognitive architecture for an anticipatory agent” section.

The anticipatory agent is able to observe the following environmental aspects: the flight schedule, the number of security lanes opened, and the number of passengers in the security checkpoint queue. Furthermore, the anticipatory agent is able to execute the following actions in the environment: opening or closing n security lanes, formalized as *openLane*(n) and *closeLane*(n).

It is assumed that the anticipatory agent is constantly aware of the environmental state through observation and updating its belief module. The anticipation process of the anticipatory agent is triggered by the observation cue *observed*(*flightSchedule*(F_2)) at $t = 5 : 10$ A.M. indicating the flight delays.

In step 1 of the anticipation process, the agent makes a prediction for the time span from 5:10 A.M. until 10:00 A.M. on whether the disrupted flight schedule will result in a saturated state and in what severity. It is based on its observations, which are the flight schedule, the number of security lanes opened, and the number of passengers in the security checkpoint queue. Specifically, the agent determines whether the queue length at the security checkpoint is above an acceptable threshold value defined based on the IATA waiting standard corresponding to a reasonable level of service. This standard specifies a maximum of 10 min waiting time for Economy Class at a security checkpoint (Stewart Head 2014).

In step 2, based on these predictions, a set of action options is generated that might overcome the negative effects of this disruption, as described below.

If a saturated state of the system is expected to be reached based on the predictions from step 1, then before it a phase transition should occur from a phase of normal performance, in which the security checkpoint performance is within the defined acceptable performance boundaries, to a phase of congestion, where the system performance is decreasing below the defined performance threshold. In this work the phase transition is characterized by an uninterrupted positive accumulation rate of passengers in the queue, resulting in a saturated state. The period of time over which this accumulation occurs is defined as the *pre-saturation phase*. Thus, the lower boundary of the pre-saturation phase corresponds to the moment in time when the accumulation rate of passengers is equal to zero. The upper boundary of the pre-saturation phase corresponds to the moment in time when the number of passengers in the queue exceeds the pre-defined threshold value, and the system reaches a saturated state. The period of time over which the system is in a saturated state is defined as the *saturation phase*.

A heuristic optimization algorithm was developed that seeks, based on the result of the initial prediction, for action options with an execution time t_{ex} restricted to the pre-saturation phase that minimizes the probability of saturation by activating more resources (security lanes). An action option is defined by three variables:

- i : the number of extra security lanes to open.
- t_{ex} : the moment in time to open the extra lane(s).
- ΔT : the time duration of opening the extra lane(s).

Probability of saturation was understood as the probability of the queue length exceeding the threshold value. The amount of resources available was constrained by the number of security lanes present at the airport. The algorithm is described in more detail in [Appendix](#).

After that, in step 3 for each of the action options, predictions were made of the action consequences for future states of the system. All predictions were computed by forward simulation of the agent-based model, while the considered conditions of the action option evaluated were applied. Predictions were based on $N = 100$ simulations of each scenario. A prediction was determined as the mean value of the number of passengers in the security checkpoint queue over time.

In step 4, the action options were valued using the results of the predictions obtained in step 3 by using a set of the following performance indicators:

- Probability of saturation: the probability of saturation is estimated as:

$$p(\text{sat}) = \frac{1}{N} \times \text{card}(M) \tag{1}$$

Here, M denotes the set of simulations where the system reaches a saturated state.

The next measures are proposed to analyze the severeness of the saturation, if the system reaches a saturated state.

- Average duration of saturation: the average duration of saturation provides insight in how long the system is saturated for a specific scenario. This is calculated as follows:

$$\bar{T}_{\text{sat}} = \frac{\sum_{m \in M} T_{\text{sat}}(m)}{\text{card}(M)} \tag{2}$$

For each simulation $m \in M$ which reaches a saturated state, the total time the system is saturated is measured as $T_{\text{sat}}(m)$. This time is summed for M and divided by the total number of saturated simulations $\text{card}(M)$.

- Average waiting time: the average waiting time is calculated based on the average saturation value, the number of passengers in the queue that exceeds the threshold value, and average processing rate of the security checkpoint at the moment of saturation. The average saturation value is calculated as follows:

$$\bar{V}_{\text{sat}} = \frac{\sum_{m \in M} \bar{V}_{\text{sat}}(m)}{\text{card}(M)} \tag{3}$$

With, $\bar{V}_{\text{sat}}(m)$ the average normalized saturation value over time for simulation $m \in M$. The saturation value is normalized by the number of lanes opened over time. The average waiting time is calculated by:

$$\bar{W}_{\text{sat}} = \bar{V}_{\text{sat}} \times r(1) \tag{4}$$

$r(1)$ denotes the average processing rate if one security lane is opened. This rate is applied as \bar{V}_{sat} is calculated based on normalized saturation values over time.

- Average number of missed flights: the average number of missed flights is a global system characteristic and provides an extra measure to indicate whether the waiting times at the security checkpoint are within reasonable limits. For each simulation $n = 1 : N$, with N the total number of simulations, the number of passengers that miss their flight is counted. The total number of passengers that miss their flight per simulation is expressed by MF_n , with n indicating the n^{th} simulation. Then for each scenario, the average number of missed flights is calculated as follows:

$$\overline{MF} = \frac{1}{N} \times \sum_{n=1}^N MF_n \tag{5}$$

- Costs: costs are expressed by the number of security lanes opened and the time duration of opening extra security lanes.

In the final step 5 an action option for execution is chosen by considering a trade-off between the cost of extra resources associated with the option, and the probability and severity of saturation acceptable for the agent.

Model initialization and calibration

The initialization of the model was done based on data from a regional airport.

Flight schedule The morning schedule of October 5th 2017 was modeled, which is the busiest time of the day at this airport, and thus the most sensitive to disruptions. The flight schedule is presented in Table 1. The number of passengers was calculated based on the capacity of the aircraft and a load factor of 0.9 which is used by the airport for their passenger forecast.

