Real-Time Activity Recognition and Assistance in Smart Classrooms

Vasileios Efthymiou[a], Maria Koutraki[b], Grigoris Antoniou[b]

[a] Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg

[b] Foundation for Research and Technology Heraklion (FORTH), Greece

KEYWORD

Ambient Intelligence
Smart Classroom
Activity Recognition
Context modeling
Real-Time Assistance

ABSTRACT

In this paper we present our work in a real-time, context-aware sys-tem, applied in a smart classroom domain, which aims to assist its users after rec-ognizing any occurring activity. We exploit the advantages of ontologies in order to model the context and introduce as well a method for extracting information from an ontology and using it in a machine learning dataset. This method enables real-time reasoning on high-level-activities recognition. We describe the overview of our system as well as a typical usage scenario to indicate how our system would react in this specific situation. An experimental evaluation of our system in a real, publicly available lecture is also presented.

1 Introduction

In a typical classroom, a lot of time and effort is sometimes spent on technical issues, such as lighting, projector set-up, photocopy distribution etc. This time could be replaced with “teaching time”, if all these issues were solved automatically.

There are many Smart Classroom systems that try to change the behavior of an environment in order to improve the conditions of a class. One of them is [O’DRISCOLL, 2008] that focuses on making real-time context decisions in a smart classroom based on information collecting from environment sensors, polices and rules. Another context aware system is [LEONIDIS, 2010] that supports ubiquitous computing in a school classroom.

In this paper, we present a system that assists instructors and students in a smart classroom, in order to avoid spending time in such minor issues and stay focused on the teaching process, by also having more studying material at their disposal. To accomplish this, we have taken advantage of the benefits that ontologies and machine learning offer, unlike other similar systems. The main problem that we intend to solve is:

***Problem**: Assist the users of a smart-classroom in real time, by supporting the teaching process.*

In order to achieve this goal, there is a set of sub-problems that need to be solved. The most important sub-problem is knowledge representation and specifically modeling the classroom environment.

Recognizing the current state of the classroom in real time is also a crucial part of the system. Assistance would be impossible if the state of the class, and therefore the current problems were unknown, or not recognized in time.

Finally, the assistance of users should be provided in such a way that it will not interfere with, or interrupt the teaching process. On the contrary, our goal is to solve the problems that interrupt the teaching process, hence making it more natural and continuous.

Ontologies play a pivotal role not only for the semantic web, but also in pervasive computing and next generation mobile communication systems. They provide formalizations to project real-life entities onto machine-understandable data constructs [KRUMMENACHER, 2007]. In our system, machine learning algorithms use these data constructs to conceive a higher level representation of the classroom environment.

1.1 Background

In this chapter we introduce some useful notions that will help the reader throughout this work. Our system is built on the notion of context. “Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves” [DEY, 2001].

To model the context we use OWL ontologies. According to [GRUBER, 1993] “an *ontology* is an explicit and formal specification of a conceptualization”. Typically, ontologies consist of a finite list of terms and the relationships between these terms. The terms denote important concepts (classes of objects) of the domain [ANTONIOU, 2004]. The relationships typically include hierarchies of classes. A hierarchy specifies a class *C* to be a subclass of another class *C′* if every object in *C* is also included in *C′*.

OWL (Web Ontology Language)[1] is a semantic web language, designed to represent ontologies. OWL is a computational logic-based language such that knowledge expressed in OWL can be reasoned with by computer programs either to verify the consistency of that knowledge or to make implicit knowledge explicit.

SWRL (Semantic Web Rule Language)[2] includes a high-level abstract syntax for Horn-like rules in both the OWL DL and OWL Lite sublanguages of OWL. A rule in SWRL has the form

$$B_1, \ldots, B_n \; \square \; A_1, \ldots, A_m$$

where the commas denote conjunction on both sides of the arrow and $A_1, \ldots, A_m, B_1, \ldots, B_n$ can be of the form C(x), P(x, y), sameAs(x, y), or differentFrom(x, y), where *C* is an OWL description, *P* is an OWL property, and x, y are variables, OWL individuals, or OWL data values [ANTONIOU, 2004].

Propositionalization is defined as a representation change from a relational representation to a propositional one [KRAMER, 2000]. Propositionalizations can be either complete or partial (heuristic). In the former case, no information is lost in the process [DE RAEDT, 1998]; in the latter, information is lost and the representation change is incomplete. Our approach could be considered as partial propositionalization, since we do not cover all the semantics from an ontology, but only some specific properties.

Case Based Reasoning (CBR) [AAMODT, 1994] seems to be an efficient approach in reasoning when it comes to AmI systems. CBR is typically described as a cyclical process comprising the four REs, as illustrated in Figure 1:

1. REtrieve the most similar case or cases,
2. REuse the information and knowledge in that case to attempt to solve the problem,
3. REvise the proposed solution if necessary, and
4. REtain the parts of this experience likely to be useful for future problem solving.

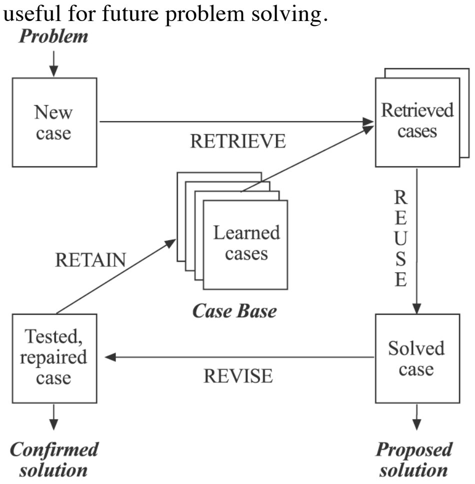


Fig. 1. The four REs of a Case-Based Reasoning cycle.

Some signal segmentation approaches are described in details in [TAPIA, 2008]. The one we chose is “not overlapping sliding windows”, since this implementation is simpler, faster and activities occurring in the edge of a time window are rather rare. Especially in a smart classroom, activities usually last longer than other domains and their number is significantly lower. The length of the time window (10 seconds) is chosen based on the nature of the activities and experimental results.

