

Automated context learning in ubiquitous computing environments

Fano Ramparany¹, Yazid Benazzouz¹, Jérémie Gadeyne¹, and Philippe Beaune²

¹ Orange Labs
Meylan, France

`fano.ramparany@orange-ftgroup.com`

² Ecole Nationale Supérieure des Mines de St-Etienne
St-Etienne, France

`philippe.beaune@emse.fr`

Abstract. Context awareness enables services and applications to adapt their behaviour to the current situation for the benefit of their users. It is considered as a key technology within the IT industry, for its potential to provide a significant competitive advantage to services providers and to give substantial differentiation among existing services.

Automated learning of contexts will improve the efficiency of Context Aware Services (CAS) development. In this paper we present a system which supports storing, analyzing and exploiting an history of sensors and equipments data collected over time, using data mining techniques and tools. This approach allows us to identify parameters (context dimensions), that are relevant to adapt a service, to identify contexts that needs to be distinguished, and finally to identify adaptation models for CAS such as the one which would automatically switch off/on of lights when needed.

In this paper, we introduce our approach and describe the architecture of our system which implements this approach. We then presents the results obtained when applied on a simple but realistic scenario of a person moving around in her flat. For instance the corresponding dataset has been produced by devices such as white goods equipment, lights and mobile terminal based sensors which we can retrieve the location, position and posture of its owner from.

The method is able to detect recurring patterns. For instance, all patterns found were relevant for automating the control (switching on/off) of the light in the room the person is located. We discuss further these results, position our work with respect to work done elsewhere and conclude with some perspectives.

1 Introduction

Context awareness is considered as a key technology within the IT industry, for its potential to provide a significant competitive advantage to services providers and to give substantial differentiation among existing services. According to a Gartner Inc. report [1], "Context-aware computing today stands where search engines and the web did in 1990".

In parallel to this, the interest of the scientific community in the context aware computing domain has gained a lot of momentum, due to the fact that with the advent of the

Internet of Thing (IoT) era, terabytes of data are bound to be produced daily by sensors and equipments.

Such data, when correctly interpreted can enrich the description of the context, which in turn makes it possible for services and applications to get context-aware, and finally to improve their efficiency in terms of personalization, and simplicity of use.

However, identifying and describing/defining relevant contexts is cumbersome. One reason is that it is generally the case that multiple contexts have to be identified and distinguished. Another is that contexts span over multiple domains such as the “user context”, the “system context” or the “environmental context”, to mention only a few.

Thus, the automated learning of contexts is a way to improve the efficiency of Context Aware Services (CAS) development.

Our approach consists of storing, analyzing and exploiting an history of sensors and equipments data collected over time. In a previous work we have used a semantic modeling language for describing context information [2] and have proved that semantic modeling makes it possible to describe heterogeneous information in a single framework. More generally, interoperability among sensors, sensors networks, and sensor based applications has been promoted by initiatives such as the Semantic Sensor Network incubation group (SSN) [3]. In the work reported here, weve sticked to that semantic modeling policy. As explained throughout this paper, this will allow us to:

- Identify parameters (context dimensions), that are relevant to adapt a service, such as the control of lights or white goods equipment. For example, the user activity is such a parameter and the next item gives an example on how this parameter is used to define contexts.
- Identify contexts that needs to be distinguished. For example, if I need more light when I read than when I watch the television, the context “I am reading” should definetely be distinguished from the context “I am watching the television”. Both contexts refer to my activity and going back to the previous item, the activity should be identified as a parameter that is relevant to our concern.
- Identify adaptation models for CAS such as the one which would automatically switching off/on of lights when needed

In the next section we introduce a simple scenario, which will illustrate a standard use case that our system supports. The details of the scenario will be used throughout the paper to provide concrete examples of the concepts involved in our approach. We then present our approach and describe the architecture of our system which implements it. The system has then been assessed on several datasets. We present the results obtained when applied on the illustrative scenario dataset. Finally, we discuss these results and position our work with respect to work done elsewhere and conclude with some perspectives.

