Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 83 (2016) 976 – 981
Procedia Computer Science

ANTIFRAGILE 2016 - the 3rd International Workshop on Computational Antifragility and Antifragile Engineering

Finding Resilient and Energy-Saving Control Strategies in Smart Homes

Stefano Marrone[a,*], Ugo Gentile[b]

[a]*Dipartimento di Matematica e Fisica, Seconda Universitá di Napoli*
[b]*Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione, Universitá di Napoli "Federico II"*

Abstract

Evolutionary computing has demonstrated its effectiveness in supporting the development of robust and intelligent systems: when used in combination with formal and quantitative models, it becomes a primary tool in critical systems. Among the modern critical infrastructures, smart energy grids are getting a growing interest from many communities (academic, industrial and political) fostering the development of a robust energy distribution infrastructure. Energy grids are also an example of critical cyber physical social systems since their equilibrium can be perturbed not only by cyber and physical attacks but also by economical and social crises as well as changes in the consumption profiles. The paper illustrates a practical framework supporting the run-time evolution of the control logic inside the Smart Meter: the centre of modern Smart Homes. By combining the modeling and analysis capabilities of Fluid Stochastic Petri Nets and the flexibility of Genetic Programming, this approach can be used to adapt the control logic of the Smart Meters to the changes of the structure and functionalities of the Smart Home as well as of the operational environment. While the main objective of the evolution is to guarantee the energetic sustainability of the Smart Home, the fulfilment of the user's requirements about the energetic need of the home allows to preserve the identity of the Smart Meter during its evolution.

Peer-review under responsibility of the Conference Program Chairs
Keywords: Genetic Programming, Computer-based Critical Infrastructures, Smart Energy Grids, Self-Adaptive Systems, Fluid Stochastic Petri Nets

1. Introduction

Home Energy Management Systems (HEMS) constitute the core of modern Smart Buildings. In this *prosumer* era, HEMS manages the need of energy of a complex buildings where traditional appliances are sided by small renewable energy plants and by battery able to store energy for future uses. In this context, sustainability is a key word since HEMSs should pursue strategies which minimise the cost of the energy of the building while guaranteeing the fulfilment of the energy needs of the appliances. Designing the logic which rules the HEMSs is, hence, a primary task in such situations.

* Viale Lincoln, 5. 81100 - Caserta, Italy. Tel.: +39 0823 27 5101; fax: +39 0823 27 4753.
E-mail address: stefano.marrone@unina2.it

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.04.195

Among the modern critical infrastructures, smart energy grids are getting a growing interest from many communities (academic, industrial and political) which foster the development of a more robust energy distribution infrastructure. Energy grids are a clear example of critical cyber physical social systems since their equilibrium can be perturbed not only by cyber and physical failures/attacks[1] but also by economical and social crises as well as changes in the consumption profiles[2].

Since both technical and economical threats can not be predicted and known a-priori, the only way to build robust systems is to embed in the system itself an ability to evolve and adapt to the changes of its operational environment. Evolutionary computing has demonstrated its effectiveness in supporting the development of robust and intelligent systems: some research studies proposed combinations with formal and quantitative models which are a primary tool in development and assessment of critical systems[3,4]. Most of these approaches limits this combination to the design phase of the system lifecycle.

The paper illustrates a practical framework supporting the run-time evolution of the control logic inside HEMSs. By combining the modeling and analysis capabilities of Fluid Stochastic Petri Nets and the flexibility of Genetic Programming, this approach can be used to adapt the control logic of the Smart Meters (the central component of HEMSs) to the changes of the structure and functionalities of Smart Buildings as well as of the operational environment.

This class of systems is also constrained to a further requirement: while the main objective of the evolution is to guarantee a proper responsiveness level to unpredicted changes, the fulfilment of the user's requirements about a energy-sustainable home forces to cope with preservation identity of the Smart Meter during its evolution. SM's control logic evolution is ever into the tracks of user's requirements. To cope to this aim, we must frame this work into the more generic context of antifragility and its related themes[5]. According to the formula *Antifragility = Elasticity + Resilience + Machine Learning*, this paper tackles with all of these topics by addressing a flexible evolving and experience-learning approach.

This paper describes a further step into a research work which counts some other publications: in[6], a full FSPN model-based approach for the energy profiling of smart buildings is described while its extension into a model-driven context as in[7]. This work starts from modelling languages introduced in these papers and focuses on the evolution within the identity preservation of control logics: hence, the theoretical and practical frameworks developed in the previous papers will be here briefly described since it is used to implement the proposed approach.

