Simulating Cyber-Physical Energy Systems: Challenges, Tools and Methods

Peter Palensky, *Senior Member, IEEE,* Edmund Widl, and Atiyah Elsheikh

***Abstract*—The energy system of the future is expected to be composed of a large variety of technologies and applications. However, the diverse nature of these components, their interlinked topology, and the sheer size of the system lead to an unprecedented level of complexity. Industry is confronted with severe problems in designing interoperable grid components, analyzing system stability, and improving efficiency. This paper describes the main challenges of continuous time-based and discrete event-based models of such cyber-physical energy systems. Using a characteristic test model, the scalability of the two approaches is analyzed. The results show the strengths and weaknesses of these two fundamentally different modeling principles that need to be considered when working with large scale cyber-physical energy systems.**

***Index Terms*—Cyber-physical energy systems, hybrid systems, modeling, simulation, software tools.**

I. Introduction

ENERGY systems are experiencing a gradual but substantial change. Electric mobility and a transition to renewable energy sources are very much welcomed but they also increase the complexity of the systems. Both the components and the systems industry face new challenges in developing new technologies because established methods and tools cannot deal with the nature of future smart grids. That is, traditional, unidirectional, and hierarchical topologies are becoming more distributed and flat. The enabling technologies are power electronics and information technology (IT), leading to cyber-physical energy systems. A short survey of these topics is provided in [1]. IT adds another unprecedented level of complexity: decentralized control, agents in the grid and on markets, smart buildings, and autonomous software make every physical system generally more complex, due to the rapidly increasing number of systems states.

To describe the energy system of the future, the following four categories of phenomena and/or domains must be considered.

1) *Physical World/Continuous Models:* Energy generation, transport, distribution, consumption, infrastructure, and their components.
2) *Information Technology/Discrete Models:* Controllers, communication infrastructure, and software.
3) *Roles and Individual Behavior/Game Theory Models:* Agents acting on behalf of a customer, market players.
4) *Aggregated and Stochastic Elements/Statistical Models:* Weather, macroview on consumers, generalized characteristics of many individual elements.

Just combining the first two categories in one model results in variable structure dynamic models [2] that are complex to analyze and implement. Consideration of the latter two categories results in large stochastic hybrid models. Such models characterize complex continuous controlled systems for instance. They also contain self-deciding agents of discrete behavior, in a fully random environment, with a large number of states [3]. These states typically correspond to a large number of operational and failure modes that are difficult to estimate with traditional approaches [4]. This introduces further challenges for model-based monitoring and diagnoses. The correct integration of model-based diagnostic techniques, together with data acquisition and modeling of faults, is therefore a key requirement to enable diagnostic reasoning [5]. Another challenge arises due to the size of energy systems. It is the scalability of the models that is particularly important, when tens of thousands of components interact.

Unfortunately, there is a lack of methods and tools. While highly specialized and useful tools for the various details and individual domains of energy systems exist, there is no method or tool that combines all of the above aspects. [6] presents promising activities in modeling cyber-physical energy systems but provides no answers to the question of how to seamlessly integrate discrete controls and roles. This paper investigates the possibilities and limits of popular tools with respect to the modeling and simulation of hybrid (i.e., continuous and discrete) energy systems.

II. Fundamental Challenges in Simulating Complex Energy Systems

Examples for energy systems with grid-friendly, agent-controlled buildings are presented in [7]. They constitute the future of our energy system with the potential to be more efficient and include more renewable energy sources. These energy systems not only contain technical infrastructure, but also smart software agents, markets, and other components, which inevitably require the employment of hybrid modeling approaches.

Manuscript received August 31, 2012; revised December 21, 2012; accepted May 10, 2013. Date of publication October 7, 2013; date of current version February 12, 2014. This paper was recommended by Associate Editor T. I. Strasser.

The authors are with the Energy Department, Austrian Institute of Technology, Vienna 1220, Austria (e-mail: peter.palensky@ait.ac.at; edmund.widl@ait.ac.at; atiyah.elsheikh@ait.ac.at).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCC.2013.2265739

From an abstract point of view, such hybrid systems can be regarded as an ensemble of concurrent finite state machines coupled via an interactive environment [8], [9]. These states are not necessarily static as in the case of the different modes of a controller. Especially for components representing physical objects, the individual states correspond to different continuous-time models, representing their dynamics according to their external and internal conditions. For a residential building the states could, for instance, correspond to the combinations of all the possible on/off-states of the HVAC sub-systems, each described by an adequate thermodynamic model. The transition between two states corresponds with a component's reaction to the alteration of the external (e.g., environmental impacts) and internal (e.g., decision by local controller or agent) conditions. Any alteration of the conditions can therefore be triggered either deterministically or stochastically.

The treatment of such hybrid systems poses various challenges.

1) *System Topology:* The exact and proper representation of a system's topology within a simulation environment is critical, especially for the treatment of complex problems. Due to the typical structure of energy systems that comprise a multitude of individual components, an object-oriented representation of the topology is preferable. In many cases a hierarchical representation is also a convenient method to characterize the dependencies between components.

2) *Data Flow and Concurrency:* Within a simulation environment the individual components have to be able to provide information concerning their internal state. This is not only the basis for information exchange between components but is also essential for data logging and monitoring. In addition, a simulation environment has to also provide the means to handle the data flow between all concurrent components such that a well-defined incremental evolution of the simulation is possible, ensuring, for example, causality and consistency. Ideally the corresponding interfaces should follow strong interoperability rules in order to enable model exchange and unit testing.

3) *Plurality of Event Types and Massive Event Occurrence:* Models of energy systems comprise of a variety of different types of events that trigger state transitions of individual objects. A simulation environment needs to be able to resolve a potentially large number of events within a short time span. Depending on the simulation's level of granularity the applied method has to achieve appropriate accuracy.

