An approach to structuring reasoning for interpretation of sensor data in home-based health and well-being monitoring applications

Herman J. ter Horst, Alexander Sinitsyn
Philips Research, High Tech Campus 34, 5656 AE Eindhoven, The Netherlands
herman.ter.horst@philips.com, alexander.sinitsyn@philips.com

***Abstract*—This paper presents an approach to structuring knowledge and reasoning for high-level interpretation of sensor data in e.g. independent living applications. The main contribution is to use generalized events, described in terms of 'space-time chunks', as a unifying and simplifying structuring principle. We use reasoning with ontologies and rules in combination with a database system, and also incorporate numerical computation. We show that an easy to use modeling formalism is obtained, and that reasoning is feasible at the time of service request, by using R-entailment, which enables efficient exploitation of ontologies and rules in the presence of RDF data. Two applications were built using the approach described in this paper, both of which are related to monitoring well-being of elderly people, and both of which use simple, low-cost sensors.**

I. Introduction

The older population is growing rapidly. It is also known that most of the elderly people who live alone, often having lost a spouse, prefer to remain independent and age at a home of their choice, whether it is a private home, apartment or group home, for as long as possible.

Pervasive technology can help age in place. A variety of new pervasive health applications is emerging. In general, these applications share a common framework involving activity monitoring using sensors, interpretation of sensor data, and providing feedback. There are three user groups, each representing a main class of applications: the elderly themselves seeking better communication with family and friends, as well as improved safety and security; the elderly people's family and relatives, who are looking for 'peace-of-mind' provided through the support of an intelligent awareness system; and care provider organizations, who can operate more efficiently, and hence more cost effectively, by having access to a remote monitoring system.

Activities of Daily Living (ADLs) concern basic activities that everybody executes on a regular basis. This concept stems from behavioral studies with the purpose of developing a generic procedure that assists human caregivers to decide whether an elderly person needs to be institutionalized or not, i.e. to decide at what level the elderly person is able to live independently. Researchers arrived at a couple of main ADLs that together are typical and diverse enough to cover the spectrum that arises in practice, such as bathing, dressing, walking, eating, toilet use, etc. [14], [15]. Automatic and reliable derivation of information on ADLs is an important goal: if ADLs could be detected automatically, an indication of well-being of an elderly person could be provided to relatives or care-providers of the person and could form a basis for the development of useful well-being monitoring applications.

Many home-based health and well-being monitoring applications and decision support applications depend on high-level interpretation of sensor data. An interpretation step needs to be made to bridge the gap between raw sensor data, which is typically of a low semantic level, and the application and user, to derive high-level information on the user and his or her context. It is desirable to realize this interpretation step in such a way that sensors are decoupled from applications [23]. Various approaches to realization of the sensor data interpretation step have been investigated, ranging from approaches that use only numerical computation, using e.g. statistically based data mining techniques, to approaches such as the one considered in this paper, which make extensive use of reasoning using explicitly represented symbolic knowledge. It should be noted that in practice, approaches that reason using explicitly represented symbolic knowledge also need to deal with numerical computation, since raw sensor data is typically of numerical form.

This paper presents an approach to structuring of knowledge and reasoning for interpretation of sensor data in applications that reason using both explicitly represented symbolic knowledge and numerical computation. The main contribution of this paper is to structure knowledge and reasoning for interpretation of sensor data using generalized events, described in terms of 'space-time chunks' which may have a long duration. This 'four-dimensional' way of looking at events, which was introduced in AI in [6], leads to a unification of static and dynamic aspects which is attractive from various perspectives, in particular from an information management point of view.

