Socio-Cyber-Physical Systems: Models, Opportunities, Open Challenges

Invited Paper

Radu Calinescu[1,2], Javier Cámara[1] and Colin Paterson[1,2]
[1]*Department of Computer Science, University of York, UK*
[2]*Assuring Autonomy International Programme, UK*
Email: {radu.calinescu,javier.camaramoreno,colin.paterson}@york.ac.uk

***Abstract*—Almost without exception, cyber-physical systems operate alongside, for the benefit of, and supported by humans. Unsurprisingly, disregarding their social aspects during development and operation renders these systems ineffective. In this paper, we explore approaches to modelling and reasoning about the human involvement in *socio-cyber-physical systems* (SCPS). To provide an unbiased perspective, we describe both the opportunities afforded by the presence of human agents, and the challenges associated with ensuring that their modelling is sufficiently accurate to support decision making during SCPS development and, if applicable, at run-time. Using SCPS examples from emergency management and assisted living, we illustrate how recent advances in stochastic modelling, analysis and synthesis can be used to exploit human observations about the impact of natural and man-made disasters, and to support the efficient provision of assistive care.**

I. Introduction

The cyber-physical systems used in smart cities, intelligent transportation, smart healthcare and other new application domains revolve around humans. As such, the development of these *socio-cyber-physical systems* (SCPS) must take human factors into account. Typically, this requires the modelling of the human involvement in the SCPS, to enable engineers to reason about the human and social aspects of the system under development. Our paper summarises key characteristics of human involvement in SCPS (Section II), describes existing and emerging paradigms for the modelling of this involvement (Section III), and discusses open challenges and opportunities for exploiting recent research to help address these challenges (Section IV).

II. Characteristics of human involvement in SCPS

More often than not, the modelling of human involvement in SCPS needs to consider the key characteristics from Fig. 1. First and foremost, it must consider the *role(s)* that humans will play in the SCPS under development. Three broadly defined (and non-exclusive) roles are typically possible.

First, humans may be *input providers* for SCPS. In this role, humans may provide information through standard computer-based interfaces (in which case the input will be *machine readable*). Alternatively, the information can be provided, directly or indirectly, through sensors that SCPS use to observe the humans it interacts with. This information is provided in a format that is typically straightforward to interpret and understand by other humans. Examples of *human-interpretable input* include voice commands or hand gestures aimed at the SCPS sensors (for directly-conveyed information), and involuntary signs of tiredness or distress (for indirectly-conveyed information). Different levels of uncertainty are associated with these types of input. Machine-readable input is likely to reflect the true intention of the humans providing it, within the confines of a typically predefined input format, and subject to humans not having mistyped the input, pressed the wrong button, etc. Direct human-interpretable input may more likely be "misunderstood" by SCPS, and indirect human-interpretable input is often associated with the highest level of uncertainty.

A second role for humans involved in SCPS is that of contributors to the functionality provided by the system. Two main categories of such contributions are *information processing*, where humans contribute to SCPS processes such as understanding (e.g., of data acquired by SCPS sensors), decision making and decision validation, and *actuation*, where they interact with the environment as required by the SCPS.

A third SCPS role for humans is that of *consumer of the service(s)* provided by the system. SCPS services such as the provision of lighting, air conditioning, heating and music to a smart building are *passively delivered*. These services do not require the SCPS to "gain the attention" of the service consumer. In contrast, services such as offering a glass of water to a patient being attended by a healthcare robot requires a physical interaction between the SCPS and the patient.

Another key characteristic that SCPS modelling should consider is the *responsibility* (level) of the humans involved in the system. Humans hired (or who formally volunteered) to support the operation of the SCPS—e.g., as information providers or system contributors—have typically signed a contract or agreement that makes them *accountable* for this support. Conversely, humans involved in the SCPS temporarily and/or anonymously are typically *unaccountable* for their interactions with the system. More than these two levels of responsibility may be appropriate for some SCPS.