Arrival times The arrival times of passengers at the security checkpoint are determined based on the forecast method of the airport. The forecast method calculates the number of passengers expected to arrive at the airport based on the departure times of the flights and the number of passengers expected for a specific flight. The number of expected passengers for a specific flight is based on historical data for that flight and integrated in the forecasting method with a load factor. Finally, per departing flight it is assumed that passengers arrive following an arrival time distribution. The distribution

Table 1 The morning flight schedule at a regional airport on October 5th 2017

Flight	Destination	Departure time	Pax
HV6035	Rome-Fiumicino	07:00	173
HV6301	Gran Canaria	07:00	134
HV6493	Venetie	07:00	173
BA4450	London City	07:00	89
HV6771	Budapest	07:30	173
HV5021	Malaga	08:00	134
BA4452	London City	10:00	89

Table 2 The arrival time distribution for passengers at the security checkpoint, based on the forecast method of the airport

Time before departure	2–1.5 h	1.5–1 h	1–30 min	30–10 min
Percentage of passengers (%)	10	40	40	10

Table 3 The arrival rate λ determined for each time span of half an hour over a time period of 05:00 A.M. until 10:00 A.M.

Time	05:00	05:30	06:00	06:30	07:00	07:30	08:00	08:30	09:00	09:30
λ (pax/s)	0.032	0.136	0.172	0.100	0.039	0.007	0.005	0.020	0.020	0.005

Table 4 The processing time distributions for each of the activities performed at the security checkpoint

Activity	Processing time (s)
Luggage drop	$N(54.60, 36.09)$
Physical check	$N(43.00, 20.96)$
ETD check	$N(34.80, 15.17)$
Luggage collect	$N(71.50, 54.95)$

is presented in Table 2. Note that this forecast method is flight schedule dependent and thus daily, weekly and seasonal variations are reflected in the flight schedule. However, the possibility disruptions might occur that either influence the flight schedule or the arrival of passengers is currently not considered by the airport in this forecast method.

The passengers arrival time is distributed following an exponential distribution $Exp(\lambda)$ as the arrivals are assumed to be independent from each other with a constant average rate. Consequently, the process modeled here is a Poisson process. The Poisson process is known to model queues, and among others, the queues existing in airports (Rue and Rosenshine 1985). In the simulator, this is done by drawing inter arrival times from the exponential distribution. Here λ , the arrival rate, varies over time as it depends on the amount of passengers that correspond to a flight and are to arrive within a specific time span, which in turn depends on the arrival time distribution presented in Table 3. The arrival rate is calculated for each half hour based on the information provided in Tables 1 and 2, following $\lambda_{\Delta T} = \frac{pax_{\Delta T}}{1800s}$. Here, ΔT denotes a specific time span of a half hour, and $\#pax_{\Delta T}$ denotes the number of passengers expected within this time span. Table 3 presents the arrival rate λ calculated for each half hour, corresponding to the flight schedule presented in Table 1.

Processing times The processing times of each of the activities at the security checkpoint were based on empirical data gathered at the airport via video analysis of the security operations. Based on this data it is determined that the processing times follow Gaussian distributions. The average and the standard deviation for the processing time of each activity are presented in Table 4. Furthermore, the probability that the WTMD initiates an ETD check was set by the airport at $p(\text{ETD check}) = 0.1$.

Table 5 Number of security lanes in use over time corresponding to the flight schedule presented in Table 1 (07:00–10:00)

Time	05:00	05:30	06:00	06:30	07:00	07:30	08:00	08:30	09:00	09:30
# of lanes	2	3	4	3	2	1	1	1	1	1

Table 6 An overview of the delayed flights

Flight	Original DT	Pax	ToA	Delay (min)	New DT
HV6035	07:00	173	0 5:10	45	07:45
HV6301	07:00	134	0 5:10	60	08:00

DT departure time, ToA time of announcement of the delay

Based on the empirical data, the probability the WTMD initiates a physical check was $p(\text{Physical check}) = 0.1$. Finally, the recovery time of the X-ray scanner was set at 3 s.

Scheduled resources At the airport, the required resources at the security checkpoint are scheduled based on a method that calculates the number of security lanes required over time. As input the scheduling method uses the number of passengers expected over time in time steps of half an hour, and the processing rate of the security lanes. The airport assumes a processing rate of 2.6 pax/min per security lane. This scheduling method is applied to obtain a resource schedule for the considered morning, which is presented in Table 5. In total five security lanes are present at the airport. Additionally, one security lane requires 6 security operators, and two security lanes requires 11 security operators.

Delayed flights Two delayed flights were introduced in the disrupted scenario, as presented in Table 6. With the shift in departure time of these flights, it is assumed that the arrival times of passengers would shift as well. As no data was available to support an estimation of how the arrival times of passengers would shift, the assumption was made that passengers that did not arrive at the airport at the time of announcement would follow the arrival distribution as presented in Table 2, assuming the new departure time. The delay scenario presented in Table 6 was selected because it resulted in a significant decrease of system performance, whereby the system was in a saturated state for a relatively long period of time. As a consequence, a relatively high number of passengers might miss or delay their flight.

