AUTOMATION AND LIABILITY

an analysis in the context of socio-technical systems

GIUSEPPE CONTISSA
UNIVERSITÀ DEGLI STUDI DI BOLOGNA
UNIVERSITÀ LUISS GUIDO CARLI

i-lex. Scienze Giuridiche, Scienze Cognitive e Intelligenza Artificiale
Rivista quadrimestrale on-line: www.i-lex.it

AUTOMATION AND LIABILITY: AN ANALYSIS IN THE CONTEXT OF SOCIO-TECHNICAL SYSTEMS

GIUSEPPE CONTISSA

Abstract: The introduction of highly automated technologies in socio-technical systems gives rise to new legal questions, especially as concerns liability for accidents, calling for new models of allocating decision-making tasks between humans and machines. In this article, I shall analyse the impact of automation in the allocation of liability within STSs. I shall also introduce the Überlingen aviation accident and analyse a lawsuit triggered by the accident, concerning the assessment of liabilities related to the development and use of automated technologies. I will conclude with some considerations on the role that the law can play with regards to the development and use of highly automated systems.

Keywords: automation, autonomy, responsibility, liability, socio-technical system

1. Introduction

Today, the main productive, administrative, and social organisations may be described as complex socio-technical systems (STSs), namely, systems that combine technological artefacts, social artefacts, and humans[1].

Technical artefacts, which to some extent involve the use of automated tools and machines, determine what can be done in and by an organisation, amplifying and constraining opportunities for action according to the level of their automated technology. Social artefacts, like norms and institutions, determine what should be done, governing tasks, obligations, goals, priorities, and institutional powers. However, norms need to be understood, interpreted, negotiated, and actuated by humans. More generally, humans play an essential role in

[1] J. K. B. Olsen et al., *A Companion to the Philosophy of Technology*, John Wiley & Sons, 2012, pp. 223-226.

Pubblicato il 3 Dicembre 2017

the functioning of STSs, providing them with governance and maintenance and sustaining their operation. The specific nature of STSs with respect to other sorts of systems, where we can observe the interaction between man and machines, is grounded in the fact that STSs involve humans not only in their role as users of the system but also in their role as operators[2].

For example, the functioning of an air traffic management system (ATM) may be described as the result of the interplay of technical artefacts (aircraft, control towers, airports, radars, etc.), human operators (pilots, air traffic controllers, airport operators, safety operators, technicians, etc.), and social artefacts that coordinate behaviours (including norms, such as aviation laws, technical procedures, manuals, and institutions, such as air carriers, air navigation service providers, safety agencies).

These systems are increasingly reliant on computer technologies, and they operate by interconnecting their information systems, as well as by employing automated technologies that sometimes replace humans, though they more often form part of organisational procedures and structures based on human-machine interaction. The complexity of automated socio-technical systems requires that the interaction between the technical component and the behaviour of the users be coordinated by sets of norms and rules applying to operators (legal norms and operative rules, determining what measures must be taken and in what situations) and to technical artefacts (standards and technical rules describing how technologies should be designed and used).

In the context of STS, the allocation of functions, responsibilities, and resulting liabilities may be viewed as a governance mechanism making it possible to enhance the functioning of the STS.

Although the introduction of automation in STSs usually ensures higher levels of efficiency and safety, their internal complexity, and the unpredictability of the environment in which they operate, often lead to accidents, which may be caused by different factors: reckless behaviour by human agents (like in the Costa Concordia accident), technical failures, inappropriate conducts (like in the Linate airport

[2] P. Vermaas et al., *A philosophy of technology: from technical artefacts to sociotechnical systems*, in *Synthesis Lectures on Engineers, Technology, and Society*, vol. VI, no. 1 (2011), pp. 1-134, p. 70.

accident)[3], and organizational defects (like in the Überlingen mid-air collision)[4].

Moreover, despite the fact that humans play a crucial role in STS, the more automation is deployed, the more are operations performed by technological tools, while humans are coupled with them in order to carry out a mere control function. In case of malfunctioning, the human being is called on to take the place of the machine, even if he has already lost the necessary experience needed to perform the task. In safety studies[5], it is shown how the STS risks can be reduced thanks to proper technological and organisational choices and thanks to a shared safety culture[6].

The development, deployment, and use of highly automated technologies in STSs gives rise to new legal questions, especially as concerns liability for accidents, calling for new models of allocating decision-making tasks between humans and machines. This change will make it necessary to critically revise the way tasks, roles, and liabilities are allocated and, as a consequence, to adapt this to the applicable legal context. However, the legal literature is still fragmented and has not yet addressed these problems on a unified approach in the context of complex STSs[7].

In this article, I shall analyse the impact of automation in the allocation of liability within STSs. I shall firstly discuss the relation between responsibility for the execution of a task and legal liability, in the context of the introduction of automation in STSs. Then, two aspects of highly automated systems (namely, the system autonomy, and the software component) that may have an impact on the liability allocation will be analysed. On the basis of such analysis, I shall present an actor-based analysis of liability allocation, focused in par-

[3] C. Johnson*, Linate and Überlingen: Understanding the Role that Public Policy Plays in the Failure of Air Traffic Management Systems,* in *Proceedings of the ENEA International Workshop on Complex Networks and Infrastructure Protection,* 2006.

[4] See *infra* section 7.

[5] J. Reason, *Human error,* Cambridge university press, 1990

[6] C. Perrow, *Normal accidents: Living with high risk systems*, New York: Basic Books, 1984

[7] Among the existing literature, see T. O. Jones, J. R. Hunziker, et al., *Product liability and innovation: Managing risk in an uncertain environment*, National Academies Press, 1994

ticular on the liability of individuals and of enterprises. Then, some considerations will be made on the final liability allocation, taking into account in particular the level of automation of the technology. In the final part of the article, I shall introduce the Überlingen aviation accident, and discuss one of the lawsuits triggered by the accident, concerning in particular the role of automated technologies in the event and the assessment of liabilities related to their development and use. I will conclude with some considerations on the role that the law can play with regards to the development and use of highly automated systems.

