Modeling for Sustainability

Benoit Combemale[*], Betty H.C. Cheng[†], Ana Moreira[‡], Jean-Michel Bruel[§] and Jeff Gray[¶]
[*]INRIA and University of Rennes 1, France
[†]Michigan State University, United States
[‡]NOVA LINCS, Universidade Nova de Lisboa, Portugal
[§]University of Toulouse, France
[¶]University of Alabama, United States

***Abstract*—Various disciplines use models for different purposes. An engineering model, including a software engineering model, is often developed to guide the construction of a non-existent system. A scientific model is created to better understand a natural phenomenon (i.e., an already existing system). An engineering model may incorporate scientific models to build a system. Sustainability is an area that requires both types of models. Both engineering and scientific models have been used to support sustainability, but largely independently of one another. Due to the inherent complex nature of sustainability that must delicately balance trade-offs between social, environmental, and economic concerns, modeling challenges abound for both the scientific and engineering disciplines. This paper offers a vision that promotes a unique approach that combines engineering and scientific models to enable broader engagement of society for addressing sustainability concerns. We introduce a research roadmap to support this vision that emphasizes the socio-technical benefits of modeling.**

I. Introduction

Computing-based technology has contributed significant capabilities and tools needed to address sustainability challenges [1]. Examples include computational modeling, large-scale data analysis, and sensor technology. In general, sustainable development should meet the needs of the present without compromising the viability of the needs of the future generations [2]. We use the term *sustainability systems* to refer to those computing-based systems that are used to support sustainable development, such as smart grids, smart homes and buildings, and other resource production and management systems. Sustainability systems differ from other types of systems in that their functionality must balance the trade-offs between the three pillars of sustainability: social, economic, and environmental [2]. In addition, as sustainability issues gain increasing attention, so will the demand for access to sustainability systems by a broader population of users. While progress has been made by leveraging modeling technology to manage the complexity of sustainability systems [1], [3], [4], numerous challenges remain as the problem complexity and scope increase, stakeholders and their needs change, and technological advances offer new options to exploit. This vision paper proposes a new approach to combine disparate sustainability models that will enable broader engagement of society, while supporting the development and the run-time management of sustainability systems.

A long-standing problem has been the inaccessibility of models and the associated data for complex systems to non-experts. This challenge emerges regardless of the model subject. *Scientific models* are used to describe existing systems or phenomena (e.g., biological systems, chemical composition, weather patterns). *Engineering models* are often used to describe systems to be built (e.g., architecture models for buildings, wiring diagrams for electrical systems, UML diagrams for a web server). Sustainability issues have been described primarily by scientific models that enable scientists to understand the impact of changes in one or more of the three pillars of sustainability. Engineering models have been used by (software) developers for the construction of computing-based systems to support various aspects of sustainability systems, such as ecosystem monitoring, power grid management, and climate-control in smart buildings. In contrast, sustainability is a global problem that must be addressed at all levels, from individual decisions to world-wide policies. It is important that different types of stakeholders (e.g., individuals, community leaders, policy makers, industrial organizations) with varying technology proficiencies be able to make well-informed decisions based on exercising the scientific models when using sustainability systems.

This vision paper describes two key insights into how modeling can be used to support sustainability, enable broader engagement of the community, and facilitate more informed decision-making. First, we observe that many of the foundational concepts used for Model-Driven Engineering (MDE) need to be reconsidered when developing sustainability systems. Instead of considering sustainability as yet another application domain, we need to analyze carefully the global nature of such systems to infer the dual and complementary needs of engineering and scientific models. Second, we propose two modeling feedback loops, one for the engineering model and another for the scientific model, which work together symbiotically to support the development and the run-time management of sustainability systems.

