Adaptive Knowledge Representation for a Self-Managing Home Energy Usage System

Martijn Warnier[1], Marten van Sinderen[2] and Frances M.T. Brazier[1]

[1]Faculty Technology, Policy and Management
Delft University of Technology, The Netherlands
[2]Faculty of Electrical Engineering, Mathematics and Computer Science
University of Twente, The Netherlands
M.E.Warnier@tudelft.nl,
m.j.vansinderen@ewi.utwente.nl,F.M.Brazier@tudelft.nl

Abstract. Automated and efficient energy management has many potential benefits for producers and consumers of energy, and the environment. Focusing on energy management on the consumer side, this paper considers two forms of energy management: minimizing energy usage in single households and avoiding peaks in energy consumption in a larger area.

A combination of context aware and autonomic computing is used to describe an automated and self-managing system that, by analyzing context information and adapting to its environment, can learn the behavior of household occupants. Based on this information, together with user defined policies, energy usage is lowered by selectively powering down devices. By powering specific thermostatically controlled devices on or off energy can also be redistributed over time. This is utilized to avoid global peaks in energy usage.

The self-managing system reasons about context and other information and acts when required. This information is the knowledge with which it can adaptively reason, about to take to ensure efficient energy usage. This paper explores the requirements that hold for representing this knowledge and how the knowledge base can continuously and adaptively be updated: to be self-managing.

1 Introduction

Efficient energy management forms an important challenge in today's society as conventional energy sources become more and more scarce and more eco-friendly alternatives are not yet evaluated at a large scale. Automated approaches to efficient energy management are currently still limited and mostly used by large power consumers such as factories [10, 16, 17], but with the advancement of the Smart Grid [2] and other recent advancements in sensor networks [1, 19], such as cognitive networks [18], this changes rapidly.

This paper addresses this challenge at two different, and possibly conflicting, levels: (i) at the level of a single household, where the goal is to lower

energy consumption and (ii) at the level of group of households (a neighbourhood), where the goal is to lower peak usage in energy consumption. In [15] a new approach based on autonomic and context aware computing is introduced. This paper extends that approach. It explores the requirements for an adaptive knowledge representation in the context of a home energy management system. The system exploits sensor information to monitor electric appliances and their surroundings. Based on this environmental information the status of appliances is updated, influencing energy consumption.

The core of the home energy management system is formed by a rule based system [5]. Such systems are typically very deterministic in nature, always producing the same outcomes for the same input: a more dynamic approach is required for self-management. Therefore, the main challenge addressed in this paper is the question how, in the context of a home energy management system, a knowledge base can continuously and adaptively be updated: to become self-managing.

The remainder of this paper is organized as follows: in the next section the approach introduced in [15] is summarized. Section 3 discusses requirements for the self-managing home energy system and Section 4 outlines an approach to adaptive and self-managing knowledge representation. The paper ends with a brief discussion and conclusions.

2 An Autonomic and Context Aware Home Energy Management System

A new green computing approach on the consumer side (demand side management) is proposed in [15]. The proposed system considers two forms of energy management: minimizing energy usage in single household and avoiding peaks in energy usage for a larger residential area. A basic architecture is proposed to achieve this goal. A service oriented framework, based on the complementary approaches of autonomic computing and context aware computing, is introduced. Context information is continuously used to analyze the energy requirements of a household. Based on this information, the home energy management system determines whether and how to influence the energy consumption of individual devices. By selectively powering thermostatically controlled devices –such as fridges, air conditioning units, electrical heating– on or off, energy consumption is redistributed over time [12, 15] avoiding peaks in energy usage. Such devices make up around 25% of the total energy consumption in the USA [9].

Residences of the household can add their own preferred energy usage profile. The resulting service oriented framework reduces energy usage in households in an intelligent and user friendly manner. Actual consumption can be analyzed with respect to consumption constraints set by the system. If either consumption constraints are (close to being) violated or user needs are unnecessarily high, then energy consumption should be decreased and corresponding control actions need to be exercised on the appliances of interest.