In the approach presented in this paper we bridge the gap between raw sensor data and user-level conclusions in two parts, called abstraction and reasoning, in such a way that sensors are decoupled from applications. The abstraction layer uses numerical computation to determine symbolic conclusions from raw sensor data while the reasoning layer uses explicitly represented knowledge to derive high-level symbolic conclusions from the output of the abstraction layer. In the applications we developed, uncertainty is handled in the abstraction layer. We show that an easy to use modeling

PervasiveHealth 2011, May 23-26, Dublin, Republic of Ireland

DOI 10.4108/pervasivehealth.2011.246099

formalism is obtained, providing a flexible means to record knowledge and configuration data for different contexts, and that reasoning is feasible in practical applications at the time of service request, using R-entailment [12], an integrated, uniform approach to reasoning with rules and ontologies which has attractive properties with respect to computational complexity.[1]

Two applications were built using the approach described in this paper, both of which are related to monitoring well-being of elderly people living alone. One application aims at enabling elderly people to live alone independently for a longer period by using in-environment sensors to monitor, detect and signal unsafe or undesirable situations; for example, a gas oven or a water tap could be on while the elderly person is present in a different room for a long period of time. The prototype system that we built deals, in particular, with tracking the elderly person's presence at specific locations (i.e. rooms) in the house. Such location information enables interesting use cases supporting care providers in assessing well-being of a person. For example, it allows checking whether an elderly person moves enough per day between rooms; if a person is not leaving the bathroom for a long time, this might indicate a bathroom fall; spending little time in the kitchen or not visiting it regularly could mean that a person is not eating or drinking enough. The second application considered in this paper supports elderly people by enhancing connectedness and by promoting 'peace-of-mind' for their remote children. The application detects, interprets and communicates an elderly person's ADLs in a way that allows the children to acquire an overview of long-term trends of their elderly parents' well-being. In particular, cooking and showering activities are being tracked. Both applications make use of input sensor data collected by simple, low-cost sensors for detecting, for example, the status of doors (open/closed) or temperature. Users do not need to wear RFID tags or other on-body sensors. No specially designed positioning systems based on e.g. GPS, RFID or WiFi are being used.

II. Related Work

Many techniques have been proposed focusing on statistical reasoning methods for pervasive health monitoring and activity recognition, as in [25] [26] [5] [16] [24], whereas this paper makes much use of symbolic reasoning using explicitly represented knowledge. A number of papers have addressed decoupling of sensors from applications through reasoning based on sensor data using explicitly represented knowledge. To the best of our knowledge, apart from earlier work by the first author [10], there has been no other work in pervasive computing or context awareness that structures knowledge and reasoning for interpretation of sensor data using generalized events, as proposed in this paper. The use of generalized events makes modeling context more generic and hence simplifies much of the context management.

[1]R-entailment is based on RDF and RDF Schema, includes a significant part of OWL, and also supports rules with RDF-like statements in IF- and THEN-sides.

The CML formalism [8] provides an approach to modeling context information to support software engineering for context-aware applications, which has been mapped to XML [21]. In this paper we use the semantic web languages RDF and OWL, defined on top of XML, which provide standardized primitives for defining data and semantics, which can be useful for context modeling.

The CARE middleware [1] supports context awareness using reasoning with ontologies and rules. Ontologies are developed using OWL DL. In CARE, ontological reasoning is only loosely coupled to reasoning with rules, and avoided at the time of service request, because of the high computational complexity of reasoning with OWL DL. In [18] the ontology language DAML+OIL is used, a predecessor of OWL DL with the same high complexity. Also in [20] reasoning using ontologies is done offline for efficiency reasons. We do not avoid reasoning with ontologies at the time of service request as the computational complexity of the R-entailment formalism that we use for reasoning with ontologies and rules is much lower than that of OWL DL (see below for further details).

ACoMS [13] provides an approach to autonomic context management which is in a certain sense complementary to our approach. While the approach described in this paper gives more attention to dealing with higher-level abstractions than to dealing with raw sensor data, [13] focuses on arriving at context facts on the basis of raw sensor data; these context facts can be compared to our basic events.

In [3] a rule-based approach is presented with applications related to elderly care. Our approach strengthens the expressivity of rules by using ontologies to provide machine-understandable definitions of terminology used in rules. It is also more expressive and flexible than that of [17], since we allow rules to use types (classes) rather than instances. In this paper we handle uncertainty in an abstraction layer, separate from the symbolic reasoning process; there are other papers where uncertainty is handled in the reasoning process, for example using Bayesian networks or fuzzy logic, as in [19].

