International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.4 (2015), pp.31-42
http://dx.doi.org/10.14257/ijmue.2015.10.4.04

An Ontology-based Problem-logic Driven Approach towards the Activity Awareness for Elderly Care

Bin Xiao[1,*], Theo Kanter[1], and Rahim Rahmani[1]

[1] *Department of Computer and Systems Sciences, Stockholm University*
Nodhuset, Borgarfjordsgatan 12, SE-164 40 Kista, Sweden
xbin, kanter, rahim}@dsv.su.se

Abstract

The same activity under different user situations in the elderly care system may lead to different intelligent system response, since user needs vary with the changing situations. Activity awareness (AA) emphasizes that systems intelligently respond to user needs by aware the user-performed activities which is denoted and comprehended according to various user situations. But most current AA technics use similar patterns retrieved from the growing historical data, neglecting the situation change, called dataset driven. Thus, an aptly approach is needed in support of AA systems to handle various activity denotations in different user situations. Responding to the problem, this paper proposes an Ontology-based Problem-logic driven approach (OPL) to enhance the AA system by denoting user activities with the problem logic according to user-oriented denoted problem domains, where the AA system can seamlessly integrate with inferring for intelligent system response. Specifically, denoted problem domain and logic are proposed to support the OPL concepts, while activity graph is formed to support the intelligent system response based on the annotated problem logic. With the OPL, systems can directly target at the changing situations by rule-based inferring. A case study is performed upon the scenario retrieved from a European elderly care project, where a proof-of-concept prototype is established to confirm the validity of the OPL approach.

Keywords: *Activity Awareness, Denoted Problem Domain, Problem Logic, Ontology, Activity Graph, Elderly care*

1. Background and Motivation

Elderly people are suffering from the dementia problem in some extent [1], which exposes them under a challenge to methodically carry out daily tasks or accurately response towards the surroundings in time. For example, the elderly people might be easy to forget taking pills regularly and with a correct amount, where an intelligent approach is needed to reminder and instruct them. The process of extracting user needs from the activity to support the intelligent response of the system is called activity awareness [2]. Activity awareness is a user-performed activity based paradigm to support those users via the intelligent service provided by the information system, which can be used for elderly care [3-4]. And researchers can conduct system assisted living by using activity awareness to capture user needs from the ubiquitous environment, with which the system can adapt to user activity and provide reminder message by retrieving the user activity and user needs [4-6]. It is always the core output to provide an intelligent response of the activity awareness system to support the elderly user with slight dementia. Thus, monitoring user status, extracting user needs and giving an intelligent system response should always be the core essences in the intelligent system for elderly care, regardless how the user needs would be retrieved.

As mentioned above, activity awareness emphasizes on capturing user needs in different situations and providing intelligent system response. Currently dataset-driven technics are

ISSN: 1975-0080 IJMUE

widely used for intelligence referring [5-6], where the data samples must be collected for computation, such as machine learning. However, in real life, even the simplest human activity is burst, various and rich-in-meaning, so the pattern based on previously collected dataset may not be able to correctly comprehend the current activity changing with the user situations. For example, towards the sudden stoop behavior of the elderly, the same activity may need to be comprehended into different intelligent notices. The elderly user maybe just stoop for greeting or it could be a symbol of sudden chock. For such case, the activity should be comprehended within a certain denoted problem domain to get the correct meaning, which is introduced in this paper to deal with the various situations. Sometimes we are not only interested in what is the activity that the user is performing, but also we need to let the system know what intelligent response the system should provide towards the current user activity status, within the various situations. Thus, we are motivated to explore how the activity awareness system can handle with various activity denotations caused by different situations and support intelligent responses towards user needs, where an aptly approach should be invented. Ontology is currently used widely for system annotating in activity awareness, however it does not focus enough on annotating the physical world. In order to adequately map the entities from physical world into the system with logic, some improvement towards the current ontology technics is needed.

In response to the problems stated above, we propose an ontology-based problem-logic driven approach (OPL), in line with the activity awareness concept, to support the intelligent elderly care under different situations. The concepts of problem logic and denoted problem domain are introduced to support the OPL approach. The OPL targets to track the activity denotations and render the intelligence response according to different denoted problem domains, with the activity awareness paradigm, which is problem-logic driven. In OPL, the denoted problem domain is a user-oriented syntax scope where the activities can be comprehended to exactly punch at the user expects, where the Problem-logic is established. Denoted problem domain defines the boundary of a context problem, where the activity shall be comprehended towards the problem needs to be solved and the user expected intelligent responses. In this paper, the author utilizes a problem-logic driven way to annotate activity data using web ontology language (OWL) [8], where the model is established above certain denoted problem domain. The activity graph is proposed based on the problem-logic driven annotating to enable the system to intelligently respond to the user needs in real time, where the system can seamlessly integrate with a rule-based inferring. As a contrast with the conventional dataset driven approach, the problem-logic driven approach closely targets at intelligent system responding to the user needs by inferring and comprehending the activity status in different situations.

