

PANARETOS, A., CORSAR, D. and VASCONCELOS, W.W. 2019. Minimality and simplicity of rules for the internet-of things. In Lujak, M. (ed.) Agreement technologies: revised selected papers from the 6th International conference on agreement technologies (AT 2018), 6-7 December 2018, Bergen, Norway. Lecture notes in computer science, 11327. Cham: Springer [online], pages 64-72. Available from: https://doi.org/10.1007/978-3-030-17294-7_5

Minimality and simplicity of rules for the internet-of-things.

PANARETOS, A., CORSAR, D., VASCONCELOS, W.W.

2019



Minimality and Simplicity of Rules for the Internet-of-Things^{*}

Athanasios Panaretos¹, David. Corsar²[0000-0001-7059-4594], and
Wamberto W. Vasconcelos¹[0000-0001-5090-7581]

¹ Dept. of Computing Science, University of Aberdeen, Aberdeen, U.K.

² School of Computing Science and Digital Media, Robert Gordon University,
Aberdeen, U.K.

athanasios.panaretos.17@aberdeen.ac.uk, d.corsar1@rgu.ac.uk,
w.w.vasconcelos@abdn.ac.uk

Abstract. Rule-based systems have been increasing in popularity in recent years. They allow for easier handling of both simple and complicated problems utilising a set of rules created in various ways (e.g., manually, or (semi-) automatically, via, say, machine learning or decision trees) depending on the situation. Despite their usefulness however, there are still improvements to be made. Knowledge representation technologies have been available for a long time and provide the means to represent domains formally and correlate entities in those domains. They also allow for ontological reasoning that can take advantage of such connections between entities. These techniques can be useful when applied on rule-based systems in order to improve the quality of rules and, hence, overall system performance. We describe and implement an approach to refine rules used in Internet-of-Things scenarios using knowledge representation and reasoning. The proposed solution uses ontological reasoning on the preconditions and postconditions of rules as it aims to reduce the total amount of rules in a system and simplify them.

Keywords: rule-based systems · internet-of-things · knowledge representation · ontological reasoning.

1 Introduction

The fast-spreading Internet-of-Things technology is creating massive changes in the way people interact with devices, but also devices interact with each other. In this quickly changing field, automation is becoming the key element, taking control from humans and giving it to machines when it comes to daily tasks.

Rule-based systems have been used for quite a while to regulate societies of agents. They are ideal for Internet-of-Things applications, however, they do have some issues. Firstly, the constant creation of rules to deal with different situations

^{*} This research is partially sponsored by the EPSRC grant EP/P011829/1, funded under the UK Engineering and Physical Sciences Council Human Dimensions of Cyber Security call (2016).

can lead to having far too many of them. Therefore, an issue of minimality of rules has arisen. Secondly, the attempt to minimize the rules in a system can result in a small set of rules that is however very complex and thus harder to maintain. Hence, there is a need for simplicity.

This paper presents a system which can tackle the issues at hand. The system is able to process events of Internet-of-Things scenarios and, along with a knowledge representation model of the domain, generate rules based on them. Furthermore, it refines these rules and through an exhaustive evaluation process it guarantees to maintain the system's accuracy while reducing the amount of rules residing in it. Explicit knowledge representation of all the required concepts is used to support the refinement process of the preconditions and postconditions of the rules.

2 Background and Related Work

Internet-of-Things has been identified as one of the emerging technologies in IT [4]. Despite the constant evolution of this technology, its main point, use of sensors and actuators based on knowledge without human intervention, remains the same [4]. This lack of solid definition of Internet-of-Things allows it to extend into multiple fields. The application of this technology ranges from personal to national making it quite efficient. Its current growth can be attributed to the adaptability of the technology itself and the many capabilities it grants its users.

Our approach makes use of explicit representation of knowledge. An ontology is an explicit specification of a conceptualization, that is the set of all the objects, entities and concepts and the relationships between them, of a domain [3]. Over time there have been a lot of successful attempts to generalize different situations and create ontological models that can satisfy a wide range of cases. The result is widely used models which greatly ease the process of inserting an ontology into any kind of system. Some relevant to Internet-of-Things are the Semantic Sensor Network³, Open Digital Rights Language⁴, Smart Appliances REference⁵ and Smart Energy Aware Systems⁶.

A closely related area of research is normative multi-agent systems. In human societies, norms have played an important role in governing the behavior of the individuals in a society [2]. Even though norms are essential in agent societies, they are not easy to synthesize. There are two approaches that deal with the problem of norm synthesis, offline and online. The offline approach tackles the problem by synthesizing all the norms required for a system during its design time. This technique however requires complete knowledge of the situation that the agents in the system will be facing.