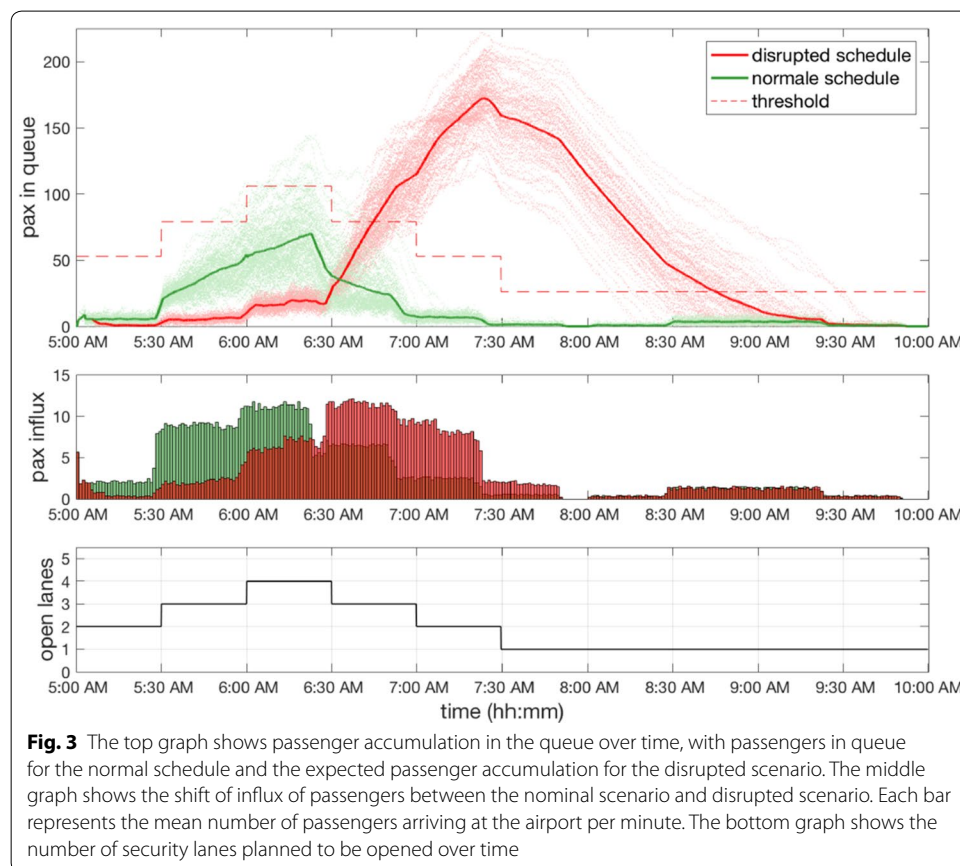
Model calibration

Model calibration was performed to ensure that the processing rate of the security checkpoint of the model fitted the actual processing rate assumed by the airport. The processing rate assumed was 2.6 pax/min per security lane. To reach this rate, the input parameters of processing times, as presented in Table 4, were multiplied by a calibration constant 0.655. The resulting processing rates for different security lane configurations are presented in Table 7.

The actual generated number of passengers was verified. The total number of passengers over all flights was 965. The mean number of passengers generated by the simulator after calibration was 968, with a standard deviation of 32, which were considered acceptable values.

Table 7 The calibrated processing rates $r(n)$ obtained for different security lane configurations, with n the number of open security lanes

n (# lanes)	$r(n)$ (pax/min)	
	μ	σ
1	2.71	0.14
2	5.20	0.22
3	7.91	0.35
4	10.40	0.44
5	13.11	0.54



Simulation results and analysis

Initial prediction and option generation The initial prediction represents the expected effects of the disrupted flight schedule, after the announcement of the two delayed flights. In Fig. 3, the evolution of passengers in the security checkpoint queue over time is presented. The graph shows in green the expected conditions with no disrupted flight schedule and in red the expected passenger numbers for the disrupted schedule, while no adaptive action is undertaken. Further, the top sub-graph shows how the influx of passengers shifts due to the disrupted flight schedule. In green the influx of passengers under nominal conditions is presented, and in red the passenger influx for the disrupted flight schedule scenario. The influx of passengers is measured as the average influx for

Table 8 Initial prediction of the performance indicators for the disrupted schedule

$p(\text{sat})$	\bar{T}_{sat} (min)		\bar{W}_{sat} (min)		\bar{MF} (pax)	
	μ	σ	μ	σ	μ	σ
1	119.8	20.96	32.58	3.29	124.88	26.08

Here, $p(\text{sat})$ denotes the probability of saturation, \bar{T}_{sat} denotes the average saturation duration, \bar{W}_{sat} denotes the average waiting time during the saturation phase, and \bar{MF} represents the average number of passengers that misses their flight

Table 9 Set of 8 action options based on the initial prediction

Action option	Number of extra lanes to open	Time of opening	Time of closing
1	1	06:26	07:23
2	1	06:26	08:05
3	1	06:26	08:27
4	2	06:30	07:00
5	2	06:35	07:04
6	2	06:45	07:14
7	2	06:45	07:23
8	2	06:35	07:23

For each action option, the number of extra lanes to open and the times of opening and closing are specified

100 simulations. Each bar in the middle graph represents the mean number of passengers arriving at the airport per minute. Finally, the bottom sub-graph shows the number of security lanes opened over time following the original resource schedule.

The shift in arrival of passengers of the delayed flights does not match with the scheduled resources in the original planning, resulting in a high number of passengers accumulating in the queue. To analyze the severity of this accumulation, the data for the disrupted schedule is compared to the threshold value defined based on the IATA level of service standards (Stewart Head 2014). The threshold value is depicted with a red dotted line. As shown by the figure, a relatively large part of the distribution exceeds the threshold value. The performance indicators quantifying the saturation are presented in Table 8. From the table it can be deduced that the probability of saturation $p(\text{sat}) = 1$, hence for all simulations, the system reached a saturated state. The severity of saturation is expressed by the average duration of saturation $\bar{T}_{\text{sat}} = 119.18$ min, and the average waiting time $\bar{W}_{\text{sat}} = 32.58$ min. Additionally, as a global performance indicator, the average number of passengers that end up missing their flight, due to the long waiting times, is presented: $\bar{MF} = 124.88$ pax. Based on these values, it can be concluded that adaptation of the system is required in order to maintain a desired performance level.

Based on the data obtained from the initial prediction, 8 action options were generated for valuation (Table 9). The action option set including action option 1 up and till 6 is generated following the action option generation algorithm as presented in Appendix. The action option 7 is based on the result of the action option 6 whereas the time duration applied for the action option 6 is extended till the time of maximum