Fig. 1 illustrates the proposed architecture of the home energy management system.
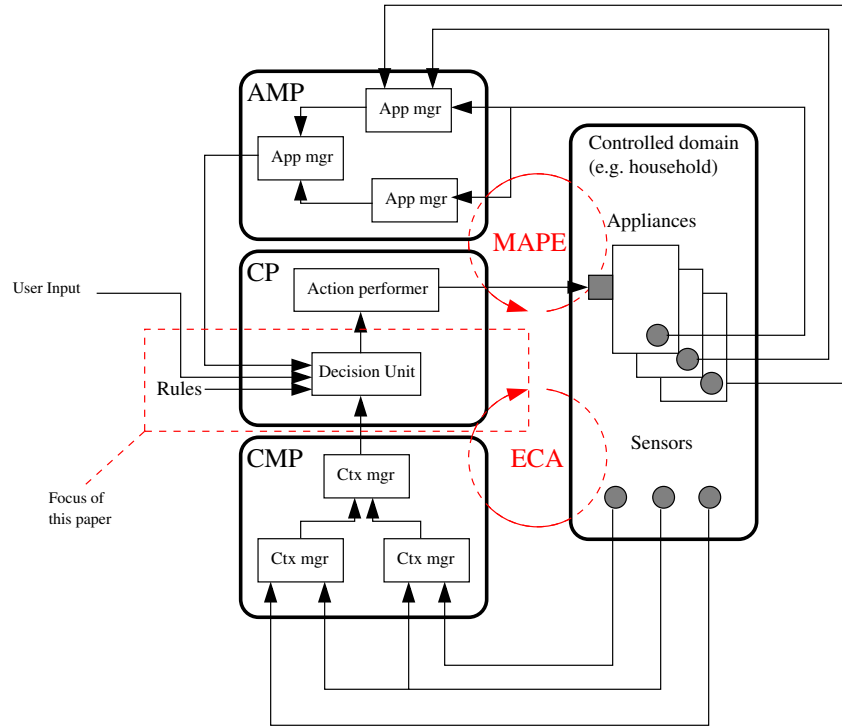
Fig. 1. The basic architecture of the home energy management system. Adapted from [15].

The MAPE (monitor, analyze, plan, and execute) control loop from autonomic computing [6] is adapted to be used to analyzing energy consumption of electrical appliances. Considering a pool of electrical appliances that are instrumented to allow monitoring and control. The monitoring consists of measuring the energy consumption of these appliances. The measurements are fed to a control process (the Appliances Management Process or AMP in Fig. 1), which interprets these as the actual consumption, and compares the consumption with the consumption constraints (in the Decision Unit in Fig. 1). If, as a result of this analysis, it is decided that control actions are needed, an action plan is produced. The action plan is derived with an algorithm that considers time-shifting of the active state of appliances. Subsequently, the plan is executed (in the Action Performer in Fig. 1) by performing the indicated control actions on the selected appliances. Figure 1 illustrates the application of the MAPE control

loop in the top of the figure. The Decision Unit, which forms the focus of this paper (see Fig. 1) is discussed in detail in Section 4.

With regard to analyzing consumer needs, the event-control-action (ECA) pattern from context-aware computing [4] is used. Figure 1 illustrates the application of the ECA pattern at the bottom of the figure. The environment of the appliances is considered. It is assumed that this environment is instrumented with sensors that are able to measure relevant conditions. For example, measurements may be used to determine context changes or situations, such as the presence of one or more persons in the house or in a particular room, the activity mode (sitting, walking, sleeping) of a person, or a person entering or leaving the house. Context situations and changes can generally not be directly or reliably measured by a single sensor. A context management process (the CMP in Fig. 1) is responsible for producing events that indicate the occurrence of a context situation or change, based on reasoning which potentially involves sensor data from several sources. Events are fed to a control process, which applies them in rules to determine actions related to perceived needs. For example, if nobody is in the house, a rule may establish the action to set the preferred value of the heating at 15 degrees Celsius. Whether the actions are really required depends on the supported needs. For example, if the preferred value of the heating is already set to 15 degrees Celsius, no action is needed. The comparison of the perceived and supported needs leads to an action plan, which, if not empty, is subsequently executed by performing the indicated control actions on the selected appliances.