In [2] a list of requirements for context modeling and reasoning is presented, together with an analysis of existing approaches in terms of these requirements. It is of interest to note that our approach provides at least a starting point for all the requirements listed in [2]. Our abstraction layer (cf. Fig. 1) ensures that heterogeneous types of sensor data can be handled, while our data management framework supports mobility of context information sources. Relationships and dependencies, and reasoning, are supported by ontologies and rules. Timeliness is handled and context histories are incorporated through our use of generalized events. In this paper we handle imperfection in the abstraction layer by using thresholds based on statistics to transform raw sensor data into symbolic 'basic events'. Our approach provides applications with an easy way of using and manipulating context information, while our modeling formalism, being based on RDF's subject-predicate-value triples, is easy to use by application designers. Finally, efficiency of context provisioning is supported by the way we use a database

for managing events and by the relatively low computational complexity of the R-entailment formalism used for knowledge representation and reasoning.

III. Approach and System

The applications considered in this paper aim at detecting activities of elderly people by interpretation of data coming from simple unobtrusive sensors such as binary sensors for detecting the status of the environment, e.g. doors (open/closed), water taps, and light switches, and digitally-sampled continuous-valued sensors for detecting parameters such as temperature and humidity. We adopt sharing and aggregation of sensor data. Data from one sensor could be used by multiple applications, while the combination of data from different sensors could improve quality of derived information. In the home our approach could be used, for example, for lighting control in combination with a well-being monitoring application using an overlapping set of sensors.

A. Outline of approach

In order to bridge the gap between raw sensor data and context-aware applications, we aim at what might be called *context determination*, i.e. determination of a high-level (i.e. user-level) description of the user context, given as input raw sensor data. As in [4] we interpret the term context (or context of use) in a broad sense, including the state of the user and the surroundings of the user, activities being undertaken, and also history. The high-level description of user context that results from context determination can be viewed as corresponding to the situation abstraction considered for example in [4] and [8] (see also the section on high-level context abstractions in [2]). We emphasize that the high-level description of context aimed for should be understandable by the end users themselves, so that any actions taken or advised on the basis of this high-level description can be explained to the user.

It should be noted that in practice, the interplay of static and dynamic aspects of context can lead to many complications. Many investigations use a notion of state or situation as a basis, considered at a specific instant in time, and consider dynamic aspects by relating states at different instants in time. We have adopted another idea which was proposed in the AI literature [6] to consider an alternative for situation calculus, a widely used technique in AI. The idea, which is also described in [22], is to consider not only instantaneously happening events, for example, but to consider *generalized events*, described in terms of 'space-time chunks' which can also have a long duration. This 'four-dimensional' way of considering events makes modeling context more generic and hence can lead to simplification. In this way, for example, not only the ringing of a doorbell can be regarded as an event, but also the presence of a person in a certain room during a certain hour can be regarded as an event. We describe context and also history as a set of such generalized events. Also, knowledge that enables drawing conclusions about context is phrased in terms of this generalized notion of event. The notion of generalized event enables easy handling of simultaneous events of different duration, which are much more difficult to handle using state-based approaches. In what follows we call generalized events simply events.

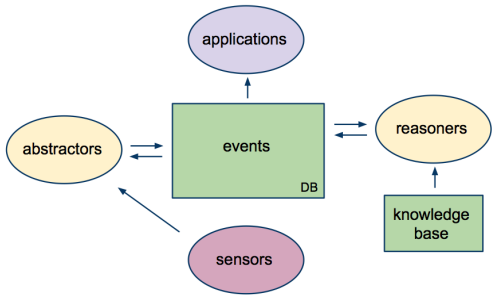

Fig. 1. Context determination using abstraction and reasoning.

