Making Sense of the World: Models for Reliable Sensor-Driven Systems

Muffy Calder, Simon Dobson, Michael Fisher, Julie McCann

Abstract—Sensor-driven systems are increasingly ubiquitous: they provide both data and information that can facilitate real-time decision-making and autonomous actuation, as well as enabling informed policy choices by service providers and regulators. But can we guarantee these system do what we expect, can their stake-holders ask deep questions and be confident of obtaining reliable answers? This is more than standard software engineering: uncertainty pervades not only sensors themselves, but the physical and digital environments in which these systems operate. While we cannot engineer this uncertainty away, through the use of models we can manage its impact in the design, development and deployment of sensor network software. Our contribution consists of two new concepts that improve the modelling process: 

frames of reference bringing together the different perspectives being modelled and their context; and the roles of different types of model in sensor-driven systems. In this position paper we develop these new concepts, illustrate their application to two example systems, and describe some of the new research challenges involved in modelling for assurance.

Index Terms—Sensor Networks, Reliability, Models, Formal Methods, Adaptive Software

I. INTRODUCTION

Sensor-driven systems are everywhere: from transportation and buildings, to smart tags, power systems and environmental monitoring. They provide data and information that facilitate real-time decision-making and autonomous actuation, as well as enabling informed policy choices by service providers and regulators. Societal challenges such as smart cities, Internet of things, big data and autonomous vehicles all depend on robust, sensor-driven systems that can be trusted to deliver useful, correct, and timely information. Our vision is smarter sensor-based systems of which stake-holders – from scientists to policy makers – can ask deeper questions while being confident of obtaining reliable answers. We are convinced that models have a central role in achieving this vision.

Given the ubiquity of sensor-rich systems, it is perhaps surprising that extracting reliable information from them remains far from straightforward: sensors are noisy, they decalibrate, may become misplaced, moved, compromised, and generally degraded over time, both individually and as a collective. This is beyond traditional software engineering: uncertainty pervades both the physical and digital environments in which these systems operate, and is even crucial to some of the adaptive algorithms employed. Furthermore, sensors themselves are increasingly multi-tenancy devices, required to fulfill both housekeeping functions (reporting to the sensor provider) and multiple sensing functions (reporting to several applications). The sensor system may also be required to support, and exhibit, increasing degrees of autonomy, self-awareness, and intelligence. Yet we understand very little about how to design, reason, and program in the face of such features and phenomena, most of which cannot simply be “engineered away”.

Our experience in the research and development of sensor-driven systems [1], [2], [3], [4], [5] has demonstrated that a critical factor impeding their wider deployment is a lack of assurance that can be provided to system users. Can we guarantee that the system will do what we expect, and therefore can we, in good conscience, make use of the information being presented, and so trust the behaviour being exhibited? Without such assurances, no business or organisation will deploy complex, semi-autonomous, adaptive sensor-driven behaviours; no decision-maker will allow their decisions to rest on a possibly unstable foundation.

It is our assertion that a crucial tool for addressing assurance is the use of models, both at design time and run-time: for specification; for explanation; and for exploration and prediction. Our contribution in this paper is develop two new concepts that contribute to a principled modelling process: frames of reference that
bring together the different perspectives being modelled together with their context; and the roles of a range of different models in assuring sensor-driven systems.

II. SENSOR-DRIVEN SYSTEMS AND MODELS

It is important to acknowledge from the outset that the uncertainty, autonomy, and complexity that are inherent in sensor-based systems will always be present: these aspects cannot simply be engineered away and subsequently ignored. Also the purpose, or mission, of the system may involve a number of trade-offs and compromises. A simple example involves controlling the duty cycle of a sensor node to preserve battery life, which might be constrained by a mission goal on minimal sensing frequency – or might even be disallowed entirely because of the need for extraction of a regular time series. Without a precisely-articulated mission, it is impossible to know how this trade-off can be made in a way that does not compromise the sensor deployment. Often such choices are implicit; formal models make them explicit, and thus amenable to analysis and verification. Modelling is at the core of enabling stake-holders to have the confidence they need to trust in – and make decisions from – the information being returned from a sensor system.