A service-orientated architecture (SOA) is used to implement the approaches outlined above, see [15] for more details.

3   Requirements for Knowledge Representation

The high level goal of the energy management system is twofold: On the one hand, on the household level, the system should *minimize the total energy consumption* of the household, within fixed boundaries set by the household owner. On the other hand, on the neighborhood level, the system should minimize fluctuations in energy demand, i.e., keep the energy consumption of the whole neighborhood as constant as possible. Note that these goals can be conflicting. Energy providers can prioritise one over the other by providing (monetary) incentives. The energy management system's high level goals provide the first system's requirement:

1. the system should be flexible enough to optimize either of the two high-level goals of the system: minimize local energy consumption or minimize global fluctuations in energy consumption

Moreover, the system needs to be highly adaptive, in particular:

2. the system should be able to adapt its behavior at runtime and change the high-level goal, depending on input from the environment

Finally, users should be able to customize the systems to their specific needs, setting limits to the adaptive behavior of the system:

3. users, i.e., home owners, should be able to customize the system to their specific needs.

Note that the last requirement is the most important one. This can potentially limit the adaptive behavior of the systems. However, it is crucial that users should be able to override the energy management system, even if this means that, for example, the air condition is set to maximum in each room. No users will allow a fully autonomous system to manage their energy usage. This issue is discussed further in Section 5.

4   Towards Adaptive and Self-Managing Knowledge Representation

The main self-managing component of the energy management system is formed by the *Decision unit* shown in Fig. 1. This unit has to (autonomously) decide how appliances are adjusted to meet the goals of the system. Fig. 2 shows the decision unit in detail.
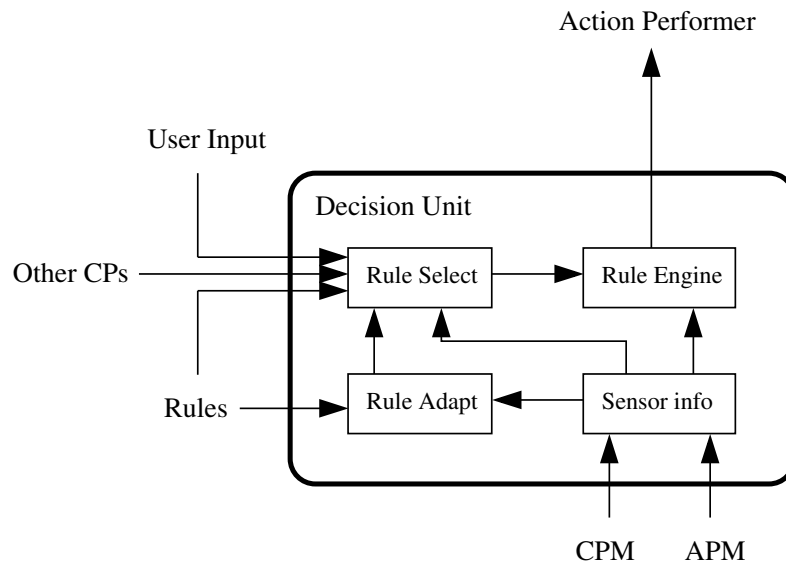
Fig. 2. The decision unit from Fig. 1.