This paper offers a modeling vision for sustainability systems that requires a broader use of modeling techniques. It also provides scientists and engineers an epistemic study of the use of models in science and engineering when considered in the context of sustainability. The dual feedback loops leverage the modeling techniques from both disciplines to achieve a more holistic MDE-based approach to supporting sustainability systems. The engineering model feedback loop comprises the models used to develop and manage the software infrastructure for a cyber-physical system for sustainability (e.g., smart grid management). The scientific modeling loop comprises a multi-view scientific modeling infrastructure for capturing the three pillars of sustainability (i.e., social, economic, and environmental) that uses an *Aggregator* to integrate multiple scientific models and incorporate information from a sustainability system and its context to enable a stakeholder to select specific "views" of sustainability to explore (e.g., pose "what if" scenarios across multiple dimensions with "zoom-in" and "zoom-out" capabilities for fine-grained or global-level views, respectively). The results of the science-based inquiries can either be used to predict the social, economic, or environmental impact of a behavior change in resource usage, or sent back through the engineering modeling feedback loop to adapt the sustainability system.

We use existing and emerging sustainability problems and systems to highlight the research challenges that emerge from our proposed vision. The remainder of this paper is organized as follows. Section II discusses the global nature of sustainability and the key modeling problems to be addressed. We characterize the dual and complementary roles of engineering and scientific models, for which the current practices are overviewed in Section III and Section IV, respectively. In Section V, we introduce our vision of how the two modeling feedback loops are intertwined to support the development and run-time management of sustainability systems, and we identify several research challenges posed by this new vision. Finally, Section VI concludes the paper with insights on the research roadmap associated with our vision.

II. Problem Statement

Sustainability has increasingly become an urgent issue over the past several decades [2]. At the same time, the pervasiveness of the Internet in our daily lives and the increasing use of global software-based systems that handle large, complex, networked, and heterogeneous systems that involve a wide range of users and hardware devices, opens new frontiers for innovative designs and solutions. The technological advances offer new user experiences and access to information that collectively have the potential to influence behavior change, including making an impact on sustainability. The next generation of development approaches should support multiple dimensions of sustainability, ranging from long-lasting dependable and dynamically adaptive software, to green software requiring less computing and fewer energy resources, to software that encourages sustainable human behavior (e.g., smart plugs and appliances, Tesla Eco-Batteries to support the Roof PV generation at Home [5], and market design and regulations that transition consumers towards more energy-saving practices; the so-called "power of the negawatt" [6]). Approaches supporting various dimensions of sustainability are likely to have a substantial positive impact on people, the economy, and the environment for the short and long term.

Traditional software abstractions and development techniques are inadequate to tackle this task in several important ways. These techniques are (computing) system focused, where abstractions are intended to manage the system complexity, provide different views of the system, and non-functional qualities refer to properties about the system, such as performance, memory usage, and reliability. In contrast, sustainability systems must also consider how that functionality has to be delivered in changing contexts as defined by the environment, economic circumstances, and social policies. For example, when control systems are used to support sustainability, it is insufficient to only consider the management of an individual element, such as a single transformer in a smart grid system. Instead, the context in which the transformer is used must be considered holistically, including the power sources (e.g., windmills, water power, solar), changing environmental conditions, the volume and types of power consumers, usage patterns, and economic factors.

Modeling has been the essential mechanism to cope with the complexity of reality. In science, models are used to describe existing phenomena of the real world [7], [8]. In engineering, models are typically used to describe a system to be developed in the future. Thus, engineering models are typically *constructive* while scientific models are *descriptive*. An engineering model may incorporate scientific models to build the systems.

