# Can We Remove the Human from Hazard Analysis?

Robert Alexander; Tim Kelly; University of York; York, England

Keywords: systems of systems, hazard analysis, simulation

## Abstract

Traditionally, hazard analysis has been a manual process performed by safety engineers following a systematic process. This has proved highly effective for existing systems. However, for Systems of Systems of the complexity now being constructed it is becoming increasingly difficult to use these conventional techniques, particularly when considering the effects of multiple concurrent and distributed failures. It therefore seems advisable to investigate how automation can assist in the hazard analysis process. In order to do this, it is necessary to clearly identify the role of the human in hazard analysis, and determine the unique contribution made by human intelligence. This paper, in particular, identifies the importance of creative input and application of domain knowledge in deriving accurate and complete results. From this, it identifies some parts of the hazard analysis process where automation could be used to improve performance. As an example of how this could work, a partially-automated approach to Systems of Systems hazard analysis is presented, and this is illustrated using a military system case study.

## Introduction

A risk-based safety process relies on the identification and analysis of the risks associated with the system. If such a process is to be effective, the analysis must be complete and accurate. If the process is to be affordable, risks must be identified and quantified (if only roughly) early in the system life cycle.

The main method for achieving this involves identifying the set of possible *hazards* that are exhibited by the system, and determining the probability and severity of their occurrence. The UK military standard Defstan 00-56 defines 'hazard' as *"A physical situation or state of a system, often following from some initiating event, that may lead to an accident"* (ref. 1). In order to acquire the set of such hazards, engineers must perform *hazard identification* (Defstan 00-56: *"The process of identifying and listing the hazards and accidents associated with a system"*). In order to determine the risk associate with each hazard, *hazard analysis* must be performed to discover the causes and effects (Defstan 00-56: *"The process of describing in detail the hazards and accidents associated with a system, and defining accident sequences"*).

At present, hazard analysis is primarily performed by engineers using manual techniques such as FHA (ref. 2) or HAZOP (ref. 3). Although this labour is time-consuming (and hence expensive) is has historically been highly effective at finding hazards and their causes, and thereby allowing safe systems to be built.

The safety-critical systems that are being developed today, however, are growing in size and complexity, while public and hence political and legal demands for increased safety are prominent. The increasing use of software is a well-documented part of this increase in complexity, but looking to current and near-future developments new problems are apparent with the increased use of ad-hoc networks and autonomous robotic systems. Systems that exhibit these latter characteristics are commonly described as 'systems of systems', and a discussion of some of the problems they present for safety is provided by the authors in references 4 and 5.

To some extent, problems with analyzing such systems can be dealt with by allocating ever-more resources to conventional hazard analysis approaches. However, given that human intelligence remains a constant, it is hard to have confidence that manual approaches will maintain the same level of completeness and accuracy in the face of increasing complexity. A similar development can be seen in the field of software, where increasing complexity has required programmers to move to progressively higher-level languages and more automated support.

Furthermore, as Venkatasubramanian notes in (ref. 6), most of the work performed by engineers in current manual approaches involves repetition of the same responses to the same situations that they have seen many times before. Such mundane, tedious work dulls awareness and may prevent engineers spending time and effort on generally

novel situations, and may cause them to miss those situations that are genuinely important. Performing manual analysis when an automated alternative is available is a waste of engineer skill and training.

This paper will discuss the potential for resolving these problems via automation of the hazard analysis process, concentrating on the systems of systems case. The following section identifies what it is that humans actually do in typical hazard analysis procedures, and the section after that discusses what parts of that can be automated. Some requirements for an effective automated hazard analysis approach for systems of systems are identified, and existing work described in the literature is then reviewed with respect to these. The authors then describe their work in progress on developing such an approach.

## What do Humans Do in Hazard Analysis?

The core of hazard analysis is the asking of a "What if?" question, where some plausible cause (such as a valve jamming shut) is evaluated in the context of the entire operational system in order to determine what effects it can have (such as causing an uncontrolled increase in pressure in the tank immediately prior to the valve). The overall goal of this questioning is to gain understanding of these cause-effect relationships such that hazards inherent in the current design of the system can be evaluated, and the design (or operational arrangements) of the system changed so that the hazards are prevented or mitigated.