The OPL method directly serves for the system inferring and intelligent responding, with the problem logics. Chapter 2 will discuss specific background for the activity awareness and the ontology based activity modelling. Chapter 3 will present the conceptual framework of problem-logic driven approach and discuss how to annotate the problem logic in activity awareness system, where the concepts of denoted problem domain and problem logic are introduced. Chapter 4 provide the activity graph to support the seamlessly integration of activity model to intelligent system inferring. In chapter 5, a case study will be performed with a scenario retrieved from an intelligent elderly European research project to show how to establish a system with a problem-logic driven approach for intelligent elderly care and confirm the validity of the OPL approach.

2. Related Work

Understanding the user status by recognizing the user activity has been regarded as an essential method for intelligent elderly care by researchers [3-4]. The existing AA approaches that retrieve activity recognition patterns from historical dataset using probabilistic or mathematic methods are called dataset driven approach. The pure

dataset driven approach puts some limit on the AA system when face with the changing user situations, which is essential for AA system. In order to support the AA system to deal with activities from different user situations, this paper will highlight the problem logic driven approach, explained in chapter three and four, where the denoted problem domain and problem logic are introduced to support the changing user situations while the activity graph is introduced for the intelligent system response. However, the currently used ontology approach cannot fully present problem logic from the physical world, where the improvement and adaption will be introduced in later discussion.

2.1. Dataset-driven Approach for Activity Awareness

"Activity awareness technics aim to let smart environments respond proactively to user needs and intentions by automatically inferring the activity being performed [7]." Researchers made a consensus to emphasize on retrieving user needs from activity to support intelligent system response, where user performed activity is one of the essential channels to capture user needs [9]. The user activity is conspicuous and various to be comprehended, which puts a challenge for the system do deal with the same activity in different user situations. Because even the same activity may have different denotations leading to different intelligent response in different situations, an approach to annotate the activity according to changing situations is needed to enhance the activity awareness. Until now, activity can be recognized by using machine vision [9-10], and other multiple sensors [11]. Usually those technics are dataset driven that calculate the collected data using mathematical processing based on history data, among which the Conditional Random Field, Markov Chain and Dynamic Bayesian Model are the fundamental and frequently used algorithms. Those technics emphasize on determining what the user is performing by capturing the activity nature [12]. The dataset driven approach suits well to the problem of recognizing the activity and determine its nature. However, it is difficult to deal with the problem that the same user activity needs to be comprehended and denoted differently in different user situations, using pure dataset driven approach [5, 11-12], where the problem is introduced. Specifically, patterns and models built on historical data cannot comprehend the activity for system responses, according to the changing user situations. It could be a meaningful improvement for the activity awareness technics, if the conventional dataset driven approach can be compensated with a logic which can be seamlessly integrated with inferring work [5, 11-12].

2.2. Ontology-based Activity Modelling

Ontology is frequently used to describe the activity concepts and the relations among those concepts [13-14]. Some researchers use ontology based methods with conventional dataset driven approach for activity processing [13-14]. And those technics mainly use ontologies to represent the system processes and their character in the engineering process [13-14]. For example, ontologies are used to describe the "sensor type", "object", and "logic confliction" [14], which are abstracted from the concept related to engineering processes. Those annotating methods help to make a tight connected system process, but it also weakens the capability to annotate physical activity status where a user needs are detected. The activities are connected with some entities in physical world, where each of them needs to be annotated with the information content and attributes instead of just the roles in engineering processes. In order to enable high capability for the problem logic present of user activities, the authors will present an object-oriented annotating approach using ontology in chapter3.

3. OPL--The Problem Logic Driven Approach

Using activity awareness to support the intelligent elderly care, the activity should aim at how systems should respond to the various stimuli in user immersed surrounding (including both physical and psychological) to directly contribute to a macro problem where the user locates. So this chapter introduces the concepts of ontology-based problem-logic (OPL), while the denoted problem domain and problem logic would be used to support the concept of OPL. The macro problem defines a scope of problem denotation, problem expectation and problem precondition, which is driven and based on the user needs and system cause. We define this macro problem scope as a denoted problem domain. Because the same user activity status may have different meaning and be comprehended differently within different denoted problem domains, it is nearly impossible to find a globally suitable definition or assumption for all the problems. In order to make the activity can be comprehended correctly and the system can provide an exactly-aim intelligent response, the activity awareness system needs to set up a denoted problem domain. For example, it is the same activity of an elderly people to sleep on the couch, but it may have different problem logics and denotations in different situations. Within some denoted problem domain, some system may need to inform the elderly to take a carpet for keeping warm. While within another denoted problem domain, the elderly might be wake up by the system, because doctor says they cannot sleep too much. In such case, the same activity might be comprehended by different meaning and aided by

