

HomeRuleML – A Model for the Exchange of Decision Support Rules Within Smart Environments

Josef Hallberg, Chris D. Nugent, *Member, IEEE*, Richard J. Davies, Kåre Synnes, Mark P. Donnelly, *Student Member, IEEE*, Dewar Finlay and Maurice Mulvenna, *Member, IEEE*

Abstract—The demands for smart environments, which can help to facilitate as well as monitor independent living are increasing. With this comes a desire for decision support rules to process data recorded from such environments. However, testing and evaluating rules can be both time-consuming and indeed stressful for the inhabitants. Within this paper we propose a model, referred to as HomeRuleML, for representing decision support rules for smart environments. The motivating factor behind such a proposal is to provide a widely and freely accessible set of rules which can be openly used and exchanged within the research domain and beyond. This model has the potential to decrease the time required for deployment, and inevitably improve the inhabitants' quality of life. In the paper we explain in detail the structure of adopting this approach and also provide an indication of the typical types of software tools required for its use.

I. INTRODUCTION

“There is no place like home” is a well known phrase which becomes more pertinent in this day of age as many people choose and indeed prefer to live at home as they become older [1], or suffer from an illness. Why should people not be able to stay at home if they want to and if they are in good health? Facilitating the need of these people to stay at home has created a greater need for smart living environments which can help assist in daily activities and support increased levels of independence. These smart environments have the inevitable and indeed desired effect of extending the period of time a person can remain in their own living environment prior to the potential need of institutionalisation.

Customizing an environment to meet the specific needs of an individual can be both time consuming and stressful.

Manuscript received April 30, 2007.

Josef Hallberg, Luleå University of Technology, Sweden, (email josef.hallberg@ltu.se)

Chris D Nugent, School of Computing and Mathematics, University of Ulster, Belfast, Northern Ireland, (tel: +44 2890 368330, email: cd.nugent@ulster.ac.uk).

Richard Davies, School of Computing and Mathematics, University of Ulster, Belfast, Northern Ireland, (email rj.davies@ulster.ac.uk)

Kåre Synnes, Luleå University of Technology, Sweden, (email kare.synnes@ltu.se)

Mark Donnelly, School of Computing and Mathematics, University of Ulster, Belfast, Northern Ireland, (email mp.donnelly@ulster.ac.uk)

Dewar Finlay, School of Computing and Mathematics, University of Ulster, Belfast, Northern Ireland, (email d.finlay@ulster.ac.uk)

Maurice Mulvenna, School of Computing and Mathematics, University of Ulster, Belfast, Northern Ireland, (email md.mulvenna@ulster.ac.uk)

Even if personalities between individuals are unique, their impairments or illnesses are often similar. Therefore, in the current work we hypothesise that using standards and models, both for storing data from the environment and for creating rules, can help reduce the time required to customise an environment in addition to the stress imposed on the inhabitant. The net effect of such an approach would provide the desired effect of making these smart environments easier to deploy, more reliable, and reduce costs.

Given the importance of this domain and the huge benefits to be gained it has attracted much attention in the past decade from both commercial, healthcare and research perspectives. The result has been the development of a number of expert systems and rule-based systems for the analysis of information recorded within smart environments [2], [3], [4]. Nevertheless, few attempts have been made to achieve a cross-system standard for decision support rules intended for smart environments where environmental and sensor-based information should be taken into account.

This work looks at creating a standard for the storage and exchange of decision support rules to be used in smart environments. The model, which is referred to as HomeRuleML, builds on previous work by the authors [5] which offers a standard (HomeML) which can be used in the storage and exchange of information recorded within such environments.

Designing a rule must be a simple process since the technical developers are not always the people with the most knowledge about the needs of the inhabitant. The rules must also have sufficient details for the technical developers to configure them to the available resources (e.g. sensors) in the environment. In addition, it must be possible to integrate the rules dynamically into a smart environment. These requirements lead to the following research questions which the authors seek to answer:

- How can rules be generated and shared in a simple manner?
- How can a narrative text describing a rule be transformed into a HomeRuleML rule?
- How can HomeRuleML rules be parsed and understood by a smart environment application?

The rest of the paper is organized as follows: Section 2 gives a brief background to the work presented in this article, in addition to some related work. Section 3 presents

the method used in developing HomeRuleML, as well as the model itself. Section 4 explains the evaluation strategy and presents an example scenario. Section 5 discusses the research questions in more detail, and Section 6 concludes and summarizes the results and gives an overview of the future for HomeRuleML.