Manual analysis techniques, by necessity, break down the task into a number of steps. This can be illustrated by considering two widely used techniques: Functional Hazard Analysis (FHA) (ref. 2) and HAZOP (ref. 3). A set of steps for each of these, as presented by Pumfrey in (ref. 7), are shown in (Table 1) and (Table 2) respectively.

Table 1 — Steps in FHA (from Pumfrey, reference 7)

| **Step** | **Description** |
|---|---|
| 1 | Identify functions. |
| 2 | For each function identified, suggest failure modes, using the three failure types as prompts. |
| 3 | For each failure mode, consider the effects (at different levels, e.g. function / subsystem / system, if necessary). |
| 4 | Identify and record any actions necessary to improve the design. |

Table 2 — Steps in HAZOP (from Pumfrey, reference 7)

| **Step** | **Description** |
|---|---|
| 1 | Select a flow in a pipeline. |
| 2 | Identify important physical attributes of the flow, such as *pressure, temperature, flow rate, chemical composition* etc. |
| 3 | Consider those deviations prompted by applying each guide word to each property for this line section. |
| 4 | Determine the possible causes of each of these deviations. |
| 5 | Investigate the expected outcome (effect on the plant) of each deviation, taking into consideration operating conditions and other causal factors where necessary, and examining the contribution of protection mechanisms and other mitigation already included in the design. |
| 6 | Decide which deviations are safety problems (i.e. those with both plausible causes and hazardous effects). |
| 7 | For deviations which are not safety problems, record a justification (i.e. explain why the design is acceptable as proposed. |
| 8 | Consider changes to the plant that will remove, or reduce the probability or severity of, hazardous deviations. |
| 9 | Determine whether the cost of the proposed changes is justified. |
| 10 | Agree actions and responsibilities. |
| 11 | Repeat steps 1 to 10 for all other lines in the plant. |
| 12 | Follow up to ensure necessary actions have been taken. |

Although the two techniques have different steps, it can be seen that they break down into similar stages as shown in table 3.

Table 3 — Combined Stages from FHA and HAZOP

| Stage | FFA Equivalent | HAZOP Equivalent |
|---|---|---|
| A) Build / acquire model | 1 | 2 |
| B) Enumerate possible deviations of model | 2 | 3 |
| C) Derive effects of deviations | 3 | 5, 6, 7 |
| D) Consider design changes to prevent or mitigate | 4 | 8, 9, 10, 12 |

HAZOP steps 1 and 11 appear not to fit into this classification, but this is because they are merely instructions to repeat certain other steps. HAZOP step 4 is genuinely excluded, but for the purpose of simplicity this stage of the technique will be ignored in this paper.

As well as performing those activities that are explicit in the given process being followed, engineers invariably perform additional actions in parallel. They validate the model, questioning the predictions that seem to arise from it. If the model is at odds with their understanding of the world and engineering judgment, they may propose a change in the model, or a change of scale or abstraction (in order to avoid repeatedly analyzing equivalent phenomena, or to drill down into detail when some prediction cannot be made at the current level of the model).

These additional activities relate to the overall ability of humans to bring their background knowledge to bear on the hazard analysis problem. An engineer faced with a paper model of a system has a great wealth of knowledge about computers, machinery and human behaviour that allows them to "cover over the gaps" in the model itself. In a manual process, the model is explicitly acknowledged as the proxy for a much more complex reality, which is fleshed out as needed by the background knowledge of the engineers. By contrast, in a fully automated system the model must *be* the reality because the system does not have such knowledge.

Given the above descriptions of what activities human engineers perform in hazard analysis, we must now determine whether, and to what extent, these activities can be automated.

## Can These Activities be Automated?

The question of "What can be usefully automated?" is certainly an important one. Problems with automation of operational manual tasks (as distinct from engineering ones) are well described, and quite familiar to safety engineers (ref. 8). Taking the stages in typical hazard analysis processes described in the previous section, we can now make some observations of the potential for automation.

An automated system cannot realistically build or acquire a model (stage A). This task is therefore deferred to human intelligence. It is possible, however, to support the development of suitable simulation models through tools and libraries of components. Engineers already use tools to produce a variety of system models, and potentially some of those tools can be reused. It can be noted, however, that many 'modelling' tools are little more than specialized diagram editors.