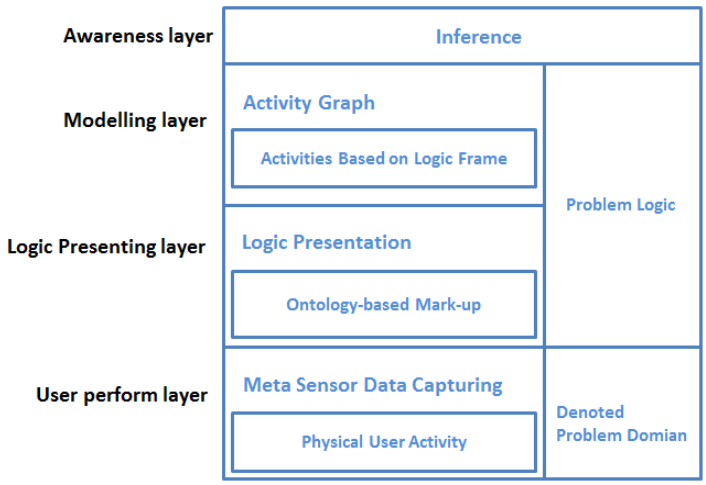

Figure 1. The Concept Framework of OPL

different system response, because they have different problem denotations and different problem expectations within their own denoted problem domains. In OPL, denoted problem domain is the tool to define user situations, which help the system correctly denote the activity. Continuing with the example above, if the problem domain is defined that the elderly needs to be monitored with limited sleep, the system will devote to a series of problem logic that wakes up the elderly when he eventually falls asleep on the couch. Further activity awareness system needs to be annotated by the problem logic in the meta-data files, which is an extension of the denoted problem domain. The definitions of denoted problem domain and problem logic are given as definition 1 and definition 2 explain.

Definition 1: Denoted Problem domain *is the scope of activity denotations for the activities within this scope to directly contribute to the user-defined services.*

The OPL is consisted with four layers as shown in Figure 1. The user activity status is detected by pervasive heterogeneous sensor clusters in the User Perform Layer, which acts as a bridge between the physical world and the intelligent system. Using the detected and

collected sensor meta-data from the physical surroundings, the problem logic retrieved from the denoted problem domain is annotated by the entity-based markup, which will be explained in the coming section. With the annotated logic in Logic Presentation Layer, an Activity Graph will be established which connects each entity and activities in the annotated file. This Activity Graph can integrate the activity model with an inferring engine seamlessly, where the Inference will be supported in the upper layer. That layer is called the Modeling layer and will be discussed in chapter 4. The Inference layer here is related to the rule-based engine, where the real-time intelligent system response can be provided. The Inference layer is the top layer of the whole framework and will be presented by an example in chapter 5.

3.1. Problem Logic within Denoted Problem Domain of OPL

Definition 2: Problem logic *indicates the service logics within a certain denoted problem domain for interpreting and comprehending the user activity for rendering intelligent system response.*

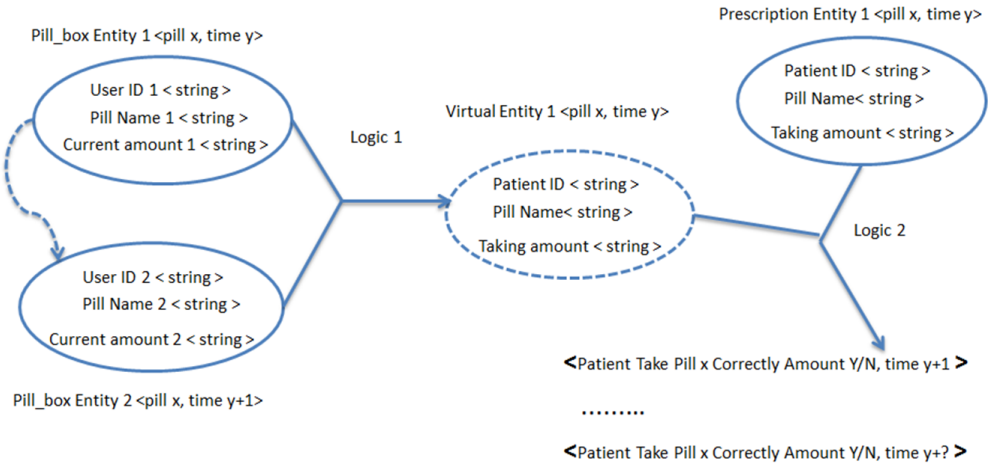

Figure 2. Present the OPL Logic with Ontology

To support the assisted living for elderly people, the activity awareness should be aligned with a set of problem logics within a certain denoted problem domain, as shown in Figure 1. The problem logic provides semantic stretches for denoted problem domain in the angle of user needs, while the denoted problem domain is a base of problem logic. The denoted problem domain is cultivated and exists in the layer of meta-data capturing via heterogeneous sensor network. Within the denoted problem domain, the positive direction of contributing to user needs is defined by the problem logic. For example, the elderly user is sleeping on the couch. If the activity is within a denoted problem domain aiming at control the sleep amount, then one of the problem logics may be generated as "if the sleep amount of this user is larger than the recommended, then inform the family". As a contrast, if the activity is within a denoted problem domain aiming at protecting from catching cold, then one of the problem logics may be "take a piece of carpet to keep warm".

