Timed Knowledge-based Modelling and Analysis: On the Dependability of Socio-technical Systems

Juliana Küster Filipe, Massimo Felici and Stuart Anderson

School of Informatics
University of Edinburgh
King's Buildings, Mayfield Road

Edinburgh EH9 3JZ, UK

+44-131-6505962
jkfilipe@inf.ed.ac.uk, massimo.felici@ed.ac.uk, soa@inf.ed.ac.uk

ABSTRACT

We are concerned with the analysis of socio-technical systems, in particular, safety-critical systems involving considerable human intervention. Experience shows that evolving knowledge distribution in socio-technical systems may trigger catastrophic events affecting system dependability. This paper presents a knowledge-based approach to model and analyse evolving scenarios in socio-technical systems. The timed knowledge-based approach captures the nature of socio-technical systems, which consist of hybrid resources continuously interacting each other. The analysis and modelling of a case study drawn from the Air Traffic Control domain shows the applicability of the proposed approach.

Keywords

Knowledge-based reasoning, timeliness, socio-technical systems, knowledge distribution, dependability

INTRODUCTION

We are concerned with the analysis of socio-technical systems that involve considerable human intervention. Distributed hybrid resources interacting each other infer emergent behaviors to socio-technical systems. Resources forming socio-technical systems represent both sources and sinks of knowledge. Interactions in socio-technical systems redistribute knowledge among resources. If the activities (e.g., Air Traffic Control) supported by social-technical systems are safety-critical, the distribution of knowledge among system resources involves high risk. Knowledge may trigger catastrophic events that affect system dependability (Leveson, 1995; Perrow, 1999).

The design of operating procedures for socio-technical systems is a key phase of the overall system development. Procedure design requires information that can be revealed only by taking a dynamic account of human activities and decision-making processes driving interactions in socio-technical systems. Gathering this information takes time and requires the involvement of domain experts (e.g., in the context of Air Traffic Control it may involve pilots, air traffic controllers, regulators, etc.). We take into account a knowledge viewpoint in order to support the modelling, the design and the analysis of operating procedures for socio-technical systems. We concentrate on questions like: What knowledge do human beings require in order to act safely? How can human beings know the effect of their actions on the state of the system? How has human knowledge evolved throughout time? Do human beings know how much time they have to act in a particular situation?

This paper shows how a logical approach can support activity analysis of socio-technical systems. A timed logic provides a framework for modelling knowledge and reasoning on related system criticalities. The logic can thus be used as an analysis tool to understand critical issues related to evolving knowledge. The analysis of a case study drawn from the Air Traffic Control domain shows how our approach can offer valuable and novel insights by supporting the identification of critical issues related to evolving knowledge.

MULTIDISCIPLINARY RATIONALE

This section identifies a rationale for the timed knowledge-based framework proposed in this paper. The rationale finds its origins in different disciplines that are relevant to the understanding of socio-technical systems.

Activity Theory (Vygotsky, 1978) helps to understand from a social viewpoint how social interactions influence human cognition. Activity Theory explains how human beings regulate their behaviors by means of inclusion of auxiliary stimuli into their activities. The stimuli take origin in external artefacts or in social interactions. In other words, Activity Theory emphasizes that human behaviors should be understood in the context of social interactions and external activities. Any subject (or individual) accomplishes specific activities (or objectives) through negotiations within a social community by the process of *Internalization* and *Externalization*. Internalization explains how individuals construct internal models about the activities to be performed. Externalization explains how individuals design and implement new activities. Complementary, *Social Learning* explains how human beings perceive machines in order to acquire computational artefacts and to accomplish specific activities (or tasks). Social Learning consists of

two main processes, namely, *Innofusion* (i.e., learning by trying) and *Domestication* (i.e., learning by interacting). Recent research further analyses Social Learning in multi-media systems. (Williams et al, 2000).