Given a suitable simulation model, can an automated system enumerate the possible deviations of that model (stage B)? Potentially, this is possible, provided that the system is built so as to be able to (i) identify points in the model where deviations can be applied (ii) manipulate the model so as to implement those faults. For a task of deriving all the permutations of a model, given a set of rules about where deviations can be applied, a computer can quickly consider a vast number possible permutations, whereas humans would take much longer to enumerate them.

For stage C, the deviations applied in stage B must be applied to the model, and their effects determined. Whether this is possible in any given case depends on the adequacy of the model and the simulation algorithms that can be

applied to it. It can be noted that in manual analysis, engineers typical only consider the effect of a single deviation at a time, and for a given deviation they can only predict the effects a short distance through the system (for example, in a process plant this might mean in devices or pipes adjacent to the point of deviation) or for a short period of time after the deviation occurs. By contrast, although an automated system cannot consider *all* possible combinations of deviation, or develop their effects for the lifetime of the system, it can project many combinations of deviations over an extended period of simulated time.

In stage D, changes to the design must be proposed and evaluated. It is not inconceivable to create an automated system that would attempt to do this, based on a library of historical design changes. Indeed, the HAZOPExpert system (ref. 6) has some ability to do this. In practice, however, once an engineer's attention has been drawn to a problem they are generally well-placed to propose solutions. The primary concern addressed in this paper is that it is difficult to identify all the hazards in the first place.

A number of additional activities were noted above that didn't fit into single process stages. All of these are far outside the capabilities of realistic automated systems. There is, therefore, a clear need for human involvement here. As we noted above, these actions occur as needed throughout the process. It follows that not only must it be possible for engineers to take such actions, but that they must have visibility of the process sufficient to see *when* they should take such actions. An analogous issue in general automation is discussed by Norman in reference 9.

It was noted that, in a manual analysis, humans may at any time propose changes to the model being used, either to correct an explicit error or to make a general change, for example to the level of abstraction. As well as being beyond the abilities of any implementable computer system, the increased "weight" of executable and analyzable models means that such a change to these models requires far more work than in the equivalent manual process. Indeed, many transforms of this nature in a manual process are performed solely in the minds of the engineers involved. This may be less of a problem in domains where levels of abstraction versus detail are well established across multiple uses of models.

## What Do We Need from Such an Approach?

It was established in the previous section that some aspects of the hazard analysis process can usefully be automated. The question then arises: "What, then, are the requirements on a method and tool that implement such an automated process?" In this paper, we will consider this in terms of general-purpose hazard analysis, then specifically in the context of systems of systems.

**Requirement 1. The approach must allow for errors and incompleteness in the model**

Given the practicalities of modelling, we can be confident that engineers will quickly find problems with any model, and will need to correct those problems or work around them. In effect, they will want to instruct the system "that's clearly not realistic; don't tell me about it again". We therefore need to allow for this, and not put explicit obstacles in its way.

In reference 10, Zhao and Venkatasubramanian describe the extension of their PHASuite hazard analysis system with a learning engine for this purpose. They note that *"Without a proper mechanism to incorporate this kind of knowledge based on user feedback, if a similar situation arises later when analyzing another process, PHASuite would generate similar inappropriate results and users would have to make the same changes to the results again."* They go on to describe an extension to PHASuite that uses a case-based learning system to allow the user to give feedback and have the system respond to it.

One corollary of this is that we need to make it relatively easy for engineers to perform an automated analysis, note (after some period of study) that some of the results are unrealistic, make changes to the model to correct this problem, and repeat the analysis. In order for this to be practical, they must be able to detect unrealistic output without performing significant manual work that will need to be duplicated once the changes are made.

This may point towards specifying an end point in the analysis where the automated work is decreed "finished" and the engineers carry on from there "on foot", i.e. drawing together the results by some manual means.

**Requirement 2. The approach must use a risk-based, whole-system model**

In a real system, accidents are rare. Typically, they require seemingly bizarre coincidences of actions and states. An accurate simulation model will manifest accidents equally rarely. Therefore, the system must provide a way to detect repeated near-misses as well as accidents. This can be achieved by recording the exposure to risk either experienced by or caused by each entity in the system.

Furthermore, as discussed by Perrow in reference 11, many accidents in complex systems stem not from single, dramatic failures but from a mixture of the unexpected interactions between normal behaviour and a number of simultaneous minor failures. In order to capture the effects of such interactions, the system must use a model that it is mechanically similar to the real system, rather than, for example, a crude functional abstraction.