Some of the novel modelling requirements of sensor-driven systems include:

- the interpretation of both sensed data and context as domain-meaningful situations [6] and vice versa, to ensure that the sensor system is sensing the information needed to identify and differentiate between high-level situations;
- the ability to handle both stochastic and multi-scale phenomena occurring in real-world sensor system deployments;
- the ability to abstract naturally in order to capture the mission (or purpose) of a sensor system in terms of the problem domain rather than forcing designers to be “bogged down” in the sensor domain;
- compositionality, allowing the mission to be defined in terms of the entire network, as well as at the device level;
- the ability to adapt behaviour autonomously at both node and network level to maintain the mission in unforeseen contexts, while using sensed data to inform and calibrate modelled behaviours; and
- amenability to formal verification not only across design and programming, but also deployment and subsequent runtime-verification.

Models are fundamental to Science and Engineering, providing insight into the behaviours of what already exists, predicting future behaviours of what might exist, and exploring the implications of the specifications and designs of what we want to exist. Studying models of systems is not new. For example, there is much recent research in pervasive and cyber-physical systems that tackles modelling, implementation, and deployment of artificial systems integrated within real-world environments. We incorporate this work, for example we draw on pervasive systems, where raw data from devices is processed as context, from which situations are inferred, and control and adaptation then depends on those situations. And we draw on cyber-physical systems to the extent to which they concentrate on the modelling and interaction of software with physical scenarios, typically using hybrid systems analysis and novel models of programming, as advocated in [8]. Lee [9] also sees models as being central to the study of such cyber-physical systems, noting that “software abstractions were not created for cyber-physical systems. They were created to run payrolls”. We agree: do not yet have the best abstractions and concepts for sensor-driven systems.

A particular modelling challenge is to combine aspects of discrete and scheduled behaviours of software, which are typically described via (discrete space) models and logics, with the continuous nature of flow and physical control, often modelled by ordinary or partial differential equations, and the stochastic nature of sensors and sensing. A straight-forward mathematical approach can lead to very complex approaches, such as stochastic hybrid systems [7], which are expressive yet hard to utilise. To help tackle this challenge, we propose a clear articulation of frames of reference that determine the environment in which the systems operate, together with different types of model and their linkages.

III. FRAMES OF REFERENCE

Environments for sensor-based systems are inherently complex, and without further decomposition it is hard to see how the different requirements and perspectives – from both environment and stake-holder viewpoints – can be managed effectively. Frames of reference allow multiple perspectives and different levels of concern to be balanced within analysis. Some common frames are the following.

Security: vulnerabilities, threats and their mitigations, such as access controls preventing unauthorised entry to a system and encryption methods that encode data so it can only be accessed via keys. These vulnerabilities include those presented by multi-tenant architectures and the spatial factors of access, for example close-proximity, remote-proximity.
Privacy: anonymity, identity, authentication of personally identifiable information, and controls on intended and unintended disclosures.

Geographic: spatial and topological relationships between sensors. Typically these may be static networks (e.g., because the sensors are fixed to lampposts or in the ground), or dynamic networks (e.g., because they are fixed to mobile people, animals or objects), or on a fluid surface or within a flow in a pipeline. The relationships may or may not include explicit communication.

Failures: relationships between the components that can fail, degrade, or operate incorrectly, including fail-safe mechanisms and redundancies.

Economic: quantitative aspects of resource consumption, production and discovery, such as energy, money, or communication bandwidth.

Legal: obligations, permissions and responsibilities for different system components and human users.

Social: communication and interaction between humans involved in the system, and between humans, the physical/natural world, and the underlying technologies. Usability and cognitive dissonance are key concepts. For example, GPS drift can cause cognitive dissonance and lack of trust in a system when a user believes they are stationary yet their GPS coordinates are fluctuating.

Uncertainty: what are the acceptable bounds of uncertainty for various aspects of the system, and how are bounds qualified, quantified, and related to each other.

Temporal: expected certainty of the model over time, for example weather forecasting becomes less certain the further we look into the future, and navigation models become less precise as we move away from the position where we last verified our location.