The rest of the paper is structured as follows: In Section 2, we present a motivating scenario for a smart classroom. Section 3 is about works similar to our system. Section 4 describes the design of our system in details. In Section 5 we analyze our activity recognition approach and in Section 6 we present and comment our experimental results.

[1] http://www.w3.org/TR/owl-features/

[2] http://www.w3.org/Submission/SWRL

2 Motivating Scenario

Scenarios in our work express the activities that can take place in a classroom e.g. lecture, exam. In order to identify these activities, a series of simpler events should be applied, in each case. After that, we undertake to assist the particular activity.

A typical scenario in a classroom is a student presentation. In this scenario, a student is giving a project presentation in the Smart Classroom. The audience consists of other students and the teacher.

Each person in the audience carries an electronic device (smart-phone/laptop) with a smart-classroom application. This application has access to the profile of the users, their personal calendars and to a GPS, if available. The calendar of the classroom is checked for lecture scheduled at this time and the audience members' personal calendars are checked for participation in this presentation, in order to know that everyone is in the right room. The lights and the projector are currently on. A student stands near the display of presentation. This student's profile matches the "presenter's" profile in the calendar of the classroom. The teacher and the rest of the students are seated and the noise level is low, indicating that there is a teaching activity in the classroom. Knowing these simple facts, "Student Presentation" activity is identified.

After the identification of the activity, the presentation is assisted by lowering the lights and adapting the projector's input in students' and teacher's devices. The microphone in the presenter's stand is activated and the audience microphones are deactivated. The audio output of the classroom is connected to the microphone and the audio output of the presenter's laptop. The audience has the chance to record the whole, or part of this presentation. In case a member of the audience has different language preferences and a translator is available, the audio output of this member's device is set to be the translator's output. The current slide of the presentation is also provided to this member translated. The teacher has the ability to take notes and grade the presentation in real time.

After the presentation, the audience has some questions. The lighting of the room is set to normal again. A student rises and makes a question. This student's microphone is activated and in case his language preferences are different, his question is translated. When the presentation is over and everyone has left, the classroom is set to its default state.

3 Related Works

Typically, Smart Classroom systems aim to modify the context of the classroom in order to improve the conditions of a class. [O'DRISCOLL, 2008] focuses on making real-time context decisions in a smart classroom based on information collected from environment sensors, policies and rules. [LEONIDIS, 2010] is a context-aware system that supports ubiquitous computing in a school.

Many systems focus on distance learning for a Smart Classroom domain. One of them is [PISHVA, 2008] that provides an overview of the technologies used in a smart classroom for distance education. It gives an example of a successful implementation of distance education technology that had been used to link university campuses in Japan and in USA. In another work [SHI, 2003] multimedia communication systems let teachers and students in different location participate in the class synchronously. Teachers can use multiple natural modalities while interacting remotely students in order to achieve the same result as a classroom with students physically present.

Beside the works in the Smart Classroom domain, there are several intelligent systems focusing on other domains. [CHEN, 2004] presents a smart meeting room system that explores the use of multi-agent systems, Semantic Web ontologies and reasoning. This work provides services and information to meeting participants based on their current situation needs.

Apart from the works that focus on improving the environment conditions in an intelligent classroom, there are many works that describe the context representation in ubiquitous environments. [HONG, 2008] presents a context-aware management architecture to support learning environments. It describes an ontology for context representation in an intelligent school domain. In [MOHAMMAD, 2007], the authors propose a context ontology model for a smart meeting space. In this work, an upper level context ontology is designed, which analyzes in seven major concepts that can be reused to build an ontology model for smart meeting space environment. There are also works that are focused on modeling specific parts of the context in intelligent environments. In the work of [GOLEMATI, 2007] a general user profile ontology is used to model user characteristics in upper level classes and it can also be used to describe several other domains.

4 Architecture for Building AmI Systems

In this section we provide a detailed description of the different tasks and components that constitute our system. The data flow is illustrated in Figure 2. Our system consists of four components: the Classroom, the Knowledge Base (KB), which is an OWL ontology, the Activity Recognition system and the Classroom Assistant. The Case Base can be seen as part of the Activity Recognition system.

A simple description of a complete cycle of our system is the following, as depicted in Figure 2. Nodes represent the different components and arrows represent the data flow:

1. Data from sensors and from services of AmI Sandbox [GRAMMENOS, 2009] are stored in the Context KB, i.e. an OWL ontology. These services provide functionality like localization searching, speech recognition, or simply providing the sensor data. In our scenario we assume that the localization searching service provides data about students' and teacher's current locations, which are stored into our ontology as well as data from lights sensors or data from RFID sensors for people identification. This step runs continuously, in order to avoid missing events/information.
2. SWRL rules are used for a first level of reasoning, to create *Simple Events*. In the "Student Presentation" example of Section 2, some *Simple Events* are: "student stands near the display of presentation", "teacher sits", "there is a scheduled lecture in the classroom" etc.
3. The *Simple Events* that occurred within a timeframe are passed to the Activity Recognition system periodically. The Activity Recognition system parses these *Simple Events*, to recognize the current Activity.
4. The Activity Recognition system loads the past cases from a database with correctly recognized activities (Case Base). Using these cases as a machine learning database, it identifies the most possible current Activity.
5. The Activity that is identified in step 4 is written in the Case Base as a new Case. In parallel, the same result is also passed to the Classroom Assistant system.
6. Depending on the current Activity, given as input, the Classroom Assistant changes the context. In our scenario it lowers the lighting, adapts the presentation file in students' and teachers' devices, activates the presenter's microphone etc.

The Classroom in this design can be replaced by any other smart-space, which would make Figure 2 generic for AmI systems. For example, we are currently using the same design to build another system, used in the Ambient Assisted Living domain. Instead of a Classroom, we have a Smart-Home and instead of a Classroom Assistant, we have a Smart-Home Assistant.

For the rest of this section, we present a brief summary for each component of our system. In the place of the Classroom, we describe the AmI Sandbox, which we use to simulate the environment of a real classroom. SWRL rules are used for both sensor-handling and assisting the final users.