New modeling approaches are needed to meet the challenges posed by the change in the nature of sustainable, global software systems that are different from the traditional MDE approaches. Existing mechanisms need to be leveraged to facilitate modeling and analysis of the multiple dimensions of sustainability system requirements and architectures. In particular, we need to explore the challenges of modeling for sustainability; explore modeling one or more dimensions of sustainability; develop innovative decision support techniques, including what kind of data processing and visualizations are needed to enable different types of stakeholders with complementary interests to understand sustainability concerns and impacts of individual decisions [9]; propose approaches that cope with the increasing demand of precision, trustability, reliability, scalability, adaptation and context-awareness, and timely acquisition and/or computation of information for decision-making; investigate cognitive models, methods and tools, that can be used by a broad range of stakeholders and users, not just engineers. Therefore, a sustainability (eco)system has to be modeled in the context in which it operates, with respect to its environment, social policies, and economic conditions.

In the following, we examine how MDE is currently applied in engineering disciplines in the context of sustainability, discuss its usefulness in science and show how the differences in model use between the two fields is decreasing given current trends. Our proposed vision is to facilitate their symbiotic use to support sustainability systems.

III. MDE for Constructive Engineering Models

Engineers, including software engineers, use models to create abstractions of the complex systems under development [10], [11]. Such systems (e.g., engine, software, building) do not usually exist at the time the model is built, because often an engineer's first goal is to build the phenomenon from the model. We identified several foundational concepts used by engineering modeling approaches to handle size and tame complexity. One of the first concepts used is decomposition or "divide and conquer," where developers focus on the smaller pieces of a problem and their corresponding solutions, and then use composition techniques to aggregate the solution pieces, through well-defined interfaces, into the overall system [10]. Decomposition alone may not always work for complex systems, but abstraction helps by omitting details that are not relevant to the task at hand. Another foundational concept is separation of concerns [10], through the creation of multiple views of the system, such as the structural and the behavioral views. Each view is described with a given modeling language and is, by definition, easier to comprehend than trying to understand the entire system at once.

MDE reduces the accidental complexity associated with developing complex software-intensive systems [12]. A primary source of accidental complexity is the gap between the high-level concepts used by domain experts to express their specific needs and the low-level abstractions provided by general-purpose programming languages [13], [14]. Manually bridging this gap is costly, both in time and effort. MDE approaches address this problem by automatically generating the major system artifacts (e.g., test cases, implementations) from models. MDE models focus on how system functionality and domain-specific concepts (e.g., automotive, medical, or financial) are modeled relative to how their behavior may be specialized for the respective domain (e.g., an automotive system has numerous sensors for onboard control systems that all communicate via a communication bus, thus requiring sensors, controllers, and a communication bus to be modeled). Domain-specific modeling languages (DSMLs) provide a vocabulary and modeling primitives to make it easier for the domain experts to create models that describe the intended behavior of systems to be developed [15]. For these domain-specific systems, when we discuss "non-functional" goals, we typically refer to qualities or properties about the functional goals or requirements. Examples include performance, reliability, and memory usage.

Figure 1 depicts a typical MDE approach where engineers use engineering models to develop software-based sustainability systems, where models of the functional objectives of the system are created, updated (*Engineering Model* in Figure 1), and transformed to code (dependency between *Engineering Model* and *Software*) that is used to manage sustainability systems (e.g., smart grid).

Current MDE approaches that can be used for sustainability systems (such as autonomic and self-adaptive systems) focus on the mechanisms or computing infrastructure (e.g., agent-based systems, self-* systems, architecture models, goal-based models, model-based testing, model-based code generation, model-based reasoning). For example, consider an autonomic power management system that monitors the power consumption of its customers. Based on usage rates of its customers, the system may autonomously route more power to the high-demand areas temporarily to prevent brownouts.