The decision unit as a whole takes input from four different sources, namely (i) sensor input, (ii) user input, (iii) other control process units (CPs) at other households and (iv) rules. The sensor input comes from the Context Management Process (CMP) and the Appliances Management Process (AMP). These units process the sensor information and provide (aggregated) context and appliances status information which forms the basis for the adaption process. The user input gives the home owner the opportunity to override the (autonomous) decision unit, for example, indicating that the air conditioning unit in a room cannot be put off.

To meet the goal of the system, global (at a neighborhood level) minimization of the fluctuations in energy consumption, the control processes (CPs) in individual households need to communicate with each other. The CPs can be organized in a virtual tree overlay [13] and work together to meet this goal, for example using the approach outlined in [12, 14].

The heart of the Decision Unit is formed by a rule based system which consist of Rule Select, Rule Adapt and Rule Engine components, see Fig. 2. It is assumed, since the application domain is known and unlikely to change rapidly, that the knowledge base of the home energy system uses a fixed ontology, i.e., all rules are formulated in the same (fixed) language. Rule based systems are traditionally very deterministic systems. They consist of rules of the form shown in Example 1 below:

Example 1 (*Rule base system*)

$$\text{matching condition } 1 \Rightarrow \text{effect } 1$$
$$\text{matching condition } 2 \Rightarrow \text{effect } 2$$
$$\ldots\ldots$$
$$\text{matching condition } n \Rightarrow \text{effect } n$$

$\triangle$

By default the ordering of the rules defines the (operational) semantics of the system. Rules are evaluated in order, and the first rule with a matching condition is executed, i.e., later rules that might match are discarded. Execution of a rule leads to an effect. In this case something like altering the status of an appliance, i.e., lowering the temperature of a fridge. Since the environment will change continuously different rules will be executed over time. However, given the same conditions the same rules will be executed, making the system completely deterministic (static). To make the system more adaptive to its environment it should evolve over time, to meet the demands of a specific household. There are several options that can be used to make the rule based system adaptive, these include: (i) **load specific set of rules based on the environment**, for example in the *rule select* unit a specific set of rules can be loaded that has as goal to minimize global fluctuations instead of minimizing the local energy use

of a household. (ii) **evolving rules**, the rules can be changed based on genetic algorithms [3] or a neural network [7]. (iii)**weighted rules**, in this approach all rules are weighted. Instead of executing the first matching rule all rules are selected and the rule with the highest weight is executed. Fuzzy logic [8] like approaches can be used for this. (iv) **hybrid approach**, combine some of the options above.

Evolving rules (item 2 above) effectively is typically difficult. Neural networks and genetic algorithms try to merge and combine existing rules to produce new (better) ones. However, a fitness function is required to determine if a newly generated rule is better then existing ones. Finding a suitable fitness function is typically very hard. Therefore this alternative is not further studied in detail here. Instead, a hybrid approach that combines item 1 and 3 above is explored. The proposed adaptive rule based system has the following properties:

- rules are weighted
- rules are bundled in a set, called the *device set*, per device
- rules can be added and removed per set
- sets can be added and removed to the *rule base*, the active rule sets that are used by the rule engine

For each device there is an associated device set, consisting of weighted rules, that determines the (adaptive) behavior of the device. Consider, for example, the following three rules, shown in Example 2 below, that are part of the device set for controlling the ac-unit in the master bedroom:

Example 2 (*Weighted Rules controlling an AC unit*)

$\ldots\ldots$
(**0.3**) #people in room ≥ 1 & $t > t_{\max}$ & AC $=$ off $\Rightarrow$ AC $=$ on & cooling=10
(**0.7**) #people in room ≥ 1 & $t > t_{\max}$ & AC $=$ off $\Rightarrow$ AC $=$ on & cooling=2
$\ldots\ldots$
(**0.5**) #people in room $= 0$ & AC $=$ on $\qquad\qquad \Rightarrow$ AC $=$ off
$\ldots\ldots$

$\triangle$

All three rules have a weight (0.3, 0.7 and 0.5 respectively). The first two rules share the same pre-conditioning, that evaluates to the value true if there is more then one person in the room, the ac-unit is turned off and the current temperature (t) is higher then some predetermined temperature ($t_{\max}$, for example 25 degrees Celsius). However, the result of executing the rules is different: executing the first rule results in turning the AC unit at setting 10, in the second case the AC unit is turned on at setting 2. A higher setting leads to a faster cooling of the room, but also a higher (at least initial) energy consumption. Since both these

rules share the same precondition, the one with the highest weight (the second rule with weight 0.7) is executed.

