View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of Bath Research Portal

UNIVERSITY OF BATH

*Citation for published version:*
Mogles, N, Padget, J & Bosse, T 2018, 'Systemic approaches to incident analysis in aviation: comparison of STAMP, Agent-Based Modelling and Institutions', *Safety Science*, vol. 108, pp. 59-71.
https://doi.org/10.1016/j.ssci.2018.04.009

*DOI:*
10.1016/j.ssci.2018.04.009

*Publication date:*
2018

*Document Version*
Peer reviewed version

Link to publication

*Publisher Rights*
CC BY-NC-ND

**University of Bath**

**Alternative formats**
If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

**General rights**
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 22. Jun. 2021

# Systemic approaches to incident analysis in aviation: comparison of STAMP, Agent-Based Modelling and Institutions

Nataliya Mogles[1], Julian Padget[1], and Tibor Bosse[2]

[1] University of Bath, Department of Computer Science,
Bath, United Kingdom
{n.m.mogles, j.a.padget}@bath.ac.uk
[2] Vrije Universiteit Amsterdam, Department of Computer Science,
Amsterdam, The Netherlands
t.bosse@vu.nl

**Abstract.** The rapid development and increasing complexity of modern socio-technical systems suggest an urgent need for systemic safety analysis approaches because traditional linear models cannot cope with this complexity. In the aviation safety literature, among systemic accident and incident analysis methods, Systems Theoretic Accident Modelling and Processes (STAMP) and Agent-based modelling (ABM) are the most cited ones. STAMP is a qualitative analysis approach known for its thoroughness and comprehensiveness. Computational ABM approach is a formal quantitative method which proved to be suitable for modelling complex flexible systems. In addition, from a legal point of view, formal systemic institutional modelling potentially provides an interesting contribution to accident and incident analysis. The current work compares three systemic modelling approaches: STAMP, ABM and institutional modelling applied to a case study in an aviation domain.

**Keywords:** event models, norms, cognitive functioning, socio-technical systems

## 1 Introduction

Nowadays, technical systems and automation processes have penetrated almost all human organisations and this tendency is set to continue. Modern transportation systems, and especially aviation and air traffic management, are characterised by a high degree of complexity due to multiple human-technology couplings and their sophisticated interactions. Safety analysis of such complex socio-technical systems (STSs) poses a real challenge since it is shaped by multiple individual, social, technical and environmental factors.

Air traffic management (ATM) typifies a complex socio-technical organisation where humans and software systems are tightly coupled and interact in a complex manner. In general, ATM is a highly organised, structured system with multiple control loops and regulations. In spite of that, undesirable events such as accidents do occasionally occur in this domain and numerous small incidents regularly occur. One of the characteristic features of this complexity is high time dynamics where several seconds or milliseconds might change the whole picture of events. Humans are very good in reasoning about time aspects on a daily basis. However, if we deal with a system with multiple distinctive components and their interdependencies, the necessity of formal automatic reasoning and analysis of such systems is inevitable.

Traditionally, the analysis of such incidents is performed by safety experts using fault and events trees [16], [41]. This approach is quite simplistic and does not allow a full analysis with respect to a temporal dimension, neither is it powerful with respect to a prospective analysis which empowers the designers of the whole system in attempting to prevent future accidents or incidents.A systems approach towards accident analyses within modern STSs, including aviation and ATM domain, is the most promising paradigm [40]. This paradigm tries to avoid the limitations of sequential accidents models based on causal accident models. The systems approach views accidents as the result of unexpected relationships between a system's parts with the requirement that systems are analysed as whole entities rather than consisting of isolated parts.

In the aviation safety literature, different accident analysis methods and models underpinned by a systems-thinking paradigm are cited, such as AcciMap[29], Systems Theoretic Accident Modelling and Processes model (STAMP) [20], [21], Functional Resonance Analysis Method (FRAM) [15], agent-based modelling (ABM) approach [4], [35], systems dynamics simulation [13]. There is a debate in different research communities whether Swiss Cheese Model (SCM) [30], [31] can be characterised as a systemic approach since it is being perceived as a sequential model rather than a holistic approach capable of analysing emergent behaviour of complex STSs. These methods are developed within different research communities and frequently have different methodological and conceptual roots which result in different knowledge representation within the underlying models. Some of these methods, for example ABM and system dynamics simulation, are formal as they utilise mathematical formalisms.

There is a strong need for formal modelling and safety analysis in modern STSs [43] in order to understand their complexity. Formal modelling methods and techniques have been widely applied to risk and safety analysis in aviation and other safety critical domains, e.g. fault and event trees, ABM, Bayesian Belief Networks (BBN), [17], [2], though not all formal approaches are systems-based. For safety analysis in complex STSs from a legal perspective, another formal systems-based approach can be interesting - institutional modelling developed in Artificial Intelligence (AI) domain. This approach combines institutional theory [34] and computational reasoning. An example of institutional modelling is the InstAL modelling language [14], which has been successfully applied to the formal analysis of socio-technical organisations from an institutional perspective, including security analysis in organisations [28].

The landscape of all approaches towards accident analysis in aviation, including systemic ones, can be roughly classified across two dimensions according to knowledge representation: quantification (qualitative-quantitative) and mathematical formalisation (formal-informal) – see Table 1. As can be seen in Table 1, safety analysis approaches denoted with asterisks are spread across the two dimensions and occupy three quadrants: quantitative-formal, qualitative-formal and qualitative-informal. Data-driven statistical analysis is categorised as a quantitative-informal approach and various model

**Table 1.** Safety analysis methods classification.

| | **Quantitative** | **Qualitative** |
|---|---|---|
| **Formal** | ABM*, fault and event trees, BBN, systems dynamics | Model verification methods, institutional modelling* |
| **Informal** | Descriptive statistics | STAMP*, FRAM*, Accimap* SCM |

* Systemic methods

validation and verification techniques [33] are classified as qualitative-formal within this framework.

There are several studies which compare different systems-based approaches. Fo example, in [32] Accimap [29], the Human Factors Analysis and Classification System (HFACS) approach [42] based on SCM and STAMP are compared. In [40] a comparison of STAMP with Accimap and the Australian Transport Safety Bureau (ATSB) approach [10] based on SCM is described. In both cases, the comparisons are done within one quadrant of the methods classification framework: qualitative-informal. In [37], the authors compare methods from two quadrants: fault tree, BBN and FRAM.

We believe that it is important to compare systemic methods with different positions within the knowledge representation dimensions. This should help safety practitioners and scientific communities form a wider view on systemic methods and to gain understanding of the diversity of these methods for the sake of broadening the safety analysis repertoire and opening opportunities for systemic methods integration. Given the arguments above, in the current work we address three quadrants within the knowledge representation framework and compare the most frequently cited in literature methods from two quadrants: STAMP within qualitative and informal, ABM within quantitative-formal quadrant and an innovative safety analysis method from the qualitative-formal quadrant: institutional modelling. Application of institutional modelling can be interesting from the legal point of view.

JAP: feel this needs some substantiation and context; sounds odd on its own here

The institutional InstAL modelling language [14],[12] has been successfully applied to the formal analysis of socio-technical organisations from an institutional perspective in different domains: see for example [26] for the analysis of interactions between smart phones and social networking platforms and [28] for the formal analysis of security and legal policies in organisations.

JAP: at the risk over-citing me (indirectly), there's a paper by Tingting [23] and her thesis [22]

The ABM paradigm, expressed in terms of the Temporal Trace Language (TTL) models in conjunction with the LEADSTO sublanguage were previously applied in formal safety analysis of human functions in the aviation domain [5,7]. The main findings of these exploratory studies revealed the feasibility and effectiveness of formal retrospective analysis of accidents and incidents in TTL/LEADSTO language, which has the potential to substantially improve the quality of safety analysis in (aviation) transportation and other safety critical socio-technical organisations, e.g. in nuclear power, pharmacology and healthcare domains. The TTL/LEADSTO language allows for hybrid qualitative and quantitative knowledge representation and covers two quadrants within the classification framework in Table 1. The TTL/LEADSTO models take the agent-based perspective which implies that an incident is caused by complex interactions between different agents in the organisation. Specifically, in the domain of aviation safety, there is another formal agent-based approach proposed in [4,38]. However, this formal analysis uses probabilistic modelling to quantify aircraft collision risks given particular configurations of agents and their environment. Due to the stochastic nature of processes modelled by this approach, the behaviour of the resulting models is not transparent and intuitive and it is difficult to track the behaviour of individual agents within them. In contrast, logic-based accident modelling and analysis might be a useful tool to analyse the emergence of safety issues in such socio-technical organisations due to the presence of qualitative and hence unambiguous and straightforward information representation, that can be easily processed both by humans and software agents.