The online approach on the other hand synthesizes norms during runtime. This gives it the advantage of not requiring complete knowledge of the situation

³ <https://www.w3.org/TR/vocab-ssn/>

⁴ <https://www.w3.org/TR/odrl-model/>

⁵ <https://sites.google.com/site/smartappliancesproject/ontologies/reference-ontology>

⁶ <https://ci.mines-stetienne.fr/seas/index.html>

on design time which also implies that the norms that will be created over time will adjust to the conditions of the system as they keep changing [6]. It is worth mentioning that norm emergence has been gaining popularity lately. It is an online norm synthesis approach that allows agents to communicate with each other, synthesize norms themselves and decide on which ones will be used by the entire agent society. There are two related research work our project borrows from, namely IRON and its predecessor BASE. They both make use of a Case Based Reasoning mechanism to deal with conflict situations. Their differences are that IRON has a norm evaluation method and a norm generalization and specialization operator, which improve its performance.

3 Architecture and Implementation

The goal of the system is to process a log of events of Internet-of-Things scenarios and knowledge representation models in order to generate norms. The system will also have to be able to be integrated in different domains with only the input changing for each different adaptation. The rules, once generated, will be processed by the system and refined through generalizations and specializations of the ontological counterparts of the preconditions and postconditions. The output will be a set of rules that will be aiming to achieve minimality and simplicity.

Based on the system objectives, we can describe the several functional requirements that are inferred.

- FR-1** The system will formulate rules based on an initial log of events and make them available in a specific format – The rules created will therefore be directly linked to the input log rather than abstract rules based on the Internet-of-Things scenario.
- FR-2** The system will refine the rules by using a generalization and a specialization operator – These two operators will be able to make the rules move towards a more uniform form that will enable their merge.
- FR-3** The system will evaluate the entire normative system after each operator action – This ensures that the norms that have been altered are kept only if they are efficient in their new form.
- FR-4** The system will post-process the rules further to aim at achieving properties such as minimality and simplicity.

The scenario the system is applied on is a smart house that operates based on Internet-of-Things technology. All the sockets of the house have their voltage measured constantly which enables the complete knowledge of operation of every device. The system built, will be offering a set of norms to the occupants of the house that if followed will have beneficial effects to them such as reduced electricity cost and power conservation.

The data set⁷ used is a result of a research project of the Distributed Systems Group of the Computing Science department of ETH in Zurich⁸. It is referred

⁷ http://rossa-prod-ap11.ethz.ch/delivery/DeliveryManagerServlet?dps_pid=IE594964

⁸ <https://www.vs.inf.ethz.ch/>

to as the ECO (Electricity Consumption and Occupancy) data set and it is a comprehensive open-source data set for non-intrusive load monitoring and occupancy detection research. It has information of 6 households for a period of 8 months with readings every second on all sockets of the house through smart meters. It additionally holds occupancy information through the use of a tablet computer in the households. [1] and [5] analyzed it in detail but due to its open-source nature it is useful to any other researcher on the field as well.

3.1 Architecture

It is important to distinguish the different components of the system so as to understand its functionality. We introduce all the main components as well as the initial inputs.

Input data – is the data on which the rules will be based and include tagged undesirable system states.

Ontologies – in the system refer to the knowledge representation of the devices as well as of the preconditions that define the rules.

Rules generator – generates rules according to data that has been given as input. The rules are created by picking at random elements from the ontologies that represent the preconditions and postconditions. If the generated rule can be applied to an undesirable state it is kept otherwise it is discarded.

Rules post-processor – receives the rules and generalizes or specializes them using the corresponding operator. Further aspects of ontological DL reasoning could be used to expand the concept hierarchies, this would give a richer set of generalisations / specialisations for use by the algorithm, but would not fundamentally alter the described approach.

Rules evaluator – evaluates the effectiveness of the generalized rules compared to their previous form. Specifically, it compares the amount of cases the rule got triggered before and after its generalization.

Rules merger – merges the generalized rules to reduce their total size.