By adapting the weights of rules, different energy consumption patterns emerge. Again looking at the two rules mentioned above, in the current situation energy consumption is low over a longer period. If the weights of the two rules are swapped, the result would be a higher energy consumption, but for a longer period. In effect, the first mechanism that is used to adapt the system is this changing (adapting) of the weights of the rules. This makes it is possible to adapt the reaction of the system to a specific situation.

Note that the third rule (with weight 0.5) has a different pre-condition. It evaluates to true if there is no one in the room and the AC unit is on. The effect of executing the rule is that the AC unit is turned off.

The *Rule Select* unit (from Fig. 2) loads selected device sets into the rule engine. By periodically loading different rules into the rule engine, i.e., by changing the *rule base*, the system adapts to its environment. Based on input from the environment different (possibly conflicting) high level goals can be met by different rule-bases. This loading of different device sets provides the second adaptation mechanism of the system.

User input, i.e., from home owners, can be mapped easily to rules that deal with a specific appliance, for example, the air conditioning unit in the bedroom. The rule is seen in the Example 3:

Example 3 (*User generated Rule controlling an AC unit*)

$$(\mathbf{1.0})\ \text{true} \Rightarrow \text{AC} = \text{on}$$

$$\triangle$$

User generated rules should always evaluate to true (hence the pre-condition in the rule above). Also note that such rules should always be loaded (unless specifically cancelled by the user) and should have high weights (in this example, the maximum value of 1.0) to ensure that user generated rules are executed. Finally, note that the weight of such rules should, in principal, not be adapted by the system.

The system provides two adaptation mechanisms: weight adaptation which is handled by the adapt rule component and selective device set loading (and unloading), which is handled by the rule select unit. Separating these mechanisms has the advantage that its easier to reason about adaptation policies at a higher (strategic) level. This is left for future work, as are specific rule adaptation policies.

In summary, the Decision Unit takes input from the user and other CPs. Based on input from the environment (sensor info), weighted rules are adapted and selected. The rule engine selects all matching rules and chooses the ones that have the highest weight per appliance. These are then send to the action performer which adapts the status of the appliances. This whole process is repeated periodically.

5 Discussion and Conclusions

This paper discusses an approach and architecture for a home energy system based on an adaptive and self-managing knowledge representation. The system is based on a weighted rule based system that adapts continuously to its environment. One of the main challenges of this system is to meet its different, possibly conflicting, goals. And while the current architecture should make this possible it remains to be seen if these goals are not too conflicting to be unifiable in practice. It might be necessary to drop the goal of lowering the global fluctuations in energy consumption to meet the user's preferences and minimize the local energy consumption of the household. Simulations and/or experiments should provide more insight on this issue. This is left for future work.

Another issue is how to scale this up to collections of households. A hierarchical structure could be used in which the architecture can be repeated at different levels of granularity. For example, a household has appliances as units that are being controlled; an apartment has living units as the units that are being controlled; a city block has apartments as units being controlled; etc. Finding the correct clustering of households [11] that are controlled by one processing unit forms another challenge.

From a technical perspective it is not very difficult, with the proposed architecture, to force control processes in different households to cooperate to reduce peaks in (global) energy usage. However, this might lead to some considerable discomfort with home owners, for example if they can use their air conditioning unit at the maximum setting, because a global reduction in energy consumption is required. Monetary incentives, provided by energy producers who benefit from reduced peak usage, might help lessen the discomfort of the home owner, as would specific policies set by local governments. However, if this will be an acceptable solution remains to be seen. This issue is further outside the scope of this paper.