SCPS modelling should also take into account the *expertise* (level) of the humans involved in the system. *Expert* humans may have received training for the role(s) they play within

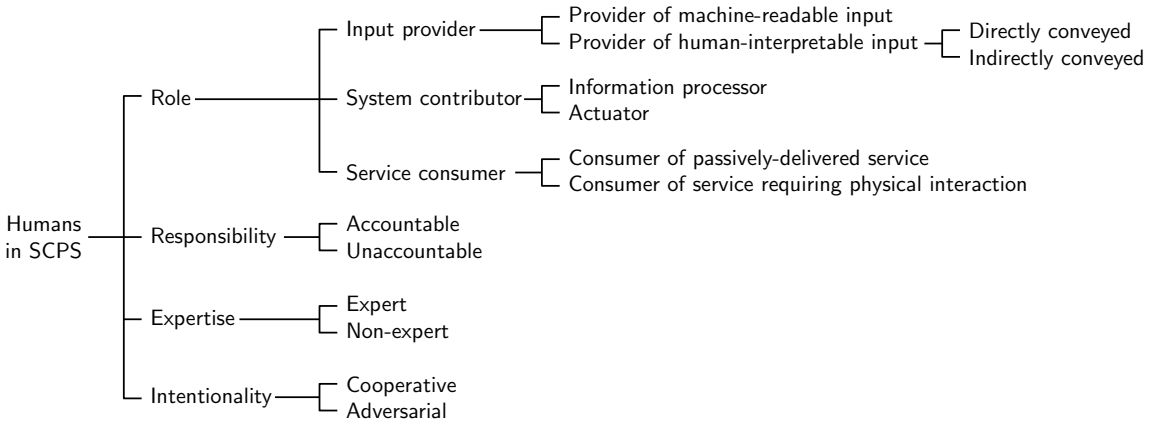

Fig. 1. Key characteristics of human involvement in SCPS

the SCPS, whereas *non-experts* have not, and therefore their interactions with the SCPS are likely to be characterised by significant uncertainty. As before, the modelling of certain SCPS is likely to be required more than two levels of expertise to be considered.

The last characteristic shown in Fig. 1 refers to the *intentionality* of the humans involved in SCPS. SCPS modelling should typically consider the presence of both *cooperative* and *adversarial* humans that interact with the system. A finer-grained characterisation of human intentionality might be required for some SCPS, e.g., by also considering a class of "neutral" humans involved in the SCPS under development.

Example 1 We illustrate the characteristics described in this section using an SCPS adapted from our recent work in [1]. This SCPS is a route-planning system for the emergency evacuation of areas affected by natural disasters such as earthquakes. The system continually re-plans the evacuation routes based on the most recent information about aftershocks, the state of the road infrastructure, the availability of water, food, medicine and fuel in different parts of the evacuated areas, etc. The SCPS obtains this information not only from sensors (used, for instance, to monitor the road infrastructure), but also from people in the area, who provide human-interpretable input as plain-text messages published on social media. Furthermore, the system relies on qualified personnel (and, in the future, on mobile robots) to help the evacuees who are in particularly critical situations, and to repair or replace failed SCPS components. The recommended routes are passively delivered to all other evacuees.

In this example, evacuees are service consumers (both of a passively-delivered service and of services requiring physical interaction). Additionally, some of them are input providers (of human-interpretable input). They are typically unaccountable, non-expert and cooperative SCPS participants. In contrast, the qualified personnel (comprising, for instance, first responders) are accountable, expert and cooperative system contributors.

III. Modelling human involvement in SCPS

The characteristics from Section II determine a broad range of interrelated human *attributes* that typically need to be explicitly captured by SCPS modelling. These attributes include:

- *Reliability*—Accountable, expert humans' involvement in SCPS is more reliable than that of unaccountable and/or non-expert humans.
- *Response time, throughput, latency*—These attributes are likely to vary greatly across SCPS tasks, across the people who perform these tasks, and even across different executions of the same task by the same person. Just how significant this variation is depends on all the characteristics from the previous section.
- *Robustness*—Expert humans may cope with change (e.g., in the information that needs processing) better than the "cyber" components of SCPS; they may even be able to handle "unknown unknowns". In contrast, non-experts may be completely unprepared for change, e.g., non-expert consumers may be unable to accommodate even small variations in how the SCPS provide their services.
- *Trustworthiness, reputation*—Cooperative, expert and accountable humans can normally be trusted to provide accurate input, to perform SCPS tasks as required, etc. Conversely, adversarial experts can inflict significant damage on the services provided by SCPS. Quantifying trustworthiness, often based on the reputation of the humans involved in SCPS, is very challenging.
- *Predictability, consistency*—Human involvement in SCPS is often associated with low levels of predictability and lack of consistency. These attributes are likely to be far better for those who are experts and/or accountable, although even these SCPS participants tire, can be temporarily distracted, etc.
- *Opportunity, willingness, capability (OWC)*—A modelling paradigm based on these high-level attributes is proposed in [2]. This OWC paradigm reduces the dimensionality of the attribute space by combining several finer-grain attributes into a single attribute (e.g., capability can subsume reliability, response time, latency, etc.). However, as we show in [3], [4], the components of the high-level OWC attributes often need to be disambiguated to enable modelling at the right level of detail.