We envision context determination to be realized by means of a two-part approach which involves a procedure for *abstraction* and a procedure for *reasoning* (see Figure 1). Abstraction aims at algorithmic determination of symbolic, qualitative conclusions. Typically, an abstraction procedure involves signal processing and statistics. Reasoning aims at drawing high-level symbolic conclusions about context using output of the abstraction procedure and using knowledge represented explicitly in knowledge bases or implicitly in algorithms. In the system described in this paper, we use a reasoner that reasons using knowledge expressed in ontologies and rules. Abstraction procedures and reasoning procedures both generate output data in the form of events and can use events as input. Abstraction procedures can therefore also use results obtained by reasoners as input. In this way, a feedback loop is possible and hybrid combinations are enabled of symbolic reasoning and reasoning through numerical, for example statistical, computation. In the applications described in this paper, a pipeline pattern is followed, with a distinction between lower-level (sensor-level) reasoning and higher-level reasoning: first an abstraction procedure generates a symbolic description of context on a low or medium semantic level, called *basic events*, then a reasoning procedure generates a high-level (i.e. user-level) symbolic description of context, called *high-level events*. In these applications uncertainty is handled in the abstraction layer; it is assumed that the basic events provided by the abstraction procedure form reliable input for the reasoning procedure.

In our view database technology can be efficiently used as a gluing component interfacing all components (i.e. abstractors, reasoners and application clients) and for providing persistent storage for example for results of abstraction algorithms and conclusions of reasoning procedures. Our system uses a database which stores events derived by abstractors and reasoners; in addition to basic events and high-level events, also intermediate-level events are stored. The use of database technology also allows performing analysis of the sensor data over a long period of time, for example through data mining. A distributed database could be more efficient than a centralized

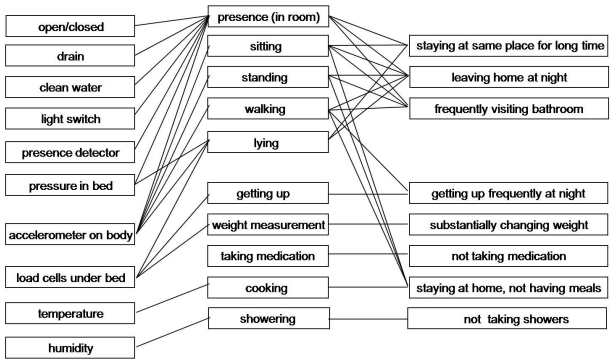

Fig. 2. Levels of information: sensors, events, application requests.

database from a reliability and scalability perspective. In the system described in this paper, however, we did not use a distributed database yet.

The core of our system consists of three main parts: abstractors, reasoners, and data management. Applications can either subscribe to and be notified about a relevant set of events and conclusions or request the status by querying the system. Abstractors and sensors could be automatically configured based on higher-level context using a feedback loop. For example, knowing that a person is not present in a room (a high-level conclusion) could be used by sensor nodes in this room to go to a lower power consumption mode.

We use a knowledge-based approach in which knowledge is explicitly represented in knowledge bases used by reasoning algorithms. An advantage of this setup is that knowledge in a knowledge base is under explicit control of people, in a relatively flexible way; knowledge bases can be developed and changed more easily than computer programs that use knowledge implicitly. A knowledge-based approach offers the perspective of flexible adaptation to many different kinds of context that can appear in practice, and could use but does not need a learning phase. In the work reported here, we used two kinds of knowledge bases: rule bases and ontologies. Rule bases contain IF THEN rules which express that certain conclusions are valid when certain conditions hold. Such rules use concepts (i.e. notions) in both IF- and THEN-sides. The meaning of these concepts can be defined in a machine-understandable way in ontologies. In this way, ontologies lead to an increase of expressivity of rules.

Figure 2 shows some examples dealing with the elderly care applications considered in this paper. The left part of this figure includes various sensors, while the right part of this figure shows various needs for information from applications. The middle part of the figure gives examples of various high-level statements on users (events) which are of interest to the applications. For example, to answer the question 'how frequently is the user visiting the bathroom', information is needed on the presence of the user in a room. Presence in rooms could be detected in many different ways, for example, using light switches or movement detectors or open/closed sensors located on doors. Figure 2 does not show that there can actually be multiple levels in the middle part: e.g., information on walking can be deduced from information on presence.