4) *Variable Structure Dynamics:* Cyber-physical systems tend to enable, disable, or otherwise alter individual parts, which leads to variable structural dynamics. Consequently, the simulation environment has to be able to identify such events and perform the corresponding actions.

5) *Modeling Language:* A modeling language has to be able to represent all the issues above in a well-established and organized way. For convenience it should also include features that simplify the tasks of model generation, component initialization, incremental model development, and capabilities for code reuse. For the sake of easy interpretation and extensibility, object-oriented approaches are again preferred. Finally, readability (both human and machine) is an important aspect.

6) *System Scalability:* Simulations of energy system models are potentially comprised of a large number of components, which have to be handled by an adequate simulation tool within a reasonable runtime, using an appropriate level of resources (e.g., number of cores, shared memory). This requires the employment of efficient algorithms as well as high-performance computing solutions. Even if the targeted systems are relatively small (e.g., micro-grids, unless fully islanding), their interplay with neighboring systems is of special interest, which again leads to scalability challenges.

III. State of the Art

A large variety of simulation tools for the individual components of energy systems are available today. This is especially true for electric power grids, where the development of tools for the simulation of generation, transmission, and distribution has been driven by the engineering requirements of providers and the industry. Proprietary simulators include NEPLAN [10], PSS Product Suite tools [11], PowerFactory [12], PowerWorld Simulator [13], PSCAD [14], or eMEGAsim (focused on HIL simulation) [15]. Free open-source implementations include OpenDSS [16], InterPSS [17], and Homer [18]. Also a variety of MATLAB tools exists, including the proprietary SimPowerSystems [19] as well as the free open-source toolboxes PSAT [20], VST [21], and MATPOWER [22]. There is an effort to promote the common information model (CIM) [23]—and derived models, such as the common distribution power system model (CDPSM) [24]—as a standard for the description of electric networks.

Similar to the situation in power engineering, is the ever-growing need for the development and extension of IT networks leading to a number of simulation tools. Proprietary network simulators include OPNET Modeler [25] and NetSim [26]. Free open-source implementations include ns-3 [27] and OMNeT++ [28].

For modeling energy consumption in buildings, a number of commercial and free tools are available. [29] gives an extensive overview of the capabilities and differences these tools possess. However, the authors mainly focus on EnergyPlus and TRNSYS, probably the two most popular and well-established tools with large validated component libraries. The capabilities of these two packages to be integrated in complex control simulation is, unfortunately, limited. [30] explicitly classifies buildings under cyber-physical energy systems but deals only with static models, while this paper focuses on dynamic systems.

Apart from specialized tools, a variety of universal multidomain modeling and simulation solutions exist. Several of these projects are based on Modelica [31], [32], an object-oriented simulation language that offers an extensive (and extensible) set of standard libraries for flexible physical modelling. Commercial vendors of Modelica-based simulation tools include Dymola [33], MathModelica [34], or MapleSim [35]. Open-source Modelica-based environments are provided by OpenModelica [36], and JModelica [37]. A conceptually similar approach has been realized with the proprietary package Simscape [19], which includes a rich set of extendable libraries for standard engineering problems.

A very generic tool for heterogeneous, concurrent modeling and design is the open-source framework Ptolemy II [38] that allows the combination of—among others—discrete and continuous models. Due to its abstract and general implementation it has been used as the basis for several other projects, such as the hybrid system simulator HyVisual [9] or the building controls co-simulation tool BCVTB [39].

Generic open-source co-simulation frameworks with a focus on energy systems are Simantics [40] and Mosaik [41]. They both provide extensive functionalities for formal model description and couplings to other simulation engines. Examples for co-simulations platforms with specific application-tailored functionalities are given in [42] (power systems and IT) and [43] (power systems and electric mobility).

An open-source project specifically dedicated to the simulation and analysis of energy systems is GridLAB-D [44]. It provides framework and toolboxes for simulating various aspects of energy systems. This software includes its own set of tools for power flow and reliability analyses of electric systems, along with simulation of residential, commercial, and industrial buildings and the implementation of custom controllers and energy market simulation.

IV. Comparison of Two Fundamentally Different Simulation Approaches

In this paper, two completely complementary simulation approaches are applied to the same problem. They correspond to two fundamentally different approaches of modeling hybrid systems. The first approach is discrete event-based, focusing on the successive handling of autonomous and concurrent objects. The second approach is continuous time-based, starting from the complete set of algebraic differential equations that define the individual objects.

The goal of this comparison is to highlight and contrast the advantages and disadvantages of these two approaches. This will give the possibility to make proposals for future developments.

A. Description of Test Model

An easily scalable test model is introduced, which allows an assessment of the two approaches' capabilities with respect to resolving component dependencies, event handling, data flow between objects, and scalability. Communication between objects is considered ideal, i.e., lossless and without delay, such that ICT-related issues are not taken into account. The functionality of the individual components is kept simple on purpose, in order to focus on the study of the concurrent evolution and interplay of the constituent parts. Even though the model topology is also kept rather simple, it already comprises features that are characteristic for energy systems, such as deterministic and stochastic events, feedback loops, and hybrid elements.

The topology of the test model is shown in Figs. 1 and 2. The image syntax is oriented toward hybrid bond graphs. The

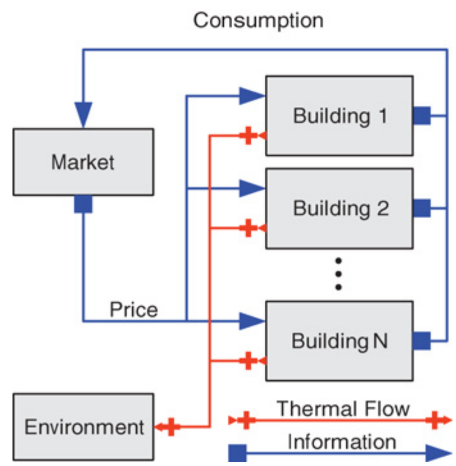

Fig. 1. Top level view of test model.