For each of these frames of reference we require appropriate concepts, principles, and techniques for expressing and reasoning about: their key distinguishing aspects; the individual node and ensemble requirements that result; and the interplay between this frame and the other frames of reference. Crucially, frames of reference cover not only the form of the sensor network, but the purpose for which it is intended (its mission) and the context(s) within which it is deployed.

IV. MODELLING TECHNIQUES

To situate our discussion on the roles of models, we here provide a brief overview of some of the main modelling techniques for sensor-driven systems.

Deterministic models always produce one specific output from a particular set of inputs or initial conditions.

Non-deterministic models can deliver several possible outputs from a given set of inputs. If you run a non-deterministic model today, and then run it again tomorrow with the same inputs, you may obtain different answers. Determinism in models is often highly valued, because it allows one to make absolute assertions. However, many aspects of both the physical world and human behaviours are fundamentally non-deterministic, and it may not be useful to try to model them in a deterministic way.

Static models describe the current state but have no inherent concept of time or change.

Dynamic models have outputs that change over time. For example, simple differential equations represent deterministic, dynamic models, because they describe the rate of change over time of specified entities. More complex dynamic models rely on important concepts such as feedback (outputs of a system are fed back into the system as subsequent inputs); equilibrium, also known as steady state (after some time, the variables no longer change); stiffness (the model’s numerical methods need to make very small step changes to provide sufficient accuracy, which consequently take a long time to run through); and non-linearity (where changes in the output are not proportional to changes in the input, potentially leading to unpredictable and counterintuitive behaviour.) Many physical systems are non-linear, which makes forecasting difficult.

Discrete models represent objects or events by values that are represented in distinct steps – a series of Integers, for example.

Continuous models involve representations that are both smooth and dense, for example representations involving the Real numbers. It is possible to combine both discrete and continuous aspects into a single hybrid model [10], [11], and this is a common occurrence in cyber-physical systems. A model may have discrete states that occur in continuous time. For example: a door has two discrete states ‘open’ and ‘closed’ and continuous time transitions between them. These transitions might occur at different speeds: one can open a door quickly, yet close it more slowly.

Stochastic (alternatively called probabilistic or statistical) models have an inherent element of random, or uncertain, behaviour and the events represented are assigned probabilities. This can be viewed as a special case of a non-deterministic model in which the probabilities are known. A stochastic model often represents, or approximates, a data generating process.

Markovian models are specific forms of stochastic models in which the next state of a process depends only on
the current state, rather than on previous states. In other words, the probability of the next step does not depend on the history of activity up to that state. The simple model of a door above is Markovian because whether or not one can open the door (and the speed) depends only on the current state of the door (open or closed), not on how many times it has been opened or closed in the past. Common Markov models, called Markov chains, can have discrete time transitions between states (discrete-time Markov chain, DTMC) or continuous time transitions between states (continuous-time Markov chain, CTMC).

**Individuals** models represent each individual entity explicitly. An individual-based model is useful when one needs to track each individual entity through a system, or if individuals vary significantly in their behaviour.

**Population** models collectively represent large groups of individuals. For example, scientists might choose to model the behaviour of several individual fishes or instead model the activities of a large shoal as a single unit. However, individual models can become unwieldy if too many individuals are modelled, in which case a population model is likely to be more useful. A common population technique, applicable when individuals exhibit a finite number of possible traits, is to develop a counter-abstraction model that records the number of individuals with each trait [12], [13].

**Logic** can be used to model by defining a set of formal logical statements. For example, a simple propositional or predicate logic can be used to represent the fact that if a certain condition holds, then a variable has a certain value. More expressive temporal logics allows statements such as: if a certain event happens, then another event will surely follow it. There are numerous logics that allow for the expression of other concepts, and it is typically the case that any logical model used to represent a real and complex scenario will involve combinations of logical dimensions [14], [15]. Automated reasoning and analysis tools exist for most logics, including computer programs such as theorem provers and model checkers.

**Automata** and process algebraic models allow simple and elegant representations of multiple processes that communicate with each other. The underlying languages are typically algebraic, meaning there are laws that define how the different operators (a sequence or choice between events, for example) relate to each other.