2 Jane ordinary day life Scenario

The scenario takes place in a simple flat and stages Jane, a 80-year-old lady who spends the first two hours of the day moving back and forth between her bedroom and her kitchen. The map of the flat is depicted in figure 5-(a). More precisely, at the beginning

of the scenario, Jane is sleeping in her bed, then she wakes up, goes to the kitchen, eventually she uses her oven to bake or reheat some food, eats it and then returns to her bedroom to take a short nap. Then she walks back to the kitchen to drink a glass of water and returns again in her bed to resume her short rest.

The flat is equipped with a sensor which keeps track of the status of the oven, i.e. if the oven is on or off, and with lights which emit signals whenever they are turned on and turned off. These devices and sensors are also pictured in in figure 5-(a). Jane keeps her mobile phone with her. The mobile phone embeds a software which is able to detect Jane's location, i.e. whether she is in her bedroom or in her kitchen. It also embeds a software which is able to detect Jane's posture, i.e. whether she is lying, standing, seating or walking.

Now by observing Jane's behaviour over a long period of time, say over a week, a human would probably notice that most of the time, if not everytime, when Jane wakes up and gets out of her bed she switches the light on, and that most of the time when Jane leaves her bedroom she switches the light off. Our claim is that we could achieve a similar analysis by applying data mining techniques on a corpus of sensors data, correlated with Jane behaviour, and collected over the same period of time.

Actually, we believe that modeling the sensors data using an appropriate representation language, storing them over time in a database and analyzing the content of this database using datamining techniques, will make it possible to discover contexts which might be relevant for adapting services in such a way that they would be personalized to Jane.

We elaborate this and introduce our approach in the following section.

3 Approach and architecture

The notion of Context is itself contextual as each application, each user, each activity has its own definition of context. For this reason there's no point considering a monolithic or centralized context management system. This lead us to opt for a context management infrastructure that each party could use to setup and manage its own context, rather than for a central context management system, which implicitly would mean that some universal contexts exists that would suit to all parties.

Moreover, the architecture as well as the information model should be flexible. More precisely, the modeling language should be able to cope with the heterogeneity of data sources as well as with the variety of nature of data produced by these data sources. For all these reasons we have based our approach on the Amigo Context Management Service (CMS)[4]. We recall here the main concepts of this framework. For more details the reader could refer to [4].

Each sensor or data source is encapsulated within a software component that we call a context source (CS). An example of this is depicted in the figure 1 where a mobile phone using Wifi based location feeds a software component called "location CS".

The connection between real sensors and its CS component is dependent on the sensor connectivity. In principle, all options can be supported, among which, the most popular ones are the serial line, PLC, Zigbee, ethernet, bluetooth connectivities. The

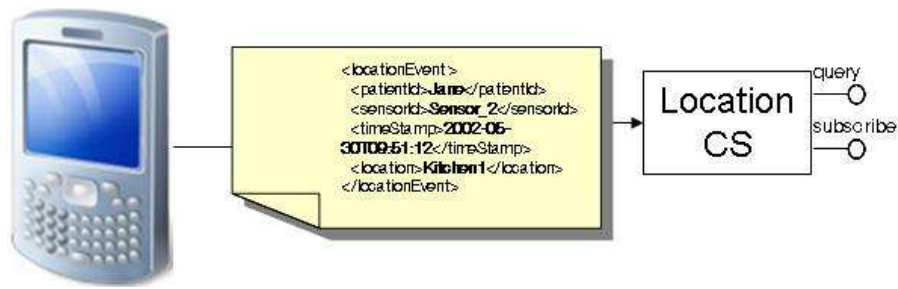


Fig. 1. Wrapping a sensor as a Context Source

point is that once this connection has been set, any access to the sensor is done through the CS component, as far as context management is concerned.

The job of “location CS” is to set semantic annotations to every bit of the sensor raw data, so that it can be automatically interpreted within the context management process later on. Figure 2 displays the result of such annotation.