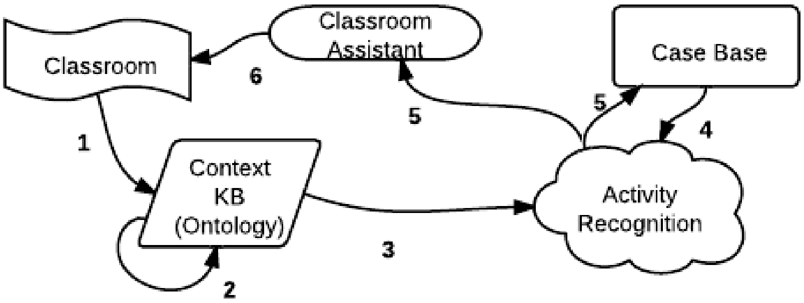


Fig. 2. System design: Sensor data are stored in the ontology (step 1). A first level of reasoning is applied to identify Simple Events (step 2). Ontology data are parsed from the activity recognition system (step 3) and the current activity is recognized, stored and provided to the classroom assistant (steps 4 and 5). The Classroom Assistant possibly changes the state of the Classroom (step 6).

4.1 Modeling Context

In this session we propose a context ontology model for an intelligent university Classroom that responds to students' and teachers' needs. Our context ontology is divided into two hierarchical parts, upper-level ontology and low-level ontologies.

The *Upper Level Ontology* or *Core Ontology* captures general features of all pervasive computing domains. It is designed in a way that can be reused in the modeling of different smart space environments, like smart homes or smart meeting spaces. The Core ontology's context model is structured by a set of abstract entities like Computational Entity, Person, Activity, Location, Simple Event and Environment (Figure 3). All these entities are widely used in context representation through ontologies, except Simple Event entity. Simple Event entity aims to capture knowledge obtained from reasoning on sensors data. More specifically in the class *CompEntity* and in their subclasses we store information that is taking by sensors.

Those data are representing the current situation of the sensor and generally the context. We can have information about the situation of a specific device e.g. if the projector is turned on or if a specific window is open. Furthermore we can take information about the persons that are located in the smart classroom through RFID sensors. We can know if a particular "rfid tag" is enabled and connected to a specific rfid reader. We apply reasoning to all the information that is stored to *CompEntity* in order to have more useful results that we can use to the scenarios. After the reasoning part on sensors data the results are stored in the *Simple Event* entity.

The way that Simple Events are represented is like 'Projector's status is "on"' or "Teacher is in front of the smart board' or "3 Persons located in Classroom".

The *Low Level* or *Domain-Specific Ontologies* are based on upper level ontology and specified by the domain. In our case the domain is an intelligent classroom in a university campus for that reason the low level ontologies are adjusting on it. Some of the domain-specific ontologies that we propose are *Person*, *Location*, *Calendar* and *Lecture* ontology.

The ontology *Person* can describes human beings that have academic nature e.g. professors, Bachelor students, Master students etc. We can store all kind of information that can describe a user of a smart classroom. Information about user's background, education, academic experience, also we can store information about user's preferences, interests or personal profile. More over in our *Person* ontology, information like hearing or visual impairments that users may have can be expressed.

We develop the ontology *Calendar* in order to store the information about user's schedule e.g. in which class has to participate on Monday from 4pm to 6pm. We also can to store information about the schedule of the smart classroom in *Calendar* ontology. For example, which lesson is gone take place at the classroom on Friday between 2 pm and 5 pm. How many will be the participants or what will be the subject of the lecture.

Into ontology *Lecture* can be stored information about the lectures that can take place in the Smart Classroom. For each lecture we can have information about the subject, the professor that giving the lecture and also the learning material that is needed. All of the ontologies are expressed in OWL.

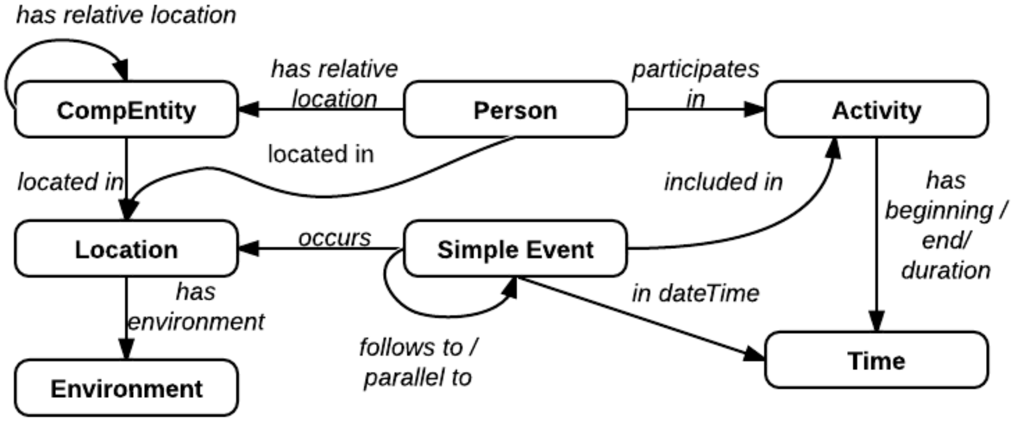


Fig. 3. Core ontology.

4.2 Reasoning: SWRL

In our implementation we try to transform our scenarios for smart classroom into rules. This rule based approach is implemented by using Semantic Web Rule Language (SWRL). The first step is to capture data from sensors and services from Smart Classroom and store them into the ontology (e.g. status of devices). Because of the nature of those data (sensors' data) SWRL rules are applied on them in order to produce more useful results for our scenarios. After the reasoning part the results are stored into the *Simple Event* class. SWRL rules are also applied for user assistance part. After the activity recognition part, the aim is to manage to assist the activity that takes place into the Smart Classroom in order to be easier for the users to participate on the activity.