3.2 Format of rules

To understand our implementation better we introduce the format of the rules in the system:

```
condition:day/precondition_1/.../precondition_X;  
action:device_1,action_1,...,device_X,action_X;  
group:groupName
```

In the condition part of the rule, the day refers to the day of the week while the preconditions refer to the time the rule will be applied. In the action part, the devices refer to any of the known devices in the house, while the actions, which have the value of 0 or 1 indicate whether the device should be turned off or on. Lastly, the groups, namely Cheap_Rate and Premium_Rate, are used to separate electricity cost depending on the time of the day. We also note that in our system we represent rules as trees and refer to nodes of these trees when

performing actions between rules. For the sake of a simple example, let us assume that there are only 2 devices in the bedroom, a TV and a lamp. Accordingly, 2 rules have been generated:

```
condition:Monday/2330;action:TV,0;group:Cheap_Rate
condition:Monday/2335;action:Lamp,0;group:Cheap_Rate
```

When generalized and merged these 2 rules can be turned into a single one of the form `condition:Monday/Late_Night;action:Bedroom,0;group:Cheap_Rate`.

3.3 Ontologies in our system

In the implementation we use as many ontologies as the preconditions and postconditions we want to apply ontological reasoning on. In this case therefore, there is one ontology for the preconditions that is used to represent time and another for the postconditions which is used to represent devices. The time ontology splits up the time in different groups based on both the time of the day and the electricity cost. The devices ontology, separates the devices into groups based on their size and location in the house.

3.4 Functions and operators

The most important functions amongst the many used in the system are the ones responsible for the merging of different rules. They are based on the criteria of the postcondition and precondition nodes being able to subsume other nodes of their type. This is detailed in Algorithm 1, establishing how we can merge nodes depending on preconditions and similarly for postconditions.

```
Data: current precondition node,list of all precondition and list of postcondition nodes
history← {};
get level of generalization of current node;
while list of preconditions is not over do
  if the current node is in the same group as a preconditions node in the list then
    get level of generalization of node in list;
    if current node's postconditions subsume node's in the list postconditions then
      compare their postconditions for contradictions;
      if no contradictions then
        add postconditions to current node;
        add node in list to history;
        delete node in list;
      end
    end
  end
end
return history;
```

Algorithm 1: Nodes merging based on preconditions

The generalization and specialization operators are the ones responsible for changing the form of the rules. There are two different kinds of operator sets as one deals with the preconditions of a rule while the other deals with the postconditions.

4 Evaluation

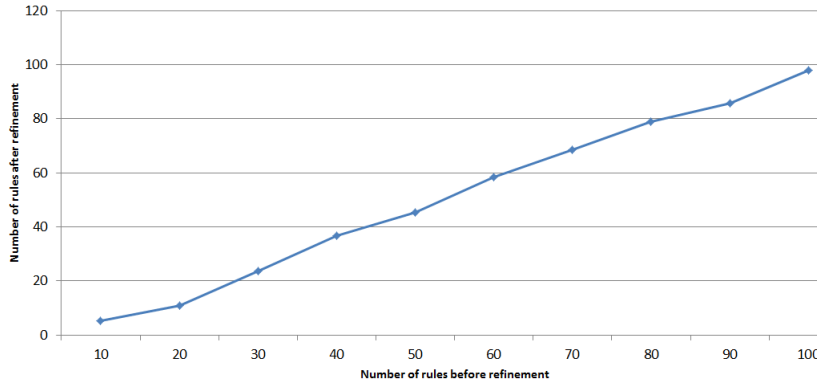


Fig. 1. Number of rules before and after refinement

For the performance and scalability evaluation our system had to undergo, some adjustments were made. The system was set on a loop to generate a specific amount of rules and continuously refine their preconditions and postconditions. Afterwards, the rules were merged and the final number of the rules was stored in a text file. It is important to mention that we ran an extensive amount of experiments (100 times for every point that appears in the graph) so as to guarantee up to a point, the statistical significance of our solution given the random nature of the rules generation.

During the evaluation of our system we attempted to cover all the parameters that should be examined. The total number of rules in the normative system before and after it was processed was the first metric we observed. Scalability testing was included as well, measuring the time required for the system to complete one iteration based on the amount of rules in the normative system on one hand and the amount of data on the other.

One of the main components that had to be checked during the evaluation of our system was the improvement on the size of the modified normative system. It is worth noticing however that the randomness of the rules generation did not always allow for rules that could be merged. We display the statistics for this metric of evaluation in Figure 1.

Scalability testing is used to evaluate the robustness of a system in situations of varying difficulty. As our system would have to be able to be integrated with Internet-of-Things scenarios that could incorporate a large set of rules or data, both of those parameters were tested.

The time required to process different amount of rules was recorded. This measurement indicates the toll multiple operations on rules can have on the system. The results are presented in Figure 2.