2. Task-responsibilities and the impact of automation

In order to introduce the analysis of liability issues in STSs, we need to refer to the concept of task-responsibility. According to Hart, "whenever a person occupies a distinctive place or office in a social organisation in which specific duties are attached to provide for the welfare of others or to advance in some specific way the aims or purposes of the organization, he is properly said to be responsible for the performance of these duties, or for doing what is necessary to fulfil them"[8]. Thus, when saying that a person x is task-responsible for an outcome O, we mean that x, given his role or task, has a duty to ensure that O is achieved. In particular, we shall use the term "task-responsibility" to cover both commitments linked to the agent's permanent position within the organisation (e.g., in aviation, the controller's task of taking care of a sector of the airspace) and commitments taken in particular situations (e.g., a controller's task of taking care of a particular aircraft having accepted a request by an unavailable colleague). Such responsibilities can concern the direct achievement of a

[8] H. L. A. Hart, *Punishment and responsibility: Essays in the philosophy of law*, Oxford University Press, 2008, p. 212. As Hart says, here we must understand the notion of a role in a very broad way, namely, as including any "task assigned to any person by agreement or otherwise" (Hart (2008).). The term "role responsibility" suggests the idea that such duties are ties to fixed positions within a permanent organisational structure. We prefer to use the term task-responsibility to clarify that these positions can depend on particular contingencies or agreements between peers, not necessarily flowing from a given organisation chart.

certain outcome, as well as supervision of the work of colleagues and collaborators. Task-responsibilities may also be assigned to groups rather than to single persons.

A crucial problem in complex STSs is that technological risks, vulnerabilities, and failures often occur because responsibilities are inappropriately assigned. Moreover, as Hollnagel and Woods have pointed out[9], when a high level of technology and automation is introduced in STSs, the increased system complexity often leads to increased task complexity.

This may seem to be something of a paradox, since it is often assumed that technological innovations tend to simplify human tasks. On the contrary, innovations may increase the overall performance of the system while reducing the human workload, but they rarely reduce task complexity, adding human supervisory-task functions to those directly discharged by humans. Therefore, without a way to effectively connect responsibilities to tasks and roles in the organisation, accountability may not be able to be ensured, since it may not be clear who had responsibility for a particular aspect of system's functioning.

However, mapping tasks and responsibilities (and assigning correlated liabilities) is not always sufficient to ensure safety and efficiency within STSs. In particular, it has been pointed out that precisely and univocally detailed task mapping might be reasonable for systems that can be completely specified, but may not be sufficiently adequate for systems that are underspecified.

In particular, complex STSs, where the technical component includes advanced technical devices and infrastructures with high levels of automation, will require a broad and flexible allocation of tasks: the relative autonomy of such technical components means that their behaviour can be difficult to predict[10], and flexible adjustments, rather than mechanical responses, are required by humans supervising or directing them.

On the one side their use by human operators, or misuse with associated failures, is relevant in assessing liabilities; on the other side,

[9] E. Hollnagel and D. D. Woods, *Joint cognitive systems: Foundations of cognitive systems engineering*, CRC Press, 2005.

[10] Vermaas et al., *A philosophy of technology: from technical artefacts to socio-technical systems*, cit., p. 78

they may often be considered not only as tools in the hands of human operators but also as agents themselves, actively contributing to carrying out the tasks within the STS, which may have an impact on the allocation of liabilities for failures.

3. Autonomous systems

The salient feature of the most advanced automated instruments is the fact that they possess a certain degree of autonomy. The notion of autonomy has been discussed in many fields (philosophy, cognitive science, computer science), leading to the distinction of several types or aspects of autonomy.

An autonomous system 1) has "the capacity to operate in the real-world environment without any form of external control, [...] for extended periods of time"[11]. Thus, a system is autonomous to the extent that it can accomplish a task by itself, without external directions. 2) It has "the capacity to learn what it can to compensate for partial or incorrect prior knowledge"[12]. Thus, the system has cognitive capacities and, in particular, the capacity to obtain new knowledge, interacting with the environment. 3) It acts to achieve its own goals, it perceives and interprets its environment, has internal states, and its behaviour also depends on such internal states[13]. Here the focus is on the cognitive architecture of the system and on the way in which this architecture mediates external inputs and system behaviour. 4) It can make decisions and take actions on its own, without the guidance of humans or other systems; roughly speaking, this means that control lies –at least partially– inside the system and not outside[14].

[11] G. A. Bekey, *Current trends in robotics: technology and ethics*, in *Robot ethics: the ethical and social implications of robotics*. MIT Press, Cambridge (2012), pp. 17-34, p. 18.

[12] S. J. Russell and P. Norvig, *Artificial Intelligence. A Modern Approach*, 3rd ed., Prentice Hall, Englewood Cliffs, N. J. 2010, p. 39.

[13] C. Castelfranchi and R. Falcone, *Founding Autonomy: The Dialectics Between (Social) Environment and Agent's Architecture and Powers*, in *Agents and Computational Autonomy: Potential, Risks, and Solutions*, vol. MMCMLXIX (2004), p. 40.

[14] M. Wooldridge, *An Introduction to MultiAgent Systems*, John Wiley & Sons, 2002, p.27.

All these aspects of autonomy are connected but do not necessarily coexist and converge in the same way in different systems. However, the more systems are endowed with autonomy in one or more of the senses described above, and therefore capable of autonomous decisions and actions (usually coupled with feedback/learning capabilities), the more it is difficult to understand and anticipate their behaviour on the basis of the working of their internal mechanisms, and in particular, considering the programming instructions implemented into them[15]. A common alternative is to conceive them adopting the "intentional stance," namely the strategy of interpreting the behaviour of such entity by using the mentalistic notions more typically applied to human agents, such as knowledge, belief, and intention[16].

In fact, a human operator interacting with such advanced devices will normally have little knowledge of their internal functional mechanisms, and not even the programmer who built them will be able to view their present and future behaviours as limited to the execution of the computational processes they consist of.

The overall interpretation of their behaviours will be based on the hypothesis that the systems are operating "rationally", by adopting determinations appropriate to the purposes that have been assigned to them, on the basis of the information available to them, and in the context in which they are going to operate. In other words, we will assume that these systems possess (in different degrees depending on the capabilities of each system) autonomous cognitive states and behaviours[17].

Therefore, the reason why the effects of what an autonomous system does will fall on the operator is not because the operator has wanted or predicted its behaviour, but rather because the operator has used the autonomous system as a cognitive tool, to delegate part of his tasks, and therefore is committed to accept the results of its cognitive activity.

Thus, since the user relies on the autonomous system's cognition, the fact that the user is responsible (in the sense that he will bear the

[15] See *infra* section 4.

[16] D. C. Dennett, *The Intentional Stance*, MIT Press, 1987.