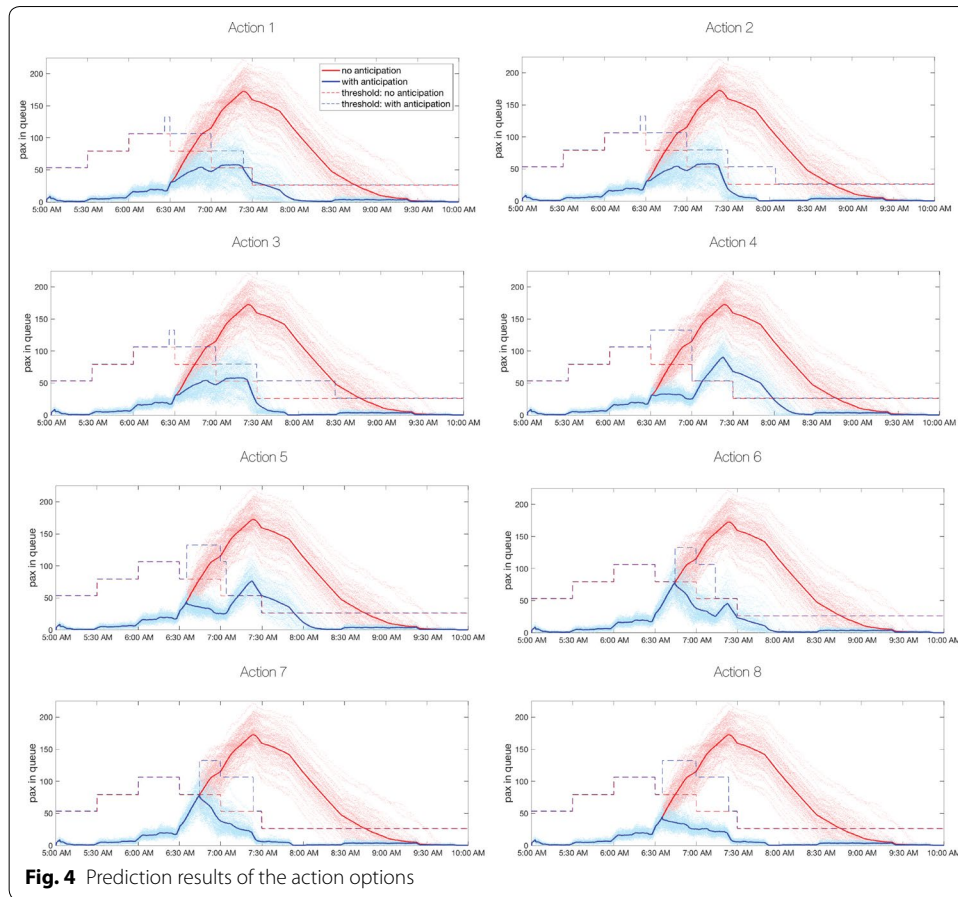


Fig. 4 Prediction results of the action options

Table 10 The valuation results for 8 action options

Action option	Number of extra lanes	Opening time (hh:mm)	Duration (min)	$p(\text{sat})$	\bar{T}_{sat} (min)		\bar{W}_{sat} (min)		\bar{MF} (pax)	
					μ	σ	μ	σ	μ	σ
1	1	06:26	58	0.57	25.78	18.53	17.57	3.19	6.47	12.21
2	1	06:26	99	0.27	13.25	9.92	10.35	0.37	0.33	1.33
3	1	06:26	121	0.27	13.25	9.92	10.35	0.37	0.33	1.33
4	2	06:30	30	1	49.52	11.36	19.89	1.54	32.35	17.99
5	2	06:35	29	0.99	36.38	14.68	10.08	0.07	18.36	15.80
6	2	06:45	29	0.63	12.22	11.70	11.45	1.32	2.06	7.09
7	2	06:45	38	0.49	2.96	2.15	10.48	0.33	0	0
8	2	06:35	48	0	0	0	—	—	0	0

Here, $p(\text{sat})$ denotes the probability of saturation, \bar{T}_{sat} denotes the average saturation duration, \bar{W}_{sat} denotes the average waiting time during the saturation phase, and \bar{MF}_{sat} represents the average number of passengers that missed their flight

saturation. Additionally, the action option 8 is based on the result of action option 5, whereas the time of duration is extended till the time of maximum saturation.

Valuation of options For each of the action options as presented in Table 9, simulations were conducted to obtain a prediction of the expected passenger demand in the queue over time. The predictions for each action option are presented in Fig. 4. The valuation results for all eight action options are provided in Table 10.

Action option 1 is defined as opening one extra lane at 06:26 A.M. and closing the extra lane at 07:23 A.M. The extra lane is thus opened for 57 min. The passenger accumulation over time is improved by this action, however, the mean number of passengers in the queue over time still exceeds the threshold value. Additionally, the probability that the system reaches a saturated state is $p(\text{sat}) = 0.57$.

Action option 2 is defined as opening one extra lane at 06:26 A.M. for 99 min. It is similar to option 1, except for the opening duration which is 41 min longer. A saturation rate of 0.60 pax/min remains to exist during the time span of 99 min. The effect of 41 min extra opening time is that the saturation is diminished compared to option 1.

Action option 3 is defined as opening one extra lane at 06:26 A.M. for 121 min. Comparing option 2 to option 3, the latter does not bring any improvements. The negative effects of the queue saturation were removed before the end of the extra lane closing of option 2. Thus, the extra duration of 22 min does not generate other beneficial effects.

Action option 4 is defined as opening two extra lanes at 06:30 A.M. for 30 min. Opening two extra lanes instead of one, as in options 1, 2, and 3, reduces immediately the number of passengers in the queue. However, the duration of opening is not long enough to overcome the saturation. The saturation is certain and the average number of passengers missing their flight is high $\overline{MF} = 32.35$.