The main goal of this paper is to compare the capabilities of three different incident and accident analysis approaches in aviation: (i) qualitative and informal CAST analysis based on STAMP; (ii) formal ABM approach represented by the hybrid qualitative/quantitative TTL/LEADSTO modelling language; and (iii) formal institutional modelling represented by the qualitative InstAL modelling language. We compare these approaches, with their different knowledge representation, in order to provide a wider view on systemic methods and to explore the potential of systemic accident analysis methods integration. The comparison is done by an application of the three approaches to the incident case study described in[5,7].

The remainder of the paper is organised as follows: in Section 2 a comparison framework is described, then a case study which was modelled according to the three approaches is presented in Section 3. A more detailed description of all three approaches is given in Section 4, while Section 5 analyses outcomes according to the three approaches are presented. A evaluative comparison of the methods is given in Section 6. Finally, Section 7 concludes the work.

## 2 Comparison aspects

In order to model and analyse behaviour of complex transportation systems, such as ATM, from global and local perspectives, the selected modelling approaches should fulfil certain requirements that may serve as criteria for assessing their suitability for performing the analysis. In the literature on safety analysis approaches, various analysis requirements are mentioned and different modelling approaches classifications and taxonomies are proposed. For example, Le Coze [19] addresses organisational dimension of accidents and classifies approaches according to a two dimensional framework with a level of abstraction (micro-meso-macro) and relations with data (normative versus descriptive). Netjasov and Janic [25] classify approaches towards safety modelling in aviation into the following types according to the type, goals and context of modelling: causal models for risk and safety assessment, collision risk models, human error models and third party risk models. Lundberg et al [24] classify accident models into simple linear system models (cause-effect models), complex linear system models – epidemiological approach (e.g. Swiss cheese [30]), complex interactions, performance variability [15]. The authors also mention various scopes of accident analyses addressed by different approaches: human, technology, organisation and information availability. Benner [3] proposes ten criteria for accident model evaluation: models must be realistic, definitive, satisfying, comprehensive, disciplining, consistent, direct, functional, non-causal, visible. Sklet [36] compares 14 selected methods for accident investigation and proposes a framework for their comparison. The framework includes the following aspects of a safety approach: provision of a complete understanding of the events leading to the accident; preferable graphical description of the accident sequence; demonstration how safety barriers influenced the accident; complexity reflected according to a classification of a socio-technical system involved in safety, comprising multiple levels (technical system, staff, management, company, regulators and Government); type of an underlying accident model used; primary or a secondary method; the need for education and training in order to use the method (expert, specialist or novice).

Salmon et al. [32] compare Accimap, HFACS (Human Factors Analysis and Classification System) – based on Swiss cheese – and STAMP. The authors mention the following comparison aspects: output of the model, model structure and taxonomy of failures, approach comprehensiveness, linkage of failures between and within the levels of analysis, reliability (between different analysts), domain independence, context consideration. Underwood and Waterson [40] compare ATSB (Swiss cheese based), Accimap and STAMP approaches. The compared approaches were evaluated against two topics of interest: coverage of systems theory concepts and usage characteristics. Within the first criteria (systems thinking), the following comparison subcriteria were identified: system structure, component relationship, system behaviour (input, outputs, feedback mechanisms); within the second topic: data requirements, validity and reliability, usability (training resources), graphical representation. Altabbakh et al [1] demonstrate that the STAMP approach focuses on a holistic system. The authors also state that the method goes beyond human performance factors and adds organisational hierarchy, working practices and roles. Moreover, it considers all levels of complex systems including environment, human error, physical component failure, the context, also interrelationships between components of the system.

JAP: this and the preceding paragraph are good in terms of literature cited, but they are mostly descriptive and lack analysis. Perhaps that is intentional, but I suggest it would help to ensure that reader expectations are properly set up at the beginning of the section with some motivation and the outcome, e.g. how it leads to the next paragraph that describes the proposal.

Taking into account the models' classifications and accident approaches evaluation criteria mentioned above and approaches classification framework described in Section 1, we propose the following incident analysis methods evaluation framework within complex socio-technical organisations:

1. *Levels of analysis*: an approach can address a certain level of granularity of processes and actors, either a micro level, like human factors approach or agent-based modelling, or a more global level of organisations and organisational units – a meso-macro level – that includes for example organisational structure, policies, norms and regulations. There exists a methodological difficulty in connecting multiple levels of analysis in one framework since different granularities are based on different conceptual roots and different definition of contexts under study [19].
2. *Taxonomy of failures*: One may argue that having a fixed taxonomy of failures within an approach will restrict the possibilities of failures beyond this taxonomy while performing analysis. However, it is important to have a taxonomy in order to be able to compare different approaches and from the scientific point of view, to improve the transparency and reproducibility of a method.

3. *Quantitative representation*: In some contexts, it is important to quantify concepts under analysis in order to provide a clear picture of events and processes and to be able to represent different variables of interest.
4. *Qualitative representation*: an ability of a method to provide a clear qualitative framework for reasoning about systems' operations and possible failures
5. *Formal semantics*: the existence of a formal semantics behind a method which utilises mathematical models for analysis and verification. This criteria will allow for formal specification and automatic property checking not only for software technical systems where it has traditionally been deployed, but in order to expand it to the analysis of complex socio-technical systems where complexity cannot be tackled by non-formal methods only.
6. *Events representation*: people are comfortable with reconstructing an incident as a chain of events, it helps to create a clear picture of an incident from a perspective of an external observer, so in many case it is important to have this type of representation within a method
7. *Time dynamics expressiveness*: the capability an approach to represent the time dynamics of organisational and local actors' processes
8. *Amount of training*: the amount of training needed for a typical safety expert not familiar with the approach to learn it
9. *Graphical representation*: effective visualisation of modelling results within an approach
10. *Data requirements*: the amount of and ease of access to the data needed for an application of an approach
11. *Time resources*: the amount of time needed for analysis given a person who performs analysis is already trained in an application of a method
12. *Additional resources (software etc)*: additional resources needed for performing safety analysis, it can be IT technologies (software, hardware), also access to organisational archives and reports, collaborations with other experts etc.
13. *Main versus complementary method*: this criteria defines whether an approach is stand-alone and can be applied for full safety analysis process or it rather strengthens or complements other safety analysis approaches

We did not aim for a particular number of aspects, rather the 13 items above emerged from an analysis of the literature. The list of the aspects for comparison can be roughly divided into two groups: aspects 1–7 roughly correspond to the expressive power of models and approaches; while aspects 8–13 refer to the practical usage of methods. The current comparison framework is based in part on the previously published work of the authors [6].Note that the order of the evaluation aspects listed above does not strictly represent the importance of these aspects. Depending on the nature, context and main goals of incident investigation, relative importance of different aspects of methods can vary.

JAP: need to make clear in what respects it builds on/enhances, in order to differentiate, nd why

## 3 Case Study

A specific case study in the domain of ATM has been modelled following three different approaches in order to get more insight into the expressive capabilities, strengths

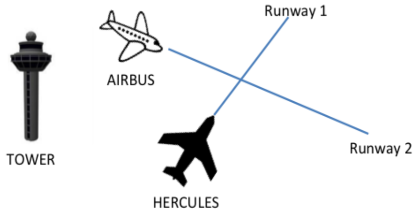

**Fig. 1.** Schematic overview of the case study.

and weaknesses of the approaches. The case study consists of (a simplification of) an existing scenario in the context of a runway incursion incident that occurred in 1995, described in more detail in [5]. A schematic overview of the scenario is provided in Figure 1, while the summary follows:

*Scenario: The Airbus was preparing for the departure. It was supposed to taxi to runway 03 in the north-east direction. The Airbus received permission to taxi and started taxiing to its runway. Approximately at the same time, a military Hercules aircraft that was ready for the departure as well received permission to taxi in the north-west direction from its parking gate. The Hercules was supposed to take off from runway 36 that crossed with runway 03 that was designated for the Airbus. Both aircraft were taxiing to their runways. During the taxiing, the Airbus received its flight route from the air traffic controllers. Some time later, when the Airbus was near the runway designated for taking off, it switched from the taxiing radio frequency to the frequency of the Tower and received permission to line up on the assigned runway. The Hercules was still at the taxiing radio frequency and also received permission to line up, while at the same time the Airbus received permission to take off at the radio frequency of the Tower. However, due to unknown reasons* [3]*, the Hercules pilot interpreted his permission for lining up as permission for taking off and started taking off on runway 36. As a result of this mistake of the pilot of the Hercules, two aircraft were taking off simultaneously on crossing runways, and none of the crews were aware of that. The air traffic controllers in the Tower observed the conflicting situation and communicated a STOP signal to the pilot-in-command of the Airbus, while the Airbus was still on the ground (but at high speed). The pilot had to make a quick decision about the termination of the take-off as there is a point in this process that one cannot safely do this anymore. After having analysed the situation, the pilot-in-command of the Airbus gave a command to the co-pilot (who controlled the aircraft) to abort the take-off and start braking on the runway. During braking, the crew of the Airbus saw the Hercules flying close in the air above their own aircraft at a distance of about 5 meters. A serious collision was prevented.*

[3] This misinterpretation might be explained by the fact that the pilot of the Hercules got used to the routine procedure of taxiing from the same military parking place at this airport where the line up clearance was often immediately followed by the take off clearance.

