TOWARDS A RESILIENCE ASSURANCE MODEL FOR ROBOTIC AUTONOMOUS SYSTEMS

Campean, Felician (1);
Kabir, Sohag (1);
Dao, Cuong (1);
Zhang, Qichun (1);
Eckert, Claudia (2)

1: University of Bradford;
2: The Open University

ABSTRACT

Applications of autonomous systems are becoming increasingly common across the field of engineered systems from cars, drones, manufacturing systems and medical devices, addressing prevailing societal changes, and, increasingly, consumer demand. Autonomous systems are expected to self-manage and self-certify against risks affecting the mission, safety and asset integrity. While significant progress has been achieved in relation to the modelling of safety and safety assurance of autonomous systems, no similar approach is available for resilience that integrates coherently across the cyber and physical parts. This paper presents a comprehensive discussion of resilience in the context of robotic autonomous systems, covering both resilience by design and resilience by reaction, and proposes a conceptual model of a system of learning for resilience assurance in a continuous product development framework. The resilience assurance model is proposed as a composable digital artefact, underpinned by a rigorous model-based resilience analysis at the system design stage, and dynamically monitored and continuously updated at run time in the system operation stage, with machine learning based knowledge extraction and validation.

Keywords: Resilience, Autonomous systems, Systems Engineering (SE), Artificial intelligence, Industry 4.0

Contact:
Campean, Felician
University of Bradford
School of Engineering
United Kingdom
F.Campean@bradford.ac.uk

Cite this article: Campean, F., Kabir, S., Dao, C., Zhang, Q., Eckert, C. (2021) 'Towards a Resilience Assurance Model for Robotic Autonomous Systems', in *Proceedings of the International Conference on Engineering Design (ICED21),* Gothenburg, Sweden, 16-20 August 2021. DOI:10.1017/pds.2021.580

1 INTRODUCTION

Since the term resilience was coined by Holling (1973) as "the persistence of systems and of their ability to absorb change and disturbance and still maintain the same relationships between populations or state variables", the concept has gained traction to describe the ability of technical and human systems to maintain core functions while suffering adverse effects and then return to an accepted state of normality. While resilience in human system refers to the ability to adapt and keep going in the face of change and adversity, technical resilience focuses on the systems ability to maintain functionality such that they can fulfil their mission. Human and technical resilience come together in cyberphysical systems (CPS), which involve both human agents and technical systems, with varying degrees of autonomy. As long as humans are in control of cyberphysical systems, the onus is on humans to ultimately hold the responsibility to monitor and achieve resilience. Most of the research on the resilience of autonomous CPS has concentrated on the software systems, which are responsible for the autonomous control features, revolving around methodologies for assurance of dependability and system safety and security. These developments originated from concerns over safety-critical systems, both military and civil applications, for which the levels of integrity and assurance of the physical systems were well known, and proving safety of the autonomous control features under non-deterministic scenarios was required.

Autonomous systems are becoming increasingly common across the range of applications, from cars, drones, manufacturing systems and medical devices, introduced to address prevailing societal changes, and, increasingly, consumer demand. In all cases, autonomous systems are expected to self-manage and self-certify against risks affecting the mission, safety and asset integrity. While such systems offer great potential for product innovation, their wider adoption by society depends upon the willingness of stakeholders (businesses, engineers, regulators, users) to engage with the development and operation / use, which is contingent upon the trust on their resilient behaviour. While significant progress has been achieved in relation to the assurance of safety and dependability attributes of autonomous CPS systems, there is no similar approach available for resilience. Specifically, we refer to a *resilience assurance* models that integrates coherently across the cyber and physical structure of the CPS system.

This aim of this paper is to introduce a conceptual model for resilience assurance, underpinned by a system of digital artefacts implementing a Digital Resilience Assurance (DRA) model, to support the management of the system for resilience, as well as the communication of resilience with stakeholders. The work is underpinned by a comprehensive discussion of resilience as it applies to ubiquitous robotic autonomous systems (RAS), underpinning the definiton of the conceptual framework and model for resilience assurance based on DRAs, focussing on the integration of resilience within both the system design and system operation lifecycle phases. The ultimate aim is to pave the way towards a comprehensive framework for resilience assurance, underpinned by a methodological and modelling ecosystem, to guide the design, development and operation of future resilient autonomous systems.

The paper starts with a review of existing concepts of resilience and assurance provision methods for computational systems. Section 3 introduces the proposed framework for resilience assurance for RAS, and section 4 provides a succinct summary of impact and further research challenges.

2 RESILIENCE OF ROBOTIC AUTONOMOUS SYSTEMS

Robotic Autonomous Systems (RAS), such as autonomous cars, are CPS designed to perform tasks with a high degree of autonomy. An autonomous system is commonly associated with a computer / computational system that is able to make its own decisions and take its own actions, without human supervision or control (Fisher et al, 2021). The architecture of a RAS includes a physical structure and computational element which control the system based on the continuous assessment of its interaction with the larger system of systems context. In most cases, RAS have significant interactions with human users and stakeholders in the operational phase. Human stakeholders define the mission (e.g. assigning destinations and journey specification to an autonomous car), as well as the potential interactions in the broader system-of-systems context (e.g. as pedestrians and other road users). The safety attributes of the system and the ability to provide assurance and guarantees for its safe operation are fundamental, and have been widely researched and discussed (Fisher et al, 2021). While AI and software systems are the default decision-maker, many current systems have a hybrid supervisory architecture, where human operators are still the final authority in critical situations. Therefore, both the autonomous systems and human are critical elements of the resilience system. This critically depends on the efficiency of the interaction between humans and RAS, which ultimately underpins the trust humans place in it.