Starting from the logic presenting layer in the OPL framework, the problem logic is presented by the ontology mark-up language. Also in the modelling layer, the problem logic is used to establish the activity graph. Finally, as a top layer of the OPL, activity graph and problem logic are cooperated to support activity awareness inference, where the system should provide an intelligent response towards the user activity. The denoted problem domain and problem logic are defined according to the different user situations, which can save the AA system from general and blur activity denotation that one fits all.

3.2. Ontology-based Problem Logic for OPL

```
<?xml version="1.0"?>


<!DOCTYPE rdf:RDF [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>


<rdf:RDF xmlns="http://www.owl-ontologies.com/Ontology1399365101.owl#"
     xml:base="http://www.owl-ontologies.com/Ontology1399365101.owl"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <owl:Ontology rdf:about=""/>
    <owl:DatatypeProperty rdf:ID="age">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:domain rdf:resource="#perscription"/>
        <rdfs:range rdf:resource="&xsd;int"/>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="alarmamount">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:domain rdf:resource="#perscription"/>
        <rdfs:range rdf:resource="&xsd;int"/>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="medicinename">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:domain rdf:resource="#perscription"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>
    <pill_box rdf:ID="my_boxa">
        <userid rdf:datatype="&xsd;string">Se19600617-6366</userid>
        <username rdf:datatype="&xsd;string">Lily</username>
        <pillgrid rdf:datatype="&xsd;string">pill1</pillgrid>
```

Figure 3. A Sample of OWL/XML Code for OPL

Activity source indicates for the sensor clusters or any other channel that input system-readable data or file to describe the user activity, which is the portal to detect and acquire the user activity status in the User Performed layer. In order to annotate the problem logics presented in OPL, the web ontology language (OWL) is used to mark up the data from each activity sources. According to the standard of W3C [15], ontology includes the classes, the instance of classes and the properties. In line with the OWL rules, we make instance of each class as an entity, which connected with a set of properties to describe its features. The meta-data input from activity sources helps to instance the class as an entity. The entities are integrated with different problem logics. Take an example from the intelligent elderly care scenario, where the elderly people taking pills from the pill boxes. The entities and relevant problem logic of this scenario are shown in the Figure 2.

Suppose the elderly with slightly dementia needs to take pills closely following the prescription from the doctor. And the caregivers will place the prescriptive pills from the doctor into the sensor-equipped pill box that the patient will take the medicine from. For this case, we define three classes: the Pill_box, the Virtual class and the Prescription. Each class is inserted with a set of properties, to annotate the feature and the functionalities of this entity. It is a temporal process for the activity resources generate a group of meta-data based entity in intervals. Based on the assumption, the denoted problem domain for this scenario can be defined as the denotation scope, where the pill taking amount of certain patient can be monitoring regularly. The problem logics are defined based on the denoted problem domain, as: (1) Logic 1: balance amount of the pills in the pill box explicates the amount taking by patients. (2) Logic 2: The pills amount balance between the pill box and prescription highlights whether the patients taking correct amount of pills. All the problem logics are in

temporal logic. Figure 3 shows the ontology file slotted by activity resources, with which both the activity sources and problem logic are annotated.