# 4 Safety Analysis Approaches

This section provides a short introduction and overview of each of the three approaches that are the subjects of the comparison.

## 4.1 STAMP: CAST

The Systems-Theoretic Accident Model and Processes (STAMP) methodology [21] is based on systems and control theory. According to STAMP, the analysis and management of any emergent property of a socio-technical system is, in its essence, a control problem. To deal with this problem, STAMP incorporates a number of tools that help analysts generate scenarios, define specifications and explain how inadequate control, in relation to an unwanted deviation of a system emergent property, could happen. In this methodology, an accident is not understood in terms of a series of events, but rather as the result of a lack of control or the constraints imposed on the system design and operations. The STAMP approach is also an example of systemic accident modelling that treats an accident as a result of failure of an entire system [21]. The approach comprises several steps according to the STAMP-based CAST methodology [21]: 1) Identify the systems hazards involved in the loss; 2) Identify the systems safety constraints; 3) Document the safety control structure; 4) Determine the proximate events leading to the loss; 5) Analyse the loss at the physical system level; 6) Analyse how each successive higher level of the system contributed to the inadequate control at a lower level; 7) Examine overall coordination and communication between the elements of the system; 8) Determine the dynamics and changes in the system and the safety control structure; 9) Generate recommendations. The analysis process is non-linear and there are no strictly defined requirements that one step must be completed before the next one is started. The present study will focus on the first seven steps of the analysis. In addition, according to the STAMP accident analysis methodology, each component of the system is described in terms of the following characteristics: safety requirements and constraints, controls, context (e.g., roles and responsibilities, environmental and behaviour-shaping factors), dysfunctional interactions and failures, reasons for the flawed control actions and dysfunctional interactions (e.g., control algorithm flaws, incorrect process models, inadequate coordination or communication, reference channel flaws, feedback flaws).

## 4.2 Agent-Based Modelling: TTL and LEADSTO

The predicate-logical Temporal Trace Language (TTL) is a hybrid language which integrates qualitative, logical aspects and quantitative, numerical aspects[8]. This integration allows the modeller to express dynamic properties at different levels of aggregation, which makes it well suited both for simulation and logical analysis. The TTL language is based on the assumption that dynamics can be described as an evolution of states over time. The notion of state as used here is characterised on the basis of an ontology defining a set of physical and/or mental (state) properties that do or do not hold at a certain point in time. These properties are often called *state properties* to distinguish them from *dynamic properties* that relate different states over time. A specific state is characterised by dividing the set of state properties into those that hold, and those that do not hold in the state. To formalise dynamic properties, explicit reference is made to time points and to traces (i.e., sequences of states). They are expressed by temporal statements using the standard first-order logical connectives and quantifiers.

To be able to perform (pseudo-)experiments, only part of the expressivity of TTL is needed. To this end, the executable LEADSTO language described in [9] has been defined as a sublanguage of TTL, with the specific purpose to develop simulation models in a declarative manner. In LEADSTO, direct temporal dependencies between two state properties in successive states are modelled by *executable dynamic properties*. The LEADSTO format is defined as follows. Let $\alpha$ and $\beta$ be state properties as defined above. Let $\alpha$ and $\beta$ be successive state properties, then, $\alpha \twoheadrightarrow_{e,f,g,h} \beta$ means:

*If state property* $\alpha$ *holds for a certain time interval with duration g, then after some delay(between e and f) state property* $\beta$ *will hold for a certain time interval of length h.*

More details on the TTL and LEADSTO languages are available in [9,8].

### 4.3 Institutional modelling: InstAL

InstAL was developed for institutional modelling [12]. It is an action language that follows the traditions of the Event Calculus [18]. The generic institutional model expressed by InstAL has the following characteristics (summarised from [27]):

1. It distinguishes between external world and internal institutional events. The former act as triggers for institutional action, while the latter can have associated permission and power to reflect whether an action is allowed and whether the performance of some action has an institutional effect.
2. The generation relation *G* implements the institutional state dependent recognition of a world event as an institutional event.
3. The consequence relation *C*, triggered by an institutional event, implements the institutional state dependent addition or deletion of (inertial) facts.
4. In addition, there are rules for non-inertial facts such that each is true only if some associated condition over the institutional state is true. This allows for the recognition of situations whose constituent facts may be initiated or terminated by events at different points in time.

The institutional state comprises four kinds of facts, namely those pertaining to the $\mathcal{D}$omain being modelled, $\mathcal{P}$ermissions and Po$\mathcal{W}$ers associated with actions and $\mathcal{O}$bligations arising from actions. Although the approach has similarities with the Event Calculus, there are two main differences: (i) InstAL uses a two-level event structure where external events generate (possibly multiple) institutional events which initiate and terminate institutional facts, and (ii) the computational model is realized in Answer Set Prolog rather than conventional Prolog, which allows exploration of all possible traces over a finite number of steps or just a single step institutional evaluation. A detailed description of the model and the language is given in [12] and in [11].

## 5 Case Study Analysis

The runway incursion scenario described in section 3 was modelled according to the three different safety analysis approaches: STAMP, Agent-Based Modelling (ABM) and

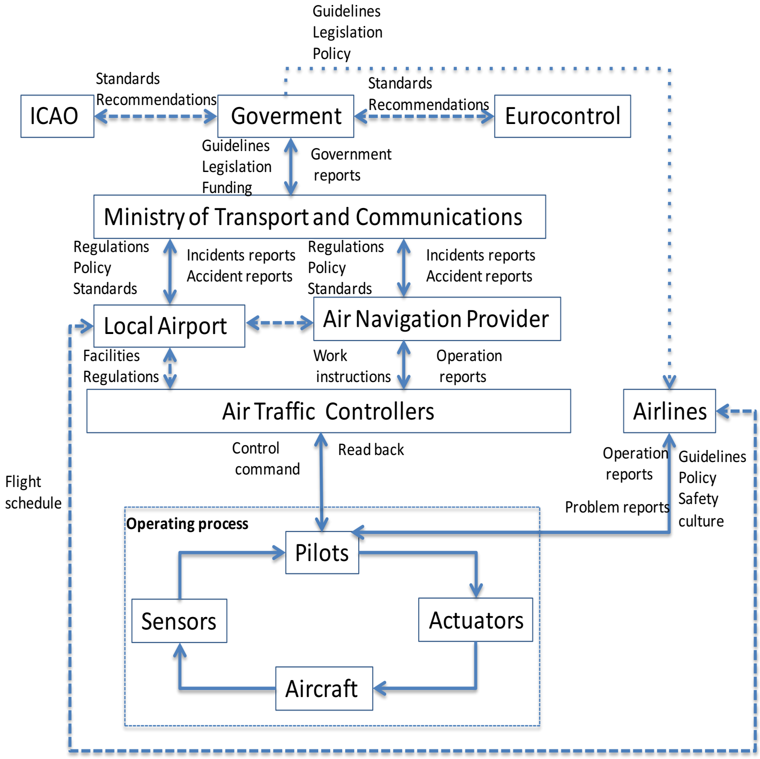

**Fig. 2.** General hierarchical safety control structure for the case study.

institutional modelling. As concrete examples of each of the approaches, CAST analysis was selected for STAMP approach, TTL in conjunction with LEADSTO language for ABM and the InstAL language for institutional modelling. The three models' outcomes are described in the following subsections.

## 5.1 CAST Model

First, the general hierarchical control structure of air traffic operations in the country of the incident was constructed according to steps 3 and 7 of CAST technique listed in subsection 4.1, starting from the government down to the aircraft (see 2).