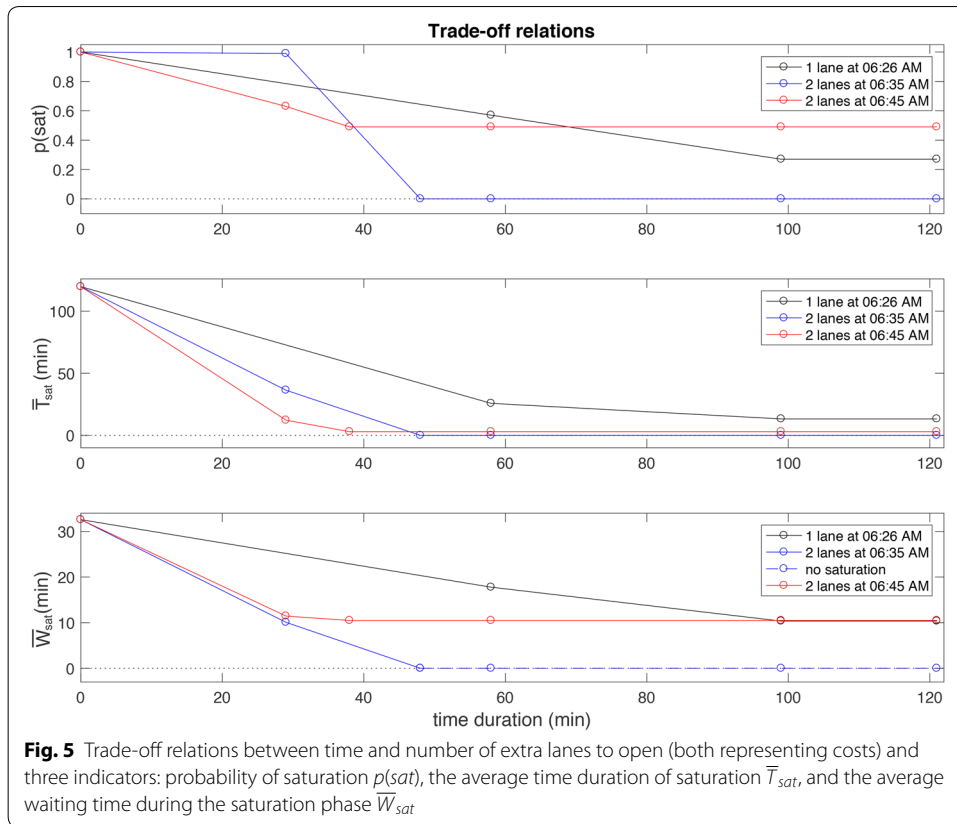
Action option 5 is defined as opening two extra lanes at 06:35 A.M. for 29 min. Although the probability of saturation of the queue is almost certain ($p(\text{sat}) = 0.99$), the severity of the saturation is lower than the action option 4. The average number of passengers missing their flight is reduced to $\overline{MF} = 18.36$.

Action option 6 is defined as opening two extra lanes at 06:45 A.M. for 29 min. It is similar to option 5, except for the time of opening which is 10 min later. Compared to option 5, the probability of saturation is smaller ($p(\text{sat}) = 0.63$), but the averaged waiting time is higher ($\overline{W}_{\text{sat}} = 11.45 \text{ min}$).

Action option 7 is defined as opening two extra lanes at 06:45 A.M. for 38 min. It is similar to option 6 except for the duration of opening which is 9 min longer. Similarly to the comparison of option 2 with option 1, the severity of the saturation is diminished. With this option, the probability of saturation is about one half, however the number of missed flight is 0.

Action option 8 is defined as opening two extra lanes at 06:35 A.M. for 48 min. It overcomes the whole saturation of the queue. This option is actually the one of the set with the highest parameter values (number of lanes, duration of the opening). It is thus the option with the highest cost.

Choice of the action option Depending on the action taken, a pre-saturation phase might result in saturation, as is the case for action option 4. However, action option 8 shows that saturation can be averted by the right action and timing. Essentially, a trade-off needs to be considered between the cost of extra resources and the probability and severity of saturation that is accepted by the agent. For instance, for a similar severity of saturation, if it is decided to choose action option 2, then only one extra lane has to be opened, but for a duration of 99 min, while choosing action option 7 would double the number of lanes for only 38 min. In Fig. 5 trade-off relations are presented that are derived from the valuation results presented in Table 10. Trade-off



relations are shown for opening one extra security lane at 06:26 A.M., and two extra lanes at 06:35 and 06:45 A.M. with respect to three performance indicators: probability of saturation $p(sat)$, the average time duration of saturation \bar{T}_{sat} , and the average waiting time during the saturation phase \bar{W}_{sat} . The number of extra security lanes and the duration of opening these lanes represent the costs. The relations presented for opening one extra security lane are based on the data corresponding to action option 1, 2 and 3. The relations presented for opening two extra lanes at 06:35 A.M. are based on action option 5 and 8. For action option 8 the average waiting time during the saturation phase is undefined as no saturation phase exists when this action is executed. This is represented in Fig. 5 by the dashed blue line. The relations presented for opening two extra lanes at 06:45 A.M. are based on action option 6 and 7. Furthermore, opening two security lanes at 06:45 A.M. for longer than 38 min will not further improve the performance indicators. The same holds for opening two extra security lanes at 06:35 A.M. for longer than 48 min as for action 8 already an optimal result is obtained with regard to probability of saturation and the average time of saturation. This is reflected in the trade-off curves.

Main conclusions from the results

Based on the results, the following conclusions were drawn:

- Opening one extra security lane does not provide enough extra resources to overcome the probability of saturation. Nevertheless, opening one security lane for a duration of approximately 100 min reduces the probability of saturation to 0.27. Additionally, the average waiting time is decreased to 10.35 min, which is just above the threshold value of 10 min. Finally, the average number of missed flights is reduced to approximately one passenger.
- Based on the results obtained for opening one extra lane, it is important to act proactively, when the resources available are not matching the demand. By mobilizing the extra resources already in the pre-saturation phase, the severity of saturation is decreased.
- The option to open two extra lanes provides enough resources to avoid a saturated state, as the average processing rate of two extra lanes is higher than the average saturation rate of the system. However, to obtain a probability of saturation equal to zero, the opening of two extra security lanes has to be executed proactively, e.g., before the system reaches a saturated state, and the lanes are opened for 48 min.
- Timing of a specific action can have significant effects on the impact on saturation risk and severity. Therefore, it can be concluded that in practice, mobilization time of resources should be taken into account while planning for an adaptive measure.

Conclusions and outlook

Understanding and improving resilience and its adaptive capacity of the air transport system is essential for the system's current performance and future growth. Formal foundations for the adaptive capacity of resilience of sociotechnical systems are currently largely missing. Farjadian et al. (2017) presented one of the first attempts to formalize and operationalize theoretical resilience concepts stemming from behavioral sciences in the context of a shared pilot-autopilot control architecture. To represent concepts of resilience the authors employed control theory, which is different from our agent-based modeling approach. Furthermore, we focus on a multiagent system environment with many interacting agents, whereas the authors focused on interaction between two agents in their work.