II. BACKGROUND

Trying to develop a common standard for representing decision support rules is not a new concept. For example, within the financial world this has been a prolific area of interest for many years where decision support rules are used for recommendations on stock market trends [6], [7]. The driving force behind a standard for rules for use within this scenario is because parameters (e.g. stock prices etc.) are constantly changing and sometimes the rules are required to be altered in order to take these changes into account. The same driving force exists for rules in smart environments where it is not desirable to update the entire application to add a rule. In addition being able to create a repository of rules for exchange between researchers is also an important factor.

In smart environments the attempts to develop a standard for decision support rules have been sparse. There are, however, a few examples. These include the first order logic used in Gaia [8], FACTS [9], and CADEL [10]. Although these examples address the dynamic nature of rules they are not developed for the purpose of sharing between smart environments. The rules are also very high level and lack detailed information about sensors in the environment and the environment itself. The general sensor abstractions provided can be useful for distribution, but will then require an additional layer which binds the sensor abstractions to

specific sensors. This is different from the approach taken with HomeRuleML, which provides the necessary information for binding sensor data to a specific sensor.

The HomeRuleML model in this article is based on HomeML [5], a standard for representing, exchanging and storing diverse data recorded within smart environments. HomeML provides detailed information about sensors (sensor events, sensor location, etc.), the environment (room information, room purpose, etc.) and the inhabitant (including any wearable sensors). HomeML can therefore be compared to a context-storage platform built upon XML with the purpose of sharing data between smart environments. Given the proposal to base HomeRuleML upon the concepts of HomeML the information in the HomeRuleML model can be kept simple and easy to understand, and it will refer to the additional information stored in HomeML.

Figure 1 shows the most relevant nodes of the HomeML definition. Apart from general information about the smart-environment and the rooms, it has a *Device* subtree which stores information about various sensors in the environment. Each device is identified by a *deviceID*, and contains a general description, information about location, device type (e.g. MovementSensor, TemperatureSensor, etc.), and which unit is being used for the sensor-data (e.g. Celcius, State, etc.). Furthermore, there is information for separating between different trials or “runs” (*RealTimeInformation*) containing sample rate of the sensor, as well as start and stop time. Finally there is room to store sensor data in the *Event* node. Each event is identified by an *eventID* but also contains a time stamp.