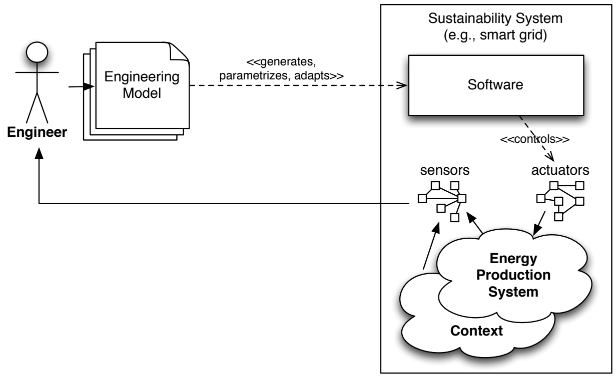

Figure 1. An engineering view of the use of MDE to support sustainability

The designer of such an adaptive system is faced with a challenging set of modeling and development tasks: anticipating how and when the system will need to adapt in the future, codifying this behavior in decision-making components to govern the adaptation, and ensuring system integrity during and after adaptation. These tasks are particularly difficult for systems that must operate reliably and safely in the face of continuous dynamics and environmental uncertainty. One approach to manage the complexity is to use MDE to model the functional system, adaptation logic mechanisms [16], and non-functional properties such as performance and resource use. Model composition and transformation techniques can be leveraged to support automated generation of the target sustainability system [4], [17]. Depending on the specific MDE approach used, models can be created manually [18], automatically created based on requirements specifications [19], [20], or a combination thereof [17].

Given the inherent uncertainty with these systems, both from the environment and from the modeled systems, it is difficult, if not impossible to anticipate all possible conditions that will occur at run-time [21]. One approach is to use models at run-time to manage the system [22], including self-healing and reconfiguration, and maintaining consistency between the changing system and the corresponding models [23] relative to specific environmental contexts [24]. In this case, the data provided by the sensors are automatically fed back to the engineering models, which are used at run-time to adapt the system (see Figure 1).

IV. MDE for Descriptive Scientific Models

Scientists also handle the complexity of the phenomena they study through modeling [25]. Of course, to be useful as a communication means, models have to be made explicit, that is, assumptions on the world must be made explicit, and communicated in a language that can be understood by stakeholders [26]. There also must be a causal connection between the models and the real-world, in order to assure the fidelity of the result when understanding the phenomena. The principle of substitution (i.e., the result obtained from a model is the same as what would be produced by observing the phenomena in the real-world) often associated with simulation, is important in this context so that the scientists have confidence in the result obtained by a query on the model.

Scientific models are used to explain and predict the behavior of real-world phenomenon (e.g., Newton's laws of gravitation to predict the time it will take for an apple to fall from a tree). That led the philosopher K. Popper[1] to the characterization of scientific theories as falsifiable models, i.e., models that can be compared to some observable reality to assess whether and where they fit. Scientists abstract away from complex details and typically the models they construct only hold within certain boundaries that need to be understood explicitly. For instance, Newton's laws of gravitation only hold nearby the surface and for objects of certain size. Some models are known to be wrong, but still explain certain phenomena quite well, e.g., Kepler's geocentric model of the solar system. Abstraction always means that certain properties are omitted while (hopefully) the relevant ones, with respect to the purpose of the model, are captured in enough detail to fulfill the model's purpose. Whether a model is helpful can therefore only be answered with the knowledge about its purpose.

Figure 2 depicts a multi-view scientific modeling infrastructure. Scientists define models to capture the various dimensions of sustainability. Then, a *simulation engine* can be used to integrate the selected models (possibly including sensor data gathered from the sustainability system itself), analyze the data, and perhaps generate visualizations that depict multiple views of the data and the analysis results. The simulation engine is intended to support two objectives: to facilitate the user in understanding (and visualizing) the impact of a candidate solution, and provide a means to optimize a solution based on analyzing trade-offs across one or more dimensions represented by the scientific models.