All these attributes associated with the involvement of humans in SCPS are affected by uncertainty. As such, expressing them in the modelling of SCPS requires the quantification of this uncertainty with its probabilistic, time-related and potentially adversarial aspects (where possible) or capturing the nondeterminism induced by the presence of uncertainty (otherwise). This need calls for the use of formal models capable of such as queueing networks [5], Petri nets [6], stochastic models [7] and timed automata [8], [9], and of tools for their simulation (e.g. Palladio [10]) and verification (e.g. PRISM [11] and UPPAAL [8]).

Example 2 In recent work, we used Markov decision processes (MDPs) to model an assisted-living SCPS developed to help dementia sufferers with the daily task of hand-washing [12]. The SCPS provided voice prompts to the sufferers in certain MDP states, to guide them through what they must do next, if they were struggling to progress. These voice prompts became increasingly detailed when the sufferers repeatedly failed to progress, with human caregivers summoned when the prompts alone were insufficient. The SCPS aimed to achieve effective trade-offs between not overloading the sufferers with overly frequent prompts and not overloading the caregivers by summoning them too often. The MDP modelled the uncertainties associated with the sufferers' progress with the hand-washing task, and with what constituted a suitable pattern of prompts from the SCPS. This enabled the synthesis of Pareto-optimal MDP policies corresponding to effective SCPS configurations.

IV. Open challenges and opportunities

New opportunities for SCPS are firstly created by technological advances. Increasingly reliable and affordable components ranging from simple sensors to mobile robots with sophisticated functionality have greatly increased the range of potential applications for SCPS, as has the emergence of 5G and energy harvesting technologies. However, architecting these SCPS and engineering their control software so they can fully exploit such components is very difficult, in particular for SCPS used in safety-critical applications and/or which must adapt to changes in their environment or requirements. This section describes several of the main open challenges (OCs) from this area. For each challenge, we discuss recent research results that may provide opportunities for addressing it.

OC1) Formally specifying the requirements of SCPS—To reason about the ability of SCPS to meet their requirements, these requirements need to be expressed formally. Recently introduced formalisms for the specification and analysis of collective adaptive systems (e.g., [13]) and for capturing both the architectural and behavioural characteristics of software systems (e.g., [14], [15]) could help with addressing this challenge, although further work is needed to extend them with the ability to capture the social aspects of SCPS.

OC2) Ensuring the accuracy of stochastic models of SCPS—When stochastic analysis and verification are applied to inaccurate models, their results can be significantly skewed. This can subsequently lead to invalid engineering decisions, which may be confidently taken due to the false sense of confidence induced by the use of these formal techniques. Examples of common practices that may lead to this pitfall include: using single-point estimates for state transition probabilities; assuming exponentially distributed state transition rates; computing estimates based on too few observations of the modelled SCPS (components); and disregarding the fact that model parameters change over time. In other application domains, such problems have been mitigated by using online observations to update these model parameters [16], [17], [18], or using data from the testing of the system components to compute confidence intervals for the attributes of interest [19], [20] or to refine the stochastic models being analysed and verified [21], [22]. We envisage that extensions of these techniques could also improve the accuracy of SCPS stochastic models.

OC3) Leveraging human-interpretable input—Exploiting this type of input from humans involved in future SCPS is essential for the delivery of many of their envisioned services, but is also very challenging. Recent advances in machine learning, and in particular in deep learning [23], are expected to support this important SCPS task, but further research is needed on how to collect the right data sets, on how to train and verify trustworthy machine-learned models using these data sets, and on how to integrate the models into SCPS effectively.