B. Reasoning with rules and ontologies: R-entailment

Before turning to the kind of knowledge represented and used in our system using rules and ontologies, we briefly discuss our format for knowledge representation. Use of the semantic web languages RDF and OWL offers the potential to make use of the growing amount of machine-processable knowledge expressed in these languages on the web. However, as was already mentioned in the section on related work, in several investigations where semantic web ontologies were used in applications of pervasive computing, the issue of computational complexity of reasoning was encountered. In addition, there is the issue that semantic web technology involves different reasoning paradigms which are not directly compatible: RDF, description logics (OWL DL), and logic programming. RDF allows data to be represented as subject-predicate-value triples, and is accompanied by a simple ontology language, RDFS (RDF Schema). RDF and RDFS include higher-order expressivity allowing for example classes to be used as instances, which is considered to be useful by many developers of ontologies. OWL DL is based on a description logic and provides primitives for specifying ontologies not provided by RDFS, but does not provide RDFS's higher-order expressivity. OWL DL has a high computational complexity: NEXPTIME [9]. OWL DL combined with basic primitives for logic programming rules is undecidable.

The three reasoning paradigms mentioned - RDF, descriptions logics, logic programming - have not been fully integrated. In order to arrive at a technically coherent and tractable way of reasoning with ontologies and rules on the semantic web, a careful selection of possibilities needs to be made. It is natural to start from RDF, the basic, standard format for expressing data on the semantic web. In this paper we use a relatively simple, uniform approach to reasoning with ontologies and rules that fully includes RDF and RDFS, that includes a significant part of part of OWL, and that also supports rules with RDF-like statements in IF- and THEN-sides. The notion of consequence is called R-entailment [12]. The part of OWL included [11] has been used by the W3C for standardizing a new variant of OWL: OWL 2 RL.[2] The relatively low computational complexity of RDFS is preserved by making the extension to R-entailment. Making the natural assumptions that rules do not introduce blank nodes (i.e. existentially quantified variables) in their THEN-sides and satisfy a bound on the number of statements in their IF-sides, R-entailment has NP-complexity [12]; R-entailment actually has polynomial-time complexity in the commonly occurring case, also applying to the applications considered in this paper, where query statements do not include blank nodes [12].[3]

[2]http://www.w3.org/TR/owl2-profiles/

[3]R-entailment can be used with Jena (see http://jena.sourceforge.net) and is also supported by the BaseVISor system (see http://vistology.com/basevisor).

C. Knowledge Representation and Reasoning

In this section we describe the way in which knowledge is used for drawing semantically high-level conclusions on context based on a lower-level description of the sensor data. The reasoner that we use to determine a user-level description of context of use, in the form of high-level events, uses a symbolic summary of sensor data as input, in the form of basic events (cf. Fig. 1 and the text discussing it). We distinguish two kinds of basic events: value events and transition events. A *value event* indicates that a certain value detected by a sensor holds without interruption between a certain start time and a certain end time. Here a value is a symbolic, qualitative value such as 'open' or 'closed' for a door sensor or 'low' or 'high' for a humidity sensor. A *transition event* indicates that the (discretized) value of the sensor has not been constant between a certain start time and a certain end time. Determination of value events and transition events requires relatively little data preprocessing and can lead to significant data reduction, which limits bandwidth usage in the system.

The two kinds of basic events for sensors are expressed as input for our reasoning system in the form of RDF statements, i.e. subject-predicate-value triples of one of the following two forms:

```
[SensorID] hasValue [SensorValue]
[SensorID] hasTransition [SensorTransition]
```

With respect to the elderly care applications considered in this paper, it is assumed that there is at most one person present in the home under consideration, identified in this section as `User`. This assumption makes sense in our application area as this involves monitoring of elderly people living alone. It is possible to detect the presence of multiple people and to automatically disable monitoring when, for example, a guest is present who can take care of the elderly person in case of abnormality.

We describe the ontology used by our reasoning system by describing its classes and properties (i.e. binary relations). The more concrete parts of the ontology are rather specific to our application setting. A central point is that the ontology could easily be set up in an alternative way so as to apply to other settings. We begin by describing the more abstract parts of the ontology, which can be expected to apply in many settings. The 'root' of the ontology consists of the class `Object`. A central class of the ontology is the subclass `Sensor` of the class `Object`. The class `Sensor` has another subclass: `SensorTP`, which is defined to be the class of sensors for which each transition indicates presence of the user.