[17] G. Sartor, *Cognitive automata and the law: electronic contracting and the intentionality of software agents*, in *Artificial intelligence and law*, vol. XVII, no. 4 (2009), p. 253

legal consequences resulting from the autonomous system's activity) does not exclude, but rather presupposes, the legal relevance of the system's cognitive states and processes: the liability of the user—in case of systems with high level of autonomy—is more similar to the liability of the employer for the employee's torts (vicarious liability) rather than the liability of a custodian. In fact, vicarious liability is not based upon the fact that the employer can foresee the behaviour of the employee, but rather on the fact that the employee accomplishes a tort while acting in the course of his employment.

Moreover, according to several studies in the area of cognitive-systems engineering[18], when automation in an STS has more or less completely taken over, humans become controllers of automated systems rather than operators. These systems perform cognitive functions, acquire information from the environment, process it, and use the knowledge so obtained to achieve the goals assigned to them, as specified by their users. It has been observed that when one or several operators and one or several automated support systems interact for the fulfilment of a task, it would be better to describe humans and technology not as two interacting "components", but as making up a joint (cognitive) system.

The term "joint cognitive system"[19] means that control is accomplished by a combination of cognitive systems and (physical and social) artefacts that exhibit goal-directed behaviour.

Several studies describe these fusions between humans and machines as "hybrids"[20]. In hybrids, the participating individuals or collective actors are not acting for themselves but are acting for the hybrid as an emerging unit, namely, the association between humans and non-humans. They do so in the same way as managers who are not acting on their own behalf but are "agents" or "actants" representing

[18] Cognitive systems engineering addresses questions such as how to make use of the power of contemporary technology to facilitate human cognition, how to understand the interactions between humans and technologies, and how to aid design and evaluation of digital artefacts.

[19] E. Hollnagel, *The human in control: Modelling what goes right versus modelling what goes wrong*, in *Human Modelling in Assisted Transportation*, Springer, 2011, pp. 3- 7.

[20] G. Teubner, *Rights of non-humans? Electronic agents and animals as new actors in politics and law*, in *Journal of Law and Society*, vol. XXXIII, no. 4 (2006), pp. 497-521.

their “principal”, which is the corporation as a social system. In these cases, agency pertains not only to humans or to machines but also to the hybrid itself, such that human machine interaction and trust play a decisive role in assessing and allocating liability. From this perspective, a relevant (and still open) question is that of how to deal with cases in which, as in the Überlingen aviation accident[21], conflicting information is provided to human agents (pilots) by other humans (controllers) and automated systems, and more generally what kinds of priorities should be given to different signals, and when humans may override automatic devices.

4. Software and liability

Automated systems are usually the result of a combination of hardware and software components. Liability for both hardware and software failures is therefore particularly relevant in this context. While hardware failures usually fall into product liability, we need to analyse separately some specific issues related to software failures.

As is well known, the use of software always implies the possibility of a failure, since not all software defects can be detected in the development and validation phases, and so it is impossible to guarantee that a piece of software will be absolutely error-free[22], even though good development methods and skilful programming can bring mistakes to a minimum. Thus, operators must be ready to mitigate the consequences of software malfunctioning, and be capable of manually handling high-priority tasks when there is a software failure. Competent effort in both preventing malfunctioning and mitigating its impact is particularly important when software is the core component of a safety-critical system (which is the case for many STSs), whose failure could result in death, injury or illness, major economic loss, mission failure, environmental damage, or property damage.

[21] See *infra* section 7.

[22] F. E. Zollers et al., *No more soft landings for software: Liability for defects in an industry that has come of age*, in *Santa Clara Computer & High Tech. LJ*, vol. XXI (2004), p. 745, p. 745.

The number and complexity of roles and tasks involved in the development, implementation, and use of software makes the assignment of responsibility a problematic issue, so it is often very difficult to identify exactly what went wrong, who is responsible towards damaged third parties, and to what extent: the developer of the software, the implementer-customiser, or the operator.

There is currently much discussion as to whether strict (no-fault) liability should be imposed on the producer/manufacturer of software for loss or injury caused by the software. Those in favour of this proposition claim that producers are in the best position to prevent defects in the products. Moreover, it may be argued that producers can best absorb or spread losses in cases where damage was caused by a software defect, even though no negligent behaviour took place during software development. Usually, to mitigate this approach, the concept of misuse (or contributory negligence) is introduced, such that a user might be held partly or fully responsible whenever he uses the software in an incorrect or improper way, and as a consequence acts negligently. Moreover, software development contracts and licenses usually include strong liability limitations or even exemptions from developers/providers' liability for losses or injuries caused by their products. Such a restriction, however, only apply to the parties to such contracts and licences, not to damaged third parties.

Others claim, on the contrary, that by making software producers liable for all damages caused by their software, we would put many producers out of the market (in particular those who are delivering free or open-source software), and reduce the creativity of the software industry, thus stifling technological progress. Moreover, insuring risks for software liability is very difficult, since these risks are very difficult to assess and quantify. Finally, making software producers liable may decrease the incentive that users have to identify defects and help to fix them in cooperation with developers.

The liability regime applicable to loss or injury resulting from defective software may vary depending on (a) whether the software is deemed a service or a product, and (b) whether liability-exclusion clauses have been included in the contractual relations between the parties.

(a) A software product's qualification as a product or as a service is significant because only fault-based liability usually applies to services, while a stricter liability for damages usually applies to producers of defective products, as established, for example, in the European Union under Directive 85/374/EEC.

In particular, the issue of whether a software may be viewed as a source of information or as a technical device is relevant because the liability standards for providing erroneous information are higher than those for providing a faulty device. Of particular interest in this context are the US "aeronautical charts" cases[23], where the courts categorised charts as products rather than as sources of information, assuming that a nautical chart or an airline chart is similar to other navigation instruments such as a compass or radar finder which, when defective, can prove to be dangerous. Since the charts were considered to be a product, the judges held their producer liable under a strict liability rule. Thus, the judges considered charts to be different from books, newspapers, or other sources of information, as well as from consultancy services, to which judges usually do not apply strict liability standards (they do not consider authors, publishers, or consultants liable for providing, without any fault, wrong information reliance on which leads people to harmful consequences). The chart cases, by analogy, support the view that software, too, may be viewed as a product (rather than as a service) and may be subject to strict liability. However, the debate in this matter is still open, as the case law is uncertain, even more so when addressing new subject matter, such as the liability of providers of GPS services for mistaken information provided to their users. However, in EU the question of how software ought to be qualified under the EU product liability regime remains partly unresolved, since in many cases the conceptual distinction between faulty software and faulty instructions remains unclear[24].