OC4) Synthesis of SCPS—Typically, SCPS development requires the exploration of huge design spaces populated with alternative SCPS architectures and configurations, many of which are unfeasible or highly suboptimal. Techniques and tools are greatly needed to automate this process or parts of it. Ideally, these techniques and tools would start from a set of SCPS requirements, and would present developers with alternative systems designs that satisfy the requirements and are Pareto-optimal with respect to multiple optimisation criteria such as cost, utility and environmental impact. Such solutions have been proposed recently for the development of software systems (e.g., [24], [25]), and can also handle uncertainty in the operational profile of the system under development [26], [27]. However, these solutions cannot yet handle some of the new types of requirements encountered in SCPS.

OC5) SCPS self-adaptation—While it makes a lot of sense for SCPS to adapt their configurations (and even their architectures) dynamically to changes in their environment or requirements [28], [29], building self-adaptation capabilities within SCPS is extremely challenging due to the multiple concerns that these systems need to consider. It has taken the research community the best of two decades to explore and advance self-adaptive software systems, and, while this research is relevant to SCPS [30], it needs to be considerably extended before it can address the non-"cyber" aspects of these systems.

OC6) SCPS assurance—SCPS assurance requires the provision of comprehensive evidence and of an (assurance) argu-

ment explaining why safety-critical SCPS can be trusted to deliver their intended services in their specific environments. All steps of the SCPS assurance process are highly challenging: obtaining compelling evidence that all SCPS components (including humans and machine-learned models) can be trusted to their assumed levels of trustworthiness; combining this evidence to support the assembly of assurance arguments; and updating the evidence and the assurance argument as the SCPS evolves through offline maintenance or online self-adaptation. As for several of the previous open challenges, previous research on assurance [31], [32] (including assurance of self-adaptive software systems [33], [34]) may provide useful starting points for the assurance of SCPS.

Example 3 Consider again the route-planning and assisted-living SCPS from Examples 1 and 2. While probabilistic temporal logics were successfully used to specify requirements associated with the risks and duration of evacuation routes [1] and with the sequence of voice prompts provided to dementia sufferers [12], these logics cannot easily express requirements such as the interactions between evacuees who use the same route, or the distress experienced by sufferers who receive too many reminders or do not see their carers for long periods of time (open challenge OC1). Furthermore, the effectiveness of these SCPS depends on the accuracy with which events (e.g., damage to the road infrastructure) in the evacuated area and sufferer response to voice prompts, respectively, are mapped to state transition probabilities within the stochastic models that underpin decision making in these systems (OC2). Also essential for the two SCPS is that the relevant events and changes in the sufferer state are correctly detected and interpreted (OC3), and that updated evacuation routes and sequences of voice prompts are efficiently synthesised in line with requirements (OC4). In doing so, both SCPS may need to dynamically adapt their architecture and behaviour, e.g., by changing the deployment of the qualified personnel who collect information about the uncertain parts of the evacuated area, and by changing the style of the voice prompts to cope with the carer becoming temporarily unavailable, respectively (OC5). Last but not least, the two SCPS are safety critical, and therefore assurances are required to confirm not only the suitability of their synthesised policies (which correspond to evacuation routes and sequences of voice prompts, respectively), but also the robustness of the sensors collecting data for updating the models used for the policy synthesis, the correctness of the software used to implement these policies, etc. (OC6).

ACKNOWLEDGEMENT

This work was partly funded by the Assuring Autonomy International Programme.

REFERENCES

[1] C. Paterson, R. Calinescu, D. Wang, and S. Manandhar, "Using unstructured data to improve the continuous planning of critical processes involving humans," in *14th IEEE/ACM International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, 2019, to appear.

[2] D. Eskins and W. H. Sanders, "The multiple-asymmetric-utility system model: A framework for modeling cyber-human systems," in *8th International Conference on Quantitative Evaluation of SysTems (QEST)*, 2011, pp. 233–242.

[3] J. Cámara, G. A. Moreno, and D. Garlan, "Reasoning about human participation in self-adaptive systems," in *10th IEEE/ACM International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS*, 2015, pp. 146–156.