This work sets the first steps towards formalization of adaptive resilience of the complex, sociotechnical air transport system with a particular focus on its ability to anticipate. This ability has been identified as essential for handling diverse disruptions in the air transport system.

In this paper first, based on theoretical findings on adaptive resilience and anticipation from behavioral sciences and by applying TTL language, a formal agent-based model of an anticipation mechanism was developed, including five steps of the anticipation process. It was demonstrated that TTL is a suitable language to formalize anticipation mechanisms in the context of a multiagent system. A different language that was used for this research as well, but not presented in this paper, is the language of TTS. It was concluded that this high-level and generic language is able to provide an even higher expressiveness, and would therefore also be a suitable language to describe the working of the mechanism. However, this language offers less powerful means of analysis than

TTL, which is important for the understanding of anticipation and other mechanisms of adaptive resilience.

In the second part of the paper, the working of the formalized anticipation mechanism and its five steps is illustrated by a simulation case study on anticipation in airport security checkpoint operations. To this end, an operational agent-based model was developed and calibrated using data from a regional airport. One of the main conclusions of this study is that the proactive allocation of security resources based on the principles of anticipation can in many cases be beneficial for both the airport and the passengers, when disruptions occur.

Currently, anticipation in the air transport system is largely done by humans. Human operators repeatedly demonstrate well developed anticipation abilities in this system. Therefore, we tried to model human anticipation using the theoretical basis from behavioural sciences to gain understanding of anticipation mechanisms and their role in air transport operations. By modeling and analysis of simulation results we discovered a formal anticipation procedure, which has proven to be beneficial for system resilience. As a next step the identified and formalized anticipation mechanism can be automated in a form of an intelligent agent, which can be used to support a human operator. We leave field tests of such intelligent agents at an airport for future research.

Furthermore, in future research other mechanisms of adaptive capacity of resilience, such as coordination, learning, and self-organisation will be studied using formal and computational approaches too. The agent-based airport simulation environment AATOM, used in this study, has a modular structure and can be extended to study local and system-wide effects of diverse resilience mechanisms and their combination. It appears to us a promising way forward to continue gradually unravelling the complexity of the adaptive capacity of resilience of the sociotechnical air transport system.

Authors' contributions

ANB elaborated theoretical foundations of anticipation, provided description of the cognitive architecture, did actual implementation of the simulation model, performed analysis of the simulation results, provided a description of the case study and conclusions. AS wrote the abstract and the introduction, elaborated theoretical foundations of anticipation, wrote formal language description and formalized the anticipation mechanisms in TTL. MV wrote a part of the description of the case study, specifically parts on scope and objective of the case study; introduction to AATOM, conceptual model specification, calibration/initialization using airport data; prediction through forward simulation; metrics for analysis; results; behavior for different action options; trade-off relations. All authors read and approved the final manuscript.

Author details

¹ Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, Delft, The Netherlands. ² ProRail, Utrecht, The Netherlands.

Acknowledgements

The authors would like to thank Alexander Dilweg for provided data and knowledge on operations at the regional airport, and Stef Janssen for support with the model implementation.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Data from a regional airport used in this research are provided in the paper.

Ethics approval and consent to participate

Not applicable.

Funding

The study was not funded.

Appendix

An action option is defined by three variables:

- i : the number of extra security lanes to open.
- t_{ex} : the moment in time to open the extra lane(s).
- ΔT : the time duration of opening the extra lane(s).

The algorithm seeks for action options with an execution time t_{ex} that lays within the period of time corresponding to the pre-saturation region. The point in time where the pre-saturation phase starts is denoted by t_0 and the point in time where the system reaches the saturation phase is denoted by t_1 . Thus $t_0 \leq t_{ex} \leq t_1$. Additionally, the moment in time where the saturation phase ends is denoted by t_2 .

The algorithm aims to minimize the probability of saturation. As first step, the algorithm determines for each possible number of extra lanes to open $i = 1 : N - 1$, with N the total number of security lanes present at RTHA, if the average processing rate r_{pro} of added resources is larger or smaller than the average saturation rate r_{sat} . Here, the saturation rate is estimated as the averaged linear increase of passengers in the queue between t_0 and t_1 . Further, the average processing rates as presented in Table 3 are assumed. Depending on whether r_{pro} is $<$ or $\geq r_{sat}$ *algorithm a* or *algorithm b* is applied to determine the variables of the action option(s). Step 1, selection of the algorithm is described below.

Step 1: Selection of algorithm

```

input
  total number of security lanes  $N$ 
  saturation rate  $r_{sat}$ 
  vector of processing rates  $\overline{r_{pro}}$ 
for  $i = 1 : N - 1$ 
   $\Delta_r = r_{sat} - r_{pro}(i)$ 
  if  $\Delta_r > 0$ 
    add number of lanes  $i$  to set  $M^1$ 
  end
  if  $\Delta_r \leq 0$ 
    add number of lanes  $i$  to set  $M^2$ 
  end
  save  $\Delta_r(i) = \Delta_r$ 
end
for  $j = 1 : end(M^1)$ 
  execute algorithm a
end
for  $j = end(M^1) + 1 : end(M^2)$ 
  execute algorithm b
end
output
  see output algorithm a and algorithm b

```

If $\Delta_r > 0$, this means that when opening i extra security lanes at t_{ex} (with $t_0 \leq t_{ex} \leq t_1$) accumulation will continue with an estimated saturation rate of Δ_r . Now, *algorithm a* determines how far in advance, before the moment of saturation at t_1 , the extra number of security lane(s) i should be opened to avoid that the threshold value will be exceeded. The execution time t_{ex} is determined based on Δ_r and the moment in time maximum saturation is reached t_{max} . To this end, *algorithm a* solves the following equation:

$$d(t) + \Delta_r(t_{max} - t) = threshold(t_{max}) \quad \text{for } t \leq t_1 \quad (6)$$

Here, $d(t)$ represent the demand, e.g. the number of passengers in the queue at time t . The threshold value that holds at t_{max} is denotes as $threshold(t_{max})$. If the obtained value $t \leq t_0$, t_{ex} is set equal to t_0 , otherwise $t_{ex} = t$. *Algorithm a* is described below.