The paper is structured as follows: Section 2 briefly presents needed background as well as framing the paper into its scientific context. Section 3 shows the model-based approach at-a-glance while Section 4 focuses on evolving control strategies. Section 5 summarises the current state of the work and draws future research directions.

2. Background and Related works

Choosing an optimal energy consumption policy can also be pursued by transferring this responsibility to a device within a HEM. Authors in[8] propose and analyse different solutions where the awareness in the energy consumptions is possible if the ICT infrastructure, related to the energy domain, is designed in an energy-aware manner, too. Our approach has a similar point of view because it aims to analyse policies adopted by SMs in order to obtain good trade-offs between energy consumptions and costs.

Some works have been focused on similar approaches. For example, the work described in[9] suggests a scheduling algorithm for SMs able to balance energy consumption within a neighbourhood on a shared electrical channel; the work also proposes a billing model, separated from the scheduler, encouraging the adoption of this shared mechanism. The work in[10] analyses a smart charging system that uses a local energy storage to provide savings in customer bill by stocking energy during low-cost periods.

Power Grids and Smart Grids modelling and simulation tools are also receiving a growing interest in both academic and industrial settings. GridLab-D[11] and EnergyPlus[12] examine the energy consumption with a focus on heat generation and thermal load of a building. PowerMatcher Simulation Tool[13] assumes that the price of energy varies throughout the day, but it only considers a static energy demand without allowing to model energy consumption patterns. The Smart Home Simulator[14] introduces the capability to model a smart home configuration, to set the energy workload starting from real-world data and to model how a Smart Meter logic behaves in this configuration.

Model-based approaches can be divided into two categories: combinatorial models and state-space models. The first category does not fit in our purposes due to the complex behavioural mechanisms underlying a Smart Grid

infrastructure: therefore our choice is to adopt a state-space formalism. An exemplary work using state-space models is[15] where a modelling framework based on Stochastic Activity Networks is used to evaluate quantitatively the effects of malfunctions in electric power systems. On the same wavelength is the work described in[16] in which a Petri Net model is used to address fault diagnosis within the Distribution domain of a Smart Grid.

Petri Nets are more explicitly addressed in the modelling and the evaluation of Smart Grids in other scientific works addressing both dependability[17] and security aspects[18]; at the best of our knowledge very few works use Petri Nets for energy related purposes[19]. Ordinary Petri Nets however does not fit well the continuous nature of the energy flows. As in the work described in[20] we adopt the FSPN formalism[21], in which thanks to the timing features and the presence of continuous places, marked by a continuous positive real, is possible to better describe energy supply domain.

3. Model-based optimisation of HEMSs

This Section defines the overall methodology and the enabling framework for the automatic adaptive construction of the control logic for a Smart Building. Essentially a control logic is an algorithm that takes as input (1) the energetic balance of the building (i.e., the difference between the intake from renewable energy sources and domestic appliances), (2) the quantity of the energy stocked into battery and (3) the economical conditions of the market (i.e., selling and buying prices per energy unit). As output, such algorithm decides to perform some actions on energy as: sell (to a customer), buy (from a supplier), store (into battery), take (from battery). The adaptive mechanism proposed in this paper has the objective to select a control logic that is capable of fulfilling the requirements of the user as well as minimising the overall costs of the Smart Building i.e., to minimise the bill of the electric furniture of the building.

The main aspects to accomplish this task are *modelling*, *evolution*, *evaluation*, *instantiation* and *monitoring*; Figure 2 shows how they are organised into a workflow of activities:

- *modelling*: when the process starts, the first performed activity is the definition of a configuration model that captures all the devices of the Smart Building to control (**Update System Model**) generating a **Home Model**. This activity is invoked also as a consequence of changes of some external conditions (reconfiguration of the structure of the Smart Building, as a device addition, and/or changes in the economical operational environment, as the decrease of the energy produced by a supplier);
- *evolution*: when the Home Model or the needs of the final users have changed, the algorithm managing energetic and economical resources (the **Control Model**) is not yet applicable and it must be updated;
- *evaluation*: a new tentative Control Model has been produced but its effectiveness has to be measured in order to verify it now fits with changed conditions and it perform better than before;
- *instantiation*: when a candidate Control Logic fits with the new environment, it replaces the old one being instantiated and translated (**Running Control Logic**) into the concrete language interpreted by the Smart Meter;
- *monitoring*: finally, a monitoring activity is performed to watch the evolution of relevant variables and to reactivate the entire process when some of these monitored variables raises some alarm. An example is constituted by monitoring the overall expense of the Smart Building, an overcome of a threshold set by the user invalidates the current Running Control Logic and opens for the research for a more efficient one.