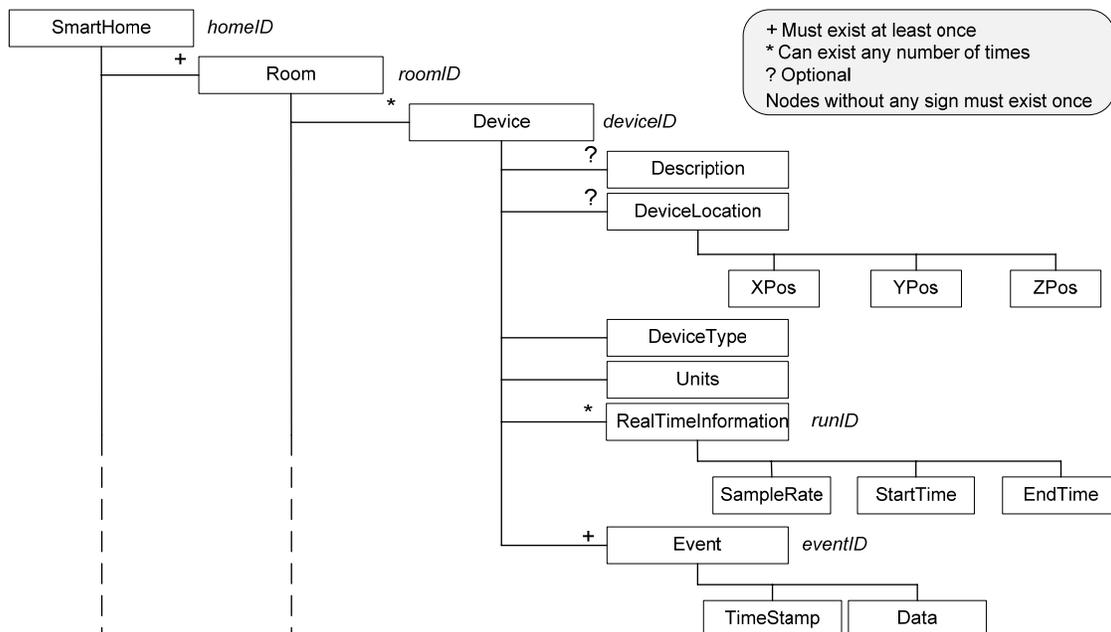


Fig 1: Excerpt of the HomeML definition

HomeRuleML as well as HomeML are created using the eXtensible Markup Language (XML) [11]. XML has gained much attention in recent years and has become a widely established solution for the expressing of syntax for data and the exchange of data via the Internet. The main benefits of employing XML relate to its platform, vendor and application independence. It is also considered to offer an easy to follow hierarchical data structure. There are also XML standards such as RuleML [12], and Drools [13] which implement the recent Java Rule Engine standard [14]. Nevertheless, these XML standards also lack support in terms of the specific details needed for a rule set for use in a smart environment.

Although there have been several successful rule-based systems, none of these have addressed the issues and needs of a truly open standard for the exchange of decision support rules intended for smart environments. Such a standard would facilitate the open exchange of data and will assist in moving one step closer towards the development of a common rule repository for use by the entire research community. It will also help to decrease the complexity of deploying smart environments. XML provides a flexible base for storing and modifying rules. It also offers a suitable standard for information exchange over the web.

III. METHODS

As previously stated, it is important to facilitate the exchange of rules between both researchers and smart environments and to simplify the deployment process for smart environments. Providing an openly accessible repository with tested and verified rules is one means that can be offered to help achieve this goal. HomeRuleML has been designed as a first attempt to offer a solution for an open standard for decision support rules within smart environments.

For sufficient flexibility and usability the model should support the application of logical expressions to sensor data, the possibility of defining the order of events, and to define time constraints. Furthermore, it should incorporate functionality for the reuse of more generic rules to simplify the creation of more complex rules. With HomeRuleML it was decided that abstractions may be made in sensor data and contexts at a lower level, but references to sensors, context providers, and other rules, will be made explicit through the use of identification numbers.

The first step in achieving the aforementioned requirements has been to design HomeRuleML as a series of hierarchical data trees (Figures 2 and 3). These graphical representations have been supplemented with Tables 1 and 2 which describe the elements and attributes within the aforementioned models.

A. The Rule Tree

Each rule starts with a root element *Rule* (Figure 2, Table 1), which is uniquely identified by its attribute *ruleID*. The

main components for each rule are *Description*, *DateRecordCreated*, *Operation*, and *Outcome*. Each rule can only have one set of each component (Table 1).

Description contains the description and purpose of the rule, and also what kind of result or action can be expected as an outcome. *DateRecordCreated* contains the date that the XML file in question was created. The *Operation* component contains logical operators which can be applied to devices, operations, or rules. It should be noted that *Operation* is a reference, in order to accommodate recursive behaviour of operations, and will be discussed later (see Table 2).

The *Outcome* is, as its name suggests, the outcome of the rule itself and is of interest if the Boolean result from the *Operation* is *true*. An *Outcome* can consist of a *Value* and an *Action*. The *Value* is a String value and is used primarily when the rule is being used as part of another rule. It also helps explain the purpose of the *Rule*. The *Action* can consist of a number of *Device* components identified by their *deviceID* attribute. Each *Device* contains a *SetValue* element which is the value to be applied to the device in question as a result of the rule and the *Operation* being *true*.

B. The Operation Tree

The *Operation* is a support component for rules (Figure 3, Table 2). It is a recursive schema which makes it possible to facilitate nested operations and rules in order to form complex expressions. The root element is *Operation* which has an *operator* attribute that can have the following logical operators: *AND*, *OR*, *NOT*, *NAND* (neither), *NOR*, *XOR* (either one or the other, but not both) and a special operator *ORDER* which makes it possible to define a sequence of events. The operator will be applied to the underlying elements, but will not traverse the tree to apply itself on nested elements.

The main components for each operation are *Time*, *Device*, *Operation*, and *Rule* (see Table 2). The *Time* component is optional and is for defining temporal logic to be applied to the elements of the current *Operation*. The attribute of the *Time* component is a *condition* which can have the following operators: *GT* (greater than), *LT* (less than), *EQ* (equals), *GEQ* (greater than or equal to), *LEQ* (less than or equal to), and *NOT*. The condition will be applied to the *Value* subcomponent which is defined as time in seconds and the result will be a Boolean value.