2.1 What is resilience?

From the original area of ecology, approaches and definitions of resilience have proliferated widely. Generic definitions include the UN (2009) resilience definition as "the ability of a system, community or society exposed to hazards to resist, absorb, accommodate to and recover from the effects of a hazard in a timely and efficient manner, including through the preservation and restoration of its essential basic structures and functions", and the US National Academy of Science definition as (Connelly et al, 2017) "the ability to prepare and plan for, absorb, recover from, and more successfully adapt to adverse events". From an engineered systems point of view, Hollnagel et al. (2006) provided the widely accepted definition of engineering system resilience as "the ability to adjust its functioning prior to, during, or following changes, disturbances, and opportunities, and thereby sustain required operations under both expected and unexpected conditions". Thus, the resilience of an engineered system is defined in a narrower, technical sense, in relation to the capability to monitor, anticipate and respond to events, and also to learn from the past and anticipate what could encounter in the future.

To make resilience an actionable concept it is important to relate resilience to other concepts that can be described and quantified. **Resilience attributes** are inherent characteristics of a resilient system (Flannery et al., 2018; Gasser et al., 2019; Mak & Clarkson, 2017), including:

- *robustness*: the ability of a system to maintain its functional performance around the expected level, in the presence of endogenous and exogenous disturbances;
- *adaptability*: denotes the ability of a system to recover following a disturbance event through an internal change agent;
- *flexibility*: denotes the ability of a system to change to accommodate influences and disturbances in the future, through interaction with an external change agent or decision maker.

Other resilience attributes discussed in the literature include functional redundancy, resourcefulness, and rapidity, but these can be regarded as sub-attributes of the three main ones discussed above.

Resilience capabilities characterise a systems' response to expected and unexpected events, and include:

- *preparation* and *absorption* denoting the system readiness to overcome perturbations without much effort, before and during an event. They can be attained through enhancing the system robustness, e.g. event mitigation, reconfiguration or re-design, control and operation strategies.
- *adaptation* relates to changes in management approach enabled by learning from previous experience of disruptions events (Connelly et al, 2017)
- *recovery* quantifies the duration to return, partially or fully, to its normal function following a disruptive event (Jackson & Ferris, 2013).
- *anticipation* to predict potential situations and future levels of control (Hollnagel, 2013).

All resilience capabilities can generate a close-loop from preparation to anticipation that continuously evolves, and there is a recursive relationship between those elements in the evolvement process.

Taking the viewpoint of CPS systems resilience, McDermott (2019) has proposed a taxonomy that revolves around *dependability*, which encapsulates a combined measure of reliability and trustworthiness. The reference dependability taxonomy defined by Avizienis et al (2004), relating to the overall ability of a system to avoid service failures, and incorporating the interrelated metrics of *availability*, *reliability*, *vulnerability*, *safety* and *maintainability*, underpins much of McDermott's CPS resilience framework. He has further considered *security* and *confidentiality* as core components of dependability, to reflect the CPS systems exposure to cyber attacks.

However, it is useful to reflect that the classic definition of dependability, see for example Ushakov (1994), emphasizes the capacity of the system to "perform its functions to within specified performance indices for the total duration of the mission"; this emphasizes the definition of the *mission* of the system and the *timescale* over which the dependability properties are expected in order to gain merited trust. This perspective is important in shaping the **operational context** for resilience; this includes: the *mission* with specific performance metrics and timescales defined by the human stakeholders; the perception of continuity of performance within specified levels that underpins *trust*; the *natural environment* and the *external events* that affect the operation of the system.

Figure 1 summarizes our framework for the definition of resilience of RAS. Like McDermott (2019) we rely on the NIST (2016) CPS definition and representation, underpinning the RAS resilience concept. Dependability provides objective metrics for characterising the resilience performance of a RAS, while resilience attributes and capabilities define the reliability behaviour of the RAS following a disturbance.

Considering the challenges with the design and development of resilient RAS, Bagchi et al (2020) distinguish between two primary aspects: (i) *resilience by design*, relating to the design effort to build in resistance to a large set of quantifiable perturbations; and (ii) *resilience by reaction*, which relates to the runtime reaction of the system that enables it to "bounce back" rapidly after a failure induced by a perturbation. Those aspects are intertwined, in the sense that RAS systems should be designed to incorporate resilience by reaction.