(b) Also relevant are contractual relations between the parties, which may include liability-limitation clauses. However, the strong liability clauses written into software development contracts and licenses can only reallocate liabilities between the parties to such con-

[23] (Aetna Casualty & Surety Co. v. Jeppesen & Co., 642 F.2d 339, 342-43 (9th Cir. 1981); Saloomey v. Jeppesen & Co., 707 F.2d 671, 676-77 (2d Cir.1983); Brocklesby v. United States, 767 F.2d 1288, 439 (9th Cir. 1985); Fluor Corp. v. Jeppesen & Co., 170 Cal.App.3d 468, 475, 216 Cal.Rptr. 68, 71 (1985))

[24] G. G. Howells, *Comparative product liability*, Dartmouth Publishing Group, 1993, pp. 34-35.

tracts, without applying to third parties damaged by the software's malfunctioning (e.g., passengers and/or their relatives seeking damages). In addition, it should be borne in mind that liability waivers are carefully reviewed by courts in light of their (potentially diverging) national standards concerning the legal validity and enforceability of such waivers in highly regulated sectors, such as ATM.

When the issue of the liability of software producers is addressed from a consumer-protection or a business-to-business perspective, while implementing different liability rules on the basis of such distinct contractual relationships, uncertainties inevitably arise with regard to computer programs released under an open source licence and whose source code is developed and shared within a community. Considering the nature of open-source software development, holding open-source developers liable for their code would imply a form of collective responsibility that would be hard to define. In addition, the developer and the user of open source software often have neither a business-to-business nor a business-to-consumer relationship with each other, such that it would be hard for the courts to decide what kind of liability rule should be applied in light of an uncertain (and possibly inexistent) contractual relationship between them.

5. Actor-based analysis of liability

Let us now analyse the impact on liability caused by the introduction of automation and the resulting changes in the allocation of tasks. In order to analyse this impact, we choose to adopt an actor-based approach, namely, to analyse the grounds for attribution of liability for the main actors involved in an STS, and to assess the impact resulting from the introduction of automation. The actors in an STS may be broadly classified into two main categories:

1. human operators working with the technologies, where the ground for attribution is mainly negligence;
2. enterprises, where the grounds for attribution are usually vicarious liability (when employees are at fault) and product liability (for designing, manufacturing, and maintaining the technologies), but also organisational liability.

There may, of course be other actors (certificators, standard setters, the government, safety bodies, etc.), but we will focus on analysing the liability risk for humans and enterprises working with technologies.

5.1. Liability of individuals

Whenever there is a failure in an STS, we try to connect the failure with the missing or inadequate execution of a task, and so with the (natural or legal) persons who were responsible for the task. As a consequence of the failure to comply with their task-responsibilities, these persons are subject to blame, penalties, and/or the payment of damages.

This approach, namely, using individual schemes of responsibility and liability, may be inadequate in complex STSs, since many people may have contributed to the failure at different levels in the organisation, and since liability should in some cases be placed on the organisation rather than (or in addition to) being placed on particular individuals.

First of all, it may not be possible to link the failure only to the inadequate performance of a single person. This is the so-called problem of many hands:[25] it may be difficult, or even impossible, to find any one person who can be said to have independently and by his own hands carried out the behaviour that resulted in a failure; on the contrary, the failure results from the sum of a number of mistakes by different human or artificial agents in different positions. This means that, as the responsibility for any given instance of conduct is scattered across more people, the discrete responsibility of every individual diminishes proportionately.

In complex organisations there is an observed tendency to manage this problem by attributing most failures to the mistakes of the human operators whose behaviour is nearest to the proximate cause of the accident. As Perrow points out: "Formal accident investigations usually start with an assumption that the operator must have failed, and if this attribution can be made, that is the end of serious inquiry. Finding that faulty designs were responsible would entail enormous shutdown and retrofitting costs; finding that management was responsible would threaten those in charge, but finding that operators were

[25] M. Bovens, *The quest for responsibility: Accountability and citizenship in complex organisations*, Cambridge Univ Pr, 1998.

responsible preserves the system, with some soporific injunctions about better training"[26]. This is reflected also in the outcomes of many legal judgements related to accidents in different activities and sectors (especially that of transportation: trains, aircraft; that of healthcare, and others).

With the introduction of automation, as task-responsibilities are progressively delegated to technology, liability for damages will tend to shift from human operators to the organisations that designed and developed the technology, defined its context and uses, and are responsible for its deployment, integration, and maintenance within the STS.

This change of paradigm becomes important not only ex post, when liability needs to be allocated, but also ex ante, when prevention and precaution need to be applied: this is how this paradigm contributes to reinforcing a "just culture" in the field of safety and security. Sanctions on operators alone cannot prevent accidents from happening: on the contrary, sanctions often cause them to take excessive cautions in order to avoid personal responsibility rather than to ensure the efficiency and safety of the outcome[27].

Individual liability of the human operator would persist only when the operator acted with an intention of causing harm or injury or with recklessness.

5.2. Liability of enterprises

Enterprise liability could be seen as quasi-strict, in the sense that it attaches to an enterprise regardless its intentions or negligent behaviour. There are two main types of enterprise liability: vicarious liability and organisational (or "systemic") fault liability.

1) Vicarious liability is the liability of the employer (enterprise) for the wrongful act of an employee, when the act is performed within the scope of the employment agreement. The law of employer liability varies among legal systems. For example, common-law systems and many civil-law systems (such as the Italian and French systems) acknowledge a no-fault liability of employers for the wrongful act of their employees, while under the German Civil Code (Art. 831 BGB)

[26] Perrow, *Normal accidents: Living with high risk systems,* cit., p. 146.

[27] S. Dekker, *The criminalization of human error in aviation and healthcare: A review*, in *Safety science*, vol. XLIX, no. 2 (2011), pp. 121-127.

an employer is only liable if it is itself at fault (the employer is presumed to be at fault, but this presumption is rebuttable).

2) Organisational ("systemic") fault[28] liability emerges when the harm is directly related to or caused by business activity[29]. There should also be a "fault" in the enterprise in its behaviour, but this has to be understood in a special way: the enterprise is at fault whenever the harm it caused could have been prevented by reasonable investment to improve the safety and security of the product or activity in question. This idea is captured to some extent by what is known as Learned Hand's formula: one is negligent when the cost of taking adequate precautions to avoid the harm is inferior to the expected cost of the injury, an expected cost obtained by multiplying the probability of injury by the cost of the injury (if it were to happen). The burden of proof for organisational fault falls on the enterprise: it is up to the enterprise to exonerate itself by proving that there was no reasonable measure of care that could have been taken to avoid the harm. Enterprise liability can go beyond "fault" liability, covering all insurable losses and injuries: those that may be linked to quantifiable risks incident to the enterprise's activity. In this way, risk can be spread in the most effective way possible (in that the enterprise can distribute the costs of expected injuries over its customers by increasing the cost of the products it sells or the services it offers). As concerns employee liability, employees are liable to third parties for intentional harm but are usually exonerated for negligent acts (except when these are reckless).