**Agent-based** models typically contain a large set of autonomous agents that each represent individuals interacting with others based on their individual attributes and behaviours. They are often used for modelling complex, and possibly emergent, systems, especially when the behaviour of the whole system cannot be derived from a simple aggregation of the behaviour of the individuals, or if the system self-organises. Agents can have differing levels of autonomy; if it is important for an agent to have explicit reasons for making one choice over another, this can be done using agents that include representations of the agent’s beliefs, desires and intentions [16].

**Game theoretic** models are based on agents that (repeatedly) play a form of game with the goal to maximise their own benefit as defined by a utility function. A game is a formal interaction where each player chooses a move from a limited number of options and gains (or loses) utility as a result. We can model cooperative games (where players form coalitions), non-cooperative games (where individuals’ decisions and their time crucially affects the game outcome), or zero-sum games (where one player’s gain is the other player’s loss). Such models may be applicable to adversarial situations; for example, many security scenarios can be characterised as a game between the system and its attacker(s).

**Adaptive** models characterise situations where optimisation occurs through feedback interaction processes with the environment. The adaptive behaviour is therefore driven by the form of the environment, and so such models have become popular for matching system behaviour to environmental patterns. Adaptive approaches are widely used across Engineering, from feedback control systems to neural networks and genetic algorithms. One of the most appealing approaches is reinforcement learning. These are based on algorithms that, in effect, learn directly from past examples, data and experience. In this area, the terms model and algorithm are often conflated.

Finally, we highlight two approaches for utilising multiple models. **Ensemble modelling** involves running two or more related (but different) models, and then combining their results, either into a single result or comparing them. Modelling techniques are often combined to provide more powerful and domain-specific techniques. For example, mixing continuous, discrete and stochastic in order to provide stochastic hybrid systems. One drawback of some combinations is that analysis can be difficult and may be poorly supported by automated tools.

### V. Role of Models

We have found it useful to reflect upon the roles that each model plays in an ecosystem of models for sensor-based systems, and organise them in the following way.

First, we distinguish models that pertain to the environment that is being sensed, from models that pertain
to the components of the sensor based system. Second, we consider how stakeholder concerns, which refer to the frames of reference, influence the abstractions we choose in the models and the assurance properties we analyse.

Fig 1 illustrates our organisation. The two main boxes on the left, surrounded by dashed lines, are the models of the environment being sensed (upper box) and system (lower box). The downward arrow between the boxes indicates that the environment models inform the system models, because changes in the environment may affect the system. For example, temperature changes may degrade individual sensor readings, or a visiting lorry may block the communication between a set of sensors. The upward arrow indicates that the purpose of a sensor based system often involves recommending actions by users or other agents that will change the environment. For example, the purpose might be to direct people away from one room and into another, or direct vehicles on to another route, or identify a sensor that should be moved. The third (dashed) upwards arrow indicates the scenario where we deduce, from discrepancies between predicted and actual sensed values/situations, that the models needs modification – for example to correct errors or to make less abstract. In our experience, this has so far been relatively rare, but as sensor based systems become more accurate and reliable, we would expect this form of model modification to be more common.

Within the system (lower box) we indicate the expected data flows between the models. We note that on the left, we have a standard sense and control loop, whereas on the right we have more semantic capabilities introduced by pervasive computing and sensor networks. Computer programs themselves can be considered as models, but usually we require aspects of functional and performance behaviour to be modelled and analysed at a more abstract level, using the techniques described in the previous section.

The chosen frames of reference influence the choice of model abstraction and the relevant laws are encoded, as appropriate (e.g. concerning how failures propagate, how and where deontics are represented, etc.). In our experience we often consider combinations of frames, e.g. uncertainty and security, or legal and economic. The assurance properties wrap-up the mission of the system along with multiple frames of reference. Their analysis aims to answer the concerns and questions posed by stake-holders and assure them they can be confident in answers obtained from the system (or to indicate how or where control and reasoning needs to be modified control when required properties are shown not to hold).