Distributed Cognition (Norman, 1999) focuses just on the interaction between representational resources, which can be located within human mind as well as external artefacts. It helps to understand how information mediated activities are carried out by distributed resources. Distributed Cognition stresses that human cognition is not isolated within human minds, but it emergently extends to external distributed artefacts. Internal and external artefacts do not exist in isolation. They form socio-technical systems. A holistic view of socio-technical systems may explain the nature of socio-technical systems. For instance, the SHEL model (Edwards, 1972) defines any (electronic mediated) productive process as performed by a combination of Hardware (e.g., any material tool used in the process execution), Software (e.g., procedures, rules, practices, etc.) and Liveware (e.g., end-users, managers, etc.) resources embedded in a given Environment (e.g., socio-cultural, political, etc.). Hence any productive process may be regarded as an instantiation of the SHEL model for a specific process execution. As the SHEL model emphasizes that any productive process relies on the different resources, Distributed Cognition and resources-based modeling (e.g., like the SHEL model) show different aspects of socio-technical systems that may be linked together in a sound way (Write et al, 2000). Simple models representing human understandings with respect to socio-technical systems have stimulated the use of mechanized verification methods in order to identify automation surprises (Rushby, 2002). The results encourage further investigations on the use of mechanized tools in order to identify inconsistencies in the design of socio-technical systems.

Starting from Activity Theory, the exemplification provided by Distributed Cognition helps to acquire a simple systemic model (e.g., the SHEL model) of socio-technical systems. *Situation Awareness* provides a knowledge-oriented interpretation of socio-technical systems (Endsley, 1995; Endsley, 2000). Situation Awareness stresses the knowledge required in order to perform a specific activity. Moreover, Situation Awareness points out how knowledge constantly evolves. Timeliness is therefore a critical aspect of Situation Awareness. For instance, from a human viewpoint, a pilot needs specific knowledge in order to safely fly an aircraft. The pilot furthermore adapts various strategies for maintaining updated Situation Awareness. Complementary, *Workflow* stresses how human beings negotiate information in order to perform specific tasks. Both Situation Awareness and Workflow point out that understanding the knowledge required in order to carry out specific tasks is important for designing socio-technical systems.

This paper presents an approach for modelling and analysing the timeliness aspects of the knowledge required in order to perform specific tasks. This provides a way of representing and assessing evolving scenarios from a knowledge viewpoint. Thus our timed knowledge-based approach allows us to model and analyse those situations in which knowledge is critical for the dependability of socio-technical systems. The remains of this paper will present the proposed approach and will show how the approach can be applied in socio-technical systems.

CASE STUDY

This section takes into account a case study drawn from the Air Traffic Control (ATC) domain. On the 1st July 2002 two airplanes, a Tupolev TU 154 M on its flight from Moscow (Russia) to Barcelona (Spain), and a Boeing B757 on its flight from Bergamo (Italy) to Brussels (Belgium), collided while on flight. The collision caused the catastrophic destruction of both airplanes and the loss of all passengers (Aviation Safety Network, 2002). Hereby, we do not want to investigate the causes of the accident. Further information about the referred accident may be found in the Aviation Safety Network web site or in other related repositories. We intend to analyse how interactions in socio-technical systems may redistribute critical knowledge to carrying out technology-mediated activities. ATC case studies provide safety-critical activities relying on a mixture of socio-technical interactions where knowledge is distributed among the involved actors. The main features of the Tupolev-Boeing (hereafter TB) accident are:

- *Timeliness*. ATC scenarios consist of timed sequences of events involving hybrid interactions (e.g., human-machine interaction; human-human interaction).
- *Knowledge*. Knowledge continuously evolves as a consequence of the occurrences of human actions and interactions.
- *Technology mediated*. Human activities are supported by available technologies.
- *Human factors*. Human beings are an integral part of (socio-technical) systems.

Notice that the above features may be identified in other contexts as well (e.g., health systems). Nowadays technology is pervasive. Human activities depend on technical systems. The basics on which socio-technical systems rely on can thus be generalised.