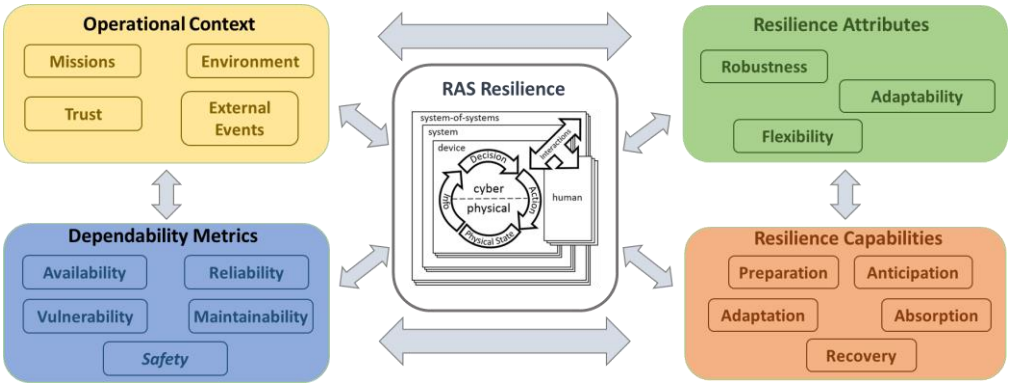

Figure 1. Summary of resilience concepts

2.2 Existing methods for Assurance Provision

RAS are often part of self-adaptive autonomous systems (SAS), where multiple heterogeneous systems collaborate to achieve common goals, such as a vehicle in a convoy. These systems dynamically reconfigure in response to changes such as unexpected failures of components/subsystems, the continuous change in the context of operation, variable workloads, and physical infrastructures. SASs need to provide assurance about their conformance with functional and non-functional requirements.

For traditional non-adaptive systems, assurance is provided through design and development activities including verification, validation, testing, conformance to standards and certification. Safety assurances are often provided through safety arguments where safety goals are defined and rationales for believing that these goals are met are dependent on a variety of assumptions. These assumptions may include aspects like failure semantics and failure rates of both hardware and software components, operating context, the efficiency of the human operator to respond to events, etc. (Knight et al, 2014). In operation the physical system and its operating environments are monitored to see if any of these safety assumptions are violated. However, for SAS design-time assurance are no longer sufficient since it is based on static assumptions about constantly evolving behaviour.

Figure 2 illustrates a high-level classification of the commonly used methods for design time and run time assurances, clustered around the primary methods used, such as goal-oriented approaches, formal methods or model checking However, it is common that an approach uses multiple methods (see Calinescu et al., 2018; Cheng et al. 2014 for a detailed discussion).

Goal-oriented methods use criteria for which assurances can be provided for CPS based on the stakeholders' expectation or regulatory standards. A high-level goal usually represents a system-level assurance claim, which is decomposed into lower-level goals (sub-claims) evidence can be gathered through monitoring, verification, testing, etc. Different formalisms or modelling language exist. For instance, RELAX (Whittle et al., 2009) allows a textual language-based specification of goal models. Similarly, FLAGS, a KAOS-based goal modelling framework was proposed by Baresi et al. (2010), which allows defining crisp goals using Linear Temporal Logic (LTL) and fuzzy goals using fuzzy logic. Goal Structuring Notation (GSN) (Kelly and Weaver, 2004) is another community standard widely used to specify assurance cases in the form of goal models.

Model checking is widely used for hardware and software systems to verify that certain properties hold for all reachable system states. Model checking approaches use formal logic such as Computational Tree Logic or LTL. For instance, Cámara et al. (2015) used probabilistic model checking to maintain trustworthy service delivery in the variable operating environment. Filieri et al. (2011) propose a runtime probabilistic model checking approach to detect harmful system reconfigurations. Kwiatkowska et al. (2009) used probabilistic model checking for reliability analysis.

Formal methods utilise graph-based formalisms to enable rule-based analysis and verification of system properties. For instance, Becker et al. (2006) used a graph transformation method for safety

assurance of multi-agent systems, where the correctness of the high-level abstract model of SAS was checked through invariant-checking and simulation techniques. To allow for the uncertainty of behaviour and operation environment various approaches have been suggested: combining design-time analysis techniques with runtime verification techniques (Rushby, 2008); safety contracts and safety cases (Calinescu et al., 2018; Schneider and Trapp, 2013); modular and composable safety guarantee models of Digital Dependability Identities (DDIs) (Schneider et al., 2015), which contains all the information required to uniquely define the dependability properties of the system or the component. The DDI of multiple components can be hierarchically combined to form the safety guarantee model of the whole system (Kabir and Papadopoulos, 2020).

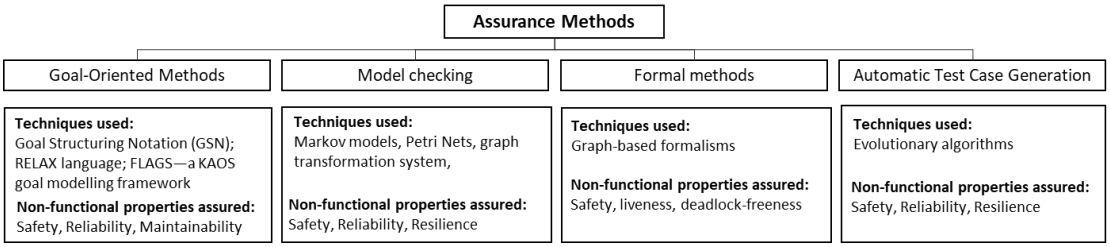

Figure 2. High-level classification of the assurance provision methods

Testing is used as a standard approach to verify that a system is behaving as expected in a known number of situations. However, it can be difficult to dynamically generate test cases applicable to the SAS evolving operational context and goals. To automatically generate test cases for the assurance of SAS, goal-based evolutionary optimisation (Nguyen et al, 2012) of the system behaviour under uncertainty can be employed (Fredericks et al., 2013); however, this cannot guarantee that test cases cover all possible scenarios. Fredericks et al. (2014) proposed an evolutionary approach for runtime adaptation to evolving operating conditions to ensure that test cases are relevant to the current operating conditions.