The property `relatedToObject` is introduced to describe the background configuration, e.g. objects and relationships which are more or less fixed. This property has two subproperties, i.e. specialized versions: `presentAt` and `presentInRoom`. The `presentAt` property is used, for example, to describe the connection between sensors and objects, e.g. between a door sensor and the door to which it belongs. The `presentInRoom` property is used, for example, to state, as a conclusion, that the user is present in a certain room.

In order to take time into account in the reasoning in a simple, symbolic way, there is also a property `previously-PresentInRoom`, which is used to indicate that the user has just been, but is no longer, in a certain room. The class `TemporalProperty` is used to define the class of properties which can only be used to make statements that have a limited duration of validity.

In addition to the class `Sensor`, the class `Object` has other subclasses, such as: `Room`, `Cupboard`, `Bed`, `Door`, and `Window`. The class `SensorTP`, i.e. the class of sensors for which each transition indicates presence of the user, has subclasses such as `CleanWaterSensor`, `CupboardDoor-Sensor`, and `LightSwitchSensor`. In addition to the class `SensorTP`, the class Sensor also has subclasses such as `WindowSensor`, `DoorSensor`, and `TemperatureSensor`, for which transitions may not indicate presence of the user.

The ontology just described is populated with instance data, which essentially describes the configuration of the context considered, in particular the floor plan and the types and placement of sensors. In order to facilitate management of this configuration data, we developed a relational database application, which can automatically provide convenient overview reports of all the data and can also generate text files that contain the instance data in the form of RDF statements.

In this paper we do not use the standard XML syntax of RDF but use another, also quite widely used, abbreviated syntax, in order to improve readability. This so-called N-Triples syntax greatly facilitates the use of the R-entailment formalism used here.[4] The following rule deals with transitions of all sensors of a type belonging to the class `SensorTP`:

```
IF ?s type SensorTP  AND  ?s hasTransition ?t
  THEN User presentAt ?s
```

By means of this rule, the class `SensorTP` defined in the ontology allows the combination of a number of 'concrete' rules into just one rule which is more abstract. For sensors of a type that does not belong to the class `SensorTP`, transitions are being handled by more specific rules.

The reasoner draws conclusions relating to the room where the user is present using the following two rules:

```
IF ?o presentAt ?p  AND  ?p presentAt ?q
  THEN ?o presentAt ?q
IF ?o presentAt ?p  AND  ?p presentInRoom ?q
  THEN ?o presentInRoom ?q
```

Conclusions relating to cooking and showering events can be drawn using the following two rules:

```
IF ?s type HumiditySensor
   AND  ?s hasTransition "LOW/HIGH"
  THEN User uses Shower
IF ?s type TemperatureSensor
   AND  ?s hasTransition "LOW/HIGH"
  THEN User uses Stove
```

With respect to handling time, all rules mentioned so far just consider the 'current' moment ('now'). The following

[4] We make various abbreviations in the standard vocabulary from RDF and OWL. For example, the property `rdf:type` is abbreviated as `type`.

TABLE I
REASONING ALGORITHM (SEE TEXT FOR H AND C).

On input of new basic events
DO
  B := set of new basic events (i.e. `hasValue` and `hasTransition` statements)
  H_0 := H
  H_1 := set of statements with `User` as subject derived from $B \cup C$
  IF H_1 contains a `presentInRoom` statement
  THEN H := H_1
  IF H_0 contains a statement of the form
    `User` p r
    which is not contained in H_1, while the ontology entails the statement
    p `type TemporalProperty`
  THEN add to H the following statement:
    `User` q r
    where the name of q is that of p with `previously` prefixed
  RETURN H

rule combines the current moment with information on the immediately preceding history. It can be used for drawing conclusions relating to walking events:

```
IF User previouslyPresentInRoom ?r
   AND User presentInRoom ?s
  THEN User walksFrom ?r AND User walksTo ?s
```

The reasoning procedure uses the RDF statements summarizing the floor plan and the types and placement of sensors. This configuration data forms instance data for the ontology; it is derived from a database in the manner indicated above. Further input for the reasoner consists of RDF statements summarizing the current status of sensors. This status is expressed using basic events in the form of `hasValue` statements and `hasTransition` statements, as was already indicated. The reasoning procedure applies the ontology and rules to the input RDF statements, i.e. configuration data combined with the current basic events, to derive `presentAt` and `presentInRoom` statements and also statements relating to activities such as showering, cooking and walking. As output, a selection is made by the reasoner of all derived statements with the user (i.e. `User`) as subject. The reasoning procedure can continuously be triggered to draw conclusions from new sensor data, using the algorithm presented in Table 1. In Table 1, the set of derived statements with `User` as subject is denoted by H, which is short for high-level events. In the algorithm, H is initialized to be empty. The set of RDF statements describing the configuration data is denoted by C. The data management system ensures that each event concluded by the reasoner is recorded in the database with correct start time and end time.