4. Activity Graph of OPL in Support of Intelligent System Response

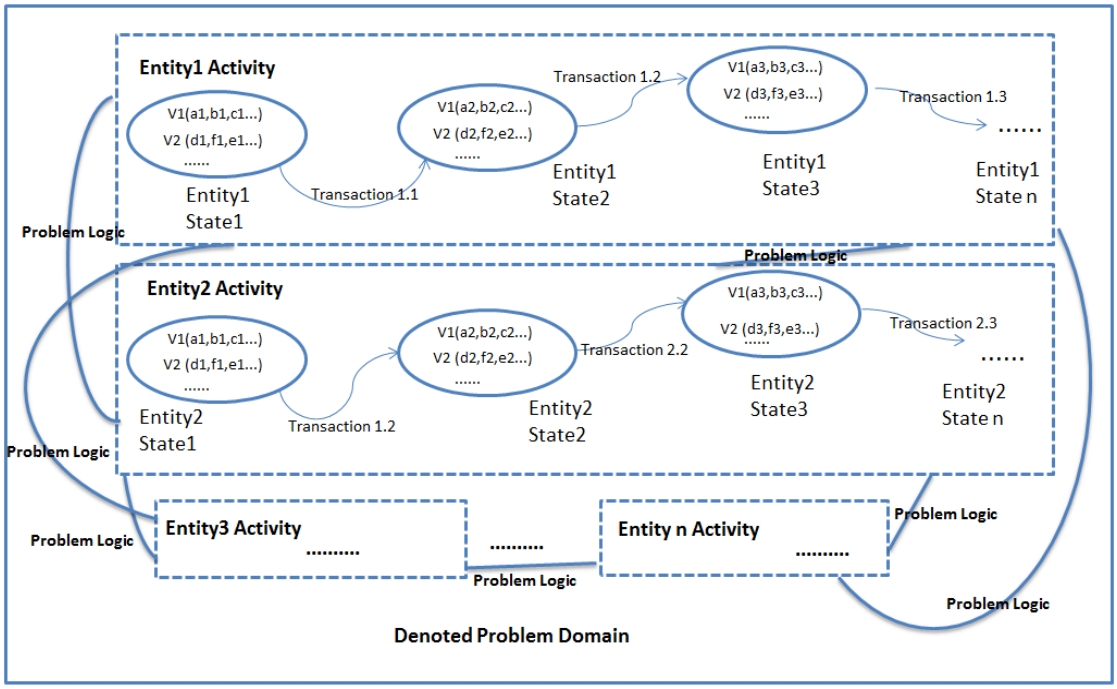

Figure 4. The Activity Graph of OPL

Definition 3: *A* ***Transaction*** *is one time status transferring by a certain entity, which is an atomic unit in OPL.*

Definition 4: *The* ***Activity*** *is a union of transactions within a certain denoted problem domain. Each activity contains at least one transaction.*

As a layer of OPL shown in Figure 1, an activity graph is formed to support the intelligent system response and awareness inference for the activity awareness in intelligent elderly care system, based on the denoted problem domain and problem logic. The activity graph is a cluster consisting of activities in the activity awareness system. Activity graph is the tool to support a system reference, directly targeting at the intelligent response. Transaction and activity are the two basic concepts to support the activity graph, which are introduced in this chapter, as definition 3 and definition 4 explain.

4.1. Ontology-based Activity Graph

The activity graph is established above the activity awareness annotating. Within this activity graph, the entity status is changing through time. Each status changing of the entity forms a transaction and the temporal and continual status changing forms an activity, as shown in Figure 4. Fore ample, the property named <pill_amount> keeps changing, in the Pill_box entity. Those transactions form an activity called X, just as an example. Actually, in OPL, we do not really care about how the system will understand the activity. The pill_amount changing of pill_entity might be because the patient takes pills regularly, or may be because the caregiver is allocating pills regularly according to the prescription. OPL just cares how the system will react towards the activity. Those system reactions will be also refer to some other activities, like the blood sugar level changing. Suppose the sugar blood

changing forms an activity called Y, where the system does not need to understand what this activity indicates. But when the activity X and activity Y are connected by certain problem logic Z, system will understand how to react that. For example, if the patient sugar blood increases and the pill amount of the pill_box is not changed, then the system will know that a health alarm should be sent to the caregivers and provide a notice that "the elderly did not take pills in time". Different from the conational activity recognition and pattern recognition, this OPL does not directly recognize what the activity is, instead it puts more attentions on how the system will react towards current activity, which can be called as indirect recognition. All the problem logic discussed above is within the denoted problem domain. For example, the problem logic Z would be meaningful, only when the denoted problem domain tries to contribute into the Pill taking activities and allocates a denotation to the system.

The cluster of activities may constitute a situation, which explains the user and surroundings over a period of time. Also those situations can be associated by the problem logic, where the heterogeneous activity resources can exchange and sharing the activity data between each other.

4.2. Ontology-based Problem-logic Driven System Intelligence

In OPL, all the activities (including all the sub elements within an activity: entities and transactions) will form a knowledge base, while all the problem logics can form a group of rules. Knowledge base and inferring rules are the two fundamental essences of the rules based reference approach [16]. In this research, the rule-based inferring language Jess [16] is used to validate the OPL, using with OWL. The ontology-annotated activity document and the logics will be mapped into Jess respectively.

As the discussion above, OPL uses Ontology to annotate the activities, where includes a classes, instances of the classes, the properties of classes and properties among those instances (including the data properties and ontology properties). While the status changes with time, the updated OWL file will be input for processing and inferring, where each file will annotate the activity with denotations. A scenario-based implementation prototype will be presented in Chapter 5 for validating and specific example.