Enterprise liability, then, covers certain special cases in which the forms of liability listed above do not apply. These special cases are (a) strict liability for technical risks, (b) product liability, and (c) statutory negligence.

(a) Strict liability for technical risks is a kind of liability specifically created to deal with risks in transportation, where we have a

[28] The term "systemic" is used here because—unlike vicarious liability, where there needs to be a wrongful act in order for an employer to be held liable—organisational liability attaches regardless of any wrongful or negligent acts on the part of any individual within the organisation, for it is a liability arising directly out of inadequate organisation of the business activities of the enterprise itself.

[29] G. Brüggemeier, *Common principles of tort law: a pre-statement of law*, British Institute of International and Comparative Law, 2006, pp. 117 – 132.

"specific risk of operating these technical transport means"[30], which may go out of control and lead to serious damage. This applies in particular to trains, automobiles, and aircraft. A recent trend has been to introduce a general clause providing strict enterprise liability for highly dangerous activities, such as those involving nuclear power production, while possibly imposing compulsory insurance. Strict liability for technical risks can be avoided only (i) in case of force majeure—covering all natural and human-caused events and phenomena which are unavoidable or uncontrollable (such as earthquakes, floods, military activities, and power failures)—or (ii) if the damage was entirely caused by a deliberate third-party act or omission specifically designed to bring about the damage in question; or (iii) if it was entirely caused by the negligence or other wrongful act of a government entity. Liability caps are usually established for strict liability. It is also possible to establish insurance pools, ensuring insurance coverage for victims by having different insurers share the burden of compensation.

(b) Another kind of special-case liability is product liability[31]. The necessary conditions for this kind of liability to emerge are a harm, a defect in a product, and a causal link between the harm and the defect. A possible line of defence against claims concerning harm caused by defective products is provided by developmental-risk doctrine, according to which an enterprise could be exonerated from liability if it proves that at the time it developed the product, the current state of the art in the field could not enable it (or anyone else) to spot the defect. In other words, the enterprise should prove that the defect was undetectable at the current state of knowledge in the particular field of business activity. Product liability also covers damages for pain and suffering inflicted on the victims, but the procedure on which basis this should be done may vary across different jurisdictions. In the European Union, it is left for individual Member States to frame according to their national laws.

(c) The last type of special case in enterprise liability is statutory negligence: if an enterprise breaches a regulation on how certain ac-

[30] G. Brüggemeier, *Modernising civil liability law in Europe, China, Brazil and Russia: texts and commentaries*, Cambridge University Press, 2011, p. 102.

[31] Introduced in EU legislation by EU Directive 85/374/EEC, concerning liability for defective products.

tivities should be organized, performed, and managed, then that enterprise is liable for the harmful effects of such activities. In other words, liability automatically follows from the violation of a standard of care established by the law for certain activities posing a danger to people, property, or the environment. Both statutory negligence and product liability are based on standards of care: the difference is that in product liability, the product must be defective, while in statutory negligence it is sufficient to infringe a standard.

6. Levels of automation and allocation of liability

When automated systems are increasingly introduced into STSs, the main effect is that liability for damage or harm is gradually transferred from humans to enterprises using the automated technology that replaced the human operator and /or to the technology developer (programmer, manufacturer) that created the technology.

While the trend of transferring liability from the individual to the enterprise has been observed for quite a long time in STSs (in ATM, for example), new technologies currently being developed and implemented will accelerate this trend, since they will deeply impact the tasks of human operators, not only quantitatively but also qualitatively, replacing human operators in their higher cognitive functions, ranging from the analysis of information, to the selection of a decision or an action, to the fully automated implementation of the chosen action.

Of course, not all advanced technological systems will possess all those cognitive functions to the same extent. For example, many currently employed automated systems are not designed to automatically implement the chosen actions, but only to suggest actions to be executed by the human operator.

In order to evaluate the final liability allocation between different actors, it will necessary to assess each technology's different levels of automation in performing different cognitive functions (acquiring information, analysing information, making decisions, and acting on them).

Different levels of automation for different cognitive functions will usually imply a different distribution of the corresponding task-responsibilities of the actors involved, including operators, as well as other actors involved in the technology (managers, producers, trainers, and maintainers). This, too, will have an impact on the final allocation of legal liability.

As a general rule, a gradual shift can be observed from personal liability to general enterprise liability and product liability. Thus, as tools become increasingly automated, liability will increasingly attributable to the organisations that use such tools, and to those that build them or are in charge of maintaining them, rather than to the operators who interact with them.

However, intermediate levels of automation are sometimes those that also create higher levels of legal risk for certain actors. This is because in these levels there is a high fragmentation of task-responsibilities between the automated technology and the operator, possibly leading in some circumstances to uncertainty in the assignment of tasks. In addition, intermediate levels of automation usually also imply greater complexity in the human-machine interface, since fragmented tasks require more interaction between a technology and its operator.

In legal terms, this may translate to an increased duty of care, resulting in a higher liability risk for (a) the operator; (b) the organisation employing the operator, both for vicarious liability and for organisational liability; and, finally, (c) the producer of the technology, since higher complexity in the human-machine interface would increase the risk of technological failure[32].

In order to limit liability risk, some legal measures may be introduced in the STS to provide evidence that the operators' actions were compliant with professional due care. In particular, task allocation should be set forth in legally relevant documentation, such as technical manuals for the use of the technology, "concepts of operations", and training manuals.

The manufacturers' liability defences can be strengthened by adopting common industry standards. To strengthen the "state of the art defence", common industry standards can be adopted in order to ensure that at least the customary standard of industry practice is met. Another factor that affects liability allocation is the scope of the man-

[32] H. Schebesta et al., *Design according to liabilities: ACAS X and the treatment of ADS-B position data*, in *Proceedings of the SESAR Innovation Days (2015)*, ed. by D. Schaefer, 2015.

ufacturer's discretion under a given standard. Generally, with less discretion for manufacturers, their liability risk decreases, even though compliance with standards and regulations does not necessarily exonerate a producer from liability. Therefore, in order to limit the liability of manufacturers, design options for automated technologies ought to be mandated or constrained for all manufacturers. Finally, in order to address warning-defect risks, it is suggested that manufacturers provide adequate warning information about the technology.