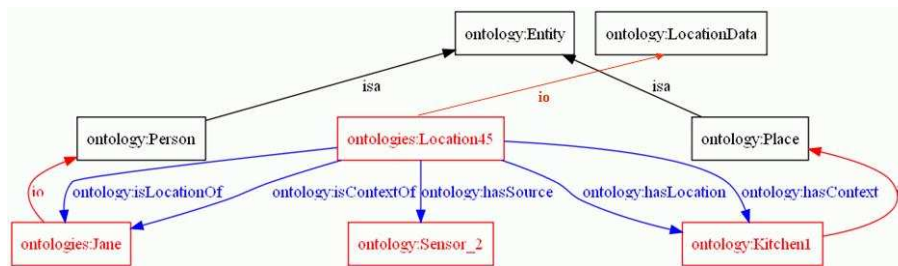


Fig. 2. Location context RDF model

For instance, “Kitchen1”, which is the location value provided by the mobile terminal, has been interpreted as a “Place”, which is a class in the context ontology. The annotation has been made explicit by linking the “Kitchen1” object to the “Place” class using a “io” (“instance of”) relation. The result of this modeling process is presented in figure 2.

Once each sensor data has been modeled, aligning and aggregating them into a integrated and consistent model is straightforward, because they have been expressed along a common ontology. This consistent model is called a situation and is described in the next paragraph 3.1. The aggregation process is handled by the ContextStorage CC component. This component is introduced later on in paragraph 3.3.

3.1 Situation

As told previously, situations are built by aggregating context data. Situations model the states of the environment. A situation could be considered as a snapshot of the environment at a given point in time, which is made of whatever information about this environment we could collect from the sensors.

The algorithm we use for computing situations is inspired from the situation calculus introduced by McCarthy in 1963 [5]. The situation calculus is a logical formalism which makes it possible to reason over dynamical environments, and provide a solution to the question “what beliefs still holds in response to actions” [6]. With respect to our problem, a sensor event creates a transition from the current situation to the new situation, whenever the information it conveys is inconsistent with the current situation (e.g. the event reports that a light is on, while it is described as off in the current situation). In this case, a new situation is created which updates the current situation by adding the new information and removing the inconsistent part.

This process is carried out by the ContextStorage CS component, so that situations can be stored persistently once they have been created.

3.2 Similarity and clustering algorithms

The next goal of the LearningComponent CC is to proceed with a classification of the situations which have been stored over time as explained in the previous section. This classification process involves a similarity function and a clustering algorithm.

A similarity function allows to measure the similarity between two situations. It helps to differentiate two situations which are quite different or to assess the similarity of two situations which are close to each other. This function is a cornerstone of the classification process. As the items we would like to measure the similarity of are graphs, we have used two discrimination criteria:

1. concepts (nodes) that appear in the graph and how often they appear
2. relations between concepts of the graph

The first criteria is evaluated using the TF-IDF (for Term Frequency-Inverse Document Frequency) method [7]. This method has been originally introduced for text data mining, but we have adapted it to our problem by drawing a parallel between texts and situation graphs.

For the second criteria we have used Rada et al. [8] similarity measurement dedicated to semantic networks. This measurement is based on “is-a” hierarchical relations. Thus, in order to evaluate the similarity between two concepts in a model the shortest path between the two concepts in the “is-a” lattice is computed. This measure is applied node per node when comparing two graphs then results are added up and normalized.

Once normalized, these two measurements have been combined using a simple weighted sum.

Clustering aims at partitioning situations into groups of situations which are similar to each other. These groups are called clusters. If several situations occurring over time are very similar to each other, they will be grouped in the same cluster.

Thus large clusters will suggest recurring patterns among situations (contexts). In order to produce such clusters we have used the Markov Clustering algorithm (MCL). MCL [9] builds a NxN distance matrix where N is the number of elements (situations) and each matrix cell contains the distance between the column element and the line element. The algorithm then proceeds by simulating random walks within the distance matrix, by alternation of expansion and inflation stages. Expansion corresponds to computing random walks of higher length (with many steps). Inflation has the effect of boosting the probabilities of intra-cluster walks and will demote inter-cluster walks.

Iterating expansion and inflation results in the separation of the graph into different segments that we call clusters in our terminology. As mentioned previously in section 2, we expect clusters to correspond to relevant contexts. Each context would then be an abstraction of all the situations contained in its cluster.

3.3 architecture

The concepts introduced previously have been implemented and integrated within a prototype, which architecture is depicted in figure 3.