[4] J. Cámara, D. Garlan, G. Moreno, and B. Schmerl, "Evaluating trade-offs of human involvement in self-adaptive systems," in *Managing Trade-Offs in Adaptable Software Architectures*, I. Mistrik, N. Ali, R. Kazman, J. Grundy, and B. Schmerl, Eds. Boston: Morgan Kaufmann, 2017, pp. 155 – 180.

[5] S. Balsamo, V. D. N. Personè, and P. Inverardi, "A review on queueing network models with finite capacity queues for software architectures performance prediction," *Performance Evaluation*, vol. 51, no. 2, pp. 269–288, 2003.

[6] C. Lindemann, "Performance modelling with deterministic and stochastic Petri nets," *Performance Evaluation Review*, vol. 26, no. 2, p. 3, 1998.

[7] V. S. Sharma and K. S. Trivedi, "Quantifying software performance, reliability and security: An architecture-based approach," *Journal of Systems and Software*, vol. 80, no. 4, pp. 493 – 509, 2007.

[8] A. Hessel, K. G. Larsen, M. Mikucionis *et al.*, "Testing real-time systems using UPPAAL," in *Formal methods and testing*. Springer, 2008, pp. 77–117.

[9] K. G. Larsen, "Verification and performance analysis of embedded and cyber-physical systems using UPPAAL," in *MODELSWARD'14*, 2014, pp. IS–11–IS–11.

[10] S. Becker, H. Koziolek, and R. Reussner, "The Palladio component model for model-driven performance prediction," *J. Syst. & Softw.*, vol. 82, no. 1, 2009.

[11] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of Probabilistic Real-time Systems," in *CAV'11*, 2011, pp. 585–591.

[12] G. Mason, R. Calinescu, D. Kudenko, and A. Banks, "Assurance in reinforcement learning using quantitative verification," in *Advances in Hybridization of Intelligent Methods: Models, Systems and Applications*, ser. Smart Innovation, Systems and Technologies, I. Hatzilygeroudis and V. Palade, Eds. Springer International Publishing AG, 2018, vol. 85, pp. 71–96.

[13] L. Bortolussi, R. De Nicola, V. Galpin, S. Gilmore, J. Hillston, D. Latella, M. Loreti, and M. Massink, "CARMA: Collective adaptive resource-sharing Markovian agents," *arXiv preprint arXiv:1509.08560*, 2015.

[14] R. Calinescu, M. Autili, J. Cámara, A. Di Marco, S. Gerasimou, P. Inverardi, A. Perucci, N. Jansen, J.-P. Katoen, M. Kwiatkowska *et al.*, "Synthesis and verification of self-aware computing systems," in *Self-Aware Computing Systems*. Springer, 2017, pp. 337–373.

[15] J. Cámara, D. Garlan, and B. Schmerl, "Synthesizing tradeoff spaces with quantitative guarantees for families of software systems," *Journal of Systems and Software*, vol. 152, pp. 33 – 49, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0164121219300445

[16] R. Calinescu, Y. Rafiq, K. Johnson, and M. E. Bakir, "Adaptive model learning for continual verification of non-functional properties," in *ICPE'14*, 2014, pp. 87–98.

[17] R. Calinescu, K. Johnson, and Y. Rafiq, "Using observation ageing to improve Markovian model learning in QoS engineering," in *ICPE'11*, 2011, pp. 505–510.

[18] A. Filieri, L. Grunske, and A. Leva, "Lightweight adaptive filtering for efficient learning and updating of probabilistic models," in *Proceedings of the 37th International Conference on Software Engineering-Volume 1*. IEEE Press, 2015, pp. 200–211.

[19] R. Calinescu, C. Ghezzi, K. Johnson, M. Pezzè, Y. Rafiq, and G. Tamburrelli, "Formal verification with confidence intervals to establish quality of service properties of software systems," *IEEE Trans. Reliability*, vol. 65, no. 1, pp. 107–125, 2016.

[20] R. Calinescu, K. Johnson, and C. Paterson, "FACT: A probabilistic model checker for formal verification with confidence intervals," in *TACAS*, 2016, pp. 540–546.

[21] C. Paterson and R. Calinescu, "Accurate analysis of quality properties of software with observation-based Markov chain refinement," in *ICSA*, 2017.