In addition to the rules explicitly described in the knowledge base, our reasoning procedure also uses certain implicitly available rules for exploiting the ontology. These implicitly available rules, called entailment rules, 'realize' the underlying semantics of RDF Schema [7] and the part of OWL used [11]. As an example of the use of these entailment rules in our reasoner, for light switch sensors, the following statement in

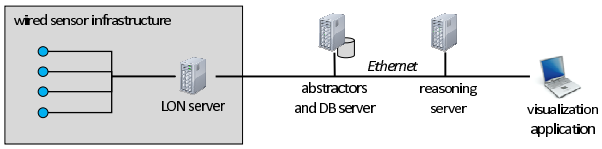

Fig. 3. Architecture of prototype.

the ontology:

```
LightSwitchSensor subClassOf SensorTP
```

is being used in combination with the following standard entailment rule for RDF Schema [7]:

```
IF ?v subClassOf ?w  AND  ?u type ?v
  THEN ?u type ?w
```

to draw the conclusion that each instance of the class `LightSwitchSensor` is also an instance of the class `SensorTP`.

To illustrate the reasoning procedure and its use of the knowledge base, we present a simple example, based on the following input statement:

```
BedPresenceSensor01 hasValue "PRESENT"
```

The system combines this statement with the following configuration data:

```
BedPresenceSensor01 type BedPresenceSensor
BedPresenceSensor01 presentAt Bed01
Bed01 presentInRoom Bedroom
```

to draw the following conclusion:

```
User presentInRoom Bedroom
```

Here the following three rules are being used:

```
IF ?s type BedPresenceSensor
   AND  ?s hasValue "PRESENT"
  THEN User presentAt ?s
IF ?o presentAt ?p  AND  ?p presentAt ?q
  THEN ?o presentAt ?q
IF ?o presentAt ?p  AND  ?p presentInRoom ?q
  THEN ?o presentInRoom ?q
```

IV. APPLICATION AND DEPLOYMENT

We implemented and tested the proposed approach in an experimental simulation of a living environment for an elderly person. Our demonstrator deals, in particular, with tracking the elderly person's presence at specific locations in the living environment. Figure 3 depicts the prototype system that draws conclusions on presence of a user given as input sensor data and shows this information on a screen in real-time.

The infrastructure (depicted in the grey box in Fig. 3) includes 147 off-the-shelf sensors, mainly simple binary sensors for detecting the status of, for example, doors, windows, and drawers, but also digitally-sampled, continuous-valued sensors, for detecting for example humidity and temperature. These sensors were already installed before the work described in this paper started. We used these sensors mainly to test our approach and system. A question to be answered in future work concerns the optimal number of sensors needed to detect a certain context in a reliable way.

All the sensors are connected with wires to a LON server,[5] which makes all the sensor readings available on the network. In Fig. 3 the distribution of physical components is chosen so that the main flow from low-level data to high-level information is from left to right: from sensor, to abstractor, event and data collection system, to database, to reasoning system and finally to application.

Our architecture uses the commonly used event-driven software pattern to provide high responsiveness. The abstractors running on the abstractors and DB server receive the sensor readings from the LON server, and update the basic events, including timing information, in the database, if necessary. Many off-the-shelf sensors already provide discretization, and for some sensors, for example humidity sensors, discretization was realized using thresholds on numerical values. Suitable values for some of these thresholds were empirically determined in a form appropriate for reuse, using statistics. In case of changes to the basic events, the database triggers the reasoner, running on the reasoning server, to perform the next reasoning cycle. The inferred conclusions (e.g. high-level events) are, combined with timing information, updated in the database, if necessary. A change in an event in the database derived by the reasoner is used to notify all clients subscribing to the event. In the setup of the localization application, there is only one client; this client puts the inferred data to the screen.