The model roles are described in more detail as follows. The roles are not a rigid prescription and for some simple sensor based systems, some roles may be absent. We note that a range of modelling techniques may be employed for each role, for example, continuous environment models, discrete environment models, etc. Environment models are the laws and problem descrip-
tions of the environment, be it physical or digital; they come from natural science, engineering, regulations, and so on. For example, they might be analytical models such as differential equations and hybrid automata, or statistical distributions, simulations, and specifications of safe and unsafe thresholds. Over time, data and situations that have been inferred may be compared with those predicted by the environment models, possibly resulting in model modifications when the empirical evidence does not align with predictions.

**Device** models specify how devices interact directly with the environment, for example sensing, actuation and housekeeping. Raw data from the devices is processed as 'context', from which 'situations' are inferred; control and reasoning (and possibly adaptation) depends on those situations. Device models may also include representations of the impact of the physical environment on the devices (e.g. degradation).

**Context** models specify how raw signals from devices are processed into reusable computer data. This is typically a syntactic transformation that may, for example, include data smoothing, outlier rejection, and the extraction and maintenance of confidence intervals.

**Network** models specify how and when devices interact with each other, and with the application (control). These are typically communications protocols, which may depend on properties of the underlying sensor topology, e.g. star topology, peer to peer, or in a fluid medium, and properties such as requirements for consistency between adjacent sensors. Sensor topologies may be modelled by metric or topological spaces. Models may also include representation of the impact of the physical environment on sets of devices and/or communications links. System data refers to operational data such as network status.

**Situation** models specify how data is interpreted semantically, typically by an inference process implemented in software. For example, data might be classified as known or a previously undetected event, or for a given time series over a set of contexts, the situation model may infer semantic quantities such as “getting hotter” or “approaching a dangerous level”.

**Control** models specify how the software controls individual devices and the network, for example, this may include powering down individual devices for periods of time, or changing communication rates. Control typically depends on a mixture of strategies determined by the reasoning component and on cleaned data.

**Reasoning** models specify how the software achieves its mission and purpose – including node-level, network-level and application-level adaptations – through strategies that may be tailored to the behavioural envelopes for individual nodes and ensembles. Behavioural envelopes can change, usually triggered by situation changes, and adaptation means that devices may be updated, enhanced or reconfigured. There is usually a close relationship between control and reasoning, hence we have indicated a data flow in both directions.

**VI. Examples**

In this section we illustrate the role of models using two short examples. The first concerns a smart water distribution network that monitors and controls pumping, valves, and communication. In water distribution systems, pumping treated water reservoirs to supply zones and storage tanks consumes most of the energy budget for a utility. Use of tanks and reservoirs, and shifting pumping schedules to cheap tariff periods, can result in savings. The mission of a smart water distribution network is to minimise pumping costs, pipe degradation and leakage, and satisfy customer demands over water pressure and quality. Treated water must be kept moving at sufficient hydraulic pressure (to maintain the treatment). Typical stake-holder questions include: what is the lowest pressure that can meet demand and keep water clean, what is the highest pressure that minimises pipe damage, what is the minimal data rate that meets legal requirements for reporting leaks? Fig. 2 illustrates some of the frames of reference and types of models we are developing for a smart water distribution network, based on our experience in [1]. The models listed are not exhaustive, for example, further control models would include maintenance of system optimisation.

The second example concerns a domestic activity monitoring system that monitors activities in the home and notifies participants, carers and clients, via text and speech messages and a variety of sound and vibration alerts. Sensors and output messaging devices are attached to coffee cups, toilet floats, bracelets etc. The activity monitor has multiple hubs, in client homes and carer offices, configured for a finite number of monitoring tasks. These are defined by sets of rules, which are configured by clients and carers. An example task is: when a hot water tap is used in the home, deliver a speech message to the carer hub. Messages can be delivered contemporaneously (e.g. generated by multiple tasks) and so usability depends on compatibilities between interaction modalities. For example, contemporaneous deployment of speech and a ring tone could be confusing to a client and so they would be considered incompatible. Typical stake-holder questions include: Are the monitoring rules consistent? Can we guarantee separation of housekeeping and application function for each sensor?
Is it possible for the system to deliver, simultaneously, messages to an individual client or carer that have incompatible interaction modalities? Can the system detect abnormal activities? Is client privacy ensured? Fig. 3 illustrates some of the frames of reference and models we developed for a domestic activity monitor [4].