[22] C. A. Paterson and R. Calinescu, "Observation-enhanced QoS analysis of component-based systems," *IEEE Transactions on Software Engineering*, vol. PP, pp. 1–1, 2018.

[23] L. Deng, "A tutorial survey of architectures, algorithms, and applications for deep learning," *APSIPA Transactions on Signal and Information Processing*, vol. 3, 2014.

[24] S. Gerasimou, G. Tamburrelli, and R. Calinescu, "Search-based synthesis of probabilistic models for quality-of-service software engineering," in *ASE*, 2015, pp. 319–330.

[25] S. Gerasimou, R. Calinescu, and G. Tamburrelli, "Synthesis of probabilistic models for quality-of-service software engineering," *Automated Software Engineering*, vol. 25, no. 4, pp. 785–831, 2018.

[26] R. Calinescu, M. Češka, S. Gerasimou, M. Kwiatkowska, and N. Paoletti, "Designing robust software systems through parametric Markov chain synthesis," in *2017 IEEE International Conference on Software Architecture (ICSA)*. IEEE, 2017, pp. 131–140.

[27] R. Calinescu, M. Ceska, S. Gerasimou, M. Kwiatkowska, and N. Paoletti, "Efficient synthesis of robust models for stochastic systems," *Journal of Systems and Software*, vol. 143, pp. 140–158, 2018.

[28] P. Jamshidi, J. Cámara, B. Schmerl, C. Kästner, and D. Garlan, "Machine learning meets quantitative planning: Enabling self-adaptation in autonomous robots," in *14th IEEE/ACM International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, 2019, to appear.

[29] J. Aldrich, D. Garlan, C. Kästner, C. Le Goues, A. Mohseni-Kabir, I. Ruchkin, S. Samuel, B. R. Schmerl, C. S. Timperley, M. Veloso, I. Voysey, J. Biswas, A. Guha, J. Holtz, J. Cámara, and P. Jamshidi, "Model-based adaptation for robotics software," *IEEE Software*, vol. 36, no. 2, pp. 83–90, 2019.

[30] A. Bennaceur, C. Ghezzi, K. Tei, T. Kehrer, D. Weyns, R. Calinescu, S. Dustdar, Z. Hu, S. Honiden, F. Ishikawav, Z. Jin, J. Kramer, M. Litoiu, M. L. G. Moreno, H. Muller, L. Nenzi, B. Nuseibeh, L. Pasquale, W. Reisig, H. Schmidt, C. Tsigkanos, and H. Zhao, "Modelling and analysing resilient cyber-physical systems," in *14th IEEE/ACM International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, 2019, to appear.

[31] R. Bloomfield and P. Bishop, "Safety and assurance cases: Past, present and possible future–an adelard perspective," in *Making Systems Safer*. Springer, 2010, pp. 51–67.

[32] R. Hawkins, I. Habli, T. Kelly, and J. McDermid, "Assurance cases and prescriptive software safety certification: A comparative study," *Safety science*, vol. 59, pp. 55–71, 2013.

[33] R. de Lemos, D. Garlan, C. Ghezzi, H. Giese, J. Andersson, M. Litoiu, B. Schmerl, D. Weyns, L. Baresi, N. Bencomo, Y. Brun, J. Camara, R. Calinescu, M. B. Cohen, A. Gorla, V. Grassi, L. Grunske, P. Inverardi, J.-M. Jezequel, S. Malek, R. Mirandola, M. Mori, H. A. Müller, R. Rouvoy, C. M. F. Rubira, E. Rutten, M. Shaw, G. Tamburrelli, G. Tamura, N. M. Villegas, T. Vogel, and F. Zambonelli, "Software engineering for self-adaptive systems: Research challenges in the provision of assurances," in *Software Engineering for Self-Adaptive Systems III. Assurances*, R. de Lemos, D. Garlan, C. Ghezzi, and H. Giese, Eds. Springer, 2017, pp. 3–30.

[34] R. Calinescu, D. Weyns, S. Gerasimou, M. U. Iftikhar, I. Habli, and T. Kelly, "Engineering trustworthy self-adaptive software with dynamic assurance cases," *IEEE Transactions on Software Engineering*, vol. 44, no. 11, pp. 1039–1069, 2018.