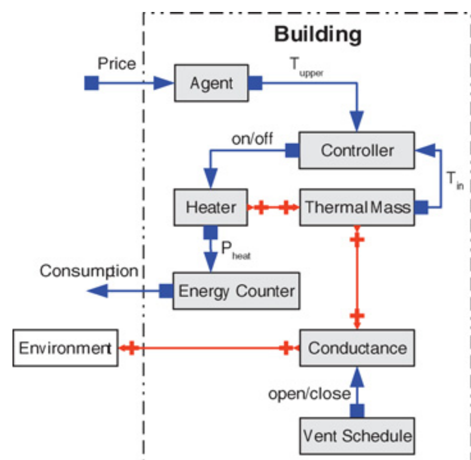

Fig. 2. Detailed view of building.

unfortunate lack of certain features (e.g., heat flow) in existing proposals, such as in [45] and [46] has lead to the new syntax use within this paper.

1) *Buildings:* The buildings are modeled as thermal capacitors (inside volume) linked to a thermal reservoir (ambient environment) via a thermal resistor (walls). The capacitor's temperature is monitored by a two-level controller (thermostat with hysteresis between $T_{\min}$ and $T_{\max}$), which can feed a constant heat flow (heating).

a) Thermal energy stored in thermal capacitor

$$Q_{\text{store}} = \rho V C_{\text{th}} T_{\text{in}}. \tag{1}$$

b) Heat flow through thermal resistor

$$\dot{Q}_{\text{loss}} = \frac{1}{R_{\text{th}}} \left(T_{\text{in}} - T_{\text{amb}}\right). \tag{2}$$

c) Heat flow regulated by two-level controller

$$\dot{Q}_{\text{heat}} = \begin{cases} 0, & \text{if heating is off} \\ P_{\text{heat}}, & \text{if heating is on.} \end{cases} \tag{3}$$

d) Heat flow balance

$$\dot{Q}_{\text{store}} = -\dot{Q}_{\text{loss}} + \dot{Q}_{\text{heat}}. \tag{4}$$

T_{in} and T_{amb} is the inside and outside temperature, ρ and V the density and the cubic content of the inside volume, C_{th} is the thermal capacity of the inside volume and R_{th} is the thermal resistance of the walls.

2) *Stochastic Events:* At random times the buildings are ventilated (windows open/closed), effectively modeled by a change of the thermal resistance[1]

$$\frac{1}{R_{\text{th}}} = \begin{cases} \frac{\lambda \phi V}{d}, & \text{if windows are closed} \\ \frac{\lambda \phi V}{d} +, G_{\text{vent}}, & \text{if windows are open.} \end{cases} \tag{5}$$

The walls' thickness and thermal conductivity are given by d and λ, ϕ is the ratio of the outside walls' area to the inside volume's cubic content (form factor), and G_{vent} is the additional heat conductance value representing ventilation.

Two different time constants, τ_{open} and τ_{close}, are used to specify the average waiting time for a window to be opened and then closed again, respectively. To model independent random events, the waiting times are drawn from exponential distributions, according to the corresponding time constants.

3) *Market:* The price per kilowatthour consumed is dependent on the average energy consumption per building. The latter is determined by reading all energy meters at regular intervals (Δt_{meter}), calculating the mean energy consumption of all buildings, and then taking the average over the last n intervals. The price p is calculated via

$$p = p_0 + \langle \bar{E}_{\text{con}} \rangle_n \times p_1. \tag{6}$$

E_{con} is the energy consumed per building between two meter readings, $\bar{E}_{\text{con}}$ is the mean value of E_{con} of all buildings, and $\langle . \rangle_n$ denotes the average over n periods. The terms p_0 and p_1 are constants.

4) *Agents:* The upper goal temperature $T_{\max}$ of a building's thermostat is controlled by an agent. In case the energy price exceeds a certain level $p_{\max}$, the upper goal temperature is reduced ($T_{\max} \to T_{\text{alt}}$).

5) *Weather:* The temperature of the buildings' ambient environment is subject to change, modeled by a sinusoidal pattern

$$T_{\text{amb}} = \bar{T}_{\text{amb}} + \Delta T_{\text{amb}} \sin(\omega t + \varphi). \tag{7}$$

The parameters ω and ϕ are chosen such that the minimum temperature occurs every day at midnight.

B. Discrete Event-Based Micro-Simulation With GridLAB-D

1) *Background:* This discrete event-based micro-simulation approach provides the means to handle the interplay of autonomous, concurrent objects. Such systems, even if composed of simple objects, can exhibit complexity and emergence.

For the purpose of this paper, the open-source tool GridLAB-D has been used. Its key features are as follows.

[1]The authors are well aware of convection and that opening a window does not change conduction. The purpose of this test model at the current state is, however, to show variable structure dynamics and not physical accuracy. Opening the window is therefore simply mapped onto a changed thermal conductance.