The rectangles correspond to actors at different levels of aggregation. Control flows are represented by the solid or dotted lines and communication flow by the dashed lines. At the highest level of this structure, the International Civil Aviation Organisation (ICAO) provides international standards for air traffic operations and communications. These guidelines are considered by local governments of the member-countries of the ICAO. Eurocontrol is a European organisation that provides safety guidelines for air traffic operations in Europe. The country where the incident occurred is a member of Eurocontrol. Recommendations concerning air traffic safety should be adopted by the

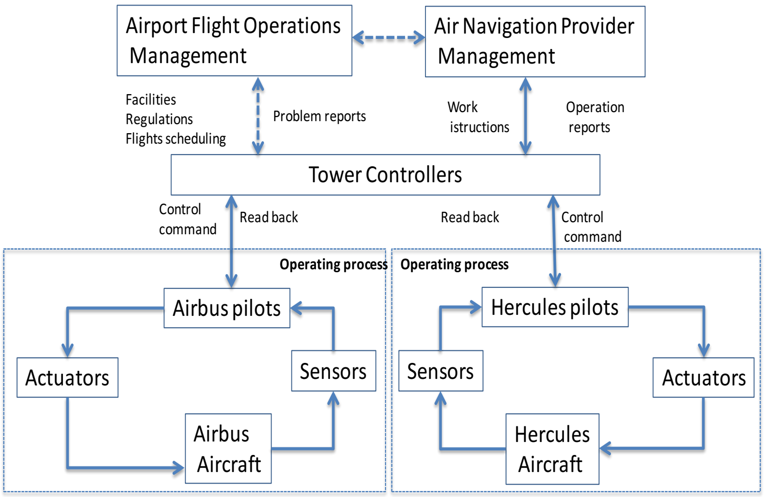

**Fig. 3.** Proximate safety control structure based on the general control structure.

governments of country-members of Eurocontrol. The government of the country participates in the decision-making processes that take part within the ICAO and Eurocontrol. This government provides the guidelines of the policy concerning transportation and funding for the airport and air traffic control organisations within the country. At a lower level, the Ministry of Transport and Communications is responsible for making uniform rules and standards for air traffic operations and functioning of airports. Air navigation providers are responsible for work instructions, procedures and guidelines regarding functioning for air traffic controllers, while airports provide the facilities for air traffic controllers. Pilots are directly controlled by air traffic controllers that communicate clearances for different operations to the pilots. The operating process of steering aircraft (see bottom of Figure 2) is directly controlled by the pilots, using sensors and actuators. Airlines execute control over their pilots by means of setting general guidelines for the adherence for procedures and for aircraft exploitation. These guidelines are in turn provided by the government of the country in question. The connection between the government and the airlines is represented by a dotted line in Figure 2, as the airlines company may be from another country.

Taking this general control structure into consideration along with the proximate events (in space and time) leading to the incident, a specific control structure for the incident was constructed (steps 4 and 5 described in Section 4.1); see Figure 3. In this structure there are two proximate operating processes: pilots controlling the Airbus and pilots controlling the Hercules. Tower controllers are involved for giving instructions and commands to the pilots of both aircraft during the incident. The Airport Flight Operation Management department of the airport communicates flight schedule changes to the controllers and in collaboration with the Air Navigation Provider Management it defines runway usage depending on traffic flow, weather conditions and work activities in the airport.

**Hercules Pilot Safety requirements and constraints**

1. Ensure safety of the aircraft, its crew, cargo and passengers while piloting aircraft
2. Complete a thorough pre-flight inspection of the aircraft
3. Ensure all safety systems are working properly
4. Ensure all information on the route, weather, passengers and aircraft is received
5. Calculate the required runway distance depending on the weather conditions
6. Consider the effects of wind and engine performance on the aircraft's fuel burn to ensure it reaches its destination safely
7. Ensure the fuel levels balance safety with economy and supervise loading and fuelling of the aircraft
8. Complete flight plans taking all information into consideration
9. Communicate with air traffic control before take-off and during flight and landing
10. Brief the cabin crew before the flight and maintaining regular contact throughout the flight
11. Report and communicate problems arising during flight to air traffic controllers
12. Ensure compliance of all laws and regulations
13. Know all limitations applicable to the aircraft (max airspeeds for gear and flaps, max takeoff & landing weights, max temps for the engines, etc)
14. Understand and interpret data from instruments and controls
15. Understand and interpret instructions of air traffic controllers
16. Follow commands of air traffic controllers
17. Make regular checks on the aircraft's technical performance and position, on weather conditions and air traffic during flight
18. Communicate with passengers using the public address system
19. React quickly and appropriately to environmental changes and emergencies

**Context**

1. Routine take off procedure at this non-busy European airport created an expectation that a line-up clearance is immediately followed by take off clearance Inadequate control actions
2. Non-compliance to take off procedure rules: pilot did not ensure that the read back of take off clearance was received by the tower controller

**Mental Model Flaws**

1. Interpretation of line-up clearance as take off clearance

The following safety constraint on the whole system according to step 1 of the CAST methodology was identified: "The system safety control structure must prevent collision of aircraft". To achieve that, a safe aircraft separation should be maintained according to existing standards. This entails two main lower level constraints (step 2): 1) Air traffic controllers should monitor traffic flow, make plans and give instructions to the pilots, ensuring that the instructions are interpreted properly; 2) Pilots should follow the commands of air traffic controllers and ensure that the commands are read back. Further analysis was performed to investigate the roles of human agents in maintaining the system's safety.

The key human components depicted in Figure 3 were selected for further analysis (steps 6 and 7): Airport Flight Operations Management, Air Navigation Provider Management, Tower Controllers, Airbus Pilot and Hercules pilot. For these components, their safety requirements and constraints, context, inadequate control actions and mental model flaws were identified. For the sake of brevity, only one example of this analysis – for the Hercules Pilot component – is shown in the Hercules Pilot box.

### 5.2 LEADSTO Model

This subsection presents an agent-based simulation model of the scenario in the LEADSTO language, which consists of a number of executable dynamic properties (EPs). The runway incursion scenario is modelled from the agent perspective with a focus on agents' observations, beliefs and communications. This model can generate a variety of alternative traces by changes to initial parameter settings. As the first step towards the formalisation of the incident, formal domain ontology was developed in TTL. The ontology includes several sorts (e.g. `AGENT`, `AIRCRAFT`, `RUNWAY`), subsorts relations, elements of sorts (e.g. `airbus` and `hercules` are instances of `AIRCRAFT`) and logical predicates over sorts. Then executable dynamic properties were defined. Figure 4 gives some examples of executable properties in formal LEADSTO notation (for simplicity, the time parameters have been left out). During the modelling process all properties have been conceptually divided into different categories related to agents characteristics: properties related to agents' observations, beliefs, communications and actions. For example, property EP1 refers to agents beliefs and states that, if an agent receives an instruction I1, while it has a strong expectation to receive a similar, but slightly different instruction I2, it will believe that it actually did receive I2. This property can be used to model the fact that the Hercules pilot interpreted his permission for line up as permission for take off. Property EP2 refers to agents communication and determines when the Tower agent communicates a permission to start taxiing to the different aircraft.

The above properties were expressed in LEADSTO language. An execution of the LEADSTO program produces only one trace. An example trace representing the scenario is given in Figure 5, while a more detailed description of the LEADSTO and TTL models and simulation results can be found in [5,7]. In Figure 5, the horizontal axis is time and the vertical axis lists the states that hold in the world. To avoid over-crowding, the atoms that represent observations and beliefs of the agents are not shown. xsHercules misinterprets the line up command from the tower and initiates take off without take off clearance, as illustrated by`performed(hercules_pilot, take_off_from (runway_36))` that holds from time point 15 until time point 21. There is no atom that states that take off clearance from the Tower is communicated to the Hercules. At the same time, the clearance for take-off is given to the Airbus aircraft that almost simultaneously initiates take off from the crossing runway at time point 20. The Tower observes the conflict situation (this atom is not shown in the trace) and communicates a "STOP" signal to the Airbus at time point 24. As a result, the pilot of the Airbus aborts the take-off at time point 27 and a severe collision is prevented by this event. It is an example of a case when a hazardous situation created by one wrong action of one agent can be corrected by another action of another agent. If atom

| | |
|---|---|
| **EP1 – Communication misinterpretation** | incoming_communication(A:Agent, I1:Action, R:Roadway)<br>& belief(A:Agent, similarity(I1: Action, I2: Action))<br>& I1 $\neq$ I2<br>& expectation(A:Agent, I2:Action)<br>$\rightarrow$ belief(A:Agent, I2:Action, R:Roadway) |
| **EP2 – Tower: Taxiing request communication** | belief(A:Agent, is_at_position(B:Aircraft, S: Startingpoint))<br>& belief(A:Agent, is_adjacent_to(T:Taxiway, S: Startingpoint))<br>& belief(A:Agent, is_available(T:Taxiway))<br>& belief(A:Agent, has_role(tower))<br>$\rightarrow$ communicate_from_to(A:Agent, B:Aircraft, start_taxiing(T:Taxiway)) |
| **GP1 – No simultaneous take-offs at crossing runways** | **Informally:**<br>There is no trace m, with time points t1, t2, agents a1, a2, and runways r1, r2, such that agent a1 performs a take-off on runway r1 at time t1 and agent a2 performs a take-off on runway r2 at time t2 and runway r1 and r2 cross and the difference between t1 and t2 is less than or equal to d.<br>**Formally:**<br>$\neg\exists$m:TRACE : $\exists$ t1,t2:TIME, a1,a2:AGENT, r1,r2:RUNWAY<br>state(m, t1) $\models$ performed(a1, take_off_from(r1))<br>state(m, t2) $\models$ performed(a2, take_off_from(r2))<br>state(m, t1) $\models$ world_state(crossing_ways(r1, r2))<br>$\mid$ t1 $-$ t2 $\mid\leq d$ |

**Fig. 4.** Examples of properties captured in the LEADSTO model

`expectation(hercules_pilot, start_take_off)` is removed from the initial declarations, a different simulation scenario is generated where Hercules does not initiate the wrong take off action. By means of changing some time parameters or initial states, the LEADSTO model can be used for the exploration of hypothetical 'what-if' scenarios.