Table 1 shows a SHEL description of the case study. The TB accident involved the following main resources, i.e., Liveware (L), Hardware (H), and Software (S) embedded in an Environment (E): Air Traffic Controller (ATCer), Crew of the Boeing B757 (CB), Crew of the Tupolev TU 154 (CT) and the two TCAS systems (respectively, TCAS_B and TCAS_T). The related documents and the ATC transcript of the original tape recording provide further information about the accident

(ARINC, 2002; Aviation Safety Network, 2002; BFU, 2002).

Table 1. A SHEL description of the case study.

Resource	Type	Description
ATCer	L	At the moment of the collision one controller was on the sector controller workplace. He had to monitor two workplaces with radar screens. The operator had provided training programs for TCAS. There were other operators, but they are not involved in the considered scenario.
CT	L	Both crews had completed the corresponding training for TCAS.
CB	L	
TCAS_T	H S	Both aircrafts were equipped with identical Collision Avoidance Systems, TCAS, (Honeywell, 2000). Neither the history of the flight nor the evaluation of the flight data recorders indicated any technical defects on the aircrafts.
TCAS_B	H S	
Operating Procedures	S	The procedures that apply in the specific case, e.g. procedures in case of TCAS resolution or other relevant procedures.
Environment	E	In the Swiss Air Traffic Control maintenance work of the system was performed in the late evening of the accident day. The horizontal separation minima had been increased from 5 to 7 nm. During this period of time the ground based collision warning system STCA (Short Term Conflict Alert) was not working. The direct phone connections to the adjacent air traffic control services were not available either. Other technology was available in the environment, but it was not directly involved in the accident.

Table 2 shows the chain of events that caused the accident. The scenario reconstruction does not report all the details, but just the main interactions. The accident scenario points out the main critical factors:

- *Knowledge.* The pilots and the controller had partial knowledge about the real situation. Each one acquired different knowledge and understanding about the situation. Hence each one differently perceived the criticality of the situation.
- *Timeliness.* The sequence of events consists of interactions that continuously modify the distribution of knowledge. The sequence of events seems to be different from many other sequences just in the final collision. The sequence of events lacks of early risk perception. The analysis with respect to the TCAS system (ARINC, 2002) points out how the prompt pilot's response may be necessary in order to maintain safety: *"The safety benefits provided by TCAS are directly dependent on a pilot's response to an RA. The pilots' instinctive reaction to an RA should always be to respond to the RA in the direction and the degree displayed."*

Table 2. Accident Scenario.

Time	Actors	Event(s)
T1	TCAS_B, CB	The TCAS on both aircrafts give a Traffic Advisory.
	TCAS_T, CT	
T2	ATCer, CT	ATCer tells CT: *"descend flight level 350, expedite, I have crossing traffic"*
T3	TCAS_B, CB	Both aircrafts get a TCAS Resolution Advisory (RA); CB complies; CT remains at FL360
	TCAS_T, CT	
T4	ATCer, CT	ATCer repeats the instruction to CT to descend; CT complies
T5	TCAS_B, CB	*"Increase descent"*
T6	CB, ATCer	CB report to ATCer that they are doing a TCAS descend
T7	TCAS_T, CT	*"Increase climb"*
T8		Collision

The following section introduces the timed-knowledge based approach and applies it to the case study.

KNOWLEDGE-BASED FRAMEWORK

The framework we consider has been largely influenced by work done in Artificial Intelligence (AI) on theoretical aspects of reasoning about knowledge using epistemic logics of knowledge and belief but is extended for our purposes in different ways that we shall describe bellow.

Epistemic logics of knowledge and belief were introduced by the philosopher Jaako Hintikka in the 1960s to capture some intuitions about the nature of knowledge (Hintikka, 1962). It was only much later that these logics received considerable attention within the AI community and found numerous applications, e.g. (Fagin et al., 1995). One such application is the modelling and analysis of knowledge requirements of robotic tasks (Brafman et al., 1994). In this case, by using these logics we can reason about the knowledge a

robot requires in order to perform a particular task. Consider the following example.