7. The Überlingen mid-air collision and the liability for automated technologies

The Uberlingen mid-air collision was a catastrophic aviation disaster, caused by a series of different failures involving automated technologies, human operators and different organisations responsible for the safe management of ATM operations. Among the many lawsuit triggered by the accident, one in particular concerned the liability of manufacturers of a highly automated technology, and may considered as an important precedent for the issues of liability and automation not only for the field of aviation but also for other socio-technical systems.

The official German Accident Investigation Report[33] described the accident in the following way:

> On 1 July 2002 at 21:35:32 hrs a collision between a Tupolev TU154M, which was on flight from Moscow/Russia to Barcelona/Spain, and a Boeing B757-200, on a flight from Bergamo/Italy to Brussels/Belgium, occurred north of the city of Ueberlingen (Lake of Constance). Both aircraft flew according to IFR (Instrument Flight Rules) and were under control of ACC Zurich. After the collision both aircraft crashed into an area north of Ueberlingen. There were a total of 71 people on board of the two airplanes, none of which survived the crash.

[33] German Federal Bureau of Aircraft Accidents Investigation (BFU), Investigation Report, AX001- 1-2/02, 3 (May 2004).

The Tupolev was the Bashkirian Airlines flight BAL2937, while the Boeing was the DHL cargo flight DHL611. Despite the fact that the accident took place over the towns of Überlingen and Owingen in southern Germany, near the Swiss border, that sector of airspace was controlled by Skyguide, the Air Navigation Service Provider which manages and monitors Swiss airspace.

The Air Traffic Controller (ATCO) responsible for monitoring the flights during the night was working in an environment below the prescribed safety standards and– because also of several other technical and organisational problems– failed to monitor the routes of the two flights, and to keep the aircraft at a safe distance from each other. He noticed only less than a minute before the accident that the two aircraft were on the same route.

The use of the technology played a crucial role in the accident dynamics. Both aircraft were equipped with TCAS (Traffic Collision Avoidance System), an advanced automated device designed as a last-resort safety net to prevent air traffic collisions.

On the basis of secondary surveillance radar transponder signals, TCAS can communicate with other TCAS-equipped aircraft within a determinate range about their respective positions, in order to identify potential collisions. TCAS systems can also automatically negotiate a mutual avoidance manoeuvre between two or more aircraft.

In particular, TCAS systems generate Traffic Advisories (TAs) and Resolution Advisories (RAs) for pilots when they receive the signal of another aircraft system in the surrounding airspace.

TAs are indications to the flight crew of a potential threat, so that pilots can try to visually acquire sight of a conflicting aircraft. RAs consists in orders for the pilots of two aircraft in collision route respectively to climb (CLIMB RA) and descend (DESCEND RA) to separate the two flights vertically. RAs shall always have priority over potential conflicting orders, including air traffic control orders.

The rules for the use of TCAS and more generally for Airborne Collision Avoidance System (ACAS) are defined in ICAO Annex 10[34]. TCAS version 7.0, the version of the standard available at the time of the accident and installed on the two aircraft, introduced in

[34] Annex 10 to the Convention on International Civil Aviation (also known as Chicago Convention) of 7 December 1944.

the TCAS logic the RA reversal function, which permits TCAS to reverse the sense of an RA, that is, change from a CLIMB RA to a DESCEND RA where there is a coordinated encounter with another TCAS aircraft. In the Überlingen accident, the issue involving the TCAS technology was that conflicting orders led the Tupolev's pilot to disregard the TCAS advisory.

When the ATCO realised the danger (less than a minute before the accident), he contacted the Tupolev (BAL2937), instructing the pilot to descend by a thousand feet to avoid collision with crossing traffic (the DHL611).

However, seconds after the Russian crew initiated the descent, their TCAS issued a RA instructing them to climb, while at about the same time the TCAS on DHL611 issued a RA instructing the pilots of that aircraft to descend.

DHL611's pilots on the Boeing followed the TCAS instructions and initiated a descent, but could not immediately inform the ATCO due to the fact that the he was dealing with BAL2937 and the radio frequency was already engaged. The Russian pilot on the Tupolev disregarded the TCAS instruction to climb and instead keep descending, following the order of the ATCO. As a result, both aircraft descended. They collided and all 71 persons on board the two aircraft died.

The accident triggered many civil and criminal lawsuit in different jurisdictions[35], and among them, one lawsuit in particular concerned the product liability of TCAS manufacturers. The decision taken by the judges has been considered an important precedent and a relevant development for the legal analysis of product defects in highly automated technology not only for the field of aviation but also for other socio-technical systems[36].

Honeywell International, Inc. and Aviation Communication & Surveillance Systems (ACSS), manufacturers of the TCAS systems

[35] For a complete list of the lawsuits triggered by the Überlingen accident, see G. Contissa et al., *Liability and automation: Issues and challenges for socio-technical systems*, in *Journal of Aerospace Operations*, vol. II, no. 1-2 (2013), pp. 79-98.

[36] H. Schebesta, *Risk Regulation Through Liability Allocation: Transnational Product Liability and the Role of Certification*, in *Air and Space Law*, vol. XLII, no. 2 (2017), pp. 107-136.

involved in the accident, were sued by relatives of the victims in front of Spanish judges.

In deciding the cases, the judges followed the Hague Convention on the Law Applicable to Products Liability of 1973[37]. Article 6 of the Convention applies the law of the manufacturer's principal place of business unless the claimant bases his claim on the law of the place of injury, while Article 11 establishes that there is no requirement for the Convention to be adopted by the country to which law Articles 6 points. On these grounds, the Spanish judges applied Arizona law to ACSS and New Jersey law to Honeywell.

In 2010, the Court of First Instance[38] held the manufacturers liable for a warning/information defect of the TCAS.

In particular, the court found that TCAS Pilot's Guide failed to clearly set forth the priority of TCAS Resolution Advisories over conflicting air traffic control orders. Consequently, the TCAS was considered defective (warning defect) and the two companies were condemned to pay damages to familiars of passengers. The Spanish judge awarded plaintiffs a total of $10,459,810.50 in damages for the deaths of 30 persons, including $6,723,639.45 as to ACSS and $3,736,171.05 as to Honeywell – an average of $348,660.35 per decedent.

In the Appeal of 2012, the Provincial Court of Barcelona[39] considered the TCAS defective not only for warning defects, but also for manufacturing defects and design defects. In particular, the TCAS design was considered unreasonably dangerous, despite it was compliant with available standards.