For the purpose of model verification, various global dynamic properties for this organisation are formalised, and implemented in the TTL Checker tool which allows for automatic verification of the properties of interest over the LEADSTO trace(s). In this particular case, a number of higher level properties of the trace in Figure 5 are checked. One example of a global property in a semiformal and formal notation is GP1 in Figure 4. The property expresses one safety related constraint imposed on the organisation: there should be no take-offs within the same time interval on crossing runways. This property does not hold in the given trace since the take-offs of Hercules and Airbus occur within the same time interval.

### 5.3 InstAL Model

We now describe the formalisation and implementation of the aviation incident scenario in InstAL. Like LEADSTO, which uses temporal logic, the basic concepts are expressed in InstAL as predicates that can be derived from other properties. Entities

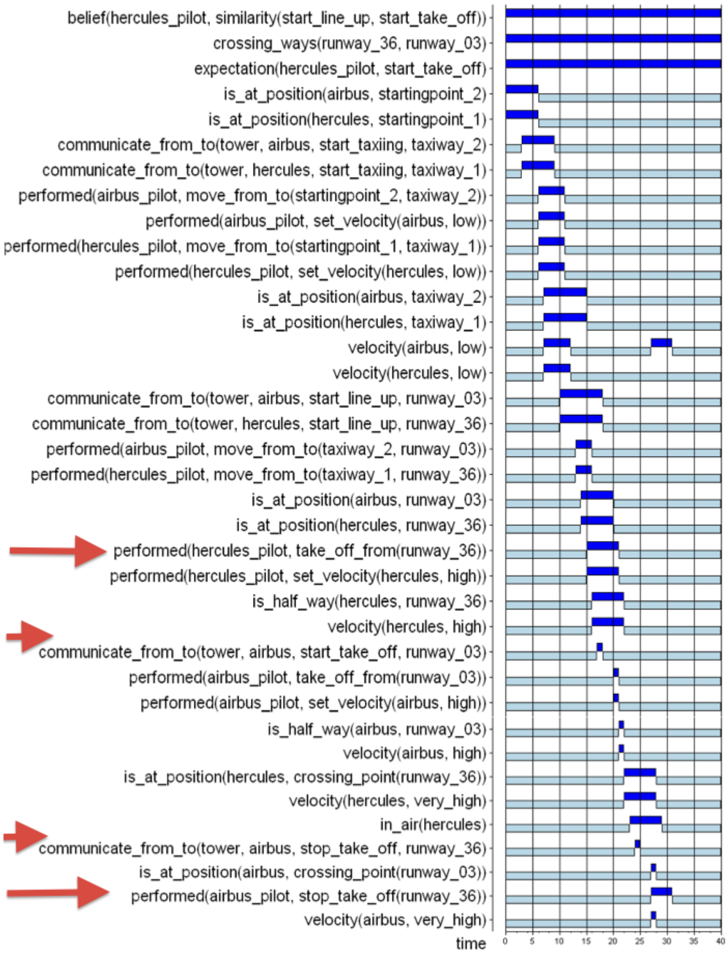

**Fig. 5.** LEADSTO trace shows hercules taking off without take off clearance.

within predicates are variables that are instantiated according to the domain declarations. Thus, in the examples below A is of type Agent, which is either `hercules1` or `airbus1`; Roadway represents one of the modelled roadways and is instantiated from the following set: {`taxiway1, taxiway2, runway1, runway2`}. Three agent actions are modelled: {`taxi, lineUp, takeOff`}. The focus of the InstAL model is on the violation of organisational norms rather than on the occurrence of (world) events, since InstAL emphasises how external events change the state of the institutional model. For this reason, the InstAL formalisation considers the violation of institutional norms by the Hercules agent and is limited to modelling the behaviour of two agents: `hercules1` and `airbus1`. The model demonstrates which sequence of events may lead to state `conflict(Agent1, Agent2)` that compromises organisational safety. The state of `conflict` may be caused by a combination of various other states. Here it is defined as a disjunction of two situations:

```
conflict(Agent1, Agent2) when situation1(Agent1,Agent2);
conflict(Agent1, Agent2) when situation2(Agent1,Agent2);
```

`Situation1` occurs when both agents are in a lining up state on the crossing runways and `Situation2` occurs when one of the agents is in a lining up state and the other one in a taking off state on the crossing runways:

```
situation1(Agent1,Agent2) when crossing(Runway1, Runway2),
                               liningUp(Agent1, Runway1),
                               liningUp(Agent2, Runway2);

situation2(Agent1,Agent2) when crossing(Taxiway1,Taxiway2),
                               liningUp(Agent1, Runway1),
                               takingOff(Agent2, Runway2);
```

JAP: can't parse this sentence

An example of a rule where line up clearance event `lclearance(A, Action)` initiates lining up and hold before runway states `liningUp(A, Roadway), rhold(A, Roadway)` is:

```
lclearance(A, Action) initiates
   liningUp(A, Roadway), rhold(A, Roadway)
   if hold(A, Roadway), runway(Roadway), not conflict(A, A2);
```

The InstAL model offers the possibility to explore which events sequences lead to the state of `conflict`. The answer set solving process first grounds the model over all values that the variables can take. While, this allows for an exhaustive analysis of all possible events in all possible orders, many of the traces do not make sense or are not of interest. By specifying a range of constraints, the grounding process is constrained and the number of traces can be reduced to those of interest. In this specific example we defined the constraints to grasp the situation where a particular sequence of events takes place.

We are interested in the trace of length 7 with 7 events: at time 0 `hercules1` receives clearance to taxi, at time 1 airbus receives clearance to taxi, then hercules and airbus stop before their runways, they receive lining up clearance further and, finally, at time point 6 hercules enters runway2. These event sequence constraints result in the construction of only one trace shown in Figure 6: this shows external and institutional events (in italics) and time instants denoted by $S_i$. The states that are initiated at time instants are in bold and the terminated states are struck through. The taxi clearance events at the beginning of the trace are omitted to enhance readability. As can be seen in Figure 6, entering the runway by `hercules1` is evaluated as an institutional violation of rules. Moreover, the `conflict(airbus1, hercules1)` fluent holds at time instants $S_6$ and $S_7$. It is caused by the two conflict situations: `situation1` when two airplanes line up on the crossing runways and `situation2` when hercules initiates take of at time instant $S_6$.

An alternative visualisation of the trace, that is more similar to the LEADSTO model appears in Figure 7. Here the dark bars represent the states (or facts) that are