In summary, the definition of the RAS resilience of RAS including attributes, capabilities and objective metrics concerning dependability and the overview of existing assurance provision methods provides the necessary underpinning for the development of a comprehensive framework for resilience assurance through verification of multiple metrics while considering different perspectives of the system operation.

3 PROPOSED FRAMEWORK FOR RESILIENCE ASSURANCE OF ROBOTIC AUTONOMOUS SYSTEMS

Based on the review of relevant related work, we first provide a deeper discussion of key challenges and requirements for a system of resilience assurance, before introducing our proposed models.

3.1 Challenges and Requirements for a RAS model of resilience assurance

RAS resilience hinges around "*mission*" and "*duration*" to demonstrate dependability. For example, an autonomous vehicle is expected to provide assurance for specific events in the short term, like joining a platoon of vehicles, or passage over a level crossing (where safe mission achievement depends both on the assurance that the vehicle will not break down as well as other vehicles not breaking down). From a computational dependability modelling perspective, this could be addressed with the DDI framework described in section 2.2. However, for the medium and long term operation and use, providing resilience assurance is much more problematic. For example, a family, who acquires an autonomous vehicle to drive their children to school needs to be assured that the vehicle safely navigates the traffic, and that their child arrives at school reliably on time, but also that this arrangement will work for whole school year. This also emphasizes that for human stakeholders it is not only important that the system is resilient, but also that *they perceive the system to be resilient*. For example, the parents need to be assured that the vehicle can deliver the school commuting mission and that the child in the vehicle must not be upset by strange noises or seemingly dangerous manoeuvres.

To enable the user to trust that the RAS is "fit for purpose", a deeper understanding of the mission parameters and specifications is required, as resilience assurance needs to reflect on all priorities of the stakeholders, and not just the safety and security. Therefore, the design and development of RAS must be underpinned by a deep understanding of the mapping of missions to system functions and requirements, integrated with the architecture development based on model-based systems engineering

(MBSE). Taking a security attribute perspective, McDermott (2019) proposed a resilience analysis model and a design flow for resilience, which integrates systematic function decomposition modelling with device design and installation. This task is however much more complicated owing to the iterative development of complex systems such as vehicles, where the physical architecture is well established through decades of evolution, and the autonomous control features are attached to an existing structure. This results in a very complex and dynamic *within-system interactions*, very often hard to trace and leading to emergent behaviour from within the system.

Such issues have been so far less consistently considered by the RAS dependability research, which mainly focusses on the complexity of external interactions in a system-of-systems context. However, there is growing recognition in industry that the trade-off between safety and availability has become critical to the development of automated driving (SaFAD, 2019) and that global reliability models for the systems are a priority for research and development, as highlighted by the industry review of reliability research challenges (Campean et al, 2020). In particular, understanding and modelling the behaviour of complex heterogeneous systems including both physical and artificial intelligence AI / machine learning (ML) "black box" components was highlighted as a priority by industry stakeholders. There are two fundamental aspects to this:

1. **Integration:** The impact of the physical system reliability and robustness on the performance of AI/ML components and thus the validity of the assurance of the system overall is critical. At present AI/ML systems largely focus on uncertainty associated with external interactions and are less adept at dealing with endogenous uncertainty, e.g. the state of health of both the sensors (and thus credibility of sensor data) and physical system. Physical systems degrade over time, and degradation monitoring is often problematic given the dynamic nature of systems behaviour and the sensing limitations for lower value RAS (including autonomous vehicles). Systems health management (including diagnostics and prognostics) are not currently systematically integrated with the dependability of autonomous control features.
2. **Obsolescence**: some physical subsystems are expected to last a very long time, whereas sensors and software will have much shorter design life. While updates to the system (e.g. software updates - including to ML/AI black box components) provide enhanced functionality and refinement, this is often done by exploiting resources within the system, with the effect of creating linkages between systems hitherto unconnected. For example, advanced autonomous navigation features within a vehicle are underpinned by complex linkages between propulsion, braking, chassis and body systems, which in the physical systems domain have been developed and engineered by independent teams. In systems resilience terms, this has two potential consequences: (i) the impact of the actual behaviour of the physical systems on the robustness of the autonomous feature needs to be thoroughly evaluated (as discussed at point (1) above); and (ii) the introduction of new dependencies within the system, introduces new vulnerabilities, and potential impact on the adaptability and flexibility of the system to recover from a significant external disturbance. Therefore, open evolution of a complex system can be regarded as a pathway for degradation of a complex autonomous system (Campean et al, 2020), which needs to be both tested and monitored.