**Requirement 3. The approach must derive qualitative rules describing cause-effect relationships**

As noted earlier, the goal of hazard analysis is to gain understanding of cause-effect relationships such that the identified hazards can be prevented or mitigated. Merely observing that the system can produce some accident is not sufficient, and nor is deriving a purely statistical relationship between some external factor and some risk metric, unless the internal-to-the-system mechanism by which such an accident can occur is eventually derived.

An effective automated approach, therefore, must provide some form of output that leads analysts *at least* to the precise combination of external circumstances that will reliably lead to an accident. Such a rule could be of the form "If it is raining heavily, and the cover for tank 4 is incorrectly seated, an explosion will occur in reactor vessel B". Engineers can then study the system model with the knowledge that there is some mechanism by which this relationship holds; in other words, it focuses their attention where it is most needed.

Ideally, the system should capture the internal process of states and events by which the system responds to the external factor so as to cause the hazard.

**Requirement 4. The approach must handle dynamic behaviour in dynamic structures**

All systems involve some form of dynamic behaviour, in that the different parts of the system have state that varies independently over time. Much traditional analysis work, however, has been concerned with systems such as process plants which have static structures; in such systems, the relationships between the parts are fixed.

Systems of systems, however, may consist of a set of entities which vary over time, and those entities may interact in an ad-hoc fashion driven as much by geographical and task-allocation happenstance as by any a priori structure. Hence, the set of interacting parts, and the relationships between those parts, varies over time. Any hazard analysis approach that is to be effective when applied to systems of systems must be able to find hazards emerging from interactions within such dynamic structures.

## What Has Been Achieved So Far?

There are many automated safety analysis techniques described in the literature. In this section, we will evaluate the relevance and effectiveness of such work in terms of the requirements identified in the previous section.

There are several existing approaches based on Monte Carlo simulation, whereby stochastic input variables (representing external conditions) are used to derive a statistical assessment of the probability of various accidents and hence of the level of safety achieved by the system. An example of this is the work of Blom et al in airspace safety (ref. 12). This information is useful, but, as we noted under requirement 3, hazard analysis requires causal rules to be derived.

Additionally, the usefulness of such purely statistical information depends heavily not just on the fidelity of the simulation but on the ability of the engineer to support that claim of fidelity. Hence, in hazard analysis, engineers

need something that they can use as a starting point for investigation; they need results that they can *use*, not results that they have to *rely* on. Dewar et al, in reference 13, describes this as "weak prediction".

There is a large body of work in the field of multi-agent simulation. An example of this applied to safety analysis is presented by Johnson in reference 14. In reference 15, Ferber gives the following explanation of the multi-agent simulation concept: "*Multi-agent simulation is based on the idea that it is possible to represent in computerised form the behaviour of entities which are active in the world, and that it is possible to represent a phenomenon as the fruit of the interactions of an assembly of agents with their own operational autonomy.*"

That is, such simulations are composed of a set of distinct agents who have their own rules of autonomous behaviour (their own program) and some simulated environment in which they interact. For example, in the work by Johnson the agents represent staff, patients and visitors in a hospital, and they act within a map of that hospital. In the scenario he studies (evacuation of the building because of a fire) the agents have behaviours that lead them to autonomously move towards the nearest exit, and they may interact in that, for example, agents representing staff may help to move wheelchair-bound or bedridden patients.

Multi-agent simulations can represent systems that exhibit complex, dynamic relationships between their components, and as such they are heavily used for modelling and simulation of military systems. An example of this is given by Ilachinski in reference 16. They can therefore meet requirement 4. As noted by Dewar et al in reference 13, such models can be built at varying levels of fidelity and completeness, thereby supporting requirement 1. Requirement 2 can be met by a multi-agent simulation, although the completeness of the model implemented depends on the time and effort expended on it.

In typical applications of multi-agent simulation, however, including the examples cited above, requirement 3 presents a problem. In addition to supporting Monte Carlo use in order to derive statistical information, many multi-agent models provide visualization of ongoing simulation runs, allowing users to "watch what happens". This is certainly valuable for comprehension of system behaviour. In a realistic hazard analysis situation, however, engineers will have the system perform many thousands of runs, and it will not be practical to watch them all in order to extract hazards. It is possible to focus on, for example, just those runs that involve particularly serious accidents, but this provides no guarantee that the user will uncover all the distinct cause-effect chains by which accidents may occur. (There may be two "equally bad" results that are reached by completely different causal pathways).