Another important test to evaluate our implementation was the time it required to go through data of different sizes. For this test, the data input was

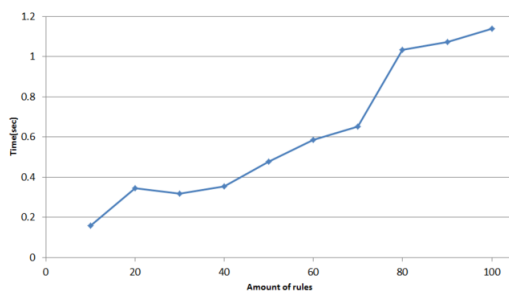


Fig. 2. Time required to process specific amount of rules

changed from a full day to increments of 3 hours and tested with 100 rules being generated. Figure 3 shows the results.

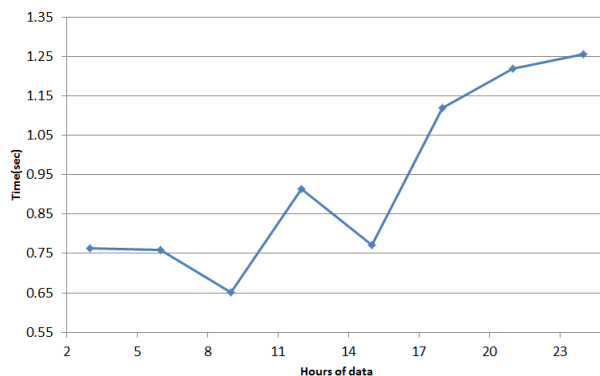


Fig. 3. Time required to process data

Besides the testing we carried out there are further experiments that could verify the consistency of our approach. Specifically, we were considering the formulation of two edge cases. The first would be an ideal, according to our understanding at least, initial set of rules with the intent of checking if our system can improve it further. The second, would be the worst possible initial set of rules in order to assess the system’s performance when handling bad input of this kind. However, due to time limitations, we could not perform either evaluation.

5 Conclusions, Discussions and Future Work

We initially aspired to create a system that would post-process and refine rules and normative systems through constant evaluations. Our goal was then transformed into a specific architectural design and our system specifications were stated. We have achieved our set objectives and presented our implementation in detail. Our analysis of the system’s performance was a thorough process that

involved, due to the random factor in the rules creation, a large number of experiments. Through scalability testing as well, we have discovered that our solution is working as intended with little variation in time required, even if the amount of rules is increased.

We have achieved the functional requirements laid out in section 3. In doing so we have made a contribution by extending and adapting an existing approach to address a specific scenario of Internet-of-Things.

In the implementation stage of the task at hand, we avoided integrating a Case-Based Reasoning mechanism into our system. This was done despite the improvement in the efficiency of the rules that it could provide for a few reasons. Firstly, the available free and open CBR systems are not easy to integrate with our system. Secondly, as our solution is focused on rule refinement rather than rule creation, we felt that it was beyond the scope of the paper to include a CBR system in the rule refinement machine.

During the development process of our system a lot of ideas were discussed and partially explored to enhance it. The enhancements can be split into system improvements and system extensions. The most significant improvement to our system would be an integration with available rule standards. This would enable our system to be easier to integrate with existing Internet-of-Things solutions. Ideally, instead of a specific technology and standard of rules, multiple options should be offered to whoever would want to incorporate our system, or parts of our system, into their own. Furthermore, additions to the system can be done to increase its capabilities. The easiest and most efficient would be a user-based rule insertion component. While the production of rules based on data is certainly as efficient as it can be, we need to consider the individual needs of every user. This would make the system more personalized and users would be more inclined to use it.

References

1. Beckel, C., Kleiminger, W., Cicchetti, R., Staake, T., Santini, S.: The eco data set and the performance of non-intrusive load monitoring algorithms. Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings (2014)
2. Durkheim, E., Simpson, G.: Emile durkheim on the division of labor in society / being a translation of his de la division du travail social, with an estimate of his work by george simpson. Macmillan (1933)
3. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies - Special issue: the role of formal ontology in the information technology (1995)
4. Gubbia, J., Buyyab, R., Marusic, S., Palaniswami, M.: Internet of things (iot): A vision, architectural elements, and future directions. FGCS (2013)
5. Kleiminger, W., Beckel, C., Santini, S.: Household occupancy monitoring using electricity meters. Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (2015)
6. Morales, J., López-Sánchez, M., Rordriguez-Aguilar, J.A., Vasconcelos, W., Wooldridge, M.: Online automated synthesis of compact normative systems. ACM Trans. Autonom. Adapt. Syst. 10, 1, Article 2 (2015)