Step 2a: algorithm a

```

input
demand: number of passengers in the queue over time  $d(t)$ 
time of maximum saturation  $t_{max}$ 
threshold value at the time of maximum saturation  $threshold(t_{max})$ 
saturation rate vector  $\bar{\Delta}_r$ 
for  $j = 1 : end(M^1)$ 
  while  $t \leq t_1$ 
    solve  $d(t) + \bar{\Delta}_r(j) \times (t_{max} - t) = threshold(t_{max})$ 
    if  $t \leq t_0$ 
       $t_{ex} = t_0$ 
    end
    if  $t > t_0$ 
       $t_{ex} = t$ 
    end
    save  $t_{ex}(j) = t_{ex}$ 
  output
  vector of execution times corresponding to opening  $j$  lane(s):  $\bar{t}_{ex}$ 
end

```

For each action option determined through *algorithm a*, defined by the number of security lanes to open and the corresponding execution time, now the time duration for opening the extra lane(s) is determined. For each action option, three different time duration are defined, calculated by the following equations:

$$\Delta T_1 = t_{max} - t_{ex}$$

$$\Delta T_2 = \frac{t_2 - t_{max}}{2} - t_{ex}$$

$$\Delta T_3 = t_2 - t_{ex}$$

Thus, for instance, if set $M^1 = \{1\}$, this means that for only opening one extra security lane $\Delta_r > 0$. A corresponding execution time t_{ex} is calculated by *algorithm a*. Now the possible action options defined for opening one extra security lane are:

- Option 1: opening 1 extra lane, at t_{ex} for a time duration of ΔT_1 .
- Option 2: opening 1 extra lane, at t_{ex} for a time duration of ΔT_2 .
- Option 3: opening 1 extra lane, at t_{ex} for a time duration of ΔT_3 .

An option is only accepted in the option set, if the resource constraint is satisfied. The resource constraint is formulated as:

$$\begin{aligned}
 &\text{if } i \leq N - n_{use}(t_{ex}) \\
 &\quad \text{action option is accepted} \\
 &\text{else} \\
 &\quad \text{action option is not accepted}
 \end{aligned}
 \tag{7}$$

Here, $n_{use}(t_{ex})$ denotes the number of security lanes in use at time of execution.

For the second case where $\Delta_r \leq 0$, this means that when opening i extra security lanes at t_{ex} (with $t_0 \leq t_{ex} \leq t_1$) accumulation is expected to come to a hold, as the added processing rate is larger or equal to the approximated saturation rate. For this scenario, *algorithm b* is applied to determine the time duration the number of i security lanes should be opened to avoid reaching a saturated state. To this end, first three execution times are selected to evaluate. These are:

- $t_{ex}^1 = t_1$: the extra security lane(s) are opened at the moment a saturated state is reached.
- $t_{ex}^2 = \frac{t_1 - t_0}{2}$: the extra security lane(s) are opened half way during the pre-saturation phase.
- $t_{ex}^3 = t_0$: the extra security lane(s) are opened at the beginning of the pre-saturation phase.

For each of these options holds that this is only possible to select the t_{ex} if constraint 7 is satisfied. If for an option the constraint is not satisfied, the algorithm seeks for the first moment in time with $t > t_{ex}$, for which the constraint is satisfied. Then, t_{ex} is set equal to t . The set of three possible execution times is denoted by $\overline{t_{ex}}$. Corresponding time duration ΔT are calculated by *algorithm b* described below.

Step 2b: algorithm b

input

demand: number of passengers in the queue over time $d(t)$
 time of maximum saturation t_{max}
 threshold value at the time of maximum saturation $threshold(t_{max})$
 saturation rate r_{sat}
 saturation rate vector $\overline{\Delta_r}$
 execution time vector $\overline{t_{ex}}$

for $j = end(M^1) + 1 : end(M^2)$

for $k = 1 : 3$

solve $d(\overline{t_{ex}}(k)) + \overline{\Delta_r}(j) \times \Delta T + r_{sat} \times (t_{max} - (\overline{t_{ex}}(k) + \Delta T)) = threshold(t_{max})$

end

 save $\Delta T(k) = \Delta T$

end

 save for each number of lanes j evaluated vector $\overline{\Delta T}$

output

 for each j evaluated and for three possible execution times, three duration values $\in \overline{\Delta T}$ are determined

Thus, for instance, if set $M^2 = \{2\}$, this means that for opening two extra security lanes $\Delta_r \leq 0$. Three execution time options are calculated as described above. For each of the execution times *algorithm b* is applied to seek for duration times that should avoid that the system will reach a saturated state. The possible action options defined for opening two extra security lanes are:

- Option 1: opening 2 extra lanes, at t_{ex}^1 for a time duration of ΔT_1 .
- Option 2: opening 2 extra lanes, at t_{ex}^2 for a time duration of ΔT_2 .
- Option 3: opening 2 extra lanes, at t_{ex}^3 for a time duration of ΔT_3 .

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 16 July 2018 Accepted: 24 September 2018