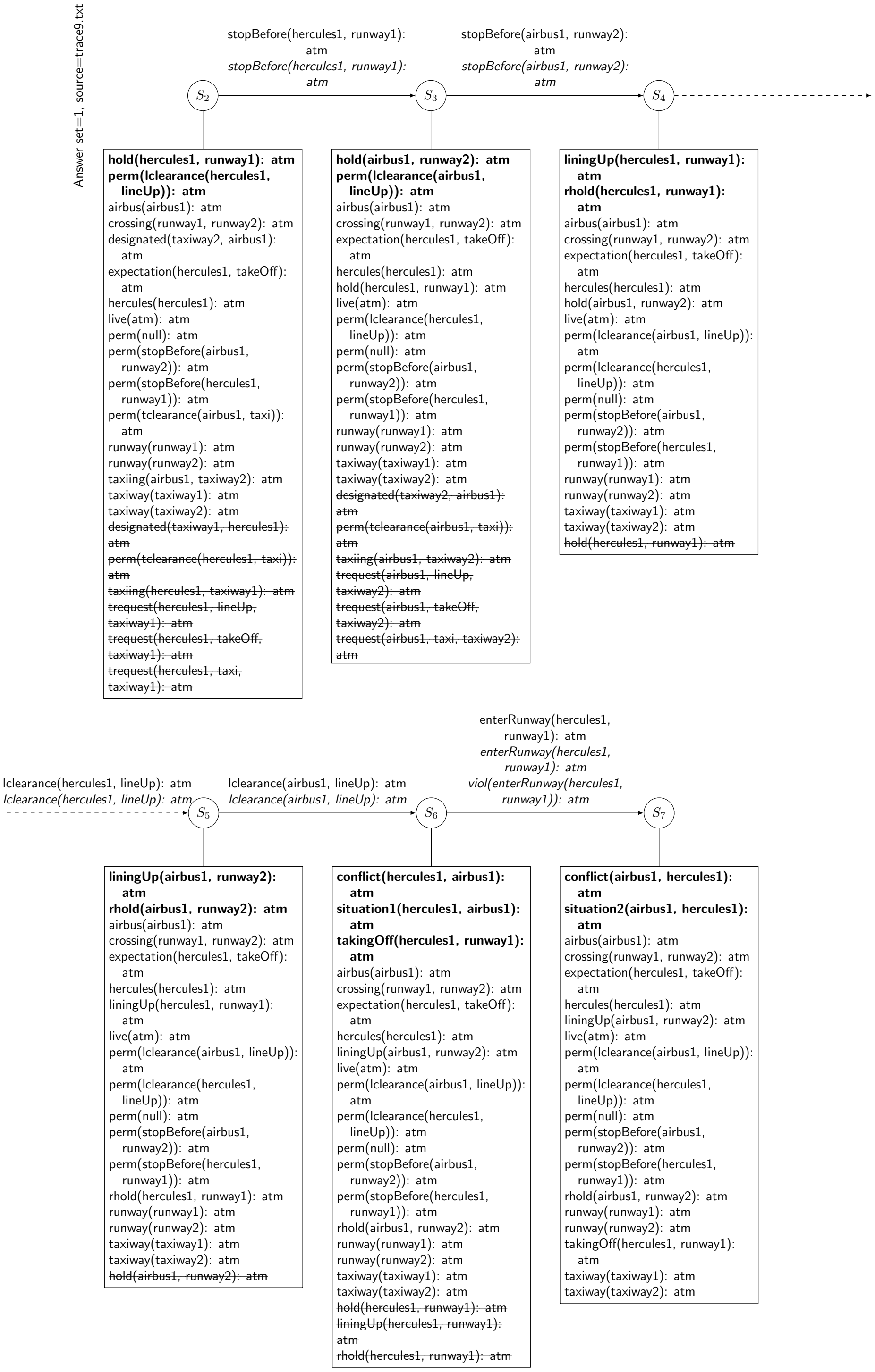
Answer set=1, source=trace9.txt
stopBefore(hercules1, runway1): atm
*stopBefore(hercules1, runway1): atm*
stopBefore(airbus1, runway2): atm
*stopBefore(airbus1, runway2): atm*
$S_2$
$S_3$
$S_4$
**hold(hercules1, runway1): atm**
**perm(lclearance(hercules1, lineUp)): atm**
airbus(airbus1): atm
crossing(runway1, runway2): atm
designated(taxiway2, airbus1): atm
expectation(hercules1, takeOff): atm
hercules(hercules1): atm
live(atm): atm
perm(null): atm
perm(stopBefore(airbus1, runway2)): atm
perm(stopBefore(hercules1, runway1)): atm
perm(tclearance(airbus1, taxi)): atm
runway(runway1): atm
runway(runway2): atm
taxiing(airbus1, taxiway2): atm
taxiway(taxiway1): atm
taxiway(taxiway2): atm
~~designated(taxiway1, hercules1): atm~~
~~perm(tclearance(hercules1, taxi)): atm~~
~~taxiing(hercules1, taxiway1): atm~~
~~trequest(hercules1, lineUp, taxiway1): atm~~
~~trequest(hercules1, takeOff, taxiway1): atm~~
~~trequest(hercules1, taxi, taxiway1): atm~~
**hold(airbus1, runway2): atm**
**perm(lclearance(airbus1, lineUp)): atm**
airbus(airbus1): atm
crossing(runway1, runway2): atm
expectation(hercules1, takeOff): atm
hercules(hercules1): atm
hold(hercules1, runway1): atm
live(atm): atm
perm(lclearance(hercules1, lineUp)): atm
perm(null): atm
perm(stopBefore(airbus1, runway2)): atm
perm(stopBefore(hercules1, runway1)): atm
runway(runway1): atm
runway(runway2): atm
taxiway(taxiway1): atm
taxiway(taxiway2): atm
~~designated(taxiway2, airbus1): atm~~
~~perm(tclearance(airbus1, taxi)): atm~~
~~taxiing(airbus1, taxiway2): atm~~
~~trequest(airbus1, lineUp, taxiway2): atm~~
~~trequest(airbus1, takeOff, taxiway2): atm~~
~~trequest(airbus1, taxi, taxiway2): atm~~
**liningUp(hercules1, runway1): atm**
**rhold(hercules1, runway1): atm**
airbus(airbus1): atm
crossing(runway1, runway2): atm
expectation(hercules1, takeOff): atm
hercules(hercules1): atm
hold(airbus1, runway2): atm
live(atm): atm
perm(lclearance(airbus1, lineUp)): atm
perm(lclearance(hercules1, lineUp)): atm
perm(null): atm
perm(stopBefore(airbus1, runway2)): atm
perm(stopBefore(hercules1, runway1)): atm
runway(runway1): atm
runway(runway2): atm
taxiway(taxiway1): atm
taxiway(taxiway2): atm
~~hold(hercules1, runway1): atm~~
lclearance(hercules1, lineUp): atm
*lclearance(hercules1, lineUp): atm*
lclearance(airbus1, lineUp): atm
*lclearance(airbus1, lineUp): atm*
enterRunway(hercules1, runway1): atm
*enterRunway(hercules1, runway1): atm*
*viol(enterRunway(hercules1, runway1)): atm*
$S_5$
$S_6$
$S_7$
**liningUp(airbus1, runway2): atm**
**rhold(airbus1, runway2): atm**
airbus(airbus1): atm
crossing(runway1, runway2): atm
expectation(hercules1, takeOff): atm
hercules(hercules1): atm
liningUp(hercules1, runway1): atm
live(atm): atm
perm(lclearance(airbus1, lineUp)): atm
perm(lclearance(hercules1, lineUp)): atm
perm(null): atm
perm(stopBefore(airbus1, runway2)): atm
perm(stopBefore(hercules1, runway1)): atm
rhold(hercules1, runway1): atm
runway(runway1): atm
runway(runway2): atm
taxiway(taxiway1): atm
taxiway(taxiway2): atm
~~hold(airbus1, runway2): atm~~
**conflict(hercules1, airbus1): atm**
**situation1(hercules1, airbus1): atm**
**takingOff(hercules1, runway1): atm**
airbus(airbus1): atm
crossing(runway1, runway2): atm
expectation(hercules1, takeOff): atm
hercules(hercules1): atm
liningUp(airbus1, runway2): atm
live(atm): atm
perm(lclearance(airbus1, lineUp)): atm
perm(lclearance(hercules1, lineUp)): atm
perm(null): atm
perm(stopBefore(airbus1, runway2)): atm
perm(stopBefore(hercules1, runway1)): atm
rhold(airbus1, runway2): atm
runway(runway1): atm
runway(runway2): atm
taxiway(taxiway1): atm
taxiway(taxiway2): atm
~~hold(hercules1, runway1): atm~~
~~liningUp(hercules1, runway1): atm~~
~~rhold(hercules1, runway1): atm~~
**conflict(airbus1, hercules1): atm**
**situation2(airbus1, hercules1): atm**
airbus(airbus1): atm
crossing(runway1, runway2): atm
expectation(hercules1, takeOff): atm
hercules(hercules1): atm
liningUp(airbus1, runway2): atm
live(atm): atm
perm(lclearance(airbus1, lineUp)): atm
perm(lclearance(hercules1, lineUp)): atm
perm(null): atm
perm(stopBefore(airbus1, runway2)): atm
perm(stopBefore(hercules1, runway1)): atm
rhold(airbus1, runway2): atm
runway(runway1): atm
runway(runway2): atm
takingOff(hercules1, runway1): atm
taxiway(taxiway1): atm
taxiway(taxiway2): atm

**Fig. 6.** InstAL trace showing the violation of an institutional norm: Hercules initiates take off without take off clearance, indicated by the violation arising from *enterRunway* occurring at time 6 and the identified **conflict** arising from **hercules1 takingOff** in $S_6$.

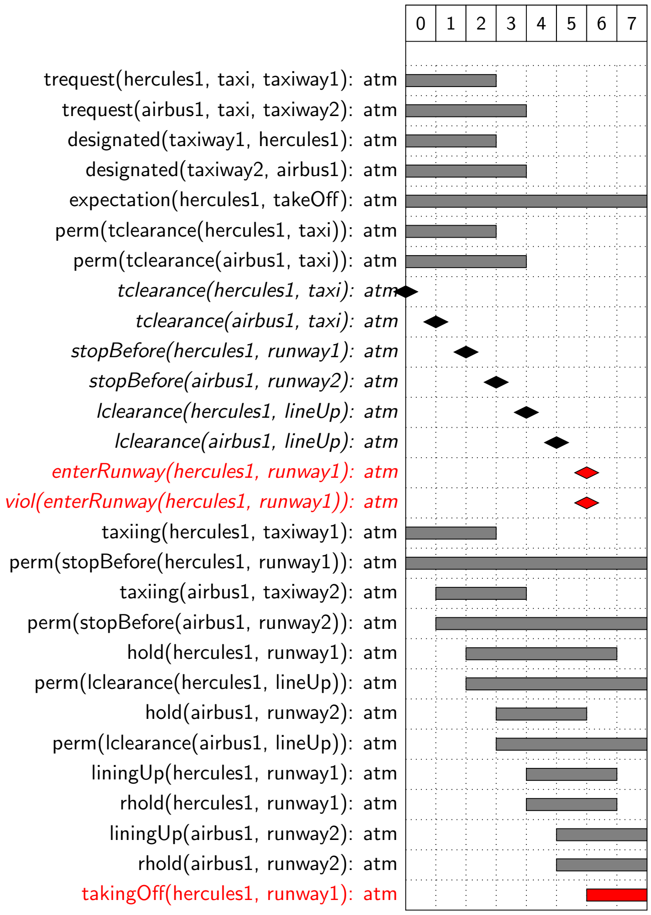

**Fig. 7.** Inst*AL* trace shows the violation of an institutional norm: hercules initiates take off without take off clearance, indicated by the violation arising from *enterRunway* occurring at time 6 and the presence of hercules1 takingOff in states 6 and 7 (key events/states are highlighted in red).