VII. NEXT STEPS

We recognise that models are just one, albeit important, component of assurance science of sensor-driven systems; modelling needs to be coupled with analysis techniques and with software development, testing and analysis of actual deployments. While techniques for context and situation recognition and inference have, and are, being researched extensively, especially in the context of big data, there are numerous new research challenges for modelling. For example:

- How can historical data be used to calibrate models for the analysis of system play-back, and predictions of future system behaviour?
- How can we use modelled predictions as inputs to control models, and use sensed contexts and situations to calibrate modelled predictions? For example, can we use model predictions (for example, of temperature or water level) to modify system goals, drive sensor change, and detect data anomalies; how should we modify models in the light of actual sensed data?
- What are the differences between the models we need to express that each node behaves as intended and that the overall system behaves as intended, and how should they be related; how can we extract models of system mission from different problem domains and frames of reference? How can we capture the variability and multiple dimensions of the underlying data, as well as various combinations of agents, temporal, spatial, performance, deontic, and probabilistic aspects? For example, what are the requirements for pipeline topologies, as opposed to sensors on fluid surfaces or stationary sensors?
- How can we model and reason about system continuity? How should we model (sub)populations of sensors that trigger discontinuities? What theories will account for heterogeneous and hierarchical sets of sensors and safe envelopes, and exceptions because of internal or external choice, with varying degrees of uncertainty?
- How can situations and situation change be inferred within acceptable bounds of uncertainty and in the presence of dynamic sub-populations of sensors?
DRAFT OF AUGUST 23, 2018

Fig. 3: Example model ecosystem for a domestic activity monitoring system.

VIII. RELATED WORK

Our intention is to improve the modelling process by providing a framework in which we can deal with the competing demands of understanding the sensed world, the computation and communication within the sensor based system, and the ways in which the devices interact with sensed world. We are unaware of ecosystem concepts similar to ours, it appears that our concepts add new insights into existing work. For example, the Ptides and Ptolomy models described in [8] and [9] refer to interplays between physical models (i.e. environment), sensors, actuators (i.e. devices), computation (i.e. control and reasoning), and network communication. The failures frame of reference is predominant, with an emphasis on timing and scheduling. Additionally, in [8] a modal, hybrid model in Ptolomy II is proposed for improved adaptation, we would classify this as a reasoning model. It would be interesting to see how our model roles would help with future modifications to these systems and/or how the models could be extended to incorporate assurance over other frames of reference.

IX. CONCLUSIONS

Sensor-driven systems will become increasingly significant over the coming decades, supporting evidence-based decision-making in the face of global challenges such as environmental change, food production, autonomous vehicles, and human ageing. But progress will be undermined by our inability to assure both decision-makers and users of the integrity and timeliness of the information being provided and the decisions taken. Without such assurance, increasingly “smart” infrastructures will become unpredictable and unacceptable, and the ability of state, scientific, and industrial actors to leverage the benefits of sensing and autonomy will be severely restricted.

We have introduced two concepts: frames of references and roles of models, that improve the modelling process, and we provided an informal survey of modelling techniques and their respective suitabilities. We outlined how these concepts may be employed through two example systems that we are engaged with; demonstration of the full benefits is the focus of further technical papers.

In conclusion, our long term vision is of smarter sensor-based systems, of which stake-holders – from scientists to policy makers – can ask deeper questions and be confident of obtaining reliable answers; models have an important part to play in achieving this vision. Further case studies, based on increasingly complex end-user applications, should include sensor-driven systems that fail, as well as succeed. A good example of a failed application is the remarkably honest assessment of a 150
wireless sensor node application for precision agriculture reported in [18]. A culture of reporting failures, and analyses thereof (e.g. the recent FAILSAFE workshop [19]) will only strengthen the case for improved modelling and for sensor-driven systems.

Acknowledgments

This work is supported by EPSRC Science of Sensor System Software programme grant (EP/N007565).

REFERENCES

[1] S. Kartakis, E. Abraham, and J. A. McCann, “Waterbox: A testbed for monitoring and controlling smart water networks,” in Proceedings of the 1st ACM International Workshop on Cyber-Physical Systems for Smart Water Networks. ACM, 2015, p. 8.