The *Device* component, identified by its *deviceID* attribute, contains an optional *Description* (describes the purpose and placement of the device), a *DeviceType* (e.g. TemperatureSensor), *Units* (e.g. Celcius), and one or more *Condition* components. The *Condition* has a *condition* attribute which has the same operators as the attribute for the *Time* component. The condition will be applied to the event values which can be found (using the *deviceID* attribute) in the HomeML standard and compared to the *Value* subcomponent, the result will be a Boolean value. If there

are more than one *Condition* components an *OR* logic operation will be applied to the results from each *Condition* in order to form a single Boolean result.

Lastly there is a reference component, *Operation*, which makes it possible to create nested expressions. Similarly to a reference, there is a *RuleReference* component which contains the *ruleID* attribute of another *Rule* (Table 1) to use as part of the *Operation*. Each *Rule* which is to be used in a

nested *Operation* is required to have its *Value* in the *Outcome* set. The *Description* and the *Condition* components in the *RuleReference* are similar to previous occurrences and the *condition* operator will be applied as a String operator. In this manner the *operator* can be applied to rules and operations in a similar way. The result from all *Operation* components will always be a Boolean value.

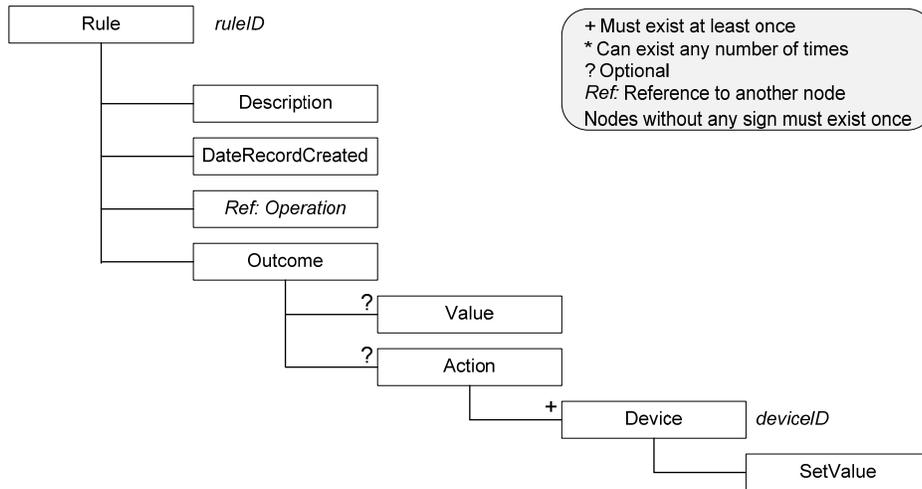


Fig 2: Hierarchical representation of HomeRuleML Rule tree

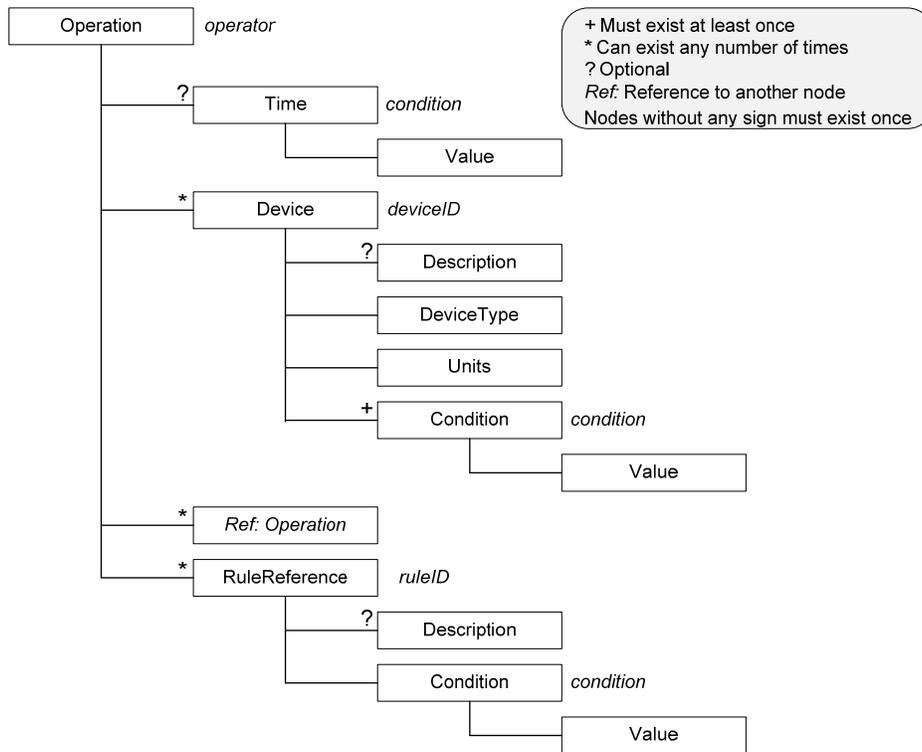


Fig 3: Hierarchical representation of HomeRuleML Operation tree

TABLE I
DESCRIPTION OF RULE ELEMENT BASED ON THE HIERARCHICAL REPRESENTATION OF HOME RULEML IN FIGURE 2

Element/Attribute	Description	Required/ Optional	Data Type	Example
Rule	The root element for XML based home rule			
<i>ruleID</i>	A unique identifier for the rule (attribute)	Required	Integer	1001
Description	General description of the rule	Required	String	Medication Management – This rule sends an alarm if the medication isn't taken at the correct times
DateRecordCreated	The creation date/time of the record	Required	TimeStamp	2007-04-23 11:56:05
Operation	A reference to an Operation component	Required	Reference	See Operation table (Table II)
Outcome	Specifies the rule's outcome	Required		