A robot is standing in front of a conveyor belt. Items are placed on the belt at the starting end. The task of the robot is to place passing items into a bag. To perform this task, the robot requires sensor information on when an item is approaching and at the right distance so that the robot can extend its arm and grab it.

From this example we can already imagine that we could similarly reason about the knowledge humans require in order to perform their tasks (in a safe manner, if we are concerned with critical systems).

Another interesting angle on logics of knowledge is that they are useful for reasoning about distributed systems where each component (it can correspond to a subsystem, an agent, an object or a human) only has a partial view or understanding (knowledge) of what is happening in the system as a whole (Fischer et al., 1986; Halpern et al., 1989).

The robot may not know when it misses an item passing in front of it on the belt. For instance, if the sensor is faulty and not sending signals to the robot on time. In this case the system is not delivering an expected quality of service but from the point of view of the robot the system is actually performing fine.

This is also what we need for socio-technical systems. The humans in the system are not omniscient, i.e., they do not have an overall view of the state of the system. Additionally, humans have limited reasoning capabilities that may vary in different situations or for different humans. An approach like described in (Fagin et al., 1988) is useful to explore this difference.

Moreover, if we have several humans in the system it may be that there is a knowledge gap between them. In many cases this gap can be highly critical. For instance, in ATC

- Air traffic controllers may often have poor information on aircraft capabilities, weather and winds that are needed for making their own assessment of whether an aircraft will be able to conform to a clearance as expected.
- Air traffic controllers do not know what the TCAS system on the airplanes is advising the pilots to do. According to table 2, ATCer only knows the TCAS_B traffic advisory at T6 when CB reports it to ATCer.

Existing approaches using logics of knowledge can express knowledge and common knowledge of different entities in the system, e.g. (Dwork et al., 1990), and consequently give us a way to explore possible knowledge gaps.

Logics of knowledge have also been enriched with time (temporal logics of knowledge) in order to be able to capture how knowledge changes throughout time. Knowledge can change due to new observations or information gained through interaction (between agents, between system components, between humans, human-machine). In either case new information is gained and current knowledge has to be revised. Notice that we are herein not interested in the way knowledge is revised. In some situations it may be reasonable to always add the newly gained information into the "set of knowledge" and revise older remaining knowledge to avoid inconsistencies - this is in fact what is done in the classic approach (Alchourrón et al., 1985). But in situations where observations can be unreliable other knowledge revision approaches have to be adopted instead, as is done for instance in (Boutilier et al., 1998).

With the added temporal aspect, temporal logics of knowledge allow us to reason about how the knowledge a human has evolves over time. What none of the existing temporal logics of knowledge has is, however, a way to express real-time requirements and properties. In some of the critical systems we want to analyse this may be of importance. For instance,

- A pilot may want to know exactly how much time is available to meet the clearance.
- In a neonatal intensive care unit, babies who suffer from a respiratory distress syndrome are connected to a ventilator. Nurses and other members of staff know that after changing the settings on the ventilator, a baby will need between 20-30 minutes to stabilise. Consequently, they know that another intervention on the baby within this period must be avoided.

Our framework is based on a real-time temporal logic of knowledge, and can be used as a reasoning tool to analyse the knowledge requirements a human has in order to be aware of possible hazards in the system and still have enough time to handle and avoid them. Also, we want to be able to reason about potential conflicting sources of knowledge.

Going into Detail

We now give a few more details on how our approach works. Even though it is based on several existing work described above, many (technical) details are new including: the adopted model, the real-time characteristics, the syntax of the logic, and expressive power.