3.2 Framework for resilience assurance

Given the open evolving nature of the architecture of RASs, the separation between "design time" and "run time", underpinning conventional approaches to assurances and guarantees in safety, dependability and reliability, does not provide a realistic representation of the distribution of the design and development effort. Our proposed framework for resilience modelling and assurance by design includes a continuous iteration between the system analysis and design and the system operation phase, as illustrated in Figure 3. This iteration is necessary to enable learning from the operational phase to feed into upgrades and refinements for the run time as well as future design iterations. We have therefore indicated a distinction between activities impacting "resilience by design" and "resilience by reaction".

As discussed by McDermott (2019), resilience analysis of complex CPS / RAS must be underpinned by a rigorous systems engineering process. The systematic and rigorous model based resilience analysis (MBRA) approach must establish a consistent and traceable linkage across the layers of analysis including: (i) the user centric *mission objectives* and goals; (ii) the use-case or event specific *operational tasks* that are expected to be associated with the mission; (iii) the structural decomposition of the s*ystem functions*, including functional and non-functional requirements and specifications associated with operational tasks and mission goals / objectives; and (iv) the system *structure*, including both physical

and cyber/information assets, that provide the behaviours required to achieve the functional requirements and specifications. Identification of the linkages within the system can be challenging given the increasing multi-disciplinary complexity, but it is required in order to evaluate the resilience attributes of flexibility and adaptability. Evaluation of the robustness attributes is underpinned by the modelling of system behaviour in relation to the dynamics of the interaction exchanges with other systems within the system-of-systems context, also taking into account the internal state of the system to account for degradation and internal faults. On a model basis, this can be supported by methodologies and tools for function failure analysis, including fault propagation, e.g. FMEA, FTA, hazard analysis. Ultimately, this facilitates a robustness driven analysis of the minimum set of information inferred from sensors data to evaluate the current state and capability of the system, i.e. robust self- and environment-awareness, also providing the "minimum data set for reliability" (Campean et al, 2020).

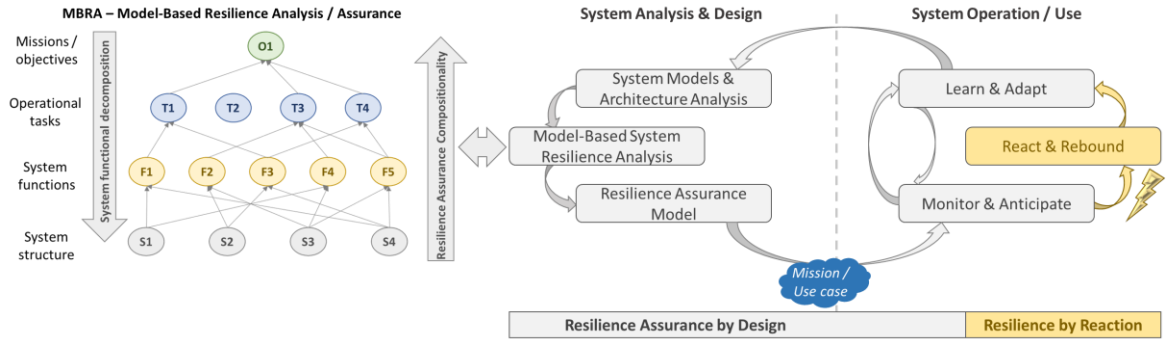

Figure 3. Framework for resilience of RAS

The system functional decomposition illustrated in Figure 3 (left side; this is a generalisation of the decomposition proposed by McDermott (2019)), also provides the compositionality structure required for a resilience assurance model. Thus, resilience assurance is underpinned by the robustness (or dependability) assessment of the system in relation to the set of functions F_i, which are required in order to assure the behaviour required for the set of operational tasks (or events) T_j, required by the mission O_k, given the assumed range of values of parameters describing the interactions with the external systems including the environment. This model of resilience assurance is bounded to the set of operational tasks or events defined in association with the missions (with defined performance indices and timescales expected by the stakeholders) for which the system has been design for.

As the system is commissioned to the operational phase, with a specific mission and the associated set of use cases, a set of *digital resilience assurance* (DRA) certificates will be issued by the system, based on the evaluation of the current internal state (including the expected evolution over the duration of the mission) and expected range of operational conditions (quantified by the range of interaction parameters values). During operation, the key internal and external parameters need to be monitored to identify the ongoing operational conditions, to update the resilience assurance in relation to the current conditions. This can be achieved with machine learning (ML) data driven diagnostics and prognostics, underpinned by the knowledge structures (e.g. knowledge graphs) available from the system MBRA model. The systems of dynamic DRAs would enable the system to anticipate future events and operational scenarios, and thereby notify the users about the potential changes in the assurance. It also enables the system to *learn* automatically by extracting knowledge from the ML analysis of the data streams, e.g. with ML supported dynamic knowledge graphs. This knowledge could lead to updates of the models underpinning the runtime control strategies to optimise the performance of the system, update the definition of the use case scenario parameters (either the definition of the range of values of the parameters, or identifying situations where the robustness of the use case models is unacceptable), as well identification of new linkages in the system (e.g. unintended linkages following a software upgrade), which might lead to undesired and unpredicted behavioural effects. Essentially, the ML-based knowledge extraction model monitors the integrity (i.e., completeness and correctness) of DRAs, and when shortcomings of the DRAs integrity are detected, alerts will be issued with a traceable set of diagnostics leading to the current state. This will prompt for an advanced warning to the supervisory control strategy system, which could react with an appropriate range of actions depending on the severity, likelihood and dynamics / timing of the negative effects potentially evolving from the current state. This could include immediate triggering of model-predictive control

routines for robust fail safe in the face of an impending disruption, or adaption of the control strategy to address weaknesses in the system to avoid the evolution towards a critical state.