Published online: 17 October 2018

References

- Abras S, Pesty S, Ploix S, Jacomino M (2008) An anticipation mechanism for power management in a smart home using multi-agent systems. In: ICTTA 2008 3rd international conference on information and communication technologies: from theory to applications, pp 1–6.
- Amsterdam Schiphol Airport Closed as Power Outage Sparks Major Flight Delays for Passengers. <https://www.independent.co.uk/travel/news-and-advice/amsterdam-schiphol-airport-closed-flights-latest-power-outage-delays-passengers-departures-arrivals-a8327886.html>. Accessed 9 May 2017.
- Baylis J, Grayson M, Lau C, Gerstell G, Scott B, Nicholson J (2015) Transportation sector resilience. National Infrastructure Advisory Council, Arlington
- Blumberg BM, Galyean TA (1995) Multi-level direction of autonomous creatures for real-time virtual environments. In: ACM (ed.) Proceedings of the 22nd annual conference on computer graphics and interactive techniques, pp 47–54
- Boos KLM. Drukte Schiphol Heeft Ons Miljoenen Gekost. <http://nos.nl/artikel/2171295-klm-boos-drukte-schiphol-heeft-ons-miljoenen-gekost.html>. Accessed 9 May 2017.
- Bosse T, Jonker CM, Van der Meij L, Sharpanskykh A, Treur J (2009) Specification and verification of dynamics in agent models. *Int J Cooperat Inf Syst* 18(01):167–193
- Butz MV, Sigaud O, Gerard P (2003) Internal models and anticipations in adaptive learning systems. *Anticipatory behavior in adaptive learning systems*. Springer, Berlin, pp 86–109
- Convertino M, Liu Y, Hwang H (2014) Optimal surveillance network design: a value of information model. *Complex Adaptive Syst Modeling* 2(1):6
- Cook A, Rivas D (2016) *Complexity science in air traffic management*. Routledge, London
- Van Dam KH, Nikolic I, Lukszo Z (2012) *Agent-based modelling of socio-technical systems*, vol 9. Springer, New York
- Damasio AR (1994) *Descartes' error: emotion, rationality and the human brain*. University of Scranton Press, Scranton
- Dastani M, Hindriks KV, Meyer J-J (2010) *Specification and verification of multi-agent systems*. Springer, Berlin
- Davidsson P (2003) *A framework for preventive state anticipation. Anticipatory behavior in adaptive learning systems*. Springer, New York, pp 151–166
- Day S, Edwards S, Fearnley C, Kilburn C, McGuire B, Stanbrough L, Wall R, Delacroix S, James A, Smith A et al (2010) Volcanic hazard from iceland. UCL Institute of Risk and Disaster Reduction, London
- Delta Malfunction on Land Keeps a Fleet of Planes From the Sky, *The New York Times*. <https://www.nytimes.com/2016/08/09/business/delta-air-lines-delays-computer-failure.html>. Accessed 8 Aug 2016.
- Farjadian AB, Annaswamy AM, Woods D (2017) Bumpless reengagement using shared control between human pilot and adaptive autopilot. *IFAC-PapersOnLine* 50(1):5343–5348
- Francis R, Bekera B (2014) A metric and frameworks for resilience analysis of engineered and infrastructure systems. *Reliab Eng Syst Saf* 121:90–103
- Germund H (2002) Conscius thought as simulation of behaviour and perception. *Trends Cogn Sci* 6:242–247
- Hoffmann J (2008) Abc: a psychological theory of anticipative behavioral control. *Workshop on anticipatory behavior in adaptive learning systems*. Springer, Berlin, pp 10–30
- Hollnagel E (2011) Rag-the resilience analysis grid. *Resilience engineering in practice: a guidebook*. Ashgate, Farnham
- Hollnagel E (2013) *Resilience engineering in practice: a guidebook*. Ashgate Publishing Ltd, Ashgate
- Hoogendoorn SP, Bovy PHL (2004) Pedestrian route-choice and activity scheduling theory and models. *Transport Res B Methodolog* 38:169–190
- Hosseini S, Barker K, Ramirez-Marquez JE (2016) A review of definitions and measures of system resilience. *Reliab Eng Syst Saf* 145:47–61. <https://doi.org/10.1016/j.res.2015.08.006>
- Jamroga W (2008) A temporal logic for markov chains. In: *Proceedings of the 7th international joint conference on autonomous agents and multiagent systems*. International Foundation for Autonomous Agents and Multiagent Systems, Vol 2, pp 697–704
- Janssen S, Blok AN, Knol A (2018) AATOM—an agent-based Airport terminal operations model. TU Delft. <https://github.com/StefJanssen/AATOM>; a tool description can be found at [https://pure.tudelft.nl/portal/en/publications/aatom--an-agentbased-airport-terminal-operations-model\(eb03ba8b-a4ec-4754-9f63-41a0774c531a\).html](https://pure.tudelft.nl/portal/en/publications/aatom--an-agentbased-airport-terminal-operations-model(eb03ba8b-a4ec-4754-9f63-41a0774c531a).html)
- Manzano M, Manzano TdLM (1996) *Extensions of first-order logic*, vol 19. Cambridge University Press, Cambridge
- Michel A, Wang K, Hu B (2001) *Qualitative theory of dynamical systems*, vol 239. CRC Press, New York
- Minch E (1986) A review of: anticipatory systems, by Robert Rosen, pergamon press, oxford, 1985, x + 436 pp. *Int J Gen Syst* 12:405–409
- Mitleton-Kelly E (2013) *Co-evolution of intelligent socio-technical systems*. Springer, Berlin
- Ouyang M (2014) Review on modeling and simulation of interdependent critical infrastructure systems. *Reliab Eng Syst Saf* 121:43–60. <https://doi.org/10.1016/j.res.2013.06.040>
- Pezzulo G, Butz MV, Castelfranchi C, Falcone R (2008) *The challenge of anticipation: a unifying framework for the analysis and design of artificial cognitive systems*. Springer, Berlin
- Richardson M, Domingos P (2006) Markov logic networks. *Mach Learn* 62(1–2):107–136
- Righi AW, Saurin TA, Wachs P (2015) A systematic literature review of resilience engineering: research areas and a research agenda proposal. *Reliab Eng Syst Saf* 141:142–152
- Rue RC, Rosenshine M (1985) The application of semi-markov decision processes to queueing of aircraft for landing at an airport. *Transport Sci* 19:154–172

- Sharpanskykh A (2011) Agent-based modeling and analysis of socio-technical systems. *Cybern Syst* 42(5):308–323
- Sharpanskykh A, Stroeve SH (2011) An agent-based approach for structured modeling, analysis and improvement of safety culture. *Comput Math Organ Theory* 17(1):77–117
- Sharpanskykh A, Treur J (2010) A temporal trace language for formal modelling and analysis of agent systems. *Specification and verification of multi-agent systems*. Springer, New York, pp 317–352
- Stewart Head of Airport Development IATA UK (2014) The new airport development reference manual (adrm). In: Passenger terminal conference 2014
- Sutcliffe KM, Christianson MK (2001) *Managing the unexpected. Assuring high performance in an age of complexity*, San Francisco
- Treur J (2016) *Network-oriented modeling: addressing complexity of cognitive, affective and social interactions*. Springer, Berlin
- Woods DD. The theory of graceful extensibility: basic rules that govern adaptive systems (exp. release 09/2018) **(in press)**

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