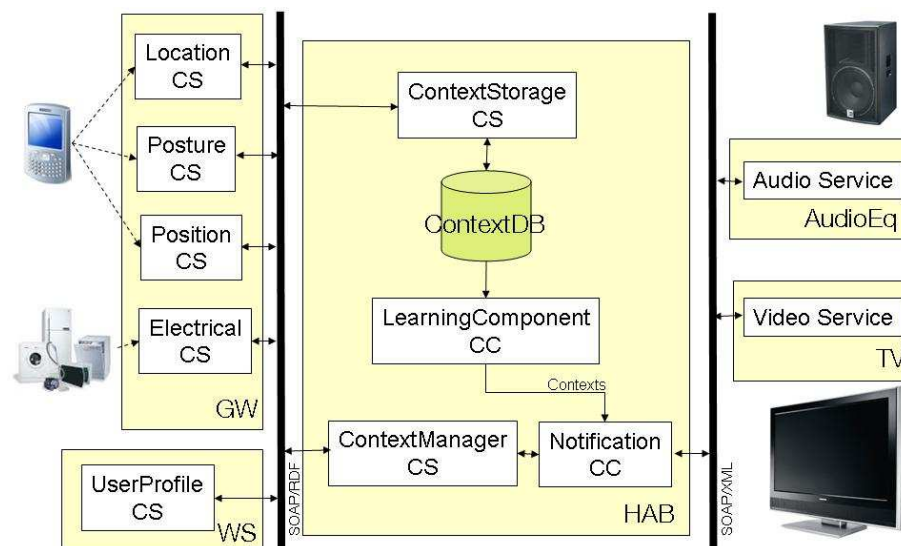


Fig. 3. Context Learning System Architecture

We simply recall and summarize the function of each component in the following:

Sensor Context Source : Provides a high level interface to sensors. A context source component can be viewed as a wrapper of the physical sensor.

Context Manager Context Source : This component subscribe to the different sensor context sources available. It integrates heterogeneous and disparate data conveyed

by the Sensor Context Source events in order to build and maintain a consistent model of the world. Such a model is called a situation. In a previous paragraph 3.1, we explained how situations are built from sensor data events.

Notification Context Consumer : Analyses the world model, identifies critical situations, plans and triggers appropriate actions

Audio and video service : Render visual and audio information

Context Storage Context Source : Collects sensor data formats them into the context data description and stores them persistently. For more details the reader could refer to [10].

Learning Component Context Consumer : Analyses the situations stored over time, discovers and extracts recurring situations (contexts)

Context Data Storing : Collects sensor data formats into the context data description and stores them persistently for retrieval and postmortem and offline analysis.

After this short introduction of our approach and the description of our context learning prototype, we present the results obtained when applying our prototype to the data generated by the illustrative scenario exposed in section 2.

4 Experimental results

Enacting the scenario introduced in section 2 yields 31 sensor data events. These events are presented in figure 4. Each column of the table represent a value of a sensor measurement. Column values are grouped per sensor. For example the first column represents the switching on of the oven whereas the second one represents its switching off. Each line of the table corresponds to an event a sensor emits. Event lines are added in a chronological order, the first event (corresponding to “oven has been switched off”) is positioned as the first line of the table. For example, event number 14 is posted by the kitchen light, which reports the switching off of the light.

Events have been also plotted on the map, at the position Jane had when they occurred. For example in figure 5-(b), we have plotted the events 5 to 13 events as circle shaped tags annotated with the number of the event. For instance, event 12 has been posted by the oven while it was switched on, whereas event 13 corresponding to its switching off.

Theses events have produced 27 situations, as resulting from the algorithm described in paragraph 3.1. Similarly to what we have done for the events, each situation has been plotted on the flat map between the couple of events that respectively initiated and terminated the situation. The 27 situations are then represented in figure 5-(c) as square shaped tags.

Although we model situations as RDF graphs, as explained in section 3.1, it is also convenient to represent them more concisely in terms of sensors measures as shown in table 6. This representation will be more suitable for evaluating the results of the algorithms as we’ll address this point in section 5.

The context learning component has identified 8 situations clusters, using the combined TF-IDF and Rada et al. similarity measure and the MCL clustering algorithm as explained in paragraph 3.2. These clusters and the situations they contain are presented in table 7.