4.2.1 Sensor handling

As discussed in the previous sessions, the data that we take from the sensors are in a very basic form. In order to draw more useful conclusions we have to process them. We manage to do that by using SWRL rules. Here, we provide some examples of such SWRL rules:

Rule 1: *Person:Teacher(?t) ^ Device:SmartBoard(?b) ^ Core:hasRelativeLocation(?t,?b) ^ Core:inFrontOf(?t,?b) ^ →*
Core: isActivated(Teacher in front of Board, "true")

Rule 1 computes if the Teacher is in front of the Board.

Rule 2: *Location:SmartClassroom(?c) ^*
Environment(SmartClassroomEnv) ^
Core:hasEnvironment(?c,SmartClassroomEnv) ^
Core:noiseLevel(SmartClassroomEnv,?noise) ^
swrlb:greaterThan(?noise,80) →
Core:isActivated(High Level Noise, "true")

Rule 2 draws a conclusion about the noise level by taking as input the sensor data about the noise.

Rule3*: Lights(ClassroomLights) ^*
Core:status(ClassroomLights, "on") ^
lightsLevel(ClassroomLights, "high") →
isActivated(Classroom_Lights_High_Level, true)

Rule 3 draws a conclusion about the level of the lights in the Classroom. Taking as input data from the lights sensor, it activates the particular Simple Event, in this case "Classroom_Lights_High_Level".

Rule 4: *SmartClassroom(SmartClassroom1) ^*
hasCalendar(SmartClassroom1, ?c) ^
Calendar:hasCalendarEvent(?c, ?evnt) ^
hasStartTime(?evnt, ?st) ^ hasEndTime(?evnt, ?et) ^
swrlb:equal(?st, CurrentTime) ^
Calendar:description(?evnt, "Lecture Slides") →
isActivated (ClassCalEvent_LectureSlides, true)

Rule 4 does not take as input data from sensors. Instead, it uses data that are stored in the *Calendar* ontology and it deduces if the event that is taking place in the classroom is a lecture with slides.

Rule 5: Projector(Projector1) ^
Core:status(Projector1, "on") →
isActivated(Projector_is_On, true)

In Rule 5, one of the simplest SWRL rules, data from projector1 are considered in order to represent as a Simple Event if proector1 is currently on.

4.2.2 Assistance

The point of the work that we described in the previous sessions is to manage to assist the Activities that can take place in a Smart Classroom. The kind of assistance that we can provide is based on the context. The adaptation of the context to humans needs. The general purpose of all the cognitive systems is to be able to change the environment conditions based on the current conditions that are given as input. That happens in order to make it easier for the users to focus on the main activity e.g. Presentation and not to be disturbed by changing the environment by themselves. More specifically, if the lights in the Smart Classroom are off and the lighting level from the outside environment is also low, the Smart Classroom Assistant will turn the lights on. Also if the temperature in the classroom is higher or lower than the normal it will adjust the temperature by operating the air conditioning.

In our particular domain, apart from the environmental conditions that can be adapted, we can also assist the activities by using information that are stored in the ontology and are related to the particular activity. For example, if the activity that takes place in the Smart Classroom is a lecture presentation we can find all the information that describes this particular lecture in the Lecture ontology. After that, we can assist each particular user by providing the file of the presentation or other additional notes related to the subject of presentation. The assistance will be different for each user according to his profile. If someone has a low level background on the presentation's subject, the Smart Classroom Assistant will provide him also with additional material tutorials, or basic notes about the subject.

In order to accomplish the adaptation of the environment for the users' needs, we apply reasoning once more to all the data that are stored into our ontologies. After this part, we configure the devices of the Smart Classroom with specific values, for example by setting the value "on" on the projector through the *projector service*.

4.3 Activity Recognition

For this part of our system, we used (an adjustment of) an activity recognition system [EFTHYMIOU, 2012], based on machine learning, described briefly in Section 5. After the SWRL rules trigger, the resulting *Simple Events* that occurred within a time frame are sent to the Activity Recognition system, building an unsolved case. One or no activity is then recognized and the solved case is added to the case base.

Based on a given set of activities that are to be recognized, a case base is initially created and classified manually. It is essential for this dataset to be a product of real observations and not just random cases. Instead of implementing the machine learning algorithms, we use WEKA [WITTEN, 2011] with default parameters, so our dataset is stored as an *arff* file.

In our domains each case represents a time-frame typically lasting 10 seconds, but generally depending on the specific domain. This means that for an hour of reasoning we need about 360 cases/lines and for a day of reasoning we need around 8640 cases/lines in our case base/arff file. Soon it becomes obvious why it is important to have some data reduction algorithms. WEKA's Resample filter can perform this task and return a new arff file as a result.

4.4 AmI Sandbox

The Institute of Computer Science of the Foundation of Research and Technology – Hellas (FORTH - ICS) has initiated a long RTD AmI program [STEFANIDIS, 2008] aiming to develop pioneering human-centric AmI technologies and Smart Environments, capable of "understanding" and fulfilling individual human needs. Under this program it created a laboratory space of about 100m^2 comprising six rooms, aiming to provide researchers the opportunity to bring along and share their know-how and resources in order to obtain hands-on experience and experiment in a highly flexible setting. In this space, various AmI technologies and applications are installed, integrated and demonstrated, and multiple ideas and solutions are cooperatively developed, studied and tested [GRAMMENOS, 2009].

For our system's needs we are going to use the Smart Office prototype environment. In this environment there are a smart office, a smart board, as well as a set of sensors of many kinds. Using this environment for the implementation of the scenarios in real time we will be able to draw conclusions about the efficiency and the accuracy of the system.

AmI services are defined through a tool called Idlematic which creates and keeps service interfaces. All services communicate through FAMINE middleware, responsible for creating, connecting and consuming these services. It is CORBA-based and provides support for C++, Java, .NET, Python, and Flash/ActionScript languages.

Sensor data are handled like mouse/keyboard events in Java. Some (already implemented) services can be used to catch the sensor signals and inform the user about the type of signal and what it means (by defining an event). In the AmI Sandbox there are sensors for sound, image, lighting, interaction, "smart" devices and more, including cameras, RFID readers, IRIS scanner, door-chair controllers, touch screens, projectors, PDAs, speakers and workstations. When an event occurs then it can be caught using event handlers. These event handlers are used to store the sensor data in the respective ontology.