Traditionally knowledge-based approaches have a possible-world semantics. We need a more powerful model with an explicit notion of time. A widely used model for real-time are timed automata (Alur, 1999; Alur et al, 1994). We omit all technical details on these models as these are irrelevant for this paper, and instead refer the interested reader to the standard references mentioned above. We also do not introduce our logic to keep the description of the approach more understandable. More details can be found for instance in (Anderson et al., 2003).

Here, it suffices to understand a timed automaton as a graph (a collection of nodes -called states- and edges -called transitions- connecting the nodes) with time

constraints attached to the states or transitions. Intuitively, a time constraint associated to a state restricts the time allowed in that state (it corresponds to a state invariant); whereas a time constraint associated to a transition tells when the transition can occur. To express time constraints we use clock variables. Examples of possible constraints are *x£10*, *x>5*, and so on. Clock assignments can also be given with a transition, and can be used for example to reset a clock (indicated by *x:=0*). Consider the next figure.

Figure 1 shows three states (S1, S2, S3) and three transitions (t1, t2, t3) between some of the states. The variable x denotes a clock. The state invariant at S1 *x£10* restricts the time allowed in that state to less than 10 time units. In order to guarantee that invariant, the state has to be left before *x=11*. To leave the state S1 with transition t2 the additional constraint *x>5* has to be true. If the state S1 is left with transition t1, the state S1 is re-entered with the clock variable x reset. In this example, possible system runs (sequences of state changes) are

(s1,5),(s1,6),(s2,-)

(s1,5),(s1,6),(s3,-)

(s1,5),(s1,6),(s1,4),(s3,-)

(s1,7),(s1,4),(s1,8),(s2,-)

where the first argument in a pair indicates the current state, and the second argument denotes the time units stayed in that state (which naturally varies from run to run). Transitions are left implicit from the runs just now, but should be clear. Because in this example, there are no transitions leaving states S2 and S3, as soon as the system enters them it remains in them forever.

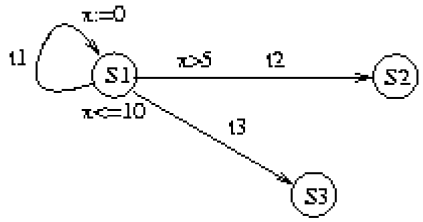

Figure 1. A simple example of a timed automaton.

Every component in a system has a model associated with it. Combining models (through parallel composition) of all the components in the system would give us a model of the overall system. But remember that this is rarely possible to achieve, as not all components in the system may either be known or have a model describing its behaviour.

The models we have for describing the behaviour of the humans in the system correspond to or are derived from activity theory models, for instance. The idea is then that if we have a model (timed automaton) representing the human behaviour, then we can explore and reason about the knowledge the human has or needs in each state, how this knowledge can evolve in a possible sequence of transitions (system run), how it can be in conflict with the knowledge of other components (human or not) in the system, and so on. We show how this applies to our case study in the next section.

APPLYING THE APPROACH TO THE SCENARIO

We consider a timed automaton indicating a partial behaviour of the CT crew that contains at least what is known to have happened between T1 and the time of collision T8. We do, nonetheless, indicate other possible states and paths just to give an idea of what could have happened instead (allowing for a simplified view of alternatives). The reason we add more paths to our model is to illustrate how our framework can be used in reasoning. In other words, a scenario just indicates a sequence of event occurrences (in this case what actually happened) but to be able to analyse what could have been improved to prevent the accident a more complete model is helpful. A model that does not consider certain possible outcomes (runs) is not a good representation of what can happen in reality. Consequently a model of the CT crew that would not allow for the sequence between T1 and T8 would have been incorrect.

Formal methods are particularly useful to explore exhaustively what can happen in accordance with a certain model. The model can then be redesigned to take into account what is missing or incorrect. Also, being able to reason about what could happen in a system taking a certain model can reveal possible hazards. Consider the model given in Figure 2.