initiated and terminated across 8 time instances and black diamonds represent institutional events. If the initial fact that Hercules pilot has the expectation of take-off clearance immediately after line up clearance is removed, it results in a trace with only one type of `conflict` fluent – the one that is caused by `situation1`, when two airplanes line up on crossing runways. By changing initial conditions or conditions of the rules where states are initiated or terminated by events in this model, one can explore how undesirable, safety-compromising states can be achieved or prevented from occurring. One should note, however, that the external events are deterministic and cannot be prevented from happening in the model, only state occurrence can be influenced by changing initial facts and by adding or removing conditions in rules.

## 6 Comparative Analysis of Results

The models developed according to the three approaches represent the occurrence of an incident from slightly different perspectives, which have their advantages and disadvantages. We put forward a brief summary of the main differences between the approaches according to the selected aspects relevant for the scope of the current work. A brief overview of the methods' performance within these selected aspects is presented in Table 2. We used the following notation within the Table: "++" represents an excellent performance of the method given an aspect of evaluation, "+" represents good performance, "+/-" stands for a moderate satisfaction of a comparison criteria, "-" denotes poor satisfaction and "--" represents very poor satisfaction.

1. *Levels of analysis*: CAST method is the most comprehensive in this respect as STAMP methodology covers all levels of analysis, starting from the roles of individuals components and going up to a macro level of an organisation and taking even a broader context at a governmental and regulatory level. LEADSTO as an example of agent-based modelling approach produces a model at a micro level of individual agents without an explicit overview of meso and macro levels; InstAL represents both agents and institutional events and norms.
2. *Taxonomy of failures*: CAST method was specifically designed to analyse safety critical systems and provides a taxonomy of failures within the STAMP framework; LEADSTO produces a model with emergent system's behaviour which is not very concretely defined in the beginning and this has no predefined taxonomy of failures; InstAL represents failures as a violation of organisational norms which in some sense does provide a taxonomy of failures.
3. *Quantitative representation*: Here only LEADSTO has an ability to express quantitative concepts, such as mathematical formulas, numbers and probabilities, though in the current model this capability was not utilised given the nature of safety issues within the case study under consideration; both CAST and InstAL do not incorporate numerical expressions or mathematical formulas
4. *Qualitative representation*: in this respect, all three methods are able to provide a qualitative representation of factors leading to an incident
5. *Time dynamics expressiveness*: Regarding the time dynamics expressivity which is one of the most important aspects of the given domain, LEADSTO/TTL appeared to be more expressive in this respect since it has quantitative time parameters that regulate temporal dependences between states; InstAL does not have such parameters, though facts initiation in InstAL always occurs immediately after the event occurrence and facts duration is regulated by other events occurrences in a sequence; CAST and STAMP framework does not focus on time dynamics, but rather on a static design and structure of a system with proper feedback and control loops
6. *Events representation*: here InstAL has a very clear events representation advantage while LEADSTO has a moderate representation which is mixed with agents' states, perceptions and actions; CAST does not focus on this aspect as it is clearly related to time aspect and STAMP framework does not consider situational time dynamics context

7. *Formal semantics*: both InstAL and LEADSTO methods have formal semantics behind them which allows for formal specification and automatic property checking which can be crucial for modern socio-technical systems where complexity cannot be tackled by non-formal methods only; note that not all agent-based approaches satisfy this criteria and there are some methods which are used for pure simulation of events without any automatic verification or property checking; CAST is not underpinned by formal semantics
8. *Amount of training*: in this respect we admit that CAST outperforms the other two methods because apart from systems thinking, for an application of LEADSTO/TTL and InstAL some background training in programming and computation is needed; however, from a point of view of safety experts specialised in areas different from engineering (e.g. human factors), none of the methods is easily comprehensible - this conclusion is drawn based on conversations with human factors focused safety experts and based on social and exact sciences interdisciplinary background of one of the co-authors of the current paper
9. *Graphical representation*: both CAST and InstAL have different types of visualisation of modelling results and outcomes while LEADSTO has only one type of visualisation which is not easily comprehensible for someone not familiar with an approach
10. *Data requirements*: since CAST comprises all levels of analysis, from micro to macro, it requires more data to complete this comprehensive analysis; data requirements for LEADSTO and InstAL models are lower since they do not focus on all levels
11. *Time resources*: all three methods are quite time consuming due to either the amount of data for processing (CAST) or due to the amount of time needed for models construction, programming or running simulations (LEADSTO and InstAL)
12. *Additional resources (software etc)*: CAST analysis does not require any additional resources apart from access to organisational archives and reports, organisational structure and hierarchy which might be easily accessible by an internal safety expert, though quite problematic for an external expert given company confidentiality issues; both LEADSTO and InstAL require IT technologies: dedicated software packages and a computer to run it on: specifically, InstAL requires several pieces of diverse software
13. *Main versus complementary method*: given the coverage of all levels of analysis by CAST method and STAMP framework, it can be used as a stand-alone approach and can be applied for full safety analysis process; as far as LEADSTO and InstAL concern, they can rather strengthen, complement or be utilised within other safety analysis approaches using their framework, inclusive STAMP

Table 2 provides a concise summary of the evaluation of the three approaches.

## 7 Discussion and Conclusions

The aim of this paper is to compare and contrast three systemic approaches for incident and accident analysis in aviation through their application to the analysis of a case study

**Table 2.** Approaches evaluation according to the specified aspects

| **Aspect** | **CAST** | **TTL/LEADSTO** | **InstAL** |
|---|---|---|---|
| 1. Levels of analysis | ++ (all) | -(micro) | +/- (micro/meso) |
| 2. Taxonomy of failures | + | - | +/- |
| 3. Quantitative representation | - | + | - |
| 4. Qualitative representation | + | + | + |
| 5. Formal semantics | - | + | + |
| 6. Events representation | - | + | ++ |
| 7. Time dynamics | - | ++ | + |
| 8. Amount of training | +/- | - | - |
| 9. Graphical representation | + | - | + |
| 10. Data requirements | +/- | + | + |
| 11. Time recourses | - | - | - |
| 12. Additional resources (software etc) | + | - | -- |
| 13. Main vs complementary | main | compl | compl |

incident in air traffic management. The selected approaches of (i) STAMP-based CAST analysis; (ii) LEADSTO simulation based on agent-based modelling; and (iii) InstAL model underpinned by institutional modelling; were applied to a real-life case study in the aviation domain. Apart from focusing on the most influential systemic approaches in safety literature, we deliberately selected qualitatively different approaches in terms of their data representation and the presence of mathematical formalisms, in order to explore not only their differences, but also their potential complementarity. An application of formal institutional modelling to safety analysis in transportation is innovative here. This type of modelling is widely applied in legal reasoningand opens an interesting perspective for accident or incident analysis from a legal point of view.

JAP: are there any citations to support this elsewhere in the paper? if not, may be criticised

In terms of an accident analysis itself, all three methods allow for qualitative representation of an incident, although two of them, TTL/LEADSTO and InstAL, have underlying formal semantics which empower these approaches to perform automatic checking and verification of complex systems where incidents occur. TTL/LEADSTO and InstAL are also capable of time dynamics modelling and representation which is very important for this particular case study that represents a highly dynamic environment. STAMP-based CAST analysis comprises the analysis of a system at all organisational levels, from actors at a micro level to governmental and regulatory bodies at a macro level and is very comprehensive. From a usability point of view, it is clear that a STAMP-based approach has advantages over TTL/LEADSTO and InstAL because it requires less training and additional resources, such as dedicated software. Given the scope of STAMP-based analysis, this approach can be used as a stand-alone accident analysis approach by safety experts, while the other two can be regarded as complementary methods in order to gain insights into some specific behaviour of a systems, rather than as a main approach for accident analysis. All three approaches are quite time consuming, which can be a disadvantage when quick and practical results are expected of a safety analyst.