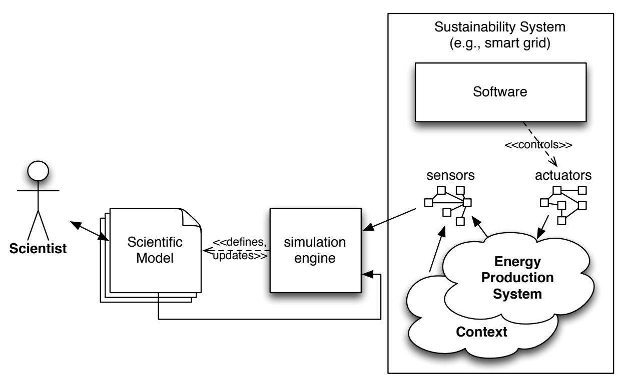

Figure 2. Use models to provide different views of the system.

Recently, we conducted an empirical study to evaluate the effectiveness of using DSMLs to support the simulation of a water management system for use in a farming context [3]. DSMLs were effective in engaging the domain experts in modeling the topological concerns (e.g., landscape and farms) and behavioral concerns (e.g., animal growth and plant lifecycle). Our investigation showed that by using DSMLs to raise the level of abstraction of modeling to domain-specific concepts (rather than programming elements), we were able to broaden the use of MDE to domain experts for scientific applications. Along with the positive feedback brought by these investigations, we had the opportunity to identify additional points that deserve to be studied more closely in order to make the experience of adding abstraction through the use of modeling techniques even more beneficial to the modeling process and the resulting analysis capabilities. Areas for further work include improving user-friendliness for tools, support for modeling strategies that are closer to the domain user's thought process and perception of the system model, use of model-based validation and verification, and promoting the idea of design as an art. The main conclusion of the work was that the current trends in MDE were progressing in the right direction with increasing support for modeling and managing a "Global problem" (e.g., zoom in/out based on science-based questions and concerns, rather than focusing on systems elements).

[1] See http://plato.stanford.edu/entries/popper/

V. Socio-Technical Modeling Shift Required for Sustainability

Computing and technology continue to play an increasing role to support sustainability research. More sophisticated ecosystem monitoring techniques have yielded petabytes of data for scientists to analyze. As scientists build increasingly more complex scientific models, additional challenges are posed to the computing disciplines to make the data and model-based analysis results more accessible and understandable to the scientists and other stakeholders, and automatically fed back to the engineering models to adapt the corresponding sustainability systems.

Building from the context of the previous figures, Figure 3 depicts an integrated approach where each model provides feedback to the other. In this approach, engineering models are dynamically adapted by interpreting the resulting impact on the sustainability system, while scientific models are also impacted by the use of the engineering models in a dynamic feedback loop that continuously adapts the system to reflect the change in priorities among the three pillars. Other stakeholders (e.g., individuals, community leaders, policy makers, industrial organizations) can select specific (personalized) views of sustainability to explore the impact of changes in social behavior, policies, and resource consumption.

We introduce the concept of *Aggregator* at the core of the approach, whose objective is threefold: i) to ensure the fusion of the various heterogeneous data sources (scientific models, personal expectations and information collected from the sensors); ii) to adapt the scientific models accordingly and to provide specific views accessible to a broader public; and iii) to (automatically) update the engineering models according to science-informed decisions taken from the scientific models. The three central roles of the Aggregator are further described in the remainder of this section.