The area of automated hazard analysis that is perhaps best represented in the literature is that related to automated HAZOP. Venkatasubramanian et al present a summary of this work in reference 6. For example, QHI (described by Catino and Ungar in reference 17) combines a plant-specific pipe and instrumentation diagram of a chemical plant with a generic library of faults, chemical processes and hazards. Permutations of the possible faults are combined with the plant model, producing modified versions of that model, and the behaviour of these variants is then simulated. The system looks for plant behaviour that is consistent with the generic set of hazard types, and reports such behaviour to the user along with the faults that caused it.

Approaches such as QHI are thus consistent with requirements 1 through 3. However, the existing automated HAZOP work is successful in part because it exploits the highly static structure of chemical process plants. As noted above, in requirement 4, this assumption does not hold for systems of systems. The work on automated HAZOP is therefore not *directly* applicable to systems of systems analysis.

## Our Approach

Given the requirements identified above, and building on the prior work discussed in the previous section, the authors have developed a partially-automated approach to systems of systems hazard analysis. This section describes the approach, and explains how it meets the requirements. The description is illustrated with examples drawn from a military system of systems case study.

The case study uses a simulation model of a military unit engaged in anti-guerrilla operations. The unit contains Unmanned Air Vehicles (UAVs), artillery pieces and helicopter-borne infantry. A single vignette has been

implemented for this model, in which the agents in the system must detect and neutralize a number of static enemy positions. Some initial results from this study were presented by the authors in reference 18.

In our approach, an engineer builds a multi-agent model of a system, and places it in a synthetic environment. Each agent is given the ability to perform a mixture of scripted and reactive behaviour. They have the ability to perceive their environment (both through organic sensors and inorganic shared situational awareness) and to act on it. In our example, there is an agent for each UAV, artillery piece and helicopter. The environment consists of a map of a mountainous desert region, and an agent for each enemy unit.

As noted above, in the automated HAZOP system QHI, which is described by Catino and Ungar in reference 17, a library of faults is provided that describes preconditions for each fault to be possible and the effect of the fault occurring. For example, the failure "valve fails shut" might have the precondition "there is a valve in a pipe between two devices" and the effect "flow rate through that pipe is set to zero". Catino and Ungar note that such a model replaces the use of guide words in manual HAZOP; the computer's ability to try a large base of fault preconditions against every possible fault location in the model partly compensates for the loss of human creativity in proposing faults to consider.

Based on this, our approach allows all sensing, communication and action channels of each agent to be open to a variety of potential faults (although the in current implementation these possible faults must be manually implemented for each entity type). For example, a given sensor can potentially be disabled, have its range reduced, or be made to perceive multiple versions of everything it detects. Given a priori probabilities of each fault occurring, a simulation engine can be built that automatically runs simulations of the system in all fault permutations that have a probability of occurrence below some user-defined impossibility threshold.

In our example, the UAV agents have two sensors, one for detecting enemies on the ground below it and one for detecting other aircraft nearby. Both of those may fail so as to provide no information. Similarly, the artillery pieces have actuators that allow them to perform indirect fire at a given coordinate; these can fail such that their fire is distributed in a wide area around their intended target. In our current work, we have assigned an arbitrary probability of $10^{-3}$ of each fault being present in given simulation run, and an impossibility threshold of $10^{-11}$. The system therefore considers combinations of up to three simultaneous faults.

As noted above, multi-agent simulations can produce vast amounts of data that is difficult to usefully interpret, and the system we describe is no exception. In reference 19, Platts et al describe a system which aims to improve the mission performance of Unmanned Air Vehicles (UAVs) by learning rules from a simulation model. In their work, a simulated UAV attacks a target using tactics determined by several input parameters (such as height, angle and velocity of approach). A large number of runs are performed, with randomly generated parameter values, and learning techniques are applied to derive rules which relate the values of those parameters to mission success or failure.