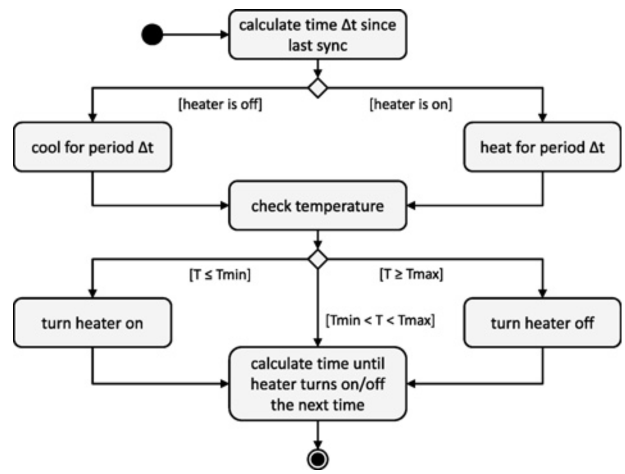

Fig. 3. Schematic view of synchronization procedure of building object in GridLAB-D.

a) GridLAB-D aims primarily at the analysis of power distribution systems. It provides a number of plug-in modules for the simulation of energy generation, distribution, and consumption as well as related topics such as controls, network communication, or markets. Custom plug-in modules can be implemented in C/C++.
b) GridLAB-D uses its own modeling language called GLM (GridLAB-D modeling language), which provides a parametric syntax for the creation of hierarchical models. It also offers tools for object initialization and data analysis.
c) GridLAB-D allows the definition of hierarchies between all objects, which are introduced via parent–child relationships. This ranking determines the scheduling of the objects, i.e., the order of execution at every simulation time step. If necessary, the relative ranking between objects can be altered on purpose.
d) Each object is by itself responsible to update its internal state according to the global simulation time. GridLAB-D's simulation core watches for changes in the states of these objects during runtime (discrete events) and enables synchronized interactions between them. Each object has to notify the simulation core when its internal state is supposedly going to change, under the assumption that all other objects remain in their current state. This implies that all objects have to be synchronized every time an event occurs.

2) *Implementation:* A custom plug-in module has been developed in C++, which includes implementations of all the components of the test model as GridLAB-D objects. Fig. 3 shows as an example the schematic view of the synchronization function of the object that represents a building's controller, heater, thermal mass, and conductance. This function updates the internal state of the object according to the current simulation time, then checks whether the heater has to be switched on/off, and finally calculates (and returns) when the heater will be switched on/off the next time. This function gets called directly by GridLAB-D's simulation core.

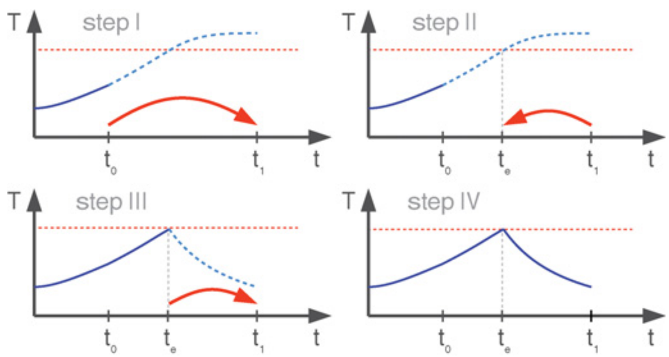

Fig. 4. Schematic view of rollback mechanism for single building object for fixed step schedule. Evolution in time of building's state is indicated by blue lines.

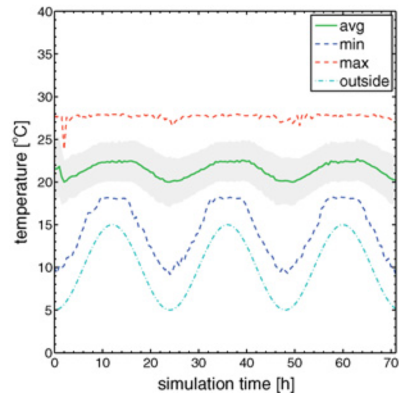

Fig. 5. Temperature profile for 1000 buildings simulated using dynamic schedule.

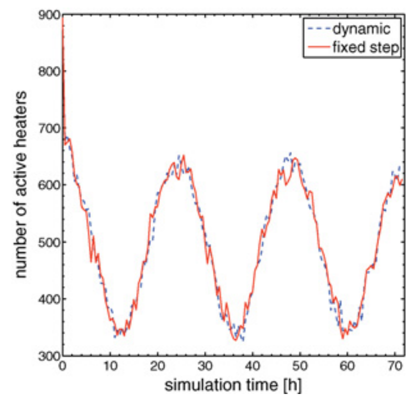

Fig. 6. Comparison of thermostat status profiles for different synchronization schedules.

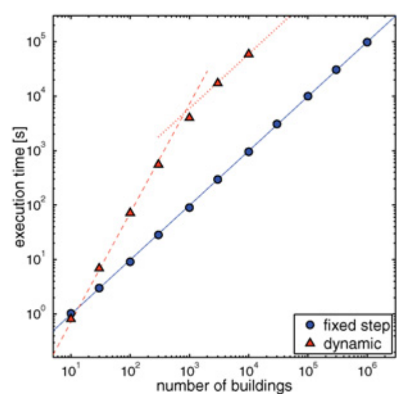

Fig. 7. Overview of elapsed computing times in GridLAB-D.

The test model has been simulated using two different scheduling mechanisms. The dynamic schedule uses the fact that the test model can be solved analytically for individual buildings such that the time for the next state transition, i.e., a change of the internal state, can be computed exactly in advance. This allows precise synchronization of all objects whenever the global state of the system changes. In the fixed step schedule, synchronizations occur with a fixed frequency. To avoid inaccuracies, the time step size has to be chosen such that it is small in comparison to the typical time span between two state transitions of an individual object.

In the dynamic scheme random events are handled explicitly. Since they cause state transitions of individual objects, they also trigger a synchronization of all components. For the fixed step schedule, events have been taken into account via a rollback mechanism. During a simulation step whenever an event is overdue for any given object, its effects are computed retrospectively.

Fig. 4 sketches the process of the rollback mechanism for a building object. After a completed simulation step at time t_0, the global simulation time progresses to t_1 (step I). If, for example, the temperature would exceed the upper controller threshold after this step, the object's state is not immediately updated. Instead the time $t_e < t_1$ of the event is computed accordingly (step II), and the evolution of the previous state is only carried out up to this time (step III). After this the internal state transition of the building object takes place (in this case the heater is turned off) and the evolution of the new state up to the time t_1 is computed (step IV). Random events are treated in an analogous manner. In this way the effects of events on individual objects are treated precisely, even though a possible impact on other components is delayed (until t_1).