3.3 Integrated RAS lifecycle model centred on resilience assurance

The concept of digital resilience assurances (DRAs) underpinned by the MBRA at the system analysis and design phase, coupled with dynamic monitoring of DRAs in system operation, affords the opportunity to conceptualise an integrated model for the design, development and operation of RAS centred on resilience assurance, illustrated by the proposed framework for in Figure 4.

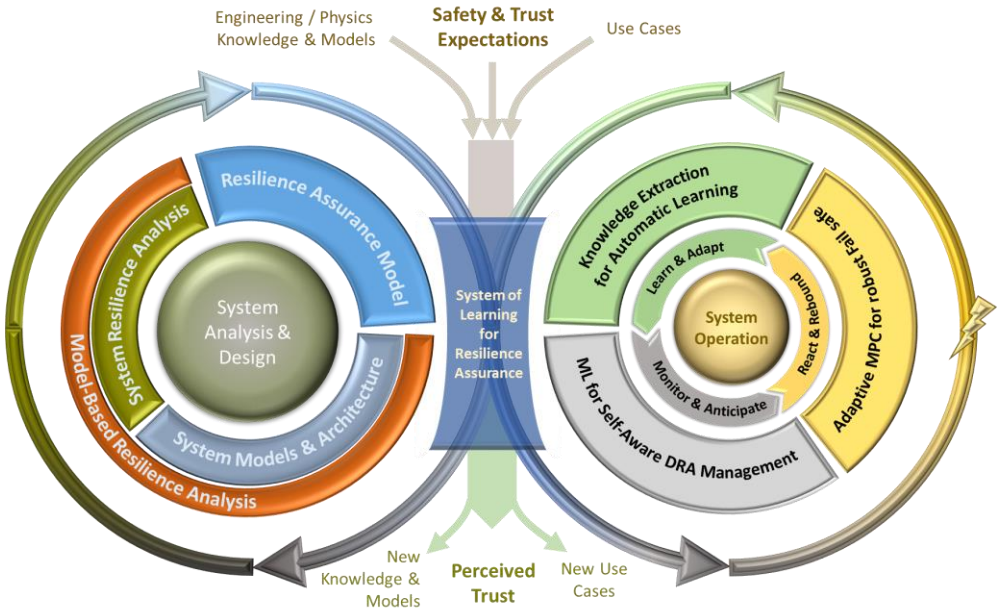

Figure 4. RAS lifecycle process model as a system of learning for resilience assurance

The model in Figure 4 reflects a continuous product development paradigm, which is necessary to reflect the open development nature of CPS in general and RAS in particular. This is consistent with the open evolving architectures of RAS prevalent in applications across the engineering fields, reflecting both evolving compositionality of the system, as well as the result of upgrades to the system (mostly software, but also hardware) throughout the life of the system. The *system analysis and design* phase (the left loop in Figure 4) consists of iterations of system architecture analysis, system resilience analysis and resilience assurance models, validated through resources available at design time (with a combination of physical systems tests, and model- / software- / hardware-in-the-loop, driven by use case data and models of the system and the environment, as well as manufacturing data of the system), and using the methods described in section 2.2. Once the system is released for use (either initial release or any system update), the *system operation* loop summarises the cycle of activities involved in the management of system centred on resilience. This starts with the issue of a system of DRAs to reflect the capability of the system in relation to the current mission and state of the system, communicated to the stakeholders. The dynamic management of DRAs needed to reflect the ensemble of stochastic uncertainties in the system operation is supported by data-driven and MBRA-enabled ML, with automatic knowledge extraction capability for online learning. The DRAs provide an effective formal mechanism for system monitoring, communicated in real time to users and stakeholders. The DRA governance provides an effective mechanism to anticipate disruption, facilitating the react and rebound control mechanisms. The traceable DRA diagnostics based on data-driven ML facilitate online learning, which supports system adaptation in real time, as well as data and models feedback for the system analysis and design loop - supporting future upgrades. The continuous iterations around and between the system design and operation loops, indicated by the external loop arrows in the model diagram in Figure 4, provide a comprehensive and integrated conceptual model for the RAS lifecycle. Taking the point of view of the vertical integration within the socio-technical context of RAS, this model can be perceived as a *system of learning for resilience assurance*. This vision is based on multiple dimensions, of which the principle one is the relationship between the users and other human stakeholders, which is one of mutual learning that ultimately revolves around the perception of *trust* in the RAS. The proposed DRA system provides a solution for communication about resilience to facilitate trust.

Information about new use cases and enhanced system knowledge and models, do not only inform development of future systems, but are also fundamental to enhancing trust for the RAS engineering and development stakeholders.