	Oven		kitchen light		Bedroom light		Location		Posture			
	on	off	on	off	on	off	kitchen	bedroom	running	standing	seating	lying
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												
24												
25												
26												
27												
28												
29												
30												
31												

Fig. 4. Sensor events

For instance, cluster 0 contains the 4 situations 2, 12, 16, 24. If we check at their synthetic representation from table 7, we can notice that they are identical as shown in figure 8. Figure 8-(a) highlights the locations of Jane during the four situations 2, 12, 16, 24, while figure 8-(b) is an excerpt of table 7 corresponding to those situations.

We can notice that this cluster can be informally described as: "The person is seating on his/her bed, while the light is on".

With a similar analysis for all the clusters found we come out with the following interpretation:

- Cluster 0** : "The person is seating on his/her bed, while the light is on"
- Cluster 1** : " The person is standing in her/his bedroom, while the light is on "
- Cluster 2** : " The person is standing in her/his bedroom, while the light is off "
- Cluster 3** : " The person is standing in the kitchen, while the light is off "
- Cluster 4** : " The person is standing in the kitchen, while the light is on "
- Cluster 5** : " The person is in his/her bed, while the light is off "
- Cluster 6** : " The person is lying on his/her bed, while the light is on"

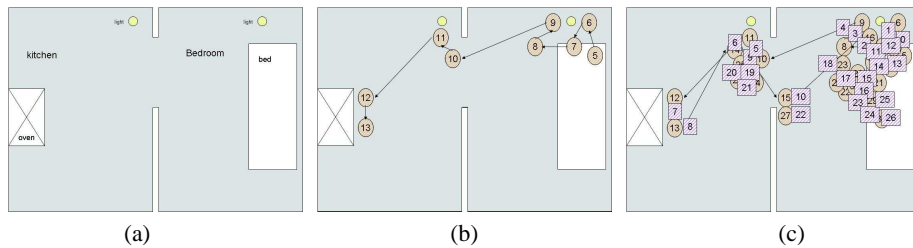


Fig. 5. Environment, sensor events and situations

Cluster 7 : " The person is seating on his/her bed, while the light is off"

Now that we've exposed the results obtained using our approach, we would like to discuss them and position our work with respect to work done elsewhere in the next section.

5 Discussion

Before evaluating our experimental results, we would like to make a general comment on the relevancy of using sensors for observing and analyzing people behaviours in their ordinary daily life.

When installing our 5 sensors (oven, kitchen light, bedroom light, location sensor, posture sensor) in Jane's two rooms flat, as each of these sensors produces measurements within ranges of size 2 ('on'/'off' for the three first sensors, 'kitchen'/'bedroom' for the location sensor) and 4 ('running'/'standing'/'seating'/'lying' for the posture sensor) we could expect situations to span over more than $2 \times 2 \times 2 \times 2 \times 4 = 64$ variants or potential combinations. However, although the scenario generates 27 situations, as seen on table 6, only few of them happen. We believe that this confirms the value of sensors, be they simple and sparsely deployed as in our experimental environment, for monitoring people behaviour. For instance, if we were to observe a concentration of situations which description fall outside those which usually happen, for example with the person lying while she/he is in the kitchen, we could consider it as an hint that something is going wrong.

Now back to our context learning research work, we can assert that our approach is able to identify clusters of similar situations which occur frequently. Although we haven't pushed the implementation of our approach that far yet, we could notice that some of these clusters correspond to contexts that are relevant to control the environment. For instance, cluster 1 and cluster 2 correspond to the context where the person is leaving her/his bedroom, and that their description suggest the bedroom light to be switched off (this is the only difference between the synthetic description of the two clusters).

Some work has addressed the extensive use of sensors measurements for learning human behaviour ([11]) but they have been limited in scope to the inference of user context (user activity/user task) from physical context information.

Oven		kitchen light		Bedroom light		Location		Posture				situations
on	off	on	off	on	off	kitchen	bedroom	running	standing	seating	lying	
												0
												1
												2
												3
												4
												5
												6
												7
												8
												9
												10
												11
												12
												13
												14
												15
												16
												17
												18
												19
												20
												21
												22
												23
												24
												25
												26

Fig. 6. Situations found

We think that these limitations principally stems from their use of the 'attribute/value' representation paradigm for representing context data. We believe that relations and structural information matter in context aware computing. For example, in a context aware building access control system, it makes sense to know the kind of relationship between the visitor and the people present in the building, and if there are several visitors it make sense to know the relationship between those visitors and to take this information into account when making a decision on which access policy to adopt.