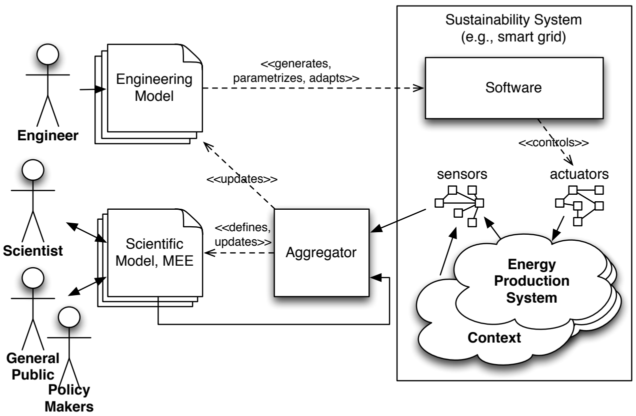

Figure 3. Intertwined use of engineering and scientific models

A. Model and Data Fusion

When considering the modeling needs of an autonomic or self-adaptive sustainability system, it must be able to monitor its environment, adapt to changing conditions, and be resilient to component failures and attacks. Many of these systems make use of wireless sensor networks and cyber-physical systems. As these systems grow in scale, complexity, and heterogeneity, they become ultra-large scale systems (ULS) [27] or Internet of Things (IoT) [28]. ULS systems comprise tens of thousands of sensor nodes, thousands of decision nodes, involve heterogeneous components (e.g., wind turbines, solar panels, small motors) that must work together collectively to produce specific results, such as energy production [27]. In the original definition of ULS systems, the concept of a *ULS Ecosystem* was introduced to capture the interplay between cyber-physical, environmental, and social.

Consider the emerging intelligent transportation system (ITS) comprising numerous intelligent vehicle systems, each containing numerous sensors and onboard controllers to provide autonomic features, such as lane keeping, smart cruise control, and collision avoidance. These cars need to communicate with each other, with satellites and roadway beacons. The objective of the entire ITS is to minimize collisions, while increasing throughput of vehicles traveling from point A to point B. MDE is used to capture the intended vehicle behavior in the context of social concerns (e.g., traffic laws) and environmental conditions (e.g., road conditions, weather). The main challenge with modeling social and environmental concerns is uncertainty. When considering sustainability systems for ULS, we need to add the economic factors to the ULS Ecosystem.

Unlike an ITS, a sustainability system requires consideration of the many trade offs between the numerous scientific models when looking for potential solutions, where solutions may involve changes to one or more of the three pillars of sustainability, as well as the sustainability system itself. For example, when considering power consumption and power production from multiple sources (e.g., wind, solar, nuclear, water), policy analysts may use the Aggregator to explore several scenarios involving the respective scientific models and the sensors from the sustainability systems to determine what to do during a drought season with record high temperatures. The Aggregator analysis may suggest how to adjust cost models and/or legislate temporary laws during the drought season (e.g., impose fines for washing cars or watering lawns). In addition, the feedback from the Aggregator to the smart grid system may be to harness more power from solar sources, decrease demand from water-based sources, and reroute power to high-demand areas and ensure that hospitals and other care facilities have sufficient resources to run air-conditioning, among other needs.

B. Personalizing Sustainability

A key insight to our vision is that sustainability approaches must be customer-centric, focusing on individuals. According to OPOWER[2], "What's at stake are tomorrow's energy consumers. To thrive for another century, utilities must capture their attention and exceed their expectations. It starts with the customer experience." To effectively address sustainability, the customer has to understand and share the vision of the providers and vice versa. Previously, we proposed the concept of *Model Experiencing Environments (MEEs)* [9] as an approach to support complex model and data integration, while offering customizable interfaces for accessing model analysis results and their visualizations. In this paper, the Aggregator is intended to support the different types of model and data integration needed by a MEE, where a MEE interface and visualization support is customizable according to a given user and their interests. The objective of MEEs is to enable different types of stakeholders to "experience" models according to their level of interests and needs. A MEE can be used as an educational medium for enabling children and others to learn about the impact of their individual actions and decisions (e.g., what is the impact of taking 4-minute showers instead of 15-minute showers over a year long period on water consumption for the household? for the town?). MEEs can also be used to study the effects of collective behavior by a community (e.g., what is the impact on environmental resources by decreasing red meat consumption down to once per week?). For land use or environmental policy analysts, MEEs can be used to pose more sophisticated "what if?" scenarios to understand the impact of specific policies or legislation. MEEs empower stakeholders to better understand the cause and effects of individual and global actions and decisions. MEEs are intended to make the complexity of the scientific models, the data and model fusion process, the distributed nature of the data, and the resolution of temporal and spatial differences between the models transparent to the MEE user, thus enabling them to focus on the science questions.