Apart from the event handlers, Idlematic and Famine also provide the ability to change the context, by calling some methods. These are methods that interact with the environment (such as the ability to close a door, turn a device on etc.)

5 Activity Recognition in the Presence of Ontologies

Since ontologies offer ways of expressing concepts and relationships between them, we found it interesting to exploit such expressiveness and assist machine learning, in order to recognize the current activity of the users of a smart space.

When it comes to real-time activity recognition, there is a need for accuracy as well as speed. Our approach aims to take advantage of the rich expressiveness that ontologies can offer and provide solid answers, using machine learning algorithms, like Support Vector Machines (SVMs) or Bayesian networks. The key factor that led to the design of a new system, using machine learning, is the lack of speed observed in the already existing systems that use algorithms such as kNN. Apart from that, the robustness and accuracy of Bayesian networks led to a system faster and more accurate than other systems that we are aware of. Coping with sensors usually means missing data and SVMs – our initial choice - lack in this field. In [JAYASURYA, 2010] it is claimed that - for a specific biological domain – BNs outperform SVMs when data are missing and perform almost the same, when all data are available. Even if we have not tested our system with missing data, this is a typical phenomenon when dealing with sensors and we wanted to be prepared.

5.1 Similar CBR tools

In this paragraph we describe some of the most similar Case Based Reasoning (CBR) tools that we found in the literature. Not all of them focus on activity recognition, but we found them very useful for the design of our system.

The most popular Case Based Reasoning (CBR) tool that supports ontologies is JCOLIBRI [RECIO-GARCIA, 2007, 2008]. As most of the CBR tools do, JCOLIBRI uses the k Nearest Neighbours (kNN) algorithm to classify a new case, so it bares the problems of kNN.

Another promising tool, CREEK [AAMODT, 1991] - later renamed as AmICREEK – is not publicly available, although some case-evaluation studies have been published [KOFOD-PETERSEN, 2009].

The most recent and similar tool that we found is SituRes [KNOX, 2010], a case-based approach to recognizing user activity in a smart-home environment. Even if this approach improves learning, however there is a great weakness; the accuracy of the system is poor, as admitted also by the authors. SituRes is based on one of the publicly available datasets that use PlaceLab [INTILLE, 2005].

PlaceLab is a 1000-square-foot sensor-rich home environment, built by MIT as a shared research facility. Unfortunately, the PlaceLab ontology is not in a form that our system can exploit. However, it is our intention to expand our system and make it capable of processing a wider range of ontologies.

COSAR [RIBONI, 2011] is a hybrid reasoning system that uses ontologies and ontological reasoning combined with statistical inferencing. In this paper a *historical variant* is introduced, which considers the past few cases when deciding on the current case, taking advantage of the fact that persons tend to perform the same activity for a certain lapse of time before changing activity. COSAR is an offline reasoner, meaning that it does not run in real-time.

5.2 Exploiting the semantics of ontologies

The input of the complete system is an ontology with instances, from which the user has to define the terms (classes) that describe the attributes, the term that describes the solution and the term where the cases are stored. For example, consider a simple ontology, where *Case, Activity, Location, Time, Winter, Summer, Autumn, Spring, Indoors*, *Outdoors*, *FirstFloor* and *SecondFloor* are classes. As expected, *Winter, Summer, Autumn* and *Spring* are subclasses of *Time*. *Indoors* and *Outdoors* are subclasses of *Location* and *FirstFloor* and *SecondFloor* are subclasses of *Indoors*. In this ontology there should be some *Case* instances like the following:

```
<Case rdf:ID="Case74">
   <has-Activity rdf:resource="#Cooking"/>
   <has-Time rdf:resource="#July"/>
   <has-Location rdf:resource="#Kitchen"/>
</Case>
```

where *July* is an instance of *Summer* and *Kitchen* is an instance of *FirstFloor*. The user should define that terms *Time* and *Location* describe the attributes of the problem, term *Activity* is used to describe the solution and each case is stored as an instance of term *Case*. The suggested solution to the problem of grasping the hierarchy information of an ontology and storing it as attributes, is keeping the whole path of each instance in Boolean values. For example *Case74* of the example above, would be stored as:
Cooking, July,0,0,0,1,Kitchen,1,0,1,0

The Boolean values following *Kitchen* mean that it belongs to *FirstFloor* and *Indoors* and not *SecondFloor* or *Outdoors*. In other words, after an attribute value, we store one Boolean number for each of its subclasses, representing that the subclass belongs or not to the path that leads to the instance value, accordind to the following formula:

$\forall S \in subclasses(C)$
$value(S) = \begin{cases} 1, if\ S \in superclasses(a) \\ 0,\ else \end{cases}$, where
$C \in range(P),\ P \in properties$ and
$a = instance(C)$.

In our first version of the activity recognition system, applied in the smart classroom domain, we used ontologies as described above and the results were satisfying. For facilitating the evaluation process we developed a simpler but quicker version without using ontologies. In this version, events occurring within a timeframe are sent as plain text (and not within an ontology) to the activity recognition system, and added as a case to the case-base. The accuracy of the simpler version was close to the first version, but time performance was better, as expected.

5.3 Reducing the dataset

One of the four challenges in CBR [KOFOD-PETERSEN, 2006] is coping with the vast number of cases, namely removing some cases occasionally to limit memory needs. We used WEKA’s *Resample* filter, which reduces the dataset to a percent (we have set this percent to be 50%) of its initial size, with zero bias, by removing random cases. The crucial part of this filter is that it keeps the same ratio for the distribution of cases, depending on their class attribute.