Value	The value of the rule if the result is true	Optional	String	Food Taken
Action	The action to be taken if the result is true	Optional		
Device	The device involved in the action	Required		
<i>deviceID</i>	The ID of the device involved (attribute)	Required	Integer	2001
SetValue	The resulting value the device should take	Required	String	Off, Alarm, etc.

TABLE II
DESCRIPTION OF OPERATION ELEMENT BASED ON THE HIERARCHICAL REPRESENTATION OF HOME RULEML IN FIGURE 3

Element/Attribute	Description	Required/ Optional	Data Type	Example
Operation	The root element for Operations			
<i>operator</i>	The logical operation for the element (attribute)	Required	String	AND, OR, NOT, NAND, NOR, XOR, ORDER
Time	Specifies time conditions for the Operation	Optional		
<i>condition</i>	The logical operator to be applied (attribute)	Required	String	GT, LT, EQ, LEQ, GEQ, NOT
Value	The time value to compare with (in seconds)	Required	Integer	3600 (seconds)
Device	A device involved in the Operation	Optional		
<i>deviceID</i>	A unique device identifier (attribute)	Required	Integer	2002
Description	General description of the device	Optional	String	Bed Occupancy Sensor
DeviceType	Details the type of the device	Required	String	PressureSensor
Units	Measurement unit	Required	String	State, Celsius, etc.
Condition	The condition for the events of the Device	Required		
<i>condition</i>	The logical operator to be applied (attribute)	Required	String	GT, LT, EQ, LEQ, GEQ, NOT
Value	The value to compare with	Required	Varies	Open, 20, etc.
Ref: Operation	A reference to the root element for Operation	Optional		
RuleReference	A reference to a different HomeRuleML rule	Optional		
<i>ruleID</i>	The unique identifier for the rule (attribute)	Required	Integer	1002
Description	General description of the rule	Optional	String	Food Sensor
Condition	The condition for the Rule	Required		
<i>condition</i>	The logical operator to be applied (attribute)	Required	String	GT, LT, EQ, LEQ, GEQ, NOT
Value	The value to compare with	Required	String	Food Taken

IV. EVALUATION OF THE MODEL

Based on the design of HomeRuleML the first version of the XML based schema has been produced. To facilitate uptake of this model, the XML schema along with documentation providing guidance on use and steps to follow to make recommendations for changes to the current version, will be made available through a series of web based resources (<http://trail.ulster.ac.uk/HomeML/>).