3) *Results:* Fig. 5 shows the temperature profiles for a scenario with 1000 buildings using the dynamic schedule. The blue, green, and red lines show the minimum, average, and maximum temperature of all the buildings, while the grey band visualizes the observed deviation from the average value (root of sample variance). Fig. 6 shows the number of active heaters as a function of time for the case of the dynamic and the fixed step schedule (with an update interval of 60 s). The comparison shows that both scheduling schemes yield virtually the same result for this simple model.

For the dynamic schedule, the number of synchronization steps per time unit is proportional to the number of buildings N and at each synchronization step all buildings are updated. This is also the case with the fixed step schedule, but the number of synchronization steps per time unit is constant. Hence, the execution time scales with N in case of the fixed step schedule, and with N^2 in case of the dynamic schedule. Fig. 7 shows how the elapsed simulation time depends on the number of included buildings and the scheduling scheme. The blue line represents a linear fit ($t \sim N$) to the execution time measurements for the fixed step schedule. The dashed red line represents a quadratic fit ($t \sim N^2$) to the first four execution time measurements for the dynamic schedule. The linear trend for larger numbers of buildings arises from a saturation effect, where the average time between two synchronization steps becomes smaller than the minimum increment in simulation time (one second).

C. Continuous Time-Based Modeling With Modelica

1) *Background:* Modelica is a modern object-oriented modeling language for DAE-based physical models with an ever increasing attention from the scientific and engineering community. It adopts the non-causal modeling approach, which in contrast to the block-diagram approach (e.g., used in Simulink) does not enforce the specification of causal input–output relations among components and variables [47].

Equation-based object-oriented modeling relies on the fact that even the most complex physical systems can be conceptually decomposed into structured hierarchies of elementary components [48]. Each of these individual components is independently described by basic laws of physics in an intuitive equation-based manner. These components are amended with interfaces that facilitate a well-defined communication with the external world (i.e., other components). Once the implementation of the individual components and their interfaces is provided, the considered model can be constructed easily.

In Modelica so-called connectors are used to establish interactions between components. A connector typically includes two types of variables, potential variable(s) (e.g., electrical potential) and flow variable(s) (e.g., electrical current). Only connectors with identical declarations can be connected (Fig. 8). In this case, connection points assemble two kinds of equations: trivial identity equations for potential variables and sum-to-zero equations for flow variables.

Advantages of employing Modelica are as follows.

1) Universal modeling concepts simplify multidisciplinary modeling, an aspect that is obviously present in energy systems (Sections I and II).
2) The Modelica language encourages the implementation of reusable, independent, and extensible components for supporting fast prototyping of physical models.
3) Modelica is complemented with a set of standardized libraries in many physical domains (e.g., thermodynamics, electrical engineering). Furthermore specialized libraries for simplifying the formulation of complex hybrid systems exist (e.g., state graphs [49], state chart [50], Petri nets [51], [52], and others).
4) Modelica simulation environments usually include advanced compilers that symbolically manipulate typically large DAE systems of higher indexes. They generate efficient real-time simulation code with the help of sophisticated algorithms, using advanced integrators (Fig. 9). This allows the focus to remain on modeling rather than paying attention to details of algorithmic implementation.
5) Since Modelica environments typically include a graphical user interface, the modeling of complex systems becomes a matter of dragging, dropping, and connecting components.

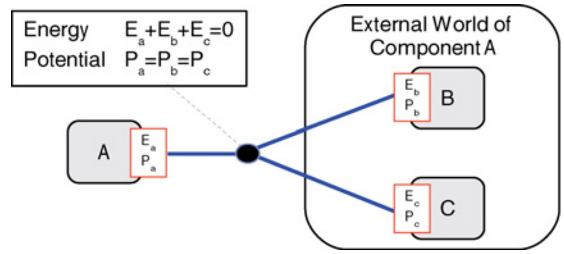

Fig. 8. Connectors provide means for components to communicate.

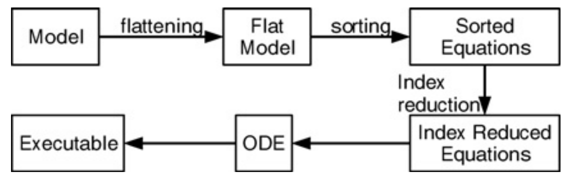

Fig. 9. Typical workflow for Modelica simulation.

2) *Implementation:* The test model has been implemented as a stand-alone Modelica library, that is divided into three packages, in order to facilitate a clear conceptual design.

1) The package "Interfaces" provides the connectors that can be utilized for building up complex models. I/O connectors involve only one potential variable and work in a similar way.
2) The package "Component" provides the basic components for the test model.
3) The package "Types" provides the type definitions of the utilized physical and I/O signals. The associated units are usually employed for performing unit checking when it comes to compilation. At simulation runtime, boundary checking can be performed.

The flexibility of the Modelica language allows for different implementation flavors enabling various perspectives of the model structure to be considered [53]. For instance, parameterizing the component type for the energy price calculation makes it possible to easily adopt different energy price models. Such an approach is particularly useful for supporting rapid prototyping on the basis of top-down modeling, when less detailed elementary components are replaced with more detailed ones.

Furthermore, many useful features and advanced constructs for model abstractions, model templates, inheritance, arrays, variable structure systems, and event handling are employed within the overall language specification.

3) *Results:* For demonstrating accuracy and performance benchmarks, the Modelica simulation environment Dymola (release version 7.4) has been employed. Dymola provides several types of integrators capable of handling DAE systems with events.