A related issue is if an (semi) autonomous home energy management system will be accepted by users. However, since there is both a monetary incentive (energy usage is lowered which in turn leads to a lower energy bill) and since users can override the behavior of the system this is probably less of an issue then the one discussed above.

References

1. I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.

2. P. Clark W. Gellings. *The Smart Grid, enabling energy efficiency and demand response.* The Fairmont press, 2009.

3. A. Corcoran and S. Sen. Using real-valued genetic algorithms to evolve rule sets for classification. *ICEC*, 94(2405):120–124.

4. P. Dockhorn Costa, L. Ferreira Pires, and M. van Sinderen. Concepts and architectures for mobile context-aware applications. In *Handbook of research on mobile multimedia*, Information Science Reference. Hershey, New York, 2008.

5. F. Hayes-Roth. Rule-based systems. *Communications of the ACM*, 28(9):932, 1985.

6. IBM Corporation. An architectural blueprint for Autonomic Computing. White paper, 2005.

7. E. Keedwell, A. Narayanan, and D. Savic. Evolving rules from neural networks trained on continuous data. In *The 2000 Congress on Evolutionary Computation CEC 00*, pages 639–645, 2000.

8. G. Klir and B. Yuan. *Fuzzy sets and fuzzy logic: theory and applications.* Prentice Hall Upper Saddle River, NJ, 1995.

9. P. Mazza. The Smart Energy Network: Electrical Power for the 21st Century. *Climate Solutions*, 2002.

10. A. Middelberg, J. Zhang, and X. Xia. An optimal control model for load shifting - With application in the energy management of a colliery. *Applied Energy*, 86(7-8):1266 – 1273, 2009.

11. E. Ogston and F. M. T. Brazier. Apportionment of control in virtual power stations. In *In the proceedings of the international conference on infrastructure systems and services 2009: Developing 21st Century Infrastructure Networks*, 2009.

12. E. Pournaras, M. Warnier, and F. M. Brazier. Local agent-based self-stabilisation in global resource utilisation. *International Journal of Autonomic Computing (IJAC)*, 2010.

13. E. Pournaras, M. Warnier, and F. M. T. Brazier. Adaptive Agent-based Self-organization for Robust Hierarchical Topologies. In *ICAIS '09: Proceedings of the International Conference on Adaptive and Intelligent Systems.* IEEE, September 2009. (to appear).

14. E. Pournaras, M. Warnier, and F. M. T. Brazier. A distributed agent-based approach to stabilization of global resource utilization. In *Complex, Intelligent and Software Intensive Systems, International Conference*, pages 185–192, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

15. B. Shishkov, M. Warnier, and M. van Sinderen. On the Application of Autonomic and Context-Aware Computing to Support Home Energy Management. In *The proceedings of the 12th International Conference on Enterprise Information Systems (ICEIS 2010)*, 2010.

16. M. Stadler, W. Krause, M. Sonnenschein, and U. Vogel. Modelling and evaluation of control schemes for enhancing load shift of electricity demand for cooling devices. *Environmental Modelling & Software*, 24(2):285 – 295, 2009.

17. G. Strbac. Demand side management: Benefits and challenges. *Energy Policy*, 36(12):4419 – 4426, 2008. Foresight Sustainable Energy Management and the Built Environment Project.

18. R. Thomas, D. Friend, L. DaSilva, and A. MacKenzie. Cognitive networks: adaptation and learning to achieve end-to-end performance objectives. *IEEE Communications Magazine*, 44(12):51–57, 2006.

19. F. Zhao and L. Guibas. Wireless sensor networks. *Communications Engineering Desk Reference*, page 247, 2009.