[2] A. R. M. Kamal, C. J. Bleakley, and S. Dobson, “Failure Detection in Wireless Sensor Networks: A Sequence-based Dynamic Approach,” ACM Transactions on Sensor Networks, vol. 10, no. 2, pp. 35:1–35:29, Jan. 2014.

[3] M. Fisher, L. A. Dennis, and M. P. Webster, “Verifying Autonomous Systems,” Comm. ACM, vol. 56, no. 9, pp. 84–93, 2013. [Online]. Available: http://doi.acm.org/10.1145/2494558

[4] M. Calder, P. Gray, and C. Unsworth, “Is my configuration any good: checking usability in a sensor-based activity monitor,” Innovations in Systems and Software Engineering, vol. 11, pp. 131–142, 2015.

[5] S. Benford, M. Calder, T. Rodden, and M. Sevegnani, “On lions, impala, and bigraphs: modelling interaction in physical/virtual spaces,” ACM Transactions on Human-Computer Interaction (TOCHI), vol. 23, 2016.

[6] J. Coutaz, J. Crowley, S. Dobson, and D. Garlan, “Context is key,” Comm. ACM, vol. 48, no. 3, pp. 49–53, 2005. [Online]. Available: doi://10.1145/1047671.1047703

[7] M. Bujorianu and J. Lygeros, “Towards a General Theory of Stochastic Hybrid Systems,” Lecture Notes in Control and Information sciences, vol. 337, pp. 3–30, 2006.

[8] P. Derler, E. A. Lee, and A. L. Sangiovanni-Vincentelli, “Modeling Cyber-Physical Systems,” Proceedings of the IEEE, vol. 100, no. 1, pp. 13–28, 2012.

[9] E. A. Lee, “The Past, Present and Future of Cyber-Physical Systems: A Focus on Models,” Sensors, vol. 15, no. 3, p. 4837, 2015.

[10] A. Tiwari, “Abstractions for Hybrid Systems,” Formal Methods in Systems Design, vol. 32, pp. 57–83, 2008.

[11] P. Tabuada, Verification and Control of Hybrid Systems: A Symbolic Approach. Springer, 2009.

[12] S. M. German and A. P. Sistla, “Reasoning about Systems with Many Processes,” J. ACM, vol. 39, no. 3, pp. 675–735, 1992.

[13] G. Delzanno, “Automatic Verification of Parameterized Cache Coherence Protocols,” in Proc. 12th International Conference on Computer Aided Verification (CAV), ser. Lecture Notes in Computer Science, vol. 1855. Springer, 2000, pp. 53–68.

[14] A. Kurucz, “Combining Modal Logics,” in Handbook of Modal Logic, ser. Studies in Logic and Practical Reasoning, J. van Benthem, P. Blackburn, and F. Wolter, Eds. Elsevier, 2007, vol. 3, pp. 869–924.

[15] S. Konur, M. Fisher, and S. Schewe, “Combined Model Checking for Temporal, Probabilistic, and Real-time Logics,” Theoretical Computer Science, vol. 503, pp. 61–88, 2013.

[16] A. S. Rao and M. Georgeff, “BDI Agents: From Theory to Practice,” in Proc. 1st Int. Conf. Multi-Agent Systems (ICMAS), San Francisco, USA, Jun. 1995, pp. 312–319.

[17] J. Ye, S. Dobson, and S. McKeever, “Situation identification techniques in pervasive computing: a review,” Pervasive and Mobile Computing, vol. 8, no. 1, pp. 36–66, 2012. [Online]. Available: doi://10.1016/j.pmcj.2011.01.004

[18] K. Langendoen, A. Baggio, and O. Visser, “Murphy Loves Potatoes: Experiences from a Pilot Sensor Network Deployment in Precision Agriculture,” in Proc. 20th International Parallel and Distributed Processing Symposium (IPDPS), 2006.

[19] “First ACM international workshop on the engineering of reliable, robust, and secure embedded wireless sensing systems (FAILSAFE).” http://wp.doc.ic.ac.uk/failsafe/, 2017.