C. Feedback to the Engineering Models

Each of the social, economic and environmental pillars must have their respective information captured in a (domain-specific) model. New integration techniques are needed to

[2]See http://www2.opower.com/moments-that-matter-whitepaper.

support a systematic, well-defined approach to integrate these disparate models to enable well-informed decisions and functionality to be provided by sustainability systems. When the Aggregator is integrating the scientific models, it must consider the temporal and spatial dependencies, as well as the granularity of the data used for the different models and sensor data from the sustainability system. Updates to the engineering models for the sustainability system must consider the type of user engaged with the system. In other words, an individual power consumer will not be allowed to effect change to the power distribution system. But the manager of a power generation company may use the MEE interface to explore different "What if" scenarios to determine the most reliable and cost-effective strategy for delivering power to its customers. The results of the analysis may propose updates to be autonomously made to the sustainability system via the feedback link from the Aggregator.

D. Additional Socio-Technical Concerns

In addition to the modeling challenges previously mentioned, several additional areas need to be considered to facilitate the socio-technical modeling shift. Approaches should include resource usage analytics techniques to enable researchers to examine complex relationships between models and variables, using the power of predictive analytics to understand behavior patterns and the impact of one pillar with respect to the other pillars. New approaches should integrate ideas from behavioral science to produce persuasive solutions to engage individuals; therefore, going beyond the traditional one-size-fits-all solutions. In such a setting, involving thousands or millions of people, data security and anonymity to preserve customer privacy needs to be planned from the very early stages of the development. Finally, the proposed models, languages and techniques need to handle the voluminous amount of disparate data coming from a wide variety of sources. In the era of Big Data, reducing large-scale problems to a scale that humans can comprehend and act upon is fundamental. Thomas and Cook [29] discuss various scalability challenges that range from information scalability, to visual scalability, display scalability, human scalability, and software scalability.

VI. Conclusion

This paper addresses the role of modeling for sustainability systems. We are convinced that models will play an essential role in promoting a broader engagement by individuals and global groups in addressing sustainability concerns. We have examined how models are used in both engineering and scientific communities. While both communities have made great advances in increasing the complexity of what can be modeled, and increased the sophistication of how models can be used, analyzed, and executed, we postulate that effectively using the modeling efforts from both communities in a much more coordinated, collaborative fashion offers the potential to address important challenges in the development of sustainability systems, including an access to a much broader group than just scientists or engineers/software developers.

References

[1] Committee on Computing Research for Environmental and Societal Sustainability, *Computing Research for Sustainability*. National Academies Press, 2012.

[2] U. W. C. o. Environment and Development., "Our common future," 1987, see http://en.wikipedia.org/wiki/Sustainability.

[3] J.-M. Bruel, B. Combemale, I. Ober, and H. Raynal, "MDE in Practice for Computational Science," in *ICCS*, 2015.

[4] B. Solomon, D. Ionescu, M. Litoiu, and M. Mihaescu, "Model-driven engineering for autonomic provisioned systems," in *COMPSAC'08*.

[5] J. Dean, "Power to the people in energy revolution: Eco-batteries will slash household bills," *The Times*, May 2015.

[6] A. B. Lovins, "The negawatt revolution," *Across the Board*, 1990.

[7] H. Stachowiak, *Allgemeine Modelltheorie*. Springer-Verlag, 1973.

[8] C. M. Pease and J. J. Bull, *Scientific Decision-Making*. Online E-Book.

[9] G. Mussbacher, D. Amyot, R. Breu, J.-M. Bruel, B. Cheng, P. Collet, B. Combemale, R. France, R. Heldal, J. Hill, J. Kienzle, M. Sch"ottle, F. Steimann, D. Stikkolorum, and J. Whittle, "The Relevance of Model-Driven Engineering Thirty Years from Now," in *MODELS'14*. Springer.