5. Case Study

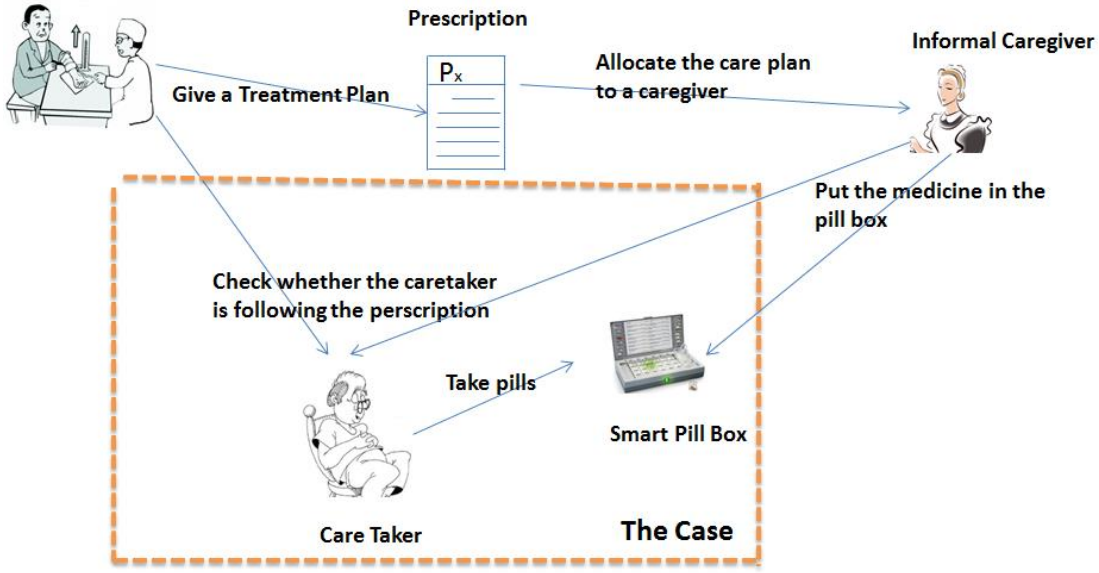

Figure 5. The Scenario Extracted from Salig++


```
Jess> (defclass medicine (is-a :THING) (slot name (type string)) (slot pilllocation (type string)) (slot takingamount (type integer)) (slot takingtimes (type integer)))
TRUE
Jess>  (make-instance medicinea of medicine (name "DBT") (pilllocation "pill1") (takingamount 3) (takingtimes 3))
<Java-Object:edu.stanford.smi.protegex.owl.model.impl.DefaultOWLIndividual>
Jess> (make-instance medicineb of medicine (name "AAD") (pilllocation "pill2") (takingamount 1) (takingtimes 2))
<Java-Object:edu.stanford.smi.protegex.owl.model.impl.DefaultOWLIndividual>
Jess> (make-instance medicinec of medicine (name "BAB") (pilllocation "pill3") (takingamount 2) (takingtimes 1))
<Java-Object:edu.stanford.smi.protegex.owl.model.impl.DefaultOWLIndividual>
Jess>  (mapclass medicine)
http://www.owl-ontologies.com/Ontology1400341742.owl#medicine
Jess>  (defclass perscription (is-a :THING) (slot patientname (type string)) (slot patientid (type string)) (slot age (type integer)) (slot medicinename (type string)) (slot alarmamount (type integer)) )
TRUE
Jess> (make-instance perscriptiona of perscription (patientname "Lily") (patientid "Se19600617-6366r") (age 60) (medicinename DBT) (alarmamount 1))
<Java-Object:edu.stanford.smi.protegex.owl.model.impl.DefaultOWLIndividual>
Jess> (make-instance perscriptionb of perscription (patientname "Lily") (patientid "Se19600617-6366r") (age 60) (medicinename AAD) (alarmamount 1))
<Java-Object:edu.stanford.smi.protegex.owl.model.impl.DefaultOWLIndividual>
Jess> (make-instance perscriptionc of perscription (patientname "Lily") (patientid "Se19600617-6366r") (age 60) (medicinename BAB) (alarmamount 1))
<Java-Object:edu.stanford.smi.protegex.owl.model.impl.DefaultOWLIndividual>
Jess>  (mapclass perscription)
http://www.owl-ontologies.com/Ontology1400341742.owl#perscription
```

Figure 6. Part of the Knowledge Base Written in Jess

In order to validate the OPL, the authors extract a use scenario from the research project Smart Assisted Living Involving Informal Care Givers (Salig++) [17] for a case study. Through this scenario case, OPL will be presented to show how the OPL can annotate the problem logic and respond to the changing situations by rule-based engine within a certain denoted problem domain. Also a prototype will be partly implemented to validate the approach by the case study.