No single method can be ideal and suitable for all possible contexts, depending on a modelling purpose and the nature of an accident of an incident, a particular approach or a combination of several approaches can be considered. For example, for analysis of a more closed and less dynamic system at all organisational levels, STAMP can be a good option. For analysis of more complex, open and highly dynamic contexts, a complementary application of formal methods and simulation (especially for a prospective analysis to predict future accidents) could be highly advantageous. To improve method capabilities and expressive power, integrated approaches can be developed which combine relevant complementary features of several different approaches. Integrated approaches are recommended for the analysis of a series of complex incidents [32]. The current paper provides a useful framework for this type of integration. For example, for a comprehensive safety analysis of complex socio-technical systems, formal methods can be combined with a STAMP model.

In future, the possibilities of using such integrated approaches for safety analysis, with the current case study and new case studies in ATM or other domains, will be explored. Regarding the application of systemic methods in general by safety experts, a serious research and practice gap exists [39], due to different contexts and goals in which academic researchers and safety practitioners communities operate. Future research should also address the possibilities of adoption and adjustment of existing systemic methods for the needs of practitioners.

## References

1. Hanan Altabbakh, Mohammad A AlKazimi, Susan Murray, and Katie Grantham. STAMP–holistic system safety approach or just another risk model? *Journal of Loss Prevention in the Process Industries*, 32:109–119, 2014.
2. Ersin Ancel, Ann T Shih, Sharon M Jones, Mary S Reveley, James T Luxhøj, and Joni K Evans. Predictive safety analytics: inferring aviation accident shaping factors and causation. *Journal of Risk Research*, 18(4):428–451, 2015.
3. Ludwig Benner. Rating accident models and investigation methodologies. *Journal of safety research*, 16(3):105–126, 1985.
4. HAP Blom, GJ Bakker, PJG Blanker, J Daams, MHC Everdij, and MB Klompstra. Accident risk assessment for advanced air traffic management. *Progress in Astronautics and Aeronautics*, 193:463–480, 2001.
5. T. Bosse and N. Mogles. Studying aviation incidents by agent-based simulation and analysis. In *Proceedings of the Fifth International Conference on Agents and Artificial Intelligence at ICAART 2013*, 2013.
6. T. Bosse and N. Mogles. Studying aviation incidents by agent-based simulation and analysis. In *BNAIC 2013: Proceedings of the 25th Benelux Conference on Artificial Intelligence, Delft, The Netherlands, November 7-8, 2013*. Delft University of Technology (TU Delft); under the auspices of the Benelux Association for Artificial Intelligence (BNVKI) and the Dutch Research School for Information and Knowledge Systems (SIKS), 2013.
7. T. Bosse and N. M. Mogles. An agent-based approach for accident analysis in safety critical domains: A case study on a runway incursion incident. In *Transactions on Computational Collective Intelligence XVII*, pages 66–88. Springer, 2014.
8. Tibor Bosse, Catholijn M Jonker, Lourens Van der Meij, Alexei Sharpanskykh, and Jan Treur. Specification and verification of dynamics in agent models. *International Journal of Cooperative Information Systems*, 18(01):167–193, 2009.

9. Tibor Bosse, Catholijn M Jonker, Lourens Van Der Meij, and Jan Treur. A language and environment for analysis of dynamics by simulation. *International Journal on Artificial Intelligence Tools*, 16(03):435–464, 2007.
10. Australian Transport Safety Bureau. Analysis, causality and proof in safety investigations aviation research and analysis report ar-2007-053. *Canberra City: Australian Transport Safety Bureau*, 2008.
11. Owen Cliffe. *Specifying and analysing institutions in multi-agent systems using answer set programming*. PhD thesis, University of Bath, June 2007.
12. Owen Cliffe, Marina De Vos, and Julian Padget. Answer set programming for representing and reasoning about virtual institutions. In *Computational Logic in Multi-Agent Systems*, pages 60–79. Springer, 2007.
13. David L Cooke. A system dynamics analysis of the westray mine disaster. *System Dynamics Review*, 19(2):139–166, 2003.
14. Marina De Vos, Julian Padget, and Ken Satoh. Legal modelling and reasoning using institutions. In *New Frontiers in Artificial Intelligence*, pages 129–140. Springer, 2011.
15. Erik Hollnagel and Orjan Goteman. The functional resonance accident model. *Proceedings of cognitive system engineering in process plant*, 2004:155–161, 2004.
16. Barry Kirwan. *A guide to practical human reliability assessment*. CRC Press, 1994.
17. Kevin B Korb and Ann E Nicholson. Chapman & hall. *CRC.?Bayesian Artificial Intelligence*, 2004.
18. R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4:67–95, 2002.
19. Jean-Christophe Le Coze. Disasters and organisations: From lessons learnt to theorising. *Safety science*, 46(1):132–149, 2008.
20. Nancy Leveson. A new accident model for engineering safer systems. *Safety science*, 42(4):237–270, 2004.
21. Nancy Leveson. Engineering a safer world: applying systems thinking to safety, 2012.
22. Tingting Li. *Normative Conflict Detection and Resolution in Cooperating Institutions*. PhD thesis, University of Bath, July 2014.
23. Tingting Li, Tina Balke, Marina De Vos, Julian Padget, and Ken Satoh. Legal conflict detection in interacting legal systems. In Kevin D. Ashley, editor, *JURIX*, volume 259 of *Frontiers in Artificial Intelligence and Applications*, pages 107–116. IOS Press, 2013.
24. Jonas Lundberg, Carl Rollenhagen, and Erik Hollnagel. What-you-look-for-is-what-you-find–the consequences of underlying accident models in eight accident investigation manuals. *Safety Science*, 47(10):1297–1311, 2009.
25. Fedja Netjasov and Milan Janic. A review of research on risk and safety modelling in civil aviation. *Journal of Air Transport Management*, 14(4):213–220, 2008.
26. Julian Padget, Emad Eldeen Elakehal, Ken Satoh, and Fuyuki Ishikawa. On requirements representation and reasoning using answer set programming. In Nelly Bencomo, Jane Cleland-Huang, Jin Guo, and Rachel Harrison, editors, *IEEE 1st International Workshop on Artificial Intelligence for Requirements Engineering, AIRE 2014, 26 August, 2014, Karlskrona, Sweden*, pages 35–42. IEEE, 2014.
27. Julian Padget, Emad ElDeen Elakehal, Tingting Li, and Marina De Vos. *InstAL: An Institutional Action Language*, pages 101–124. Springer International Publishing, Cham, 2016.
28. W. Pieters, J. Padget, F. Dechesne, V. Dignum, and H. Aldewereld. Effectiveness of qualitative and quantitative security obligations. *Journal of Information Security and Applications*, 22:3–16, 2015.
29. Jens Rasmussen. Risk management in a dynamic society: a modelling problem. *Safety science*, 27(2):183–213, 1997.
30. James Reason. *Human error*. Cambridge university press, 1990.

31. James Reason. *Managing the risks of organizational accidents*. Routledge, 2016.
32. Paul M Salmon, Miranda Cornelissen, and Margaret J Trotter. Systems-based accident analysis methods: A comparison of Accimap, HFACS, and STAMP. *Safety science*, 50(4):1158–1170, 2012.
33. Robert G Sargent. Model verification and validation. In *Modeling and simulation in the systems engineering life cycle*, pages 57–65. Springer, 2015.
34. W Richard Scott. Organizations and institutions. *Foundations for Organizational Science; Sage Publications: Thousand Oaks, CA, USA*, 1995.
35. Alexei Sharpanskykh and Sybert H Stroeve. An agent-based approach for structured modeling, analysis and improvement of safety culture. *Computational and Mathematical Organization Theory*, 17(1):77–117, 2011.
36. Snorre Sklet. Comparison of some selected methods for accident investigation. *Journal of hazardous materials*, 111(1):29–37, 2004.
37. Doug Smith, Brian Veitch, Faisal Khan, and Rocky Taylor. Understanding industrial safety: Comparing fault tree, bayesian network, and fram approaches. *Journal of Loss Prevention in the Process Industries*, 45:88–101, 2017.
38. S. H Stroeve, H. AP Blom, and GJ Bert Bakker. Systemic accident risk assessment in air traffic by monte carlo simulation. *Safety science*, 47(2):238–249, 2009.
39. Peter Underwood and Patrick Waterson. Systemic accident analysis: examining the gap between research and practice. *Accident Analysis & Prevention*, 55:154–164, 2013.
40. Peter Underwood and Patrick Waterson. Systems thinking, the Swiss Cheese Model and accident analysis: a comparative systemic analysis of the Grayrigg train derailment using the ATSB, AcciMap and STAMP models. *Accident Analysis & Prevention*, 68:75–94, 2014.
41. HA Watson et al. Launch control safety study. *section VII Vol*, 1, 1961.
42. DA Wiegmann and SA Shappell. A human error approach to aviation accident analysis: The human factors analysis and classification system.[book].-[sl]: Farnham. *UK: Ashgate Publishing*, 2003.
43. Jianwen Xiang, Kokichi Futatsugi, and Yanxiang He. Fault tree and formal methods in system safety analysis. In *Computer and Information Technology, 2004. CIT'04. The Fourth International Conference on*, pages 1108–1115. IEEE, 2004.