[10] D. L. Parnas, "On the criteria to be used in decomposing systems into modules," *Commun. ACM*, vol. 15, no. 12, pp. 1053–1058, 1972.

[11] J. M. Atlee, R. France, G. Georg, A. Moreira, B. Rumpe, and S. Zschaler, "Modeling in software engineering," in *ICSE*. IEEE, 2007.

[12] D. C. Schmidt, "Model-driven engineering," *IEEE Computer*, vol. 39, no. 2, pp. 25–31, 2006.

[13] R. France and B. Rumpe, "Model-driven Development of Complex Software: A Research Roadmap," in *FOSE'07*. IEEE, 2007.

[14] J. Gray, J.-P. Tolvanen, S. Kelly, A. Gokhale, S. Neema, and J. Sprinkle, "Domain-specific modeling," in *Handbook of Dynamic System Modeling*. CRC Press, 2007, pp. 1–20.

[15] F. Fleurey and A. Solberg, "A domain specific modeling language supporting specification, simulation and execution of dynamic adaptive systems," in *MODELS'09*, ser. LNCS. Springer Berlin Heidelberg, 2009, vol. 5795, pp. 606–621.

[16] T. Vogel, S. Neumann, S. Hildebrandt, H. Giese, and B. Becker, "Model-driven architectural monitoring and adaptation for autonomic systems," in *ICAC'09*. ACM, 2009.

[17] B. Morin, F. Fleurey, N. Bencomo, J.-M. Jézéquel, A. Solberg, V. Dehlen, and G. Blair, "An aspect-oriented and model-driven approach for managing dynamic variability," in *MODELS'08*. Springer, 2008.

[18] J. Zhang and Betty H.C. Cheng, "Model-based development of dynamically adaptive software," in *ICSE'06*. ACM, 2006.

[19] H. J. Goldsby, Betty H.C. Cheng, P. K. McKinley, D. B. Knoester, and C. A. Ofria, "Digital evolution of behavioral models for autonomic systems," in *ICAC'08*. IEEE, 2008.

[20] H. J. Goldsby and Betty H.C. Cheng, "Automatically generating behavioral models of adaptive systems to address uncertainty," in *MODELS'08*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 568–583.

[21] B. Cheng, A. Ramirez, and P. McKinley, "Harnessing evolutionary computation to enable dynamically adaptive systems to manage uncertainty," in *CMSBSE*. IEEE, 2013, pp. 1–6.

[22] N. Bencomo, "On the use of software models during software execution," in *MISE'09*. IEEE, 2009.

[23] T. Vogel, S. Neumann, S. Hildebrandt, H. Giese, and B. Becker, "Incremental model synchronization for efficient run-time monitoring," in *Models in Software Engineering*, ser. LNCS. Springer Berlin Heidelberg, 2010, vol. 6002, pp. 124–139.

[24] M. Acher, P. Collet, F. Fleurey, P. Lahire, S. Moisan, and J.-P. Rigault, "Modeling Context and Dynamic Adaptations with Feature Models," in *4th International Workshop Models@run.time at Models 2009 (MRT'09)*, 2009, p. 10.

[25] R. Frigg and S. Hartmann, "Models in science," in *The Stanford Encyclopedia of Philosophy*, 2012.

[26] A. Quarteroni, "Mathematical models in science and engineering," *Notices of the AMS*, vol. 56, no. 1, pp. 10–19, 2009.

[27] L. Northrop, P. Feiler, R. P. Gabriel, J. Goodenough, R. Linger, T. Longstaff, R. Kazman, M. Klein, D. Schmidt, K. Sullivan, and K. Wallnau, "Ultra-Large-Scale Systems - The Software Challenge of the Future," Software Engineering Institute, Tech. Rep., 2006.

[28] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[29] J. Thomas and K. Cook, Eds., *Illuminating the Path: Research and Development Agenda for Visual Analytics*. IEEE Press, 2005.