Analogous to the results from GridLAB-D, Fig. 10 shows the average temperature profile of 1000 buildings simulated using the variable step size integrator LSODAR with a relative tolerance of 10^{-6}. Fig. 11 shows the performance of a selection of some of the available methods with different settings. While the fixed-step integrators with relatively large-step sizes need

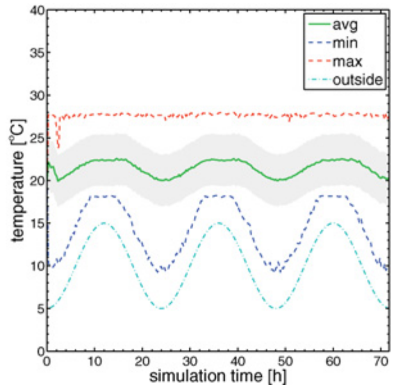

Fig. 10. Temperature profile for 1000 buildings simulated using variable step size LSODAR integrator with relative tolerance 10^{-6}.

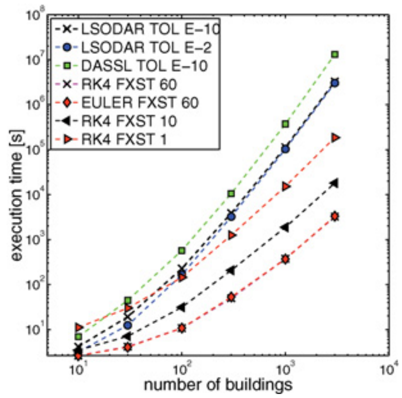

Fig. 11. Overview of elapsed computing times in Dymola.

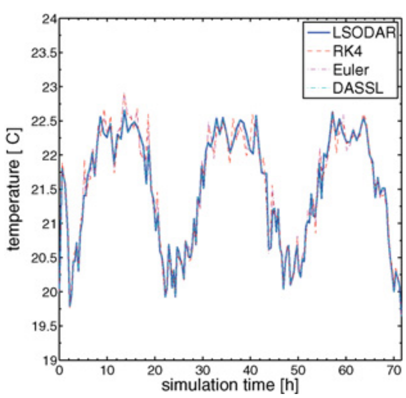

Fig. 12. Comparison of contentious average temperature profile for different integrators.

much less computing time than the variable-step integrators, the accuracy of the results needs to be examined. For instance, Fig. 12 shows a comparison of different methods with strict settings (relative tolerance of 10^{-6} for dynamic integrators and a step size of 1 for fixed-step integrators). The comparison reveals that the choice of method is relatively insignificant and for this type of model, the fixed-step methods can be employed without facing stability or accuracy problems. While the enlargement of step-sizes leads to improved runtime performance, it comes with the price of less accurate results. The selection of the optimum integrator with the correct settings depends largely on whether performance or accuracy is the top priority.

V. Discussion of Results

Both approaches offer their own solutions to (most of) the requirements regarding the simulation of hybrid systems presented in Section II. Even though both approaches yield comparable simulation results, there are basic differences in performance, usability and flexibility.

For the considered test model, the discrete event-based approach performs better with respect to runtime performance and memory usage. With an increasing number of components the equation-based method produces very large executables and shows inferior scaling capabilities. The benefits of Dymola's advanced equation solving and integration algorithms were outweighed by the loss of runtime performance caused by the large number of events and state transitions, for which the discrete event-based approach is clearly more suitable.

On the other hand, the equation-based approach is very well suited to scenarios containing more sophisticated and complex physical models. This is especially true in cases where no analytical solutions exist or where the underlying system is of higher index. In this situation the usage of an equation-based approach becomes more suitable, since GridLAB-D itself offers no generic tools for physical modeling in general.

Overall, the implementation details and the results demonstrate that a discrete event-based modeling approach can be used conveniently and flexibly for coupling discrete and continuous systems with hybrid systems. Equation-based modeling approaches on the other hand offer very effective and intuitive ways to model and solve even complex continuous time-based systems described by several stages of conceptual hierarchies. Obviously, it would be desirable if a simulation environment could combine both aspects, while still retaining their benefits.

VI. Conclusion

Two very different ways of modeling and simulating complex energy systems were presented. The investigated test model was simple but contained the main ingredients: physical elements, discrete elements, asynchronous events, and variable structure dynamics. Moving toward reality, the system was extended one step at a time with more complex parts, like full-physical electric and thermal grids, dynamic energy markets, or individual user behavior.

The consequences of each extension step was analyzed with respect to scalability and numeric performance. In addition to the two simulation environments described above, other tools were added to the benchmark, most notably Mathworks' Simscape and Ptolemy II based cosimulation.

The work with the given two tools also showed that it will make sense to combine these two worlds. The advantages were complementary and a joint setup benefitted from transparent and accessible physical models and flexible event processing.

It was, therefore, the goal to set up a usable high-performance environment to investigate complex energy systems with application scenarios that served as test cases. These were smart grid functions, e.g., demand response, multiagent energy balance communities, or energy storage management, along with long term scenarios, e.g., the impact of renewable energies and other new technologies.

REFERENCES

[1] C. Macana, N. Quijano, and E. Mojica-Nava, "A survey on cyber physical energy systems and their applications on smart grids," in *Proc. IEEE PES Conf. Innovative Smart Grid Technol.*, Oct. 2011, pp. 1–7.

[2] D. Zimmer, "An application of SOL on variable-structure systems with higher index," in *Proc. 7th Modelica Conf.*, Como, Italy, Sep. 2009.

[3] E. Benazera and L. Trave-Massuyes, "Set-theoretic estimation of hybrid system configurations," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 5, pp. 1277–1291, Oct. 2009.

[4] M. Hofbaur and B. Williams, "Hybrid estimation of complex systems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 5, pp. 2178–2191, Oct. 2004.