In our approach we have used RDF which makes relational and structural information explicit, to model the instances of the population, we've learned recurrent context from. There are some existing learning techniques which are dedicated to structured data such as structural learning, multi-table learning, inductive logic programming (ILP).

Within a preliminary stage of our work we have evaluated and compared various clustering algorithms including the Kmean algorithm, the hierarchical classification and MCL. These methods are unsupervised classifiers, which basically means that no oracle is required to declare which class a sample belongs to. Kmean algorithm places each element of the population iteratively in one of K distinct classes which minimizes the its distance to the class. Each class is represented by a prototype (or centroid) which is itself an element that represents the class. This prototype is updated at each iteration so as to ensure a good representation of the class. This iterative process completes as soon as an iteration doesn't change neither an element to class assignment, nor a prototype change in a class. There are two major drawbacks with the Kmean algorithm. One is

Clusters	Situations
C0	2 12 16 24
C1	3 11 17 23
C2	4 10 18 22
C3	5 9 19 21
C4	6 7 8 20
C5	0 14 26
C6	1 15
C7	13 25

Fig. 7. Clusters extracted

that K , the number of classes, has to be fixed arbitrarily, the other is that its results are very sensitive to the choice of the prototype at the bootstrapping stage.

We have evaluated another clustering algorithm called Hierarchical agglomerative clustering [12] that doesn't present the first drawback. This algorithm starts with singleton clusters where each element forms a cluster. The algorithm then proceeds by iteratively merging (agglomerating) pairs of clusters that are close to each other (in terms of similarity measure), until all clusters have been merged into a single cluster that contains the whole population. The result of this algorithm is a hierarchy of clusters, which can be represented as a dendrogram. This algorithm shares the second drawback of the Kmeans algorithm because the number of clusters depends on the level at which the dendrogram is cut.

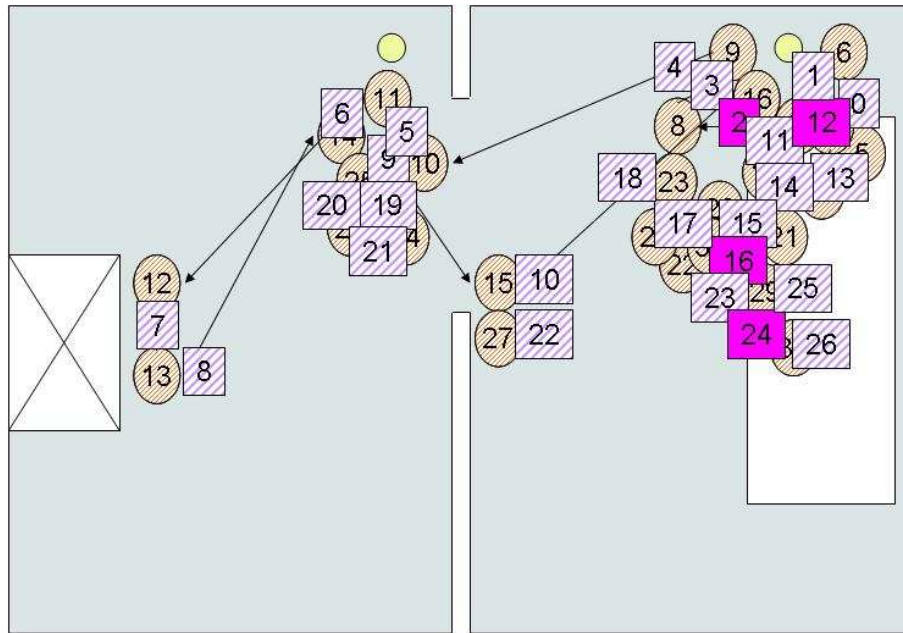
The MCL algorithm which we finally retained just ignores this second drawback. As we've seen, this algorithm had good performance on our scenario dataset.