Reducing the dataset is not necessarily limited to reducing the cases. We noticed that with our “full-path” approach to exploiting the semantics of an ontology, we also gather some redundant information for each case; values that are always the same, given some other values. For example if a student can only be described as undergraduate, or postgraduate and we know that a certain student is not an undergraduate, we can safely conclude that he is a postgraduate student. In this approach we take as granted that only “leaf” classes can have instances, which means that if a class has subclasses, then direct instances of this class are not taken into consideration. To model this attribute reduction, we use the following rule:

```
for each subclass S of a given property’s range class
    if S only has one father F, and F is marked
        add S to the attributes
    else
        mark F
```

This removes only one, the firstly-found subclass of each “level”. By the term “level” we mean the siblings of the same father (only its direct subclasses).

5.4 Limitations

The main limitations of the activity recognition system are the inability to recognize more than one activity at a time, or the activities of more than one person.

Every case in our scenarios has only one solution. This means that no parallelism is achieved when it comes to activities. For example, a teacher might be answering questions and giving a lecture at the same time. Our system would (ideally) return only one of these activities as a result for a given timeframe.

Similarly, the returned activity is only referring to one individual. In an environment with more than one individuals, like a smart classroom, the activity can be described as a “team” or “general” activity, like slide presentation, lecture etc. This means that we have not found a way to return one activity for each of the individuals present.

Another deficiency of the presented work is the lack of a higher-level reasoning based on the returned results. This could include different level of details, when it comes to results, conclusions on the results (e.g. the user is acting strangely, in confusion etc). However, we found it impractical to perform such kind of reasoning on a real-time system.

6 Experimental Evaluation

In the absence of a real dataset for a smart classroom, we decided to create one, in order to evaluate our system. The precision of our system is mostly based on the activity recognition's precision, since everything else is rule based. Therefore we present here the evaluation results for a dataset that we built based on our observations on a publicly available video from a lecture[3].

Our observations – which act as sensor data – include the position of the lecturer, the lighting, the persons that speak etc. In this video the activities observed are four: lecture with slides, lecture with whiteboard, question and conversation. A 10 fold cross validation based on this dataset is illustrated in Tables 1 and 2.

Total Number of Instances	326	
Correctly Classified Instances	313	96.0123 %
Incorrectly Classified Instances	13	3.9877 %

TP Rate	FP Rate	Precision	Recall	Activity
0.995	0.077	0.959	0.995	lecture_slides
0.955	0.008	0.977	0.955	lecture_wb
0.955	0.007	0.913	0.955	question
0	0	0	0	conversation

Table. 1. Detailed accuracy by class

classified as →	lecture slides	lecture whiteb	question	conversation
lecture slides	208	1	0	0
lecture whiteb	4	84	0	0
question	1	0	21	0
conversation	4	1	2	0

Table. 2. Confusion matrix

Table 2 can be read as "y activity was actually classified as x activity". It would ideally contain zeros only in the non-diagonal positions. So it appears that no "conversation" was classified correctly. Further evaluation experiments of the activity recognition system have been presented in [EFTHYMIOU, 2012].

The time performance of our system is based on the performance of SWRL reasoning time plus the performance of the activity recognition system. As stated in §4.3, we have set a typical reasoning cycle (case) of 10 seconds. This means that for a typical activity recognition dataset we would need around 43000 cases, which correspond to 5 (working) days of data, namely a week. We have reproduced the same 326 cases acquired from the video to create a 43000-case-large dataset, just to simulate the time performance of our system, ignoring accuracy. The average time performance of the activity recognition system is 0.757401 seconds in a rather outdated machine[4]. Similarly, the average time performance of SWRL rules is 0.686215 seconds. Summing these numbers, we get an average cycle processing time of 1.443616 seconds.

6.1 Demo

In order to run a real-world demonstration of our system, we have used the AmI Sandbox, described in Subsection 4.4. The data are real, and they were not based on assumptions. Of course, the ideal evaluation of this system would require its use in a real classroom with people that will be its final users, but this demo is the closest that we could get. To be more accurate, the data are based on the physical presence of a human in the AmI Sandbox, who acts according to a scenario. This scenario is designed based on an initial, manually created dataset. Another way to create a dataset would be to record the data of a user's arbitrary actions and then label each "timeframe" with an activity. We preferred the manually created dataset, for time's and human effort's sake.

This demo is based on the ontology presented in Subsection 4.1, in which *Activities* are recognized based on *SimpleEvents*. The available resources (that could be useful for us) in the AmI Sandbox at the time of the demo were:

- a hospital bed, capable to recognize whether someone is on it or not
- four televisions, capable of returning and setting their state (on/off, volume, channel, etc)

[3] http://videolectures.net/mlss08au_hutter_isml. Part 2

[4] Intel Pentium 4, 3.40GHz, 2GB RAM, with Microsoft Windows XP Professional SP3, using NetBeans IDE 7.0.1.

- window blinds, capable of returning and setting their state (open/closed)
- two projectors, capable of returning and setting their state (on/off)
- lights, capable of returning and setting their intensity
- RFID readers, capable of returning and (identifying) if an RFID tag is close

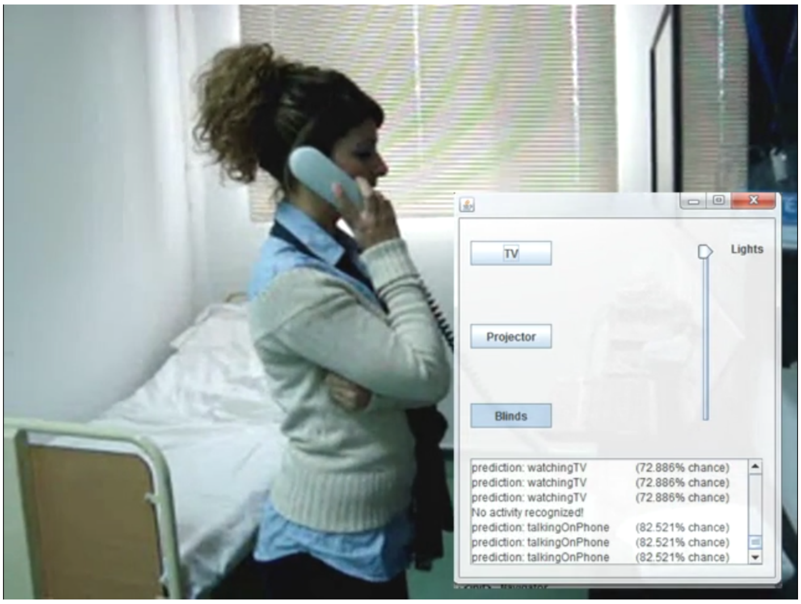
Unfortunately, we had to build our demonstration using only these resources, so the activities presented are limited and not all related to our motivating scenario. However, our intention is to show that real-time activity recognition with our AmI architecture is feasible. The activities that take place in the demonstration are:

- sleeping
- watchingTV
- presentation
- talkingOnPhone

The sensor data are stored in the ontology and then periodically parsed for activity recognition. The only non-obvious activity is "talkingOnPhone", based on the above sensors. In order to recognize whether the phone is on/off we used the RFID readers; an antenna (RFID reader) was stuck under a desk and a magnetic sticker (RFID tag) to the earpiece of a phone. The phone was placed on top of this desk, just above the reader. This way, when the earpiece was picked, the RFID tag on it was away from the reader and we could assume that the phone was on.

The demo is recorded with the help of two videos, a screenshot of which is depicted in Figure 4. The first video shows a person in the AmI Sandbox doing some activities, following a scenario. Initially she enters the room and (after a very tiring day at work) decides to rest for a while. Therefore she lowers the window blinds and lies on bed. But after so much tension she cannot sleep and decides to stand up, turn the lights and the TV on. She watches a music show for a while, but then the phone rings. She turns the TV off, and picks the phone up. It's her boss, asking her to review the presentation that she will have tomorrow! She decreases the lights' level and walks into the main room. There, she turns the projector on and reviews her slides presentation.

The second video is a recording of the Graphical User Interface (GUI) controller that appears on the screen of the laptop that this person holds. In this controller there are three toggle buttons for turning on and off the TV, the projector and the window blinds. There is also a slider for setting the lights' intensity. In the bottom of this GUI there is a text area in which the current activity and its likelihood are printed, as well as the last activities. This is a GUI that is only created for the needs of the demo, and it is not a part



```
prediction: watchingTV          (72.886% chance)
prediction: watchingTV          (72.886% chance)
prediction: watchingTV          (72.886% chance)
No activity recognized!
prediction: talkingOnPhone      (82.521% chance)
prediction: talkingOnPhone      (82.521% chance)
prediction: talkingOnPhone      (82.521% chance)
```

Fig. 4. A screenshot from the demo. On the lower right corner is the GUI available to the user for the needs of this demo. The current activity – along with the likelihood that it is actually happening - is the last shown in the text area of the GUI.

of the system. When the likelihood is below a threshold (70% for this demo), no activity is returned and the text area reads “No activity recognized”. This occurs usually in the transition between two activities.

The second video starts when the person entering the room starts recording the screen, but the system was initialized before that. That explains why the text area is not initially blank, but contains some “No activity recognized” cases. Figure 4 shows a screenshot of the video, some seconds after the phone is picked up.

Start Time(sec)	End Time(sec)	Activity
0	21	no activity
21	39	sleeping
39	57	no activity
57	75	watchingTV
75	78	no activity
78	90	talkingOnPhone
90	140	presentation

Table. 3. Activities occurring in the demonstration of the activity recognition system

The video[5] lasts 140 seconds and the first activity (sleeping) occurs on the 23rd second. The timeframes for this demo last 3 seconds. The results of the demo were really satisfying, since all activities were recognized correctly and instantly. There is only one case which is recognized falsely and that is when the phone is off, the suggested activity is still “talkingOnPhone”. This is due to the sensor signal that lags for almost 1 second. Therefore, if we start measuring the accuracy from the 21rst second (the start of the first timeframe in which an activity occurs) until the 138th and divide the time into 3-second timeframes, we have a total of 39 timeframes (the recognition result for the 40th timeframe does not appear in the video, since it stops at the 140th second, one second before the 40th timeframe ends). Since 38 of them are correctly recognized activities (including the recognition of no activity), the accuracy of the system is 38/39 = 97.44%. Table 3 summarizes the demo, illustrating the activities that occurred and at which timeframe.

7 Conclusions and Future Work

In this paper, we have introduced the use of a real-time AmI system in a smart classroom. We have presented how an ontology can be used to model the context in a smart environment and how we can take advantage of this modeling to assist activity recognition.

With a simple scenario, we have illustrated how such a system could assist its users and we have also provided experimental results on the performance. Although the first results are promising, we still have some work to do in order to test our system in a real smart classroom environment.

Apart from that, we have already started to work on the Ambient Assisted Living domain and particularly on the assistance of the elderly. We also plan to extend our Activity Recognition system, in order to grasp more of the information that ontologies can offer, reduce the case-base’s size efficiently and finally verify which activities were recognized correctly.

Acknowledgment

We would like to thank Ioannis Hryssakis and Dimitra Zografistou for their support and advice in the preparation of this paper, as well as their eagerness to help whenever needed.

[5] http://www.ics.forth.gr/isl/_videos/Real_time_activity_recognition_in_AmI.avi

8 References

[AAMODT, A. 1991] AAMODT A. A knowledge-intensive, integrated approach to problem solving and sustained learning. Dissertation, University of Trondheim, Norwegian Institute of Technology, Department of Computer Science, University Microfilms PUB 92-08460, 1991.

[AAMODT, A. 1994] AAMODT A, PLAZA E. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications. IOS Press, vol. 7: 1, pp. 39-59, 1994.

[ANTONIOU, G. 2004] ANTONIOU G, VAN HARMELEN F. A Semantic Web Primer, MIT Press, Cambridge, MA, 2004.

[CHEN, H. 2004] CHEN H, FININ T, JOSHI A, KAGAL L, PERICH F, CHAKRABORTY D. Intelligent agents meet the semantic web in smart spaces IEEE Internet Computing, 8 (6), pp. 69–79, 2004.