In 2015, the Supreme Court[40], partially modified the decision of the appeal judgment, so that only the information and manufacturing defects were upheld, and consequently reduced the amount of damag-

[37] 22nd Convention on the Law Applicable to Products Liability, signed in The Hague on 2 October, 1973. The Convention is currently in force in 11 European countries (Spain, France, the Netherlands, Croatia, Finland, Luxembourg, Montenegro, Norway, Serbia, Slovenia, and Macedonia).

[38] Judgment of first Instance of the Provincial Court of Barcelona n. 34 of 7 May 2012, Juicio Ordinario 424/2007, ECLI:ES:APB:2012:6351.

[39] Judgement of the Provincial Court of Barcelona of 7 May 2012, Sentencia n. 230/2012, ECLI:ES:APB:2012:6351.

[40] Judgment of the Supreme Court of 13 Jan. 2015, Sentencia n. 649/2014, ECLI:ES:TS:2015:181.

es payable by manufacturers. However, this decision still represents the largest compensation ever awarded by a European Court in favour of a group of aviation victims.

Concerning product liability in relation to automated technologies, the key issues are usually the proof of (1) defectiveness;and (2) the causal link.

1) Concerning products defects, they are usually classified as manufacturing defects, design defects, and warning/information defects. With complex automated systems, design defects may pose a very high burden of proof to the injured party, as he may be requested to prove that a reasonable alternative design would have prevented the damage (which can be very difficult and expensive for complex technologies), while the producer - depending on the legal system - may rely on the "state of the art" defence or on the "regulatory defence", when the manufacturer cannot implement the alternative design without deviating from a mandatory standard.

2) In complex technological environment, the causation link is often the key point to establish liability, since establishing a causal link between the fault of complex automated technology and the damage may be very difficult. The position of the damaged party can be facilitated in two ways: by inverting the burden of proof (putting upon the producer the burden of excluding the causality), or by diminishing the standard of proof. In particular in case of injuries (as opposed to economic losses), courts tend to be less rigorous in the proof of causal link.

These hurdles were in large part overcome by the Spanish Judges, who condemned the producers of the TCAS to compensate part of the damage.

In particular, the Court of Appeal extended the assessment of TCAS technology under over all the three categories of design defect, manufacturing defect, and warning/information defect. The Supreme Court dismissed the issue of design defect, but uphold the existence of the manufacturing defect and warning/information defect.

Concerning the design defect, the appeal judgment was focused on some important problems of design in the TCAS II system: under some circumstances (that occurred in the Überlingen accident) the system was affected by a software error, so that it failed by not generating the RA reversal signal. The problem, designated as "SA01 error" had been known to manufacturers before the accident, and a software solution, called CP112 update, had been already proposed but not implemented. According to the judge, in case of error SA01, the risk of collision with TCAS when a pilot did not follow his RAs was higher than without TCAS. Therefore, he concluded that the product was defectively designed because it was unreasonably dangerous, so that the risks of the product exceed its benefits. Moreover, he noticed that there existed an alternative design that could have reduced the risk (the one implementing the CP112 update), and that for four years the manufacturers of TCAS knew about the severity of the software design problems and they did nothing. For these reasons, the judge concluded for the defectiveness of the design. This despite the fact that TCAS followed and was certified under standard TSO c119b of the FAA, and that the report of the Technical Center of the FAA had found that the TCAS functioned in a way in which it should have according to the mandatory standard applicable at that time. Instead, the Supreme Court admitted the applicability of the regulatory compliance defence, observing that the manufacturers could not implement the alternative design when complying with the applicable standard, and dismissed the claim for design defect.

Concerning the manufacturing defect, the judgement in Court of Appeal focused on TCAS slowness in processing data, so that it in processed altitude data with a 3 seconds time lag, allowing reversal RAs based on erroneous information. On this regard, the judge noted that the mandatory standard (the DO-185A) required an update every second, so that the minimum rule established was not complied with, providing evidence of a manufacturing defect in the TCAS II, which resulted it into a product unreasonably dangerous, since it had not been manufactured in accordance with its design specifications, with the consequence that the product was not safe and showed an unreasonable danger. The decision was confirmed by the Supreme Court.

Concerning the information/warning defect, the judgement in Court of Appeal focused on the information deficiencies of the manual provided to the Tupolev crew, where information about TCAS functioning was not clear, and incoherent. In particular, the information and the terminology used did not make clear to ignore the ATCO orders or that the TCAS was to be used as the "ultimate line of defense". On this regard, the judge highlighted that The US Federal

Aviation Regulation FAR 91.123, states that pilots are authorised to deviate from the authorisation issued by the Air Traffic Control if the "deviation is made as a response to a traffic alert and to a resolution advisory generated by the collisions avoidance system", and that this ought to have been established with clarity in the manual for the product, that is used as a basis for producing the pilot manual. The fact that the manufacturers had no direct responsibilities for the pilot manual, but only for the product manual, according to the judge did not exempt the defendants from complying with their own and first obligation of informing completely and duly. This point was confirmed by the Supreme Court.

Concerning the causal relationship, the debate concerned whether the causal relationship with the collision should have been established in relation the TCAS defects, or rather in relation to the negligence of other parties, such as the Tupolev crew, the ATCO and Skyguide, being debated in other criminal and civil proceedings. On this regard, the judge highlighted that without the TCAS the accident would not have occurred, since no RA would have been emitted. Moreover, he affirmed that despite the fact that Skyguide's fault in particular lead to the situation resulting in the accident, TCAS was designed precisely for this type of instances: to provide a remedy, in case of other failures in the system. This point was confirmed by the Supreme Court as well.

Analysing the proceedings against the TCAS manufacturers and their outcomes, we may also observe the following:

1) The courts analysed the TCAS technologies functions and its software and hardware components in deeper detail than the Investigation Report's analysis of the technology. In fact, the Report did not consider at all the RA reversal function, and did not analyse its involvement in the causation of the accident. Instead, the courts considered the RA reversal as one of the most significant functions of the technology, and it resulted decisive in evaluating the TCAS's design as defective.

2) The decisions confirm that product liability is likely to become more important when automation is introduced in STSs. This would put manufacturers on the first line of liability even is sectors like aviation, where extensive sets of rules (including international conventions and supra-national rules) establish a system based on strict lia-

bility, designed to put other actors on the first line (namely, air carriers and Air Navigation Service Providers).

3) Design duties remain important with respect to the technologies, and may not be limited to the compliance with mandatory standards. Additionally, the manufatcurers has important duties of information. When a product has inherent risks, warnings must be issued which can minimise or eliminate those risks.