These activities can be automated according to the reference architecture shown in Figure 2.

Modelling. Modelling facilities are supported by the definition of proper modelling languages and manipulation toolset according to model-driven principles. The model of an SB is constituted by two views: the **Home Model**, related to the physical architecture of the SB, and the **Control Model** where such entities concur in the definition of a pseudo-program expressed in a well defined language. These modelling languages are introduced in[7]: in brief, a single domain model is defined considering the two views, then the first view is implemented by instantiation a graphical language (Smart Grid Modelling Language - SGML), for the second a EBNF grammar is defined to implement a textual language (Smart Meter Programming Language - SMPL).

Evolution. The evolution is made by an **Evolution Manager** that is in charge of starting from the current solution, changing its structure to produce new candidate control logics, calling evaluation methods for each candidate solution

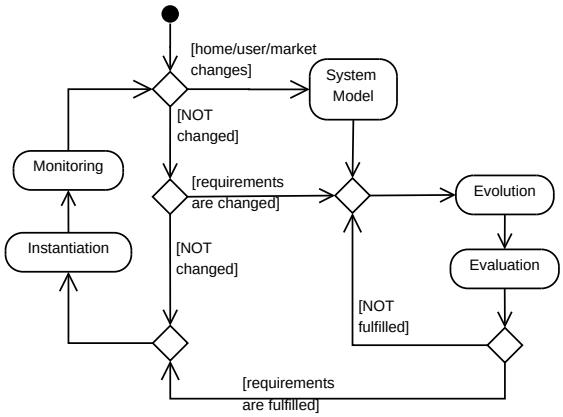

Fig. 1. Process model.

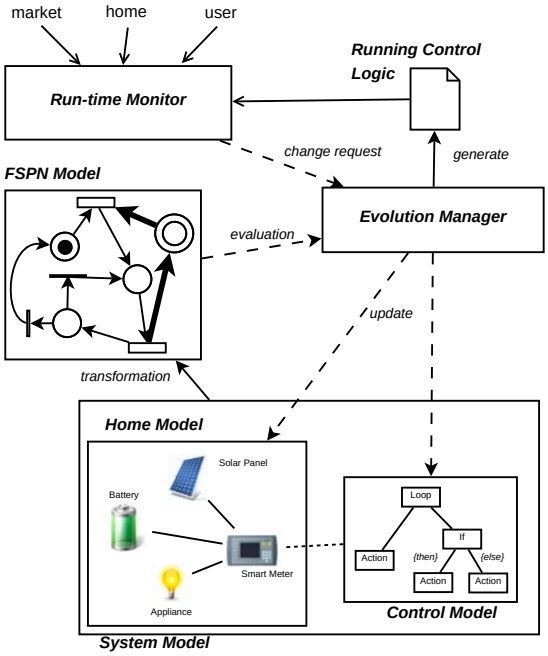

Fig. 2. Overview of the approach.

and, in the end, substituting the current one with the best (i.e., most energy saving) fitting (i.e., fulfilling user's requirements) among the candidates. This component is detailed in Section 4.

Evaluation. Evaluating a candidate Control Model means to evaluate, given a Home Model, the money spent and how much the logic fulfils the user's requirements. As explained in [6], Fluid Stochastic Petri Nets (FSPN) [21] is a formalism that fits into this scope due to its ability to cope with hybrid (both discrete and continuous) concepts. By translating both the SGML and SMPL models constituting the System Model into a **FSPN Model**, a simulation analysis of the entire model can be performed by proper tools. A energy/money consumption profile.

Instantiation. The instantiation of the chosen Control Model into the Running Control Logic is made by a simple serialisation of Control Model as well as by a model-to-text transformation according to the model-driven practises.

Monitoring. The **Run-time Monitor** is a software tool that watches the temporal evolution of external variables. It watches: changes in price *market* exchange with external partners (official suppliers, other prosumers, etc.), *user* needs (i.e., if the user changes the Quality of Service or sets a new billing profile), changes in the *home* structure (e.g., due to an addition of a new device of the failure of an existing one).

4. The Evolution Manager

This Section gives the methodological and technical bases for the construction of the Evolution Manager. According to the overall approach in Figure 2, this component is invoked by the Run-time Monitor when meaningful changes occur in the home structure, in the market or in the user requirements.