4 DISCUSSION, CONCLUSIONS AND FURTHER WORK

The main contribution of this paper is the introduction of a conceptual framework and model for resilience assurance of future ubiquitous robotic autonomous systems. The cornerstone of the proposed approach is the system of Digital Resilience Assurances (DRAs), which provide the backbone of the communication about resilience between the system and its stakeholders. From the point of view of their computational implementation, the DRAs are conceptually similar to the DDIs developed for dependability of autonomous systems. However, the broader focus on resilience capability of the system requires the DRAs to integrate both the current state of the system in light of the time-dependent degradation in capacity of the system (both physical and software) to react to mission and environment events, as well the characteristic of the threat or attack. Current research in CPS resilience tends to focus on the latter (see for example Moura & Hutchinson (2019) and Bagchi et al (2020)), with a primary concern for the software systems. As RAS become increasingly common for more diverse applications, the users' and stakeholders' trust in the resilience of the systems will depend on their experience of the system, and the way the system is able to fulfil the expected "fit for purpose" criteria. From a user perspective, the physical interactions are still likely to have a high bearing on the overall experience. We see that the integrated approach to both physical and cyber systems is essential, and the discussion in this paper emphasizes the need for interdisciplinary engagement around resilience of RAS.

Further research aims to develop and validate models and blueprints for DRAs, and the MBRA ecosystem of methods and tools, validated with case studies. The implementation and management of DRAs in conjunction with a data-driven ML ecosystem coupled with the MBRA for monitoring system robustness and efficient integration of automatic learning, also provides scope for further research. While we recognise that there is a wealth of research in ML based resilience and dependability, e.g. summarised in review papers such as Moura & Hutchinson (2019), for this paper we limited the scope to the development of the conceptual aspects of the resilience assurance model and DRAs.

The system of learning for resilience assurance provides an important model for the integration of RAS lifecycle view based on a continuous product development paradigm, with the broader socio-technic systems perspective of users and stakeholders, where the trust in resilience of the system is ultimately evidenced. This broader context reveals the further significance of the proposed DRAs to provide a general mechanism for both managing the resilience behaviour of the system in operation, as well as the communication about resilience within the socio-technical context of the system. Ultimately, we see the impact of the proposed framework in guiding the development of future resilient autonomous systems.

REFERENCES

Avizienis, A., Laprie, J., Randell, B., Landwehr, C., (2004). Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. Dependable Secure Comput. 1, 11–33. https://doi.org/10.1109/TDSC.2004.2

Bagchi S. et al. (2020). Vision Paper: Grand Challenges in Resilience: Autonomous System Resilience through Design and Runtime Measures, IEEE Open Jrnl Comp Soc, 1:155-172, https://doi.org/10.1109/OJCS.2020.3006807

Baresi, L., Pasquale, L., & Spoletini, P. (2010). Fuzzy goals for requirements-driven adaptation. In 18th IEEE International Requirements Engineering Conference, pp. 125-134, IEEE. https://doi.org/10.1109/re.2010.25

Becker, B., Beyer, D., Giese, H., Klein, F., & Schilling, D. (2006). Symbolic invariant verification for systems with dynamic structural adaptation. In Proc 28th Int Conf Soft Eng, https://doi.org/10.1145/1134285.1134297

Calinescu, R., Weyns, D., Gerasimou, S., Iftikhar, M. U., Habli, I., & Kelly, T. (2018). Engineering trustworthy self-adaptive software with dynamic assurance cases. IEEE Trans Soft Eng, 44(11), 1039-1069. https://doi.org/10.1109/tse.2017.2738640

Cámara, J., de Lemos, R., Laranjeiro, N., Ventura, R., & Vieira, M. (2015). Robustness-driven resilience evaluation of self-adaptive software systems. IEEE Trans Dep & Secure Computing, 14(1), 50-64. https://doi.org/10.1109/tdsc.2015.2429128

Campean, F, Delaux, D, Sharma, S., Bridges, J., (2020) Reliability Research Roadmapping Workshop: Implications for Engineering Design, Proc Des Soc, https://doi.org/10.1017/dsd.2020.337

Cheng, B.H., Eder, K.I., Gogolla, M., Grunske, L., Litoiu, M., Müller, H.A., Pelliccione, P., Perini, A., Qureshi, N.A., Rumpe, B. and Schneider, D. (2014). Using models at runtime to address assurance for self-adaptive systems. In Models@ run. time, pp. 101-136. https://doi.org/10.1007/978-3-319-08915-7_4