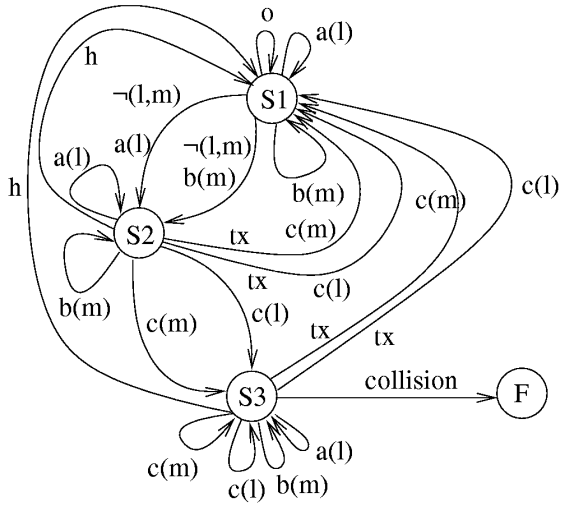

Figure 2. A timed automaton for CT crew.

To increase the readability of the figure, some event abbreviations on the transitions were made:

- Label o corresponds to an interaction with TCAS_T, where the system is indicating traffic advisory.
- Label a(l) corresponds to an interaction with ACTer, where the parameter is the advice received.

- Label b(m) corresponds to an interaction with TCAS_T, where the parameter corresponds to the RA received.
- Label c(m) or c(l) correspond to compliance of CT with respect to advisory m or l respectively.
- Label h corresponds to an external event to CT, in this case an action by other aircrafts, which may solve the conflict.
- ¬(l,m) indicates that there is a conflict between the advices l and m.
- Clock resets (x:=0) are assumed in each transition.
- Label tx corresponds to the time constraint x<t where t is a real-time value (acceptable time to comply to an advice and still be able to know the effect). For instance, if CT is at state S3 and there is not enough time to meet a clearance then even though CT is performing the complying action the transition reenters state S3. Moreover, if the other plane does not in act in an appropriate opposite manner then the collision becomes unavoidable.

The state S1 corresponds to a normal operation state. Transitions leaving from S1 to S2 can only happen if there was a sequence of advices by ATCer and TCAS_T with conflicting information (in our abbreviated notation given by ¬(l,m)). Notice that this model is still incomplete: we do not indicate compliances to advices in the normal situation (where there are no conflicts, etc). If CT reaches state S2 then several things are possible: CT can remain in state S2 (does not do anything); complies to the TCAS_T RA (as soon as possible being captured by the time constraint tx and provided that this advice was correct reaches state S1); complies to ATCer (as soon as possible and provided that this advice was correct reaches state S1); receives another recommendation from either TCAS_T or ACTer; complies to a "wrong" recommendation by ATCer reaching S3, and so on. Reaching state S3 again several options are possible. In particular the case that there is not enough time to meet a clearance, and in which case performing the complying action does not make CT leave state S3. Rather, it increases the likelihood that CT reaches state F, which is only possible from S3 (again in our simplified behavioural model) corresponding to the occurrence of a collision.

The scenario can be reconstructed from this model and is given by the sequence

At state S1, occurrence of o followed by a(l) followed by b(m) with ¬(l,m). State S2 is entered, occurrence of a(l) followed by c(l) with x>t. State S3 is entered, occurrence of b(m) followed by collision. State F is entered (failure).

We emphasize two situations arising in the scenario, conflicting information gained from different interactions, and timeliness.

In the scenario CT reached state S2, because conflicting information was given by the TCAS_T on the one side, and by ACTer on the other. The following statements reflecting knowledge by CT can be made at state S2:

CT does not know whether advice l or m is right.

CT does not know if following advice l will bring the system back to a normal state and would thus correspond to a transition into state S1.

Similarly, we could express that at state S1 (in a situation where two non-conflicting advices were received):

CT knows that complying to advice l or m is safe.