The Evolution Manager is responsible of keeping the Home Model consistent. In brief, the Home Model is a high level model built upon the SGML domain specific modelling language where real objects as lamps, washing machines, solar panels, stocking batteries, etc. To this purpose, building such model by the Evolution Manager is an easy task since it receives from the Run-time Monitor. Some examples of these data are: consumption profiles of devices (i.e., how much energy a lamp consumes during the day), recharging profile of a battery (e.g., maximum capacity or energy

leakage) and energy production profiles (i.e., how much energy a solar panel produces during daylight). These data are collected directly from the Smart Devices or by measuring them in time.

The second task the Evolution Manager accomplishes is to produce a new Control Model starting from the existing one; differently from the first case, this is not a trivial task and can be reported to the general problem of automatic program synthesis[22]. This paper explores the possibility to generate such programs by means of well-known evolutionary algorithms and in particular by Genetic Programming (GP)[23]. In brief, GP techniques work on the same principles of the Genetic Algorithms (GAs) since they manipulate candidate solutions encoded in "chromosomes". This notwithstanding, difference between them is that the chromosomes in GAs are mainly structured as list of variables (binary or not binary encoded) while in the second case, the chromosome are structured as trees since this is a form programs could be easy manipulated.

In order to provide some useful details, Listing 1 reports a simple SMPL program of a Smart Meter: the program is constituted by a bootstrap section (`INIT`) in which all the commands given to the Smart Building are reset (`clean()`). Then, according cyclic semantics typical of control programs (`LOOP`), the program checks if the overall energy balance is active (it produces more energy than consumed): in this case if the overall energy bill (`BILL`) exceeds a threshold set by the user, is sell (`SELL`) the energy on the market. In the other case it simply store the energy in the local battery for future uses (`STORE`). Figure 3 depicts the Control Model of such program in terms of its Abstract Syntax Tree (AST) or, more precisely, using two ASTs one for the `INIT` section and one for the `LOOP` section.

Listing 1. A SMPL sample program.

```
APPLICATION domain {
  INIT:
    clean();
  LOOP:
    if (ENERGY > 0) {
      if (BILL > 50.0) {
        sell();
      } else {
        store();
      }
    } else {
      buy();
    }
    wait(3,min);
}
```

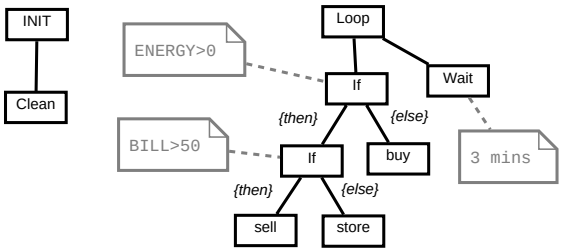

Fig. 3. Control Model of the sample program.

As in all the GA/GP approaches, proper mutation and crossover operators should be defined. The proposed mutation operators are:

- **addition**: a leaf is substituted by a new subtree;
- **deletion**: a subtree collapses into a leaf;
- **substitution**: can be seen as a consecutive application of deletion and addition;
- **local mutation**: while the structure of the subtree does not change, its data-related information mutate in order to produce a slightly different version of the program; this means that some GA-style mutation operators are applied to the data represented by the grey notes linked to an AST node. An example is constituted by increasing the waiting time (the '3 mins' label linked to the *wait* node).

Crossover operators are:

- **total**: two ASTs exchanges their INIT and LOOP sections, i.e., either INIT or LOOP section of the first moves to the second and viceversa;
- **partial**: the same as before but not on entirely section but on a section subtree.

5. Conclusions

This paper presented a model-based framework for the evolution of control strategies in smart building environments. At current state, the defined framework is partially implemented: while the definition of the languages for the Home Model and Control Model and the model transformations between such languages and FSPN have been completed in previous work, this paper focuses on Evolution Manager which implementation is an ongoing work.

First future research efforts will be oriented to the completion of the framework implementation by (1) testing the Evolution Manager, (2) implement the Monitor component. Final assessment of the approach would be accomplished first with a simulated environment and then with real devices and appliances.