5.1. A Scenario for Intelligent Elderly Care

It is a scenario in Salig++ where an elderly people with slight dementia problem needs to take medicine from the pill box, according to the prescription and treatment plan from the physical doctor. The smart pill box is equipped with sensors, which can periodically send data to the system to report the pill status inside the pill box. When a treatment plan or prescription is generated by the doctor or other professional health staff, informal caregivers will take the responsibility to put pills into the smart pill box. The caretakers usually live alone, and they will be reminded to take pills regularly from the pill box. But sometimes they may forget to correctly take pills in time, as the dementia problem.

A case can be extracted from the scenario shown in the orange box in Figure 5. Thus there are mainly two activities are included: caretaker taking pills and controlling the treatment status of the caretaker. The denoted problem domain is to control the status of the pills taken by the elderly. Also two problem logics can be defined: (1) If the pill amount in the box is larger than the prescription required, then it is that the caregivers put wrong pill amount in the box. (2) If the medicine is decreasing, then it is that the elderly is taking pills. (The pill box can be only used by the caretaker). (3) If the pill amount decreases slower than prescription required, then it is that the caretaker did not take pill correctly. (4) If the pill amount decreases quicker than prescription required, then it is the caretaker did not take pills correctly. Of course, those are just for a simplified scenario of concept-proof. In practical using, more problem logic will be generated, such as activities describing sugar blood level. As a contrast to the dataset driven approach, the denoted problem domain and problem logic are defined according to the current user situations that the elderly user needs to be cared for pill taking, where the amount and frequency are monitored. The system can avoid the one-fit-all and general activity denotation for the various situations.

```
(defrule medicinegaurd3
  (object (is-a medicine) (alarmamount ?x))
  (object (is-a medicine) (takingamount ?a&: (< ?a ?x)))
  =>
  (printout t "The medicine has not been taken enough" crlf))

(defrule medicinegaurd4
  (object (is-a pill_box)
  (username ?m)
  (pill ?c&: (< ?c 2)))
  =>
  (printout t "The caregivers, please provide mode medicine into the pill box of " ?m crlf))
```

Figure 7. Sample of the Denoted Problem Logic Written in Jess

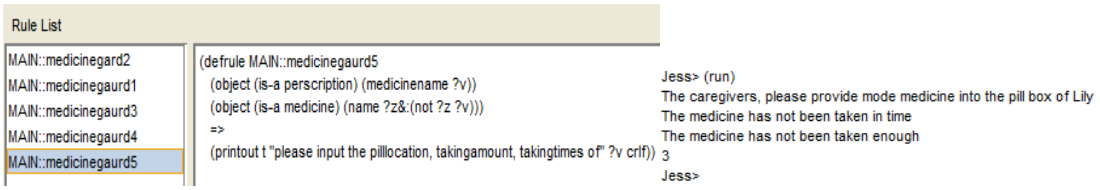

Figure 8. Depict the Denoted Problem Logic with Jess Rule (Left) and the Inferring for Intelligent Response (Right)

5.2. Ontology-based Problem-logic Driven Prototype

This scenario and the activities can be annotated by the OWL files, as shown in Figure 3. In this paper, a tiny prototype is written by Jess [18], which is one of the most light-weight and quick rule-based referring languages. In order to map the OPL into Jess logic [18], the researchers generates two parts of code, the facts and the rules. The class and properties in the classes are mapped into a Jess class [18], while each entity in the system will be mapped into the Jess instance [18], as shown in Figure 6. Also the Problem logic can be described with by Jess rules, as shown in Figure 7. Thus, each activity is under the problem logic control to form a knowledge base of activities, which is in the support of intelligent system response. The Jess code shown in Figure 7 generates the rules in JessTab [19] like shown in Figure 8 (left), as a proof-of–concept prototype. Run the prototype, a result shown in Figure 8 (right) can be output. If the user has not taken enough pills or does not taken in time, a message will be printed by the system for alarming or noticing.

6. Conclusion

Activity awareness is facing with a challenge to deal with the various activity denotations a various situations, using current dataset-driven approaches. In order to support the AA system to correctly denote and comprehend activities from different user situations, an ontology-based problem-logic-driven (OPL) approach is proposed to allow the activity awareness system for directly and seamlessly integrating with intelligent system response. The concepts of problem logic and denoted problem domain are introduced in support of OPL. OPL utilizes the problem logic within a certain denoted problem domain to let the system comprehend user activities rationally. In this paper, the authors utilize web ontology language (OWL) to annotate problem logic for activities in different denoted problem domains. Activity graph is set up based on the activity annotated with problem logic to integrate with rule based inferring. The OPL enables the system to respond towards users while the activity status and situations are changing. In order to confirm the validity of the proposed approach, a case is extracted from Salig++ (an intelligent elderly care research project) scenarios, where a proof-of-concept prototype written by Jess [16] is implemented to present how the OPL

can be utilized on an intelligent elderly care system. For future research, we will focus on exploring approaches to enhance AA system adaptivity, using OPL.