[5] F. Zhao, X. Koutsoukos, H. Haussecker, J. Reich, and P. Cheung, "Monitoring and fault diagnosis of hybrid systems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1225–1240, Dec. 2005.

[6] M. Ilic, L. Xie, U. Khan, and J. Moura, "Modeling future cyber-physical energy systems," in *Proc. IEEE Power Energy Soc. Gen. Meet.–Convers. Del. Elect. Energy 21st Century*, Jul. 2008, pp. 1–9.

[7] P. Palensky and D. Dietrich, "Demand side management: Demand response, intelligent energy systems, and smart loads," *IEEE Trans. Ind. Inf.*, vol. 7, no. 3, pp. 381–388, Aug. 2011.

[8] T. Henzinger, "The theory of hybrid automata," in *Proc. 11th Annu. IEEE Symp. Logic Comput. Sci.*, Jul. 1996, pp. 278–292.

[9] H. Zheng, "Operational semantics of hybrid systems," Ph.D. dissertation, EECS Dept., Univ. California, Berkeley, CA, USA, 2007.

[10] BCP Busarello Cott Partner AG, Zurich, Switzerland [Online]. Available: http://www.neplan.ch

[11] Siemens PTI, Munich, Germany [Online]. Available: http://www.energy.siemens.com

[12] DIgSILENT GmbH, Gomaringen, Germany [Online] Available: http://www.digsilent.de

[13] PowerWorld Corporation, Champaign, IL, USA [Online]. Available: http://www.powerworld.com

[14] Manitoba HVDC Research Centre, Winnipeg, Manitoba, Canada [Online]. Available: http://pscad.com

[15] Opal-RT Technologies, Montréal, Québec, Canada [Online]. Available: http://www.opal-rt.com

[16] R. Dugan and T. McDermott, "An open source platform for collaborating on smart grid research," in *Proc. IEEE Power Energy Soc. Gen. Meet.*, Jul. 2011, pp. 1–7.

[17] M. Zhou and S. Zhou, "Internet, open-source and power system simulation," in *Proc. IEEE Power Eng. Soc. Gen. Meet.*, Jun. 2007, pp. 1–5.

[18] F. Farret and M. Simões, *Integration of Alternative Sources of Energy*. New York, NY, USA: Wiley-IEEE Press, 2006, pp. 379–418.

[19] MathWorks, Natick, MA, USA [Online] Available: http://www.mathworks.com

[20] F. Milano, "An open source power system analysis toolbox," *IEEE Trans. Power Syst.*, vol. 20, no. 3, pp. 1199–1206, Aug. 2005.

[21] S. Ayasun, C. Nwankpa, and H. Kwatny, "Voltage stability toolbox for power system education and research," *IEEE Trans. Educ.*, vol. 49, no. 4, pp. 432–442, Nov. 2006.

[22] R. Zimmerman, C. Murillo-Sánchez, and R. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

[23] *IEC 61968-11: Application Integration at Electric Utilities—System Interfaces for Distribution Management—Part 11: Common Information Model (CIM) Extensions for Distribution*, IEC Std. ed1.0, 2010.

[24] *IEC 61968-13: Application Integration at Electric Utilities—System Interfaces for Distribution Management—Part 13: CIM RDF Model Exchange Format for Distribution*, IEC Std. ed1.0, 2008.

[25] OPNET Technologies, San Francisco, CA, USA [Online]. Available: http://www.opnet.com

[26] Tetcos, Bangalore, India [Online]. Available: http://www.tetcos.com

[27] E. Weingartner, H. vom Lehn, and K. Wehrle, "A performance comparison of recent network simulators," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2009, pp. 1–5.

[28] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proc. 1st Int. Conf. Simulation Tools Tech. Commun., Netw. Syst. Workshops*, 2008, pp. 1–10.

[29] D. B. Crawley, J. W. Hand, M. Kummert, and B. T. Griffith, "Contrasting the capabilities of building energy performance simulation programs," *Building Environ.*, vol. 43, no. 4, pp. 661–673, 2008.

[30] J. Kleissl and Y. Agarwal, "Cyber-physical energy systems: Focus on smart buildings," in *Proc. 47th ACM/IEEE Des. Autom. Conf.*, Jun. 2010, pp. 749–754.

[31] P. Fritzson, *Introduction to Modeling and Simulation of Technical and Physical Systems With Modelica*, 1st ed. New York, NY, USA: Wiley-IEEE Press, 2011.

[32] P. Fritzson, *Principles of Object-Oriented Modeling and Simulation With Modelica*. New York, NY, USA: Wiley-IEEE Computer Society Press, 2003.

[33] Dassault Systèmes, Vélizy-Villacoublay, France [Online]. Available: http://www.dymola.com

[34] Wolfram MathCore AB, Linköping, Sweden [Online]. Available: http://www.mathcore.com

[35] Maplesoft–Cybernet Systems Company, Waterloo, Ontario, Canada [Online]. Available: http://www.maplesoft.com

[36] P. Fritzson, P. Aronsson, A. Pop, H. Lundvall, K. Nystrom, L. Saldamli, D. Broman, and A. Sandholm, "OpenModelica: A free open-source environment for system modeling, simulation, and teaching," in *Proc. IEEE Int. Conf. Control Appl., IEEE Conf. Comput.-Aided Control Syst. Des., IEEE Int. Symp. Intell. Control*, Oct. 2006, pp. 1588–1595.

[37] J. Åkesson, K.-E. Årzén, M. Gäfvert, T. Bergdahl, and H. Tummescheit, "Modeling and optimization with Optimica and JModelica.org—Languages and tools for solving large-scale dynamic optimization problem," *Comput. Chem. Eng.*, vol. 34, no. 11, pp. 1737–1749, Nov. 2010.