References

1. Sridhar, S., Hahn, A., Govindarasu, M.. Cyber-physical system security for the electric power grid. *Proceedings of the IEEE* 2012; **100**(1):210–224. doi:10.1109/JPROC.2011.2165269.
2. Agency, I.E.. The impact of Financial and Economic Crisis on Global Energy Investment. Tech. Rep.; International Energy Agency (IEA); 2009. URL: https://www.iea.org/publications/freepublications/publication/impact.pdf.
3. Struck, S., Güdemann, M., Lipaczewski, M., Ortmeier, F.. Multi-objective optimization of formal specifications. In: *IEEE 14th International Symposium on High-Assurance Systems Engineering (HASE 2012)*. 2012, .
4. Hamada, M., Martz, H., Reese, C., Graves, T., Johnson, V., Wilson, A.. A fully bayesian approach for combining multilevel failure information in fault tree quantification and optimal follow-on resource allocation. *Reliability Engineering and System Safety* 2004;**86**(3):297–305. doi:10.1016/j.ress.2004.02.001.
5. De Florio, V.. Antifragility = elasticity + resilience + machine learning: Models and algorithms for open system fidelity. vol. 32. 2014, p. 834–841. doi:10.1016/j.procs.2014.05.499.
6. Gentile, U., Marrone, S., Mazzocca, N., Nardone, R.. Cost-energy modelling and profiling of smart domestic grids. *International Journal of Grid and Utility Computing (IJGUC), Inderscience* 2016;.
7. Gentile, U., Marrone, S., Mazzocca, N., Nardone, R.. μGRIMOIRE: a tool for smart micro grids modelling and energy profiling. *The Open Cybernetics & Systemics Journal* submitted to;.
8. Ricciardi, S., Palmieri, F., Fiore, U., Careglio, G., Santos-Boada, G., Sol-Pareta, J.. *Handbook of Green Information and Communication Systems*; chap. Towards Energy-Oriented Telecommunication Networks. Academic Press; 2013, p. 491–512.
9. Mohsenian-Rad, A.H., Wong, V., Jatskevich, J., Schober, R.. Optimal and autonomous incentive-based energy consumption scheduling algorithm for smart grid. In: *Innovative Smart Grid Technologies (ISGT), 2010*. 2010, p. 1–6. doi:10.1109/ISGT.2010.5434752.
10. Mishra, A., Irwin, D., Shenoy, P., Kurose, J., Zhu, T.. Smartcharge: Cutting the electricity bill in smart homes with energy storage. In: *Proceedings of the 3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet*; e-Energy '12. 2012, p. 29:1–29:10. doi:10.1145/2208828.2208857.
11. GridLAB-D, . Gridlab-d simulation software. 2011. URL: http://www.gridlabd.org/.
12. US Department of Energy, . Energyplus: Getting started with energyplus, in basic concepts manual - essential information you need about running energyplus. 2011.
13. PowerMatcher Smartgrid Technology, . Basic structure and agent roles. 2015. URL: http://www.powermatcher.net/get-technical/powermatcher-technology/basic-structure-and-agent-roles/.
14. Nysteen, M., Mynderup, H., Poulsen, B., Træholt, C.. Simulation tool for energy consumption and production: The development of a simulation tool for measuring the impact of a smart grid on a building. In: *Proceedings of ENERGY 2013*. IARIA; 2013, .
15. Chiaradonna, S., Di Giandomenico, F., Lollini, P.. Definition, implementation and application of a model-based framework for the analysis of interdependencies in electric power systems. *International Journal of Critical Infrastructure (IJCIP), Elsevier* 2011;**4**(1):24–40.
16. Calderaro, V., Hadjicostis, C., Piccolo, A., Siano, P.. Failure identification in smart grids based on petri net modeling. *Industrial Electronics, IEEE Transactions on* 2011;**58**(10):4613–4623. doi:10.1109/TIE.2011.2109335.
17. Zeng, R., Jiang, Y., Lin, C., Shen, X.. Dependability analysis of control center networks in smart grid using stochastic petri nets. *Parallel and Distributed Systems, IEEE Transactions on* 2012;**23**(9):1721–1730. doi:10.1109/TPDS.2012.68.
18. Chen, T., Sanchez-Aarnoutse, J., Buford, J.. Petri net modeling of cyber-physical attacks on smart grid. *Smart Grid, IEEE Transactions on* 2011;**2**(4):741–749.
19. Pandey, A.. *Petri Net Model for Smart Grid System*. Master's thesis; North Dakota State University; 2014.
20. Darienzo, M., Iacono, M., Marrone, S., Nardone, R.. Estimation of the energy consumption of mobile sensors in WSN environmental monitoring applications. 2013, p. 1588–1593. doi:10.1109/WAINA.2013.33.
21. Gribaudo, M., Sereno, M., Horváth, A., Bobbio, A.. Fluid stochastic Petri nets augmented with flush-out arcs: Modelling and analysis. *Discrete Event Dynamic Systems* 2001;**11**(1-2):97–117.
22. Srivastava, S., Gulwani, S., Foster, J.. From program verification to program synthesis. 2010, p. 313–326. doi:10.1145/1706299.1706337.
23. Koza, J.. Genetic programming as a means for programming computers by natural selection. *Statistics and Computing* 1994;**4**(2):87–112. doi:10.1007/BF00175355.