Acknowledgements

This research is supported by the European Union founded research project—Smart Assisted Living Involving Informal Care Givers (Salig++), which targets at supporting the self-care and informal care for the elderly citizens via multimedia and immersive participation approach within ubiquitous surroundings.

References

[1] J. E. Graham, K. Rockwood, B. L. Beattie, R. Eastwood, S. Gauthier, H. Tuokko and I. McDowell, "Prevalence and severity of cognitive impairment with and without dementia in an elderly population", The Lancet, vol. 349, no. 9068, **(1997)**, pp.1793-1796.
[2] J. Favela, "Behavior-aware computing: applications and challenges", Pervasive Computing, vol.12, no.3, **(2013)**, pp.14–17.
[3] H. B. Christensen, "Using logic programming to detect activities in pervasive healthcare", In Logic Programming, Springer Berlin Heidelberg, **(2002)**, pp. 421-436.
[4] H. B. Christensen and J. E. Bardram, "Supporting human activities—exploring activity-centered computing", In UbiComp 2002: Ubiquitous Computing, Springer Berlin Heidelberg, **(2002)**, pp. 107-116.
[5] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data", In Pervasive computing, Springer, Berlin Heidelberg **(2004)**, pp. 1-17.
[6] E. M. Tapia, S. S. Intille and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors", In Pervasive Computing, **(2004)**, Springer, Berlin Heidelberg, pp. 158-175.
[7] M. Tentori and J. Favela, "Activity-aware computing for healthcare", Pervasive Computing, IEEE, vol. 7, no. 2, **(2008**), pp. 51-57.
[8] S. Bechhofer, "OWL: Web ontology language, In Encyclopedia of Database Systems", Springer US, **(2009)**, pp. 2008-2009.
[9] P. Turaga, R. Chellappa, V. S. Subrahmanian and O. Udrea, "Machine recognition of human activities: A survey. Circuits and Systems for Video Technology", IEEE Transactions on, vol. 18, no. 11, **(2008)**, pp. 1473-1488.
[10] R. Poppe, "A survey on vision-based human action recognition", Image and vision computing", vol. 28, no.6, **(2010)**, pp. 976-990.
[11] T. Van Kasteren, A. Noulas, G. Englebienne and B. Kröse, "Accurate activity recognition in a home setting", In Proceedings of the 10th international conference on Ubiquitous computing, **(2008)**, September, pp. 1-9.
[12] E. Kim, S. Helal and D. Cook, "Human activity recognition and pattern discovery", Pervasive Computing, IEEE, vol. 9, no. 1, **(2010**), pp. 48-53.
[13] D. Riboni, L. Pareschi, L. Radaelli and C. Bettini, "Is ontology-based activity recognition really effective?", In Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on, **(2011)**, March, pp. 427-431.
[14] L. Chen, C. D. Nugent and H. Wang, "A knowledge-driven approach to activity recognition in smart homes. Knowledge and Data Engineering", IEEE Transactions on, vol. 24, no. 6, **(2012)** pp. 961-974.
[15] D. L. McGuinness and F. Van Harmelen, "OWL web ontology language overview", W3C recommendation, vol. 10, no. 2004-03, **(2004)**, pp. 1-22.
[16] K. Ilgun, R. A. Kemmerer and P. A. Porras, "State transition analysis: A rule-based intrusion detection approach", Software Engineering, IEEE Transactions on, vol. 21, no. 3, **(1995)**, pp. 181-199.
[17] Ambient Assisted Living Programme, The European join research project, http://www.aal-europe.eu/projects/salig/, **(2013).**
[18] E. Friedman-Hill, "JESS in Action", Greenwich, CT: Manning, **(2003)**.
[19] H. Eriksson, "Using jesstab to integrate protégé and jess", IEEE Intelligent Systems, vol. 18, no. 2, **(2003**), pp. 43-50.

Authors

Theo Kanter, he earned a Ph.D. in computer communications, from the Royal Institute of Technology. Theo has held a number of leading positions in telecommunications research. From 1999 to 2007, he was a senior scientist at Ericsson Research. He is now a professor at the Department of Computer and System Sciences at Stockholm University, where he leads research in adaptive and context-aware mobile communication, service architectures and self-organizing application infrastructures.

Rahim Rahmani, he is an Associate Professor of Computer Science at the Department of Computer and System Sciences of Stockholm University, where his research focuses on QoS and optimization for management of cloud computing infrastructure and wireless sensor networks. He has served as reviewer and in technical committees of international conferences. He is a member of the editorial board of International Journal of Wireless Networking and Communications.

Bin Xiao, he Bin received his M.sc in Information Processing Science from the University of Oulu Finland, during which Bin also worked as a research assistant at Mediateam Oulu. Now he is working towards the Ph.D. at Department of Computer and System Science in Stockholm University. Bin's research focuses on supporting dynamic associations of context relations between objects from visual and other sensing methods in Internet of Things.