In addition to providing the schema for HomeRuleML the authors are also planning to make available a suite of open source tools to assist users in exploiting the general concept. In the first instance this includes the HomeML standard (XML schema) for storing and sharing data from smart environments. There are also a number of assistive tools such as a browser and data editor which will help display and edit/populate HomeML data, a SQL database for efficient data storage and access on site, and tools for generating the HomeML files from the database. In addition a graphical interface for creating rules [15], will also be made available.

A. Example scenario

The creation of rules can be performed in many different ways. One way would be to use a graphical tool to generate the rule and then parse the graphical representation and generate the desired XML code. However, having to develop a rule from a narrative is a common situation. Of course a graphical tool could be used in this situation as well [15], but creating the desired XML code directly from the narrative might be preferred by some. HomeRuleML is designed to be self explanatory to anyone with basic knowledge of XML, and hence the creation of rules can be performed directly in XML.

Below is a short example of a narrative of a typical situation to be managed within a smart environment. This serves to offer a useful exercise in demonstrating the creation of the XML code:

Medication Management – A person must take their medication within 1 hour after getting out of bed. Within this period they must also ensure that they have had something to eat prior to taking the medication.

The first step to develop a rule would be to identify all the information which must be recorded. Detecting bed activity can be performed with a bed sensor and detecting if someone has taken their medication can be performed by using a smart pill dispenser [16]. However, detecting if someone has eaten might require a number of different sensor recordings. Here we assume that it is sufficient to detect if there has been movement in the kitchen and the fridge door has been opened in order to determine that someone has eaten. Hence, the XML representation to ascertain if food has been taken could be expressed as shown in Figure 4 (note that we choose to use an outcome value so this rule can be used within the main rule).

```
<Rule ruleID="102" xmlns:xsi="http://..."
  xsi:noNamespaceSchemaLocation="file:HRML.xsd">
  <Description>Food Sensor</Description>
  <DateRecordCreated>
    2007-04-23 11:58:05
  </DateRecordCreated>
  <Operation operator="AND">
    <Device deviceID="204">
      <Description>
        Kitchen Movement Sensor
      </Description>
      <DeviceType>
        Movement Sensor
      </DeviceType>
      <Units>State</Units>
      <Condition condition="EQ">
        <Value>Movement</Value>
      </Condition>
    </Device>
    <Device deviceID="205">
      <Description>
        Fridgedoor Sensor
      </Description>
      <DeviceType>DoorSensor</DeviceType>
      <Units>State</Units>
      <Condition condition="EQ">
        <Value>Open</Value>
      </Condition>
    </Device>
  </Operation>
  <Outcome>
    <Value>Food Taken</Value>
  </Outcome>
</Rule>
```

Fig. 4. The HomeRuleML representation of the Food Sensor rule.

The narrative describes a certain order in which the events must happen, as well as some time limitation for these events. First the person needs to get out of bed, and from this event the person needs to first eat, and then take their medication, all within one hour (or 3600 seconds). If these steps are not fulfilled in the specified order or within the time constraints the rule should trigger an action. In this case we trigger an alarm on one of the devices in the environment. The XML representation for this part of the scenario, using the Food Sensor rule as previously presented (Figure 4), can be represented as shown in Figure 5.

```
<Rule ruleID="101" xmlns:xsi="http://..."
  xsi:noNamespaceSchemaLocation="file:HRML.xsd">
  <Description>
    Medication Management
  </Description>
  <DateRecordCreated>
    2007-04-23 11:56:05
  </DateRecordCreated>
  <Operation operator="NOT">
    <Operation operator="ORDER">
      <Time condition="LT">
        <Value>3600</Value>
      </Time>
      <Device deviceID="201">
        <Description>
          Bed Occupancy Sensor
        </Description>
        <DeviceType>
          PressureSensor
        </DeviceType>
        <Units>State</Units>
        <Condition condition="EQ">
          <Value>Out of Bed</Value>
        </Condition>
      </Device>
      <RuleReference ruleID="102">
        <Description>
          Food Sensor
        </Description>
        <Condition condition="EQ">
          <Value>Food Taken</Value>
        </Condition>
      </RuleReference>
      <Device deviceID="202">
        <Description>
          Pill Dispenser
        </Description>
        <DeviceType>
          TiltSensor
        </DeviceType>
        <Units>State</Units>
        <Condition condition="EQ">
          <Value>Pill Taken</Value>
        </Condition>
      </Device>
    </Operation>
  </Operation>
  <Outcome>
    <Action>
      <Device deviceID="203">
        <SetValue>
          Alarm - Medication Alert
        </SetValue>
      </Device>
    </Action>
  </Outcome>
</Rule>
```