A similar approach can be applied here; if we replace the "mission success" criteria with a description of some accident, then we have rules that may lead engineers to discover an important hazard in the system. For example, in our case study, for the accident *"landed helicopter is hit by artillery fire"* the system learned the rule *"IF UAV #2 has not lost communications AND UAV #4 has lost communications AND artillery #3 is firing inaccurately THEN the accident will occur."*

Clearly, it is necessary to perform this learning attempt once for each possible accident. This is reasonable, as the set of all possible distinct accidents can be derived from the set of agents in the system and the possible direct interactions between them or between them and their environment (for example, attempted fratricide by each armed entity, collision between any two mobile entities, collision between any mobile entity and the terrain). An extension to this is that risk metrics could be collected for each accident (for example, for midair collision accident the metric could be based on incidents of unacceptable proximity) and rules learned for those cases where the metric exceeds some predefined threshold.

The resulting rules are then studied by engineers in order to determine which of them are realistic (as opposed to being simulation artefacts) and how the mechanisms of the system gave rise to the accident. This is a two stage process:

In the first stage, the analysts must try to understand the rules, and how they relate to the modelled system (rather than the model itself). Important questions at this stage include “Why did the algorithm learn this rule?”, “Is this realistic, or is it merely a simulation artefact?”, “Why (in terms of the mechanisms of the model) are these particular features/causes so important?”

The output of this first stage includes a revised set of rules, which have been manually filtered (to remove rules that have no apparent correspondence to the real system) and augmented with explanations and references to particular simulation runs which express the behaviour well. In the second stage, analysts must consider the implications of the identified hazards for the real system. Key questions are “Is this hazard serious and plausible enough to warrant our attention?” and “What are we going to do about it?”

Having described our approach to automate analysis, the question remains as to whether it meets the identified requirements. As noted in the discussion of previous work, multi-agent simulation allows systems with dynamic structure to be modelled, and the results of their structural changes observed. This satisfies requirement 4. The use of machine learning to derive qualitative cause-effect rules covers requirement 3. With regard to requirement 1, it has been noted that multi-agent simulations can be built at a variety of levels of detail, and that detail can be added over time as more complex entity models become available or are needed. We do not yet, however, have explicit support for user feedback, for example in the manner of the PHASuite extension described earlier. This will be addressed in future work.

Through support for tracking of incidents as well as accidents, and for collecting risk metrics related to entity behaviour, part of requirement 2 can be met. As for the other part (simulating whole-system behaviour), this is in the hands of the engineer implementing the model. The need for an executable model, however, in which agents achieve the overall system’s objectives through interaction with other agents and their environment, pushes the developer towards a level of completeness that is not necessarily required by other approaches.

## Conclusions

It is clear that we need an automated hazard analysis approach using multi-agent simulation in order to perform analysis of systems that exhibit dynamic interactions over dynamic structures. These systems fall well outside the scope of current manual methods. But there is a problem with removing the human from the process. If we ask “What if?” and only let engineers ‘see’ the end results arising from the simulation models, it is likely that they will struggle to understand the cause-effect relationships sufficiently to propose adequate mitigation and prevention actions. In other words, we don’t want to “cut them out of the understanding loop”. Hence, we need to combine the simulation with machine learning techniques in order to extract the causal rules we need from the mass of data generated by the simulation.

This paper has presented the requirements for such a hazard analysis approach, introduced a candidate method, and evaluated the method against the requirements. Future work will include applying the approach to a range of systems and scenarios, and experimenting with multiple machine learning algorithms and ways to introduce deviations. In order to evaluate the real-world value of the approach it will be applied in realistic industrial case studies.

## Acknowledgements

The authors would like to thank Dimitar Kazakov for his assistance in the work described in this paper. The work was funded under the Defence and Aerospace Partnership in High Integrity Real Time Systems (Strand 2).

## References


1. MoD. Defence standard 00-56: Safety management requirements for defence systems. Technical report, Ministry of Defence, 1996.

2. SAE. Aerospace recommended practice 4761 - guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment. Technical report, Society of Automotive Engineers, 1996.

3. T. Kletz. HAZOP and HAZAN: Identifying and Assessing Process Industry Hazards. Institution of Chemical Engineers, 3rd edition, 1992.

4. R. Alexander, M. Hall-May, and T. Kelly. Characterisation of systems of systems failures. In Proceedings of the 22nd International Systems Safety Conference (ISSC), pages 499-508. System Safety Society, 2004.

5. R. Alexander, M. Hall-May, G. Despotou, and T. Kelly. Using simulation to evaluate safety policy for systems of systems. In M. Barley, F. Massacci, H. Mouratidis, and A. Unruh, editors, Proceedings of the Second International Workshop on Safety and Security in Multiagent Systems, pages 5-21, 2005.