The system has been assessed on several datasets, some of them involved a large amount of data. These experiments have revealed that some optimization in the data management and algorithm is required, if we need to increase the number of context sources, or if we need to store over a longer period of time, say several weeks. We now conclude and outline some perspectives of our work.

6 Conclusion and perspectives

In this paper, we have presented a system for archiving and mining data collected from sensors deployed in a home environment. The sensors we have used in our MIDAS project include white goods equipment and mobile terminal based sensors. From the data produced by these sensors we can retrieve the location, position and posture of their owners.

However, the flexibility of the data representation language we have adopted makes it possible to support a large variety of data sources, such as web services or personal



(a)

Oven		kitchen light		Bedroom light		Location		Posture				situations
on	off	on	off	on	off	kitchen	bedroom	running	standing	seating	lying	
												2
												12
												18
												24

(b)

Fig. 8. Position and description of situations in cluster 0

productivity tools (agenda, phonebook,...). From this archive we have applied data mining tools for extracting clusters of similar data. We have applied the system to a simple but realistic scenario of a person moving around in her flat. The method is able to detect recurring patterns. More over, all patterns found are relevant for automating the control of some devices. For instance, among the 8 patterns found, 4 of them describe a context where the light of the room the person is located in, should be switched off, whereas the other 4 describe a context where the light should be switched on.

Beyond context aware home automation, we believe that our approach is applicable to domains where similarity based clusters should be found out of structures of heterogeneous and disparated data. Hence the following application domains are potential targets of our system:

- Customer Relationship Management (Learn customers habits)
- Content search and casting (Learn customers preferences)

- SmartCity, SmartHome, SmartBuilding (Discover hidden correlations)
- Web services (context aware WS)

There are some issues remaining that we are currently addressing. They include scalability and the possibility to learn service context adaptation. For the second point, we expect machine learning mechanisms will allow the identification of correlation between service configuration parameters and context descriptions.

References

1. Lapkin, A.: Context-aware computing: A looming disruption. Research report, Gartner Inc. (2009)
2. Ramparany, F., Benazzouz, Y., Chotard, L., Coly, E.: Context aware assistant for the aging and dependent society. In et al., J.A., ed.: Workshop Proceedings of the 7th International Conference on Intelligence Environments, Nottingham, UK, University of Trent, IOS Press (2011) 798–809
3. Lefort, L., Henson, C., Taylor, K., Barnaghi, P., Compton, M., Corcho, O., Garcia-Castro, R., Graybeal, J., Herzog, A., Janowicz, K., Neuhaus, H., Nikolov, A., Page, K.: Semantic sensor network xg final report, W3C Incubator Group Report (2011) Available as <http://www.w3.org/2005/Incubator/ssn/XGR-ssn/>.
4. Ramparany, F., Poortinga, R., Stikic, M., Schmalenströer, J., Prante, T.: An open Context Information Management Infrastructure - the IST-Amigo Project. In of Engineering, I.I., Technology, eds.: Proceedings of the 3rd IET International Conference on Intelligent Environments (IE'07), Germany, University of Ulm (2007) 398–403
5. McCarthy, J.: Situations, actions and causal laws. Technical Report Memo2, Stanford Artificial Intelligence Project (1963)
6. McCarthy, J., Hayes, P.: Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, B., Michie, D., eds.: Machine Intelligence. Volume 4. Edinburgh University Press (1969) 463–500
7. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill International Editions (1983)
8. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man and Cybernetics **19** (1989) 17–30 ISSN 0018-9472.
9. van Dongen, S.: Graph Clustering by Flow Simulation. Phd thesis, University of Utrecht (2000)
10. Benazzouz, Y., Beaune, P., Ramparany, F., Boissier, O.: Modeling and storage of context data for service adaptation. In Sheng, Q.Z., Yu, J., Dustdar, S., eds.: Enabling Context-Aware Web Services: Methods, Architectures, and Technologies. Chapman and Hall/CRC (2010) 469–494
11. Brdiczka, O., Langet, M., Maisonnasse, J., Crowley, J.L.: Detecting human behavior models from multimodal observation in a smart home. IEEE transactions on automation sciences and engineering **6** (2009) 588–597
12. Day, W.H., Edelsbrunner, H.: Efficient algorithms for agglomerative hierarchical clustering methods. Journal of Classification **1** (1984) 1–24