[DE RAEDT, L. 1998] DE RAEDT L. Attribute-value learning versus Inductive Logic Programming: The missing links. In: Proc. ILP-98, Springer, 1998.

[DEY, A. 2001] DEY AK. Understanding and Using Context. Personal and Ubiquitous Computing. vol. 5, issue 1, pp. 4-7, 2001.

[EFTHYMIOU, V. 2012] EFTHYMIOU V. A real-time semantics-aware activity recognition system, Master Thesis, University of Crete, Computer Science Department, 2012.

[GOLEMATI, M. 2007] GOLEMATI M, KATIFORI A, VASSILAKIS C, LEPOURAS G, HALATSIS C. Creating an Ontology for the User Profile: Method and Applications. First IEEE International Conference on Research Challenges in Information Science (RCIS), Morocco 2007.

[GRAMMENOS, D. 2009] GRAMMENOS D, ZABULIS X, ARGYROS A, STEFANIDIS C. FORTH-ICS Internal RTD Programme Ambient Intelligence and Smart Environments. Proceedings of the 3rd European Conference on Ambient Intelligence (AMI), 2009.

[GRUBER, TR. 1993] GRUBER T.R. A translation approach to portable ontology specifications. Knowledge Acquisition, vol. 5, pp. 199-220, 1993.

[HONG, M. 2008] HONG M, CHO D. Ontology context model for context-aware learning service in ubiquitous learning environments. International Journal of Computers, 2, July 2008.

[INTILLE, SS. 2005] INTILLE SS, LARSON K, BEAUDIN JS, NAWYN J, TAPIA EM, KAUSHIK, P. A Living Laboratory for the Design and Evaluation of Ubiquitous Computing Technologies. In Proceedings of CHI Extended Abstracts 1941-1944, 2005.

[JAYASURYA, K. 2010] JAYASURYA K, FUNG G, YU S, DEHING-OBERIJE C, DE RUYSSCHER D, HOPE A, DE NEVE W, LIEVENS Y, LAMBIN P and DEKKERA ALAJ. Comparison of Bayesian network and support vector machine models for two-year survival prediction in lung cancer patients treated with radiotherapy. Med. Phys. 37 1401–7, 2010.

[KNOX, S. 2010] KNOX S, COYLE L, DOBSON S. Using ontologies in case-based activity recognition. In FLAIRS 2010 Proceedings, AAAI Press, 336-341, 2010.

[KOFOD-PETERSEN, A. 2006] KOFOD-PETERSEN A. Challenges in Case-Based Reasoning for Context Awareness in Ambient Intelligent Systems. In: 1st Workshop on Case-based Reasoning and Context Awareness (CACOA'06), 2006.

[KOFOD-PETERSEN, A. 2009] KOFOD-PETERSEN A, AAMODT A. Case-Based Reasoning for Situation-Aware Ambient Intelligence: A Hospital Ward Evaluation Study, Case-Based Reasoning Research And Development, Lecture Notes In Computer Science, 2009.

[KRAMER, S. 200] KRAMER, S. Relational learning vs. propositionalization, Investigations in inductive logic programming and propositional machine learning. AI Communications, vol. 3, pp. 275-276, IOS Press, 2000.

[KRUMMENACHER, R. 2007] KRUMMENACHER R,STRANG T. Ontology-based Context Modeling. In Proceedings Third workshop on Context-Aware Proactive Systems (CAPS), 2007.

[LEONIDIS, A. 2010] LEONIDIS A, MARGETIS G, ANTONA M, STEPHANIDIS C. ClassMATE: Enabling Ambient Intelligence in the Classroom. World Academy of Science, Engineering and Technology, issue 66, pp. 594 - 598, 2010.

[MOHAMMAD, RH. 2007] MOHAMMAD RH, NGUYEN TTT, YOUNG-KOO L, BYEONG-SOO J, SUNGYOUNG L. Modeling an ontology for managing contexts in smart meeting space. In SWWS '07: Proceedings of the 2007 International Conference on Semantic Web and Web Services, 2007.

[O'DRISCOLL, C. 2008] O'DRISCOLL C, MOHAN M, MTENZI F, WU B. Deploying a Context Aware Smart Classroom. International Technology and Education Conference, INTED, 2008. Valencia

[PISHVA, D. 2008] PISHVA D, NISHANTHA G. Smart Classrooms for Distance Education and their Adoption to Multiple Classroom Architecture. Journal of Networks (JNW) , 3 (5), 54-63, 2008.

[RECIA-GARCIA, JA. 2007] RECIO-GARCIA JA, DIAZ-AGUDO B, GONZALEZ-CALERO P, SANCHEZ-RUIZ-GRANADOS A. Ontology based CBR with jCOLIBRI. Applications and Innovations in Intelligent Systems Xiva, 2007.

[RECIO-GARCIA, JA. 2008] RECIO-GARCIA JA. jCOLIBRI: A multi-level platform for building and generating CBR systems. Dissertation, Universidad Complutense de Madrid, 2008.

[RIBONI, D. 2011] RIBONI D, BETTINI C. COSAR: hybrid reasoning for context-aware activity recognition. In: Personal and Ubiquitous Computing, vol. 15 (3), pp. 271-289, 2011.

[SHI, YC. 2003] SHI YC, XIE WK, XU GY, SHI RT, CHEN EY, MAO YH, LIU F. The smart classroom: merging technologies for seamless tele-education. IEEE Pervasive Computing. (2), pp. 47–55, 2003.

[STEFANIDIS, C. 2008] STEFANIDIS C, ARGYROS A, GRAMMENOS D, ZABULIS X. Pervasive Computing @ ICS-FORTH. In: Proceedings of Pervasive 2008 Workshop, 119-124. 2008

[TAPIA, EM. 2008] TAPIA EM. Using Machine Learning for Real-time Activity Recognition and Estimation of Energy Expenditure. Dissertation, Massachusetts Institute of Technology, 2008.

[WITTEN, IH. 2011] WITTEN IH, FRANK E, HALL MA. Data Mining: Practical Machine Learning Tools and Techniques. 3rd ed.: Morgan Kaufmann, 2011.