Fig. 5. The HomeRuleML representation of the Medication Management rule

Representation of the medication management rule in the HomeRuleML format means that it could be stored on a central repository and openly shared with others. The goal is that, assuming the same device IDs are used for sensors, deployment of such a rule would simply work without any need for change in their application. In cases where device IDs mismatch the descriptions provided in HomeRuleML should be clear enough that it is clear which device ID to

include. Note that a rule can also consist of other more generic rules (e.g. Food Sensor) which is good for hiding the complexity of “lesser” activities.

V. DISCUSSION

Creating rules is often associated with a complex procedure. In many cases this results in rules being created by developers and not actually by the people with the insight and domain knowledge [15]. The desire is of course to make the rule construction so simple that anyone could do it, yet so descriptive that the end result could be used as a component in an existing model monitoring a smart environment.

In parallel to the HomeRuleML model a graphical front-end [15] has been developed by the authors. Initial evaluations have shown that this interface is a viable and appropriate method for healthcare professionals to create rules for smart environments. However, turning this graphical representation into a HomeRuleML representation is a different matter. Currently the rules are stored graphically and a parser needs to be developed in order to generate HomeRuleML code from it. Given that the schema is well defined and HomeCI is created for making rules for smart environments it should be possible.

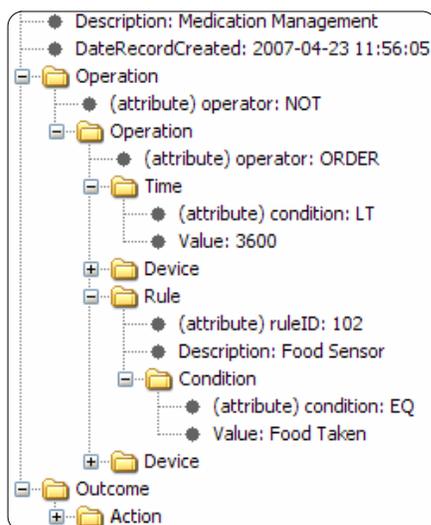


Fig 6. An example of the HomeRuleML browser.

Another way to make the creation of rules easier would be to create a browser which is based on the HomeRuleML schema. By using such a tool it would then be possible to edit the rules in this browser, which would make it possible for people without a programming background to design and update the rule (Figure 6). In addition, there are already a number of powerful XML authoring tools on the market which would simplify the generation of HomeRuleML code.

Once a HomeRuleML file is created and validated it is desirable to be able to plug the rule into a system without having to modify the application. As previously mentioned use of the Java Rule Engine [14] will help to make this

possible. HomeRuleML is also built on standard logic operators and the hierarchic structure of XML; hence it makes it easy to parse the elements in a logical order. The Simple API for XML [17] is an event driven parser which makes it possible to get events in both the beginning and the end of a node. This would mean that HomeRuleML code would yield a logical expression with the operator first, followed by the parameters (the sub-elements). Written down it would look something like AND (A, B, C), which means that the AND operator will be applied to elements A, B and C. Taking advantage of the hierarchical structure of XML and the Java Rule Engine it is possible to parse HomeRuleML files, and make them understood by a smart environment application.

VI. CONCLUSIONS AND FUTURE WORK

Being able to share decision support rules for smart environments offers great potential to decrease deployment time, increase reliability, and decrease cost. Even if individuals are different, they often share similar impairments or illnesses, which would suggest that they would require similar support from the environment. Such support can greatly improve the quality of life of this cohort of people. To facilitate this will require a number of thoroughly tested and evaluated rules.

This article has presented HomeRuleML, a model for creating decision support rules for use within smart environments. Usage of HomeRuleML offers the potential to create a repository of rules which can be easily deployed. HomeRuleML has been realized by using the strengths of XML, such as the hierarchical structure and suitability for web sharing, and as such has resulted in a XML based schema.

Access to the HomeRuleML source files will be made available through web based resources or can be obtained directly by contacting the authors of this paper. With this we hope to inform the community of these developments and maintain the user-driven notion of using XML. Furthermore, we plan to further develop and evaluate HomeRuleML, as well as create the necessary implementation for making it possible to dynamically add HomeRuleML rules to an existing smart environment application. Finally, it is our intention to secure a number of collaborators who can help evaluate the model as well as help create the first set of rules in order to establish a repository which would become widely available within the general research community. It is our belief that this will help further improve HomeRuleML for the benefit of both research community and the inhabitants of smart environments.