6. V. Venkatasubramanian, J. Zhao, and S. Viswanathan. Intelligent systems for HAZOP analysis of complex process plants. Computers and Chemical Engineering, 24:2291-2302, 2000.

7. D. Pumfrey. The Principled Design of Computer System Safety Analysis. DPhil Thesis, University of York, 2000.

8. N. B. Sarter, D. D. Woods, and C. E. Billings. Handbook of Human Factors & Ergonomics, chapter Automation Surprises. Wiley, second edition, 1997.

9. D. A. Norman. The 'problem' with automation: Inappropriate feedback and interaction, not 'over-automation'. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences, 327(1241):584-593, April 1990.

10. C. Zhao and V. Venkatasubramanian. Learning in intelligent systems for process safety analysis. In Proceedings of ESCAPE-15, 2005.

11. C. Perrow. Normal Accidents: Living with High-Risk Technologies. Basic Books, New York, 1984.

12. H. A. P. Blom, S. H. Stroeve, and H. H. de Jong. Safety risk assessment by monte carlo simulation of complex safety critical operations. In F. Redmill and T. Anderson, editors, Proceedings of the Fourteenth Safety-critical Systems Symposium, pages 47-67, Bristol, UK, 2006. Safety-Critical Systems Club, Springer.

13. J. A. Dewar, S. C. Bankes, J. S. Hodges, T. Lucas, D. K. Saunders-Newton, and P. Vye. Credible uses of the distributed interactive simulation (DIS) system. Technical Report MR-607-A, RAND, 1996.

14. C. Johnson. The glasgow-hospital evacuation simulator: Using computer simulations to support a risk-based approach to hospital evacuation. Technical report, University of Glasgow, 2005. Submitted to the Journal of Risk and Reliability.

15. J. Ferber. Multi-Agent Systems: an Introduction to Distributed Artificial Intelligence. Addison-Wesley, 1999.

16. A. Ilachinski. Exploring self-organized emergence in an agent-based synthetic warfare lab. Kybernetes: The International Journal of Systems & Cybernetics, 32(1):38-76, February 2003.

17. C. A. Catino and L. H. Ungar. Model-based approach to automated hazard identification of chemical plants. AIChE Journal, 41(1):97-109, 1995.

18. R. Alexander and T. Kelly. System of Systems Hazard Analysis using Simulation and Machine Learning. In J. Gorski, editor, Proceedings of the 25th International Conference on Computer Safety, Reliability and Security (SAFECOMP '06), Gdansk, Poland, September 2006. Springer-Verlag LNCS.

19. J. T. Platts, E. Peeling, C. Thie, Z. Lock, P. R. Smith, and S. E. Howell. Increasing UAV intelligence through learning. In AIAA Unmanned Unlimited, Chicago IL, 2004.

## Biography

Robert Alexander, Department of Computer Science, University of York, Heslington, York, YO10 5DD, UK, telephone – +44 1904 432792, facsimile – +44 1904 432767, e-mail – rda@cs.york.ac.uk

Robert Alexander is a Research Associate in the High Integrity Systems Engineering (HISE) group in the University of York's Computer Science Department. Since October 2002 he has been working on methods of safety analysis for Systems of Systems. Robert graduated from Keele University in 2001 with a BSc (Hons) in Computer Science. Prior to joining the research group, he worked for Sinara Consultants Ltd in London, developing financial information systems.

Dr Tim Kelly, Ph.D., Department of Computer Science, University of York, Heslington, York, YO10 5DD, UK, telephone – +44 1904 432764, facsimile – +44 1904 432708, e-mail – tpk@cs.york.ac.uk

Dr Tim Kelly is a Lecturer in software and safety engineering within the Department of Computer Science at the University of York. He is also Deputy Director of the Rolls-Royce Systems and Software Engineering University Technology Centre (UTC) at York. His expertise lies predominantly in the areas of safety case development and management. His doctoral research focused upon safety argument presentation, maintenance, and reuse using the Goal Structuring Notation (GSN). Tim has provided extensive consultative and facilitative support in the production of acceptable safety cases for companies from the medical, aerospace, railways and power generation sectors. Before commencing his work in the field of safety engineering, Tim graduated with first class honours in Computer Science from the University of Cambridge. He has published a number of papers on safety case development in international journals and conferences and has been an invited panel speaker on software safety issues.