4) Product liability on automated technologies may foster forum shopping, pushing litigation to be commenced in those jurisdictions, usually the United States, where most of the manufacturers have established their principal place of business, and where laws or procedures are open to methods of damages quantification by judges or juries that are more favourable to the plaintiff or include also punitive damages. This may result in even higher liability risks for manufacturers, potentially creating a disincentive for the development and introduction of safety-enhancing automated technologies.

8. Conclusions

Within an STS system, the distribution of functions, tasks, and liabilities has turned into a governance mechanism used to enhance the functioning of the entire system. This is so because in the complex interaction between human, technological, and normative elements lies the engine responsible for the functioning of the entire system.

From this perspective, the STS paradigm makes it possible to understand the problems that need to be (legally and technically) regulated in light of the new theoretical perspective receptive to technical and psychological studies and based on human-machine interaction.

Furthermore, this paradigm deals with general aspects, such as causality, subjective states and risk. Currently, the legal debate on risk (especially in law and economics) focuses on single policies rather than on a systemic approach. This systemic approach in dealing with issues in STS makes it possible to consider the topic of liability from a new perspective: no longer as a static concept—focused on an exclusive subject matter, namely, the application of legal norms to a particular situation—but rather as a dynamic perspective which takes into account the distribution of tasks and functions within the system.

In this context, by regulating risks that are untenable for society and in particular for the legal system, law creates these certainties by making new technologies compatible with the public health and safety and successfully responding to the needs of equity, deterrence, and efficiency.

The law will be increasingly required to intervene directly on the design of automated technologies: addressing liability early on in the design and development process will make it easier, less costly, and less controversial to address legal issues and to apply a more uniform approach to the attribution of liability across technological projects.

To this end, it may be useful to adopt specific methodologies, such as the Legal Case[41], aimed at addressing liability issues arising from the interaction between humans and automated tools, ensuring that these issues are clearly identified and dealt with at the right stage in the design, development, and deployment process.

Besides, in order to govern highly automated and autonomous systems, there is the need to make the law computable, so that is may become an internal component of computational processes rather than an external constraint over them; legal norms and principles must be mapped onto, and partially translated into, computable representations of legal information and reasoning, so that autonomous systems will be capable of directly processing this knowledge and operate effectively on the basis of it.

References

G. A. Bekey, *Current trends in robotics: technology and ethics*, in *Robot ethics: the ethical and social implications of robotics*. MIT Press, Cambridge (2012).

M. Bovens, *The quest for responsibility: Accountability and citizenship in complex organisations*, Cambridge Univ Pr, (1998).

G. Brüggemeier, *Common principles of tort law: a pre-statement of law*, in *British Institute of International and Comparative Law*,

[41] G. Contissa et al., *Classification and argumentation maps as support tools for liability assessment in ATM*, in *Proceedings of the SESAR Innovation Days*, (2013).

(2006).

G. Brüggemeier, *Modernising civil liability law in Europe, China, Brazil and Russia: texts and commentaries*, in *Cambridge University Press*, (2011).

C. Castelfranchi and R. Falcone, *Founding Autonomy: The Dialectics Between (Social) Environment and Agent's Architecture and Powers*, in *Agents and Computational Autonomy: Potential, Risks, and Solutions*, vol. MMCMLXIX, Springer Science & Business Media, (2004).

Contissa, G., Sartor, G., Laukyte, M., Schebesta, H., Lanzi, P., Marti, P. et al. *Classification and argumentation maps as support tools for liability assessment in ATM*, in *Proceedings of the SESAR Innovation Days,* (2013).

Dekker, S., *The criminalization of human error in aviation and healthcare: A review*, in *Safety science*, vol. XLIX, no. 2. Elsevier, (2011).

Dennett, D. C. *The Intentional Stance*. MIT Press, (1987).

Hart, H. L. A. *Punishment and responsibility: Essays in the philosophy of law*, in *Oxford University Press*, (2008).

Hollnagel, E. *The human in control: Modelling what goes right versus modelling what goes wrong,* in *Human Modelling in Assisted Transportation*. Springer, (2011).

Hollnagel, E. & Woods, D. D. (2005). *Joint cognitive systems: Foundations of cognitive systems engineering*, in *CRC Press*, (2005).

Howells, G. G. *Comparative product liability*, in *Dartmouth Publishing Group*, (1993).

Johnson, C. Linate and Überlingen: *Understanding the Role that Public Policy Plays in the Failure of Air Traffic Management Systems,* in *Proceedings of the ENEA International Workshop on Complex Networks and Infrastructure Protection,* (2006).

Jones, T. O., Hunziker, J. R. & others. *Product liability and innovation: Managing risk in an uncertain environment,* in *national Academies Press*, (1994).

Olsen, J. K. B., Pedersen, S. A. & Hendricks, V. F. *A Companion to* *the Philosophy of Technology*. John Wiley & Sons, (2012).

Perrow, C. *Normal accidents: Living with high risk systems*. New York: Basic Books, (1984).

Reason, J. *Human error*. Cambridge university press, (1990).

Russell, S. J. & Norvig, P. *Artificial Intelligence. A Modern Approach* (3. Auflage). Englewood Cliffs, N. J.: Prentice Hall, (2010).

Sartor, G. *Cognitive automata and the law: electronic contracting and the intentionality of software agents,* in *Artificial intelligence and law*, *17* (4), 253. Springer (2009).

Schebesta, H. *Risk Regulation Through Liability Allocation: Transnational Product Liability and the Role of Certification,* in *Air and Space Law*, *42* (2), 107–136. Kluwer Law International, (2017).

Schebesta, H., Contissa, G., Sartor, G., Masutti, A., Paola, T. & Taurino, D. *Design according to liabilities: ACAS X and the treatment of ADS-B position data,* in D. Schaefer (Hrsg.), *Proceedings of the SESAR Innovation Days* (2015).

Teubner, G. *Rights of non-humans? Electronic agents and animals as new actors in politics and law,* in *Journal of Law and Society*, *33* (4), 497–521. Wiley Online Library, (2006).

Vermaas, P., Kroes, P., Poel, I. van de, Franssen, M. & Houkes, W. *A philosophy of technology: from technical artefacts to sociotechnical systems,* in *Synthesis Lectures on Engineers, Technology, and Society*, *6* (1), 1–134. Morgan & Claypool Publishers, (2011).

Wooldridge, M. *An Introduction to MultiAgent Systems. John Wiley*; Sons, (2002).

Zollers, F. E., McMullin, A., Hurd, S. N. & Shears, P. *No more soft landings for software: Liability for defects in an industry that has come of age,* in *Santa Clara Computer & High Tech. LJ*, *21*, 745. HeinOnline, (2004).