The TCAS system works under the assumption that a pilot complies as soon as possible to its advice. Consequently, we could express that at any state of the model

If CT waits for an arbitrarily long period, CT does no longer know whether it is safe to comply to the recommendation or not.

DISCUSSION AND CONCLUSIONS

We believe that our approach can offer valuable and novel insights by supporting the identification of critical issues related to knowledge and evolution of knowledge throughout time. The modelling and analysis of the case study identify the different ways the timed knowledge-based approach may contribute in the design of socio-technical systems, in particular, safety-critical systems involving considerable human intervention.

- *Human-Machine Coupling*. Research in Human-Machine Interaction (Hollnagel, 1995) points out that the coupling between human and machine cannot be improved just by design solutions (e.g., improved machine interface). The coupling between human and machine can be effectively improved by supporting the understanding process. People who interact with computer-based systems often have difficulties in understanding the real situations by the limited information provided by the systems. From a systemic viewpoint the interactions occurring in the socio-technical systems constantly modify the knowledge distribution. Thus the modelling, design and analysis of procedures (interactions) allow us to identify flaws and lack of timely knowledge. The timed knowledge approach then enhances our ability to model, design and analyse procedures in socio-technical systems, hence improving coupling between human and machine.
- *Multidisciplinary Design.* The proposed timed knowledge-based approach has origins in the basics of socio-technical systems. Differently from many formal approaches that often model unreal situations by restricting socio-technical systems in a combination of resources (or components), the focus of the proposed approach is on interactions in socio-technical systems. The focus on the interactions allows us to take a systemic viewpoint complying with the basic theories explaining the nature of socio-technical systems. This differs from classical system engineering that often takes a machine centric

viewpoint. The timed knowledge-based analysis and modelling of the Air Traffic Control accident show how the approach enhances our ability in understanding timed knowledge and interactions in socio-technical systems. It then represents a step forward towards multidisciplinary design. The approach is a considerable contribution in bridging classical system engineering with theories grounded in social aspects of the electronic mediated society.

- *Knowledge Dependability.* The analysis of the Air Traffic Accident points out two main criticalities, Timeliness and Knowledge, in socio-technical systems. The timed-knowledge based approach enhances our ability to analyse those situations in which untimely knowledge may affect the overall system dependability.
- *Knowledge-based Tools.* The timed knowledge-based approach represent a basis for the devise of tools supporting the modelling, design and analysis of operating procedures for socio-technical systems. Timed knowledge-based tools may support two main phases during the design of socio-technical systems. The former is the modelling and design of operating procedures. The latter is the validation of operating procedures by simulated trials with end-user involvements. This allows us to support by a single approach different phases in the system life cycle.

In conclusion, this paper introduces a timed knowledge-based approach for the modelling, design and analysis of procedures in socio-technical systems. The use of the approach in a case study drawn from the Air Traffic Control domain shows the applicability of the approach. The proposed approach furthermore provides us with new insights in the design of socio-technical systems.

ACKNOWLEDGMENTS

We would like to tank Alberto Pasquini and Simone Pozzi (Deep Blue, http://www.dblue.it/) for their explanations about the Avionics context and case study, Mark-Alexander Sujan (Human Reliability Associates, http://www.humanreliability.com/) and Gordon Baxter (University of York) for their comments on the multidisciplinary aspects of the paper. This work has been supported by the UK EPSRC DIRC project, http://www.dirc.org.uk/, Interdisciplinary Research Collaboration in Dependability of Computer-Based Systems, grant GR/N13999.

REFERENCES

Alchourrón, C.E., Gärdenfors, P. and Makinson, D. (1985). On the logic of theory change: partial meet functions for contraction and revision. *Journal of Symbolic Logic*, 50, 510-530.

Alur, R. (1999). Timed Automata. In NATO-ASI 1998 Summer School on Verification of Digital and Hybrid Systems.

Alur, R. and Dill, D.L. (1994). A theory of timed automata. Theoretical Computer Science, 126, 183-235.