VII. ACKNOWLEDGEMENTS

This work has been partly supported through the Nestling Technologies Project, the MyMates Project, and the Centre for Distance-Spanning Healthcare (CDH).

VIII. REFERENCES

- [1] D. Cook, S.K. Das Cooke, "How Smart are our Environments? An Updated Look at the State of the Art," *Journal of Pervasive and Mobile Computing*, vol 3, num 2, 2007, pp. 53-73
- [2] J.P. Sousa, D. Garlan, "Aura: an Architectural Framework for User Mobility in Ubiquitous Computing Environments," *Proceedings of the IFIP 17th World Computer Congress - TC2 Stream / 3rd IEEE/IFIP Conference on Software Architecture (WICAS3)*, 2002
- [3] C.D. Kidd, R. Orr, G.D. Abowd, C.G. Atkeson, I.A. Essa, B. MacIntyre, E.D. Mynatt, T.Starner, W. Newstetter, "The Aware Home: A Living Laboratory for Ubiquitous Computing Research," *Proceedings of the Second International Workshop on Cooperative Buildings, Integrating Information, Organization, and Architecture*, 1999
- [4] A.K. Dey, D. Salber, G.D.Abowd, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," *Anchor article of a special issue on context-aware computing in the Human-Computer Interaction (HCI) Journal*, vol 16, 2001, pp. 97-166
- [5] C.D. Nugent, D. Finlay, R.J. Davies, H.Y. Wang, H. Zheng, J. Hallberg, K. Synnes, M.D. Mulvenna, "homeML – An Open Standard for the Exchange of Data Within Smart Environments", *5th International Conference on Smart homes and health Telematics (ICOST 2007)*, Lecture Notes in Computer Science, Vol. 4541, 2007, pp. 121-129
- [6] J.K. Lee, M.M. Sohn, "The eXtensible Rule Markup Language", *Communications of the ACM*, vol 46, num 5, 2003, pp. 59-64
- [7] B.N. Groszof, "Representing E-Commerce Rules Via Situated Courteous Logic Programs in RuleML," *Electronic Commerce Research and Applications (ECRA)*, Spring 2004, pp. 2-20
- [8] A. Ranganathan, R.H. Campbell, "An infrastructure for context-awareness based on first order logic," *Personal Ubiquitous Computing*, vol 7, num 6, 2003, pp. 353-364
- [9] K. Terfloth, G. Wittenburg, J. Schiller, "FACTS – A Rule-based Middleware Architecture for Wireless Sensor Networks," *First International Conference on Communication System Software and Middleware, 2006*, 2006, pp. 1-8
- [10] K. Nishigaki, K. Yasumoto, N. Shibata, M. Ito, T. Higashino, "Framework and rule-based language for facilitating context-aware computing using information appliances," *Distributed Computing Systems Workshops*, 2005
- [11] Extensible Markup Language (XML), <http://www.w3.org/XML/> (accessed April 2007).
- [12] RuleML, the Rule Markup Initiative, <http://www.ruleml.org/> (accessed April 2007).
- [13] Jboss Rules (Drools 3.0), <http://labs.jboss.com/jbossrules/docs/> (accessed April 2007).
- [14] JSR 94: Java Rule Engine API, <http://jcp.org/en/jsr/detail?id=94> (accessed April 2007).
- [15] C.D. Nugent, R.J. Davies, J. Hallberg, M.P. Donnelly, K. Synnes, M. Poland, J. Wallace, D. Finlay, M. Mulvenna, D. Craig, "HomeCI – A visual editor for healthcare professionals in the design of home based care," *Proceedings of the IEEE Engineering in Medicine and Biology Society*, 2007
- [16] C.D. Nugent, D. Finlay, R.J. Davies, M.D. Mulvenna, J.G. Wallace, C. Paggetti, E. Tamburini, N.D. Black, "The next generation of mobile medication management solutions," *International Journal of Electronic Healthcare*, vol. 3, no. 1, pp. 7-31, 2007
- [17] Simple API for XML (SAX), <http://www.saxproject.org/> (accessed April 2007)