Connelly, E. B., Allen, C. R., Hatfield, K., Palma-Oliveira, J. M., Woods, D. D. and Linkov, I. (2017) Features of resilience, Environ. Syst. Decis., 37:1: 46–50, https://doi.org/10.1007/s10669-017-9634-9.
Fisher, M., Mascardi, V., Rozier, K.Y., Schlingloff, B.H., Winikoff, M. and Yorke-Smith, N., (2021). Towards a framework for certification of reliable autonomous systems. Autonomous Agents and Multi-Agent Systems, 35(1), pp.1-65. https://doi.org/10.1007/s10458-020-09487-2
Fredericks, E. M., Ramirez, A. J., & Cheng, B. H. (2013). Validating code-level behavior of dynamic adaptive systems in the face of uncertainty. In Int Symp Search Based Software Engg, pp. 81-95, Springer. https://doi.org/10.1007/978-3-642-39742-4_8
Fredericks, E. M., DeVries, B., & Cheng, B. H. (2014). Towards run-time adaptation of test cases for self-adaptive systems in the face of uncertainty. In Proc 9th Int SEAMS symposium, pp. 17-26. https://doi.org/10.1145/2593929.2593937
Filieri, A., Ghezzi, C., & Tamburrelli, G. (2011). Run-time efficient probabilistic model checking. In 33rd International Conference on Software Engineering (ICSE), pp. 341-350. https://doi.org/10.1145/1985793.1985840
Flannery, A., Pena, M.A., Manns, J. (2018). Resilience in Transportation Planning, Engineering, Management, Policy, and Administration. Transportation Research Board, DOI:10.17226/25166
Gasser, P., Lustenberger, P., Cinelli, M., Kim, W., Spada, M., Burgherr, P., Hirschberg, S., Stojadinovic, B., Sun, T.Y., (2019). A review on resilience assessment of energy systems. Sustain. Resilient Infrastruct. 0, 1–27. https://doi.org/10.1080/23789689.2019.1610600
Holling, C.S., (1973). Resilience and Stability of Ecological Systems. Annu. Rev. Ecol. Syst. 4, 1–23. https://doi.org/10.1146/annurev.es.04.110173.000245
Hollnagel, E., Woods, D.D., Leveson, N., (2006). Resilience Engineering: Concepts and Precepts. Ashgate Publishing, Ltd.
Hollnagel, P.E., (2013). Resilience Engineering in Practice: A Guidebook. Ashgate Publishing, Ltd.
Jackson, S., Ferris, T.L.J., (2013). Resilience principles for engineered systems. Syst. Eng. 16, 152–164. https://doi.org/10.1002/sys.21228
Kabir, S., & Papadopoulos, Y. (2020). Computational Intelligence for Safety Assurance of Cooperative Systems of Systems. Computer, 53(12), 24-34. https://doi.org/10.1109/mc.2020.3014604
Kelly, T., & Weaver, R. (2004). The goal structuring notation–a safety argument notation. In Proceedings of the dependable systems and networks workshop on assurance cases, p. 1-6. https://doi.org/10.1.1.66.5597
Knight, J., Rowanhill, J., & Xiang, J. (2014). A safety condition monitoring system. In International Conference on Computer Safety, Reliability, and Security, pp. 83-94. https://doi.org/10.1007/978-3-319-24249-1_8
Kwiatkowska, M., Norman, G., & Parker, D. (2009). PRISM: probabilistic model checking for performance and reliability analysis. ACM SIGMETRICS Perf Eval Review, 36(4), 40-45. https://doi.org/10.1145/1530873.1530882
Mak, W.H.J, Clarkson, P.J. (2017). Towards the Design of Resilient Large-Scale Engineering Systems, Procedia CIRP, 60: 536-541, https://doi.org/10.1016/j.procir.2017.01.034.
McDermott, T.A. (2019). A Rigorous System Engineering Process for Resilient Cyber-Physical Systems Design. In 2019 Int Sym on Systems Engineering (ISSE), pp. 1-8, IEEE. https://doi.org/10.1109/isse46696.2019.8984569
Moura, J., Hutchinson, D., (2019) Cyber-physical systems resilience: state of the art, research issues and future trends, Computer science, Arxiv. https://arxiv.org/abs/1908.05077v1
Nguyen, C. D., Miles, S., Perini, A., Tonella, P., Harman, M., & Luck, M. (2012). Evolutionary testing of autonomous software agents. Autonomous Agents and Multi-Agent Systems, 25(2), 260-283. https://doi.org/10.1007/s10458-011-9175-4
NIST (2016) National Institute for Standards and Technology (NIST) Framework for Cyber-Physical Systems Release 1.0: Cyber Physical Systems Public Working Group (Rep.). May 2016.
Rushby, J. (2008). Runtime certification. In International Workshop on Runtime Verification, pp. 21-35, Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-89247-2_2
Schneider, D., & Trapp, M. (2013). Conditional safety certification of open adaptive systems. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 8(2), 1-20. https://doi.org/10.1145/2491465.2491467
Schneider, D., Trapp, M., Papadopoulos, Y., Armengaud, E., Zeller, M., & Höfig, K. (2015). WAP: digital dependability identities. In 2015 IEEE 26th ISSRE Symposium, pp. 324-329. https://doi.org/10.1109/issre.2015.7381825
SaFAD (2019) Safety First for Automated Driving, white paper, retrieved from https://connectedautomateddriving.eu/mediaroom/framework-for-safe-automated-driving-systems/
United Nations, (2009). UNISDR Terminology and Disaster Risk Reduction.
Ushakov, I.A., (1994) Handbook of Reliability Engineering, Wiley & Sons. https://doi.org/10.1002/9780470172414
Whittle, J., Sawyer, P., Bencomo, N., Cheng, B. H., & Bruel, J. M. (2009). Relax: Incorporating uncertainty into the specification of self-adaptive systems. In 17th IEEE International Requirements Engineering Conference, pp. 79-88, IEEE. https://doi.org/10.1109/re.2009.36