Anderson, S. and Küster-Filipe, J. (2003). Guaranteeing temporal validity with a real-time logic of knowledge. In Proceedings of IEEE ICDCS 2003 Workshops.

ARINC (2002). TCAS Transition Program (TTP) Industry Alert Bullitin. August, 2002, ARINC.

Aviation Safety Network (2002). Aircraft accident description 01 JUL 2002 Tupolev 154M. Available at http://aviation-safety.net/

BFU (2002). Status Report AX 001-1/-2/02. August, 2002, Bundesstelle für Flugunfalluntersuchung.

Boutilier, C., Friedman, N. and Halpern, J.Y. (1998). Belief Revision with unreliable observations. In *AAAI-98 (Proceedings of the Fifteenth National Conference on Artificial Intelligence).*

Brafman, R., Halpern, J. and Shoham, Y. (1998). On the knowledge requirements of tasks. *Artificial Intelligence*, 98(1-2), 317-349.

Dwork, C. and Moses, Y. (1990). Knowledge and common knowledge in a Byzantine environment: crash failures. *Information and Computation*, 88 (2), 156-186.

Edwards, E. (1972). Man and machine: Systems for safety. In Proceedings of British Airline Pilots Associations Technical Symposium, London, 1972, British Airline Pilots Associations, pages 21-36.

Endsley, M.R. (1995). Towards a Theory of Situation Awareness. *Human Factors*, 37(1), 32-64.

Endsley, M.R. (2000). Theoretical underpinning of situation awareness: A critical review. Endsley, M. R., and Garland D. J. (Eds.), Situation Awareness Analysis and Measurement. Lawrence Erlbaum Associates.

Fagin, R. and Halpern, J.Y. (1988). Belief, awareness and limited reasoning. *Artificial Intelligence*, 34, 39-76.

Fagin, R., Halpern, J.Y., Moses, Y. and Vardi, M.Y. (1995). *Reasoning about Knowledge*. Cambridge, Mass., MIT Press.

Fischer, M.J. and Immerman, N. (1986). Foundations of knowledge for distributed systems. In J.Y. Halpern (Ed.), *Theoretical Aspects of Reasoning about Knowledge: Proc. 1986 Conference*, pp. 44-64. San Francisco, California, Morgan Kaufmann.

Halpern, J.Y. and Fagin, R. (1989). Modelling knowledge and action in distributed systems. *Distributed Computing*, 3(4), 159-179.

Hintikka, J. (1962). Knowledge and Belief. Ithaca, N.Y., Cornell University Press.

Hollnagel, E. (1995). The art of efficient man-machine interaction: Improving the coupling between man and Machine. In J.-M. Hoc, P.C. Cacciabue, E. Hollnagel (eds.), Expertise and Technology:

Cognition & Human-Computer Cooperation, Lawrence Erlbaum Associates.

Honeywell (2000). TCAS II/ACAS II. Collision Avoidance System User's Manual. ACS-5059 Rev.-5-02/2000. Honeywell.

Leveson, N.G. (1995). SAFEWARE: System Safety and Computers. Addison-Wesley.

Norman D.A. (1999). The Invisible Computer. The MIT Press.

Rushby J. (2002). Using model checking to help discover mode confusions and other automation surprises. Reliability Engineering and System Safety 75(2):167-177.

Williams, R., Slack, R., Stewart, J. (2000). Social Learning in Multimedia. Final report, EC targeted socio-economic research, project: 4141 PL 951003, Research Centre for Social Sciences, January 2000, The University of Edinburgh.

Wright, P., Fields, B., Harrison, M. (2000). Analysing Human-Computer Interaction as Distributed Cognition: The Resources Model. Human Computer Interaction Journal 51(1):1-41.

Perrow, C. (1999). Normal Accident: Living with High-Risk Technologies. Princeton University Press.

Vygotsky, L. (1978). Mind in Society. Harward University Press.