[38] J. Eker, J. Janneck, E. Lee, J. Liu, X. Liu, J. Ludvig, S. Neuendorffer, S. Sachs, and Y. Xiong, "Taming heterogeneity—The Ptolemy approach," *Proc. IEEE*, vol. 91, no. 1, pp. 127–144, Jan. 2003.

[39] M. Wetter, "Co-simulation of building energy and control systems with the building controls virtual test bed," *J. Building Perform. Simul.*, vol. 4, no. 3, pp. 185–203, 2011.

[40] H. N. T. Karhela and A. Villberg, "Open ontology based integration platform for modeling and simulation in engineering," *Int. J. Model., Simul., Sci. Comput.*, vol. 3, no. 2, p. 1250004, Jun. 2012.

[41] S. Schutte, S. Scherfke, and M. Troschel, "Mosaik: A framework for modular simulation of active components in smart grids," in *Proc. 1st Int. IEEE Workshop Smart Grid Model. Simul.*, Oct. 2011, pp. 55–60.

[42] H. Georg, S. C. Müller, C. Rehtanz, and C. Wietfeld, "A HLA based simulator architecture for co-simulating ICT based power system control and protection systems," in *Proc. 3rd IEEE Int. Conf. Smart Grid Commun.*, Nov. 2012, pp. 264–269.

[43] M. Galus, R. Waraich, F. Noembrini, K. Steurs, G. Georges, K. Boulouchos, K. Axhausen, and G. Andersson, "Integrating power systems, transport systems and vehicle technology for electric mobility impact assessment and efficient control," *IEEE Trans. Smart Grid*, vol. 3, no. 2, pp. 934–949, Jun. 2012.

[44] D. Chassin, K. Schneider, and C. Gerkensmeyer, "GridLAB-D: An open-source power systems modeling and simulation environment," in *Proc. IEEE/PES Transmiss. Distrib. Conf. Exp.*, Apr. 2008, pp. 1–5.

[45] P. J. L. Cuijpers, J. F. Broenink, and P. J. Mosterman, "Constitutive hybrid processes: A process-algebraic semantics for hybrid bond graphs," *Simulation*, vol. 84, no. 7, pp. 339–358, 2008.

[46] F. Slomka, S. Kollmann, S. Moser, and K. Kempf, "A multidisciplinary design methodology for cyber-physical systems," in *Proc. 4th Int. Workshop Model Based Arch. Construction Embedded Syst.*, Oct. 2011, pp. 23–37.

[47] S. E. Mattsson and H. Elmqvist, "Modelica—An international effort to design the next generation modeling language," in *Proc. 7th IFAC Symp. Comput. Aided Control Syst. Design*, Gent, Belgium, Jun. 1997, pp. 16–19.

[48] H. Elmqvist and S. E. Mattsson, "Modelica—The next generation modeling language: An international design effort," in *Proc. 1st World Congr. System Simulation*, Singapore, Sep. 1997, pp. 1–6.

[49] K.-E. M. Otter and I. D. Årzén, "Stategraph: A Modelica library for hierarchical state machines," in *Proc. 4th Int. Modelica Conf.*, Hamburg, Germany, Mar. 2005, pp. 569–578.

[50] J. Ferreira and J. Estima de Oliveira, "Modelling hybrid systems using statecharts and Modelica," in *Proc. 7th IEEE Int. Conf. Emerging Technol. Factory Autom.*, vol. 2. Oct. 1999, pp. 1063–1069.

[51] S. Proß and B. Bachmann, "A petri net library for modeling hybrid systems in OpenModelica," in *Proc. 7th Int. Modelica Conf.*, Como, Italy, Sep. 2009, pp. 454–462.

[52] P. J. Mosterman, M. Otter, and H. Elmqvist, "Modeling Petri nets as local constraint equations for hybrid systems using Modelica," in *Proc. Summer Comput. Simul. Conf.*, Reno, NV, USA, Jul. 1998, pp. 314–319.

[53] W. Wiechert, S. Noack, and A. Elsheikh, "Modeling languages for biochemical network simulation: Reaction versus equation based approaches," in *Biosystems Engineering II*, vol. 121 (Advances in Biochemical Engineering/Biotechnology Series). Berlin, Germany: Springer, 2010, pp. 109–138.

Peter Palensky (M'03–SM'08) was born in 1972 in Austria. He received the Ph.D. degree in electrical engineering from the Vienna University of Technology, Vienna, Austria, in 2001.

He is currently a Principal Scientist at the Energy Department, Austrian Institute of Technology (AIT), Vienna, Austria. Before this he was the Head of Sustainable Building Technologies at the AIT, an Associate Professor at the University of Pretoria, South Africa, a University Assistant at the Vienna University of Technology, and a Researcher at the Lawrence Berkeley National Laboratory, CA, USA. His current research interests include complex energy systems and smart buildings.

He is an AdCom member of the Industrial Electronics Society of the IEEE.

Edmund Widl received the M.Sc. and Ph.D. degrees in physics from the Vienna University of Technology, Vienna, Austria, in 2004 and 2008, respectively.

From 2003 to 2011, he was with the Institute of High Energy Physics (HEPHY), Vienna, Austria, involved in several projects for the CMS Collaboration, one of the particle physics experiments at the Large Hadron Collider (LHC) at CERN, Geneva, Switzerland. In 2011, he joined the Complex Energy Systems group of the Austrian Institute of Technology, Vienna, Austria.

Atiyah Elsheikh received the M.Sc. degree in systems software engineering from RWTH Aachen University, Aachen, NRW, Germany, in 2005.

From 2006 to 2010, he was with the Department of Modeling and Simulation, University of Siegen, Siegen, NRW, Germany, and with the Institute of Biotechnology, Research Center of Jülich, Germany. In 2011, he joined the Complex Energy Systems group of the Austrian Institute of Technology, Vienna, Austria.