We use MonetDB,[6] a relational database management system which fulfils our data management requirements: support for active database functionality, client/server architecture to support multiple client programs accessing our database simultaneously, and small footprint to run on embedded hardware with limited resources. The data is retrieved and managed using SQL. The reasoning server uses Jena,[7] which supports RDF and OWL and also rules of the type used here. Jena also supports the N-Triples syntax which we used in the examples above and which helps to make our approach easy to use.

Figure 4 shows the screen of the application that visualizes output of the system. The top part of the screen is occupied by the map of the living environment showing all sensor firings and conclusions (e.g. room localization, walking, cooking, showering detection) as they happen. The bottom part shows three images from the video cameras installed: kitchen view (left), hall and bathroom view (middle) and bedroom view (right). The camera images on Fig. 4 were only used to evaluate the correctness of conclusions of the set-up against reality. The camera images were not used by the system.

After having done a number of tests with colleagues, we asked a woman aged 82 to perform normal daily activities like walking to the kitchen, preparing a cup of tea (see Fig.4), going to the dining table and drinking the cup of tea, going back to the kitchen and cleaning the cup, going to the bathroom to wash hands, and finally going to the bedroom to lie on the bed. All activities described in our knowledge base

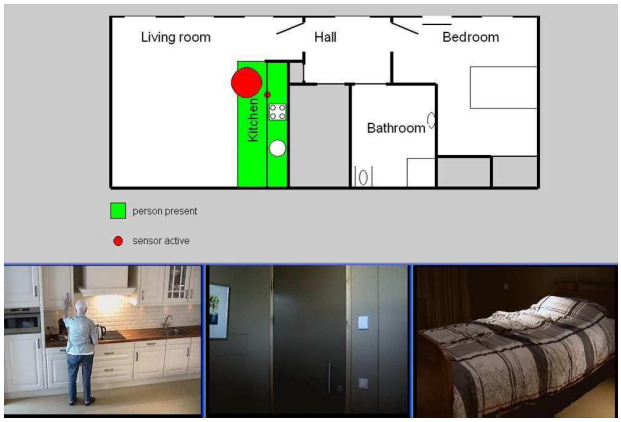

Fig. 4. Room localization application.

where detected correctly, no known activities were missed.

We can conclude from the performance analysis of the prototype that there are no bandwidth issues. In our deployed system, events are passed and processed fast enough to support activity recognition of a person living alone without undue delay. The largest computational load is caused by the reasoning process. This is expected as this process forms the most complex task of the system. One way to resolve this bottleneck is by running abstractors and local reasoners on embedded hub nodes. A local reasoner could then make conclusions based on the partial knowledge available at its end. An example of such knowledge partitioning could involve geometrical or topological information (e.g. perform reasoning with basic events happening in one room). One reason for the computational load caused by the reasoner is that Jena is a general purpose semantic web tool which is not optimized for R-entailment reasoning. Optimized implementations of R-entailment are emerging, e.g. BaseVISor.[8]

We are currently integrating our system in a broader infrastructure for pervasive health monitoring - this involves integrated, distributed data management - and continuing our work on validation. Figure 5 describes the configuration which we have recently introduced to homes of five elderly people and their caregivers. The system consists of a number of subsystems, one for each home, consisting of a relatively powerful hub/gateway node with connections to a number of heterogeneous sensors and possibly actuators. The optimal number of sensors depends on the desired use cases (e.g. activities to be detected), topology of the house or apartment, and sensor area coverage. We use wireless sensor nodes from Sensite Solutions.[9] As hub nodes we use small-size devices, gumstix,[10] which act as the event vault for progress monitoring and as internally used portals for distributed event handling. Each hub node runs abstractors and a database management

[5]http://www.echelon.com/products/lonworks
[6]http://www.monetdb.com
[7]http://jena.sourceforge.net
[8]http://vistology.com/basevisor/basevisor.html
[9]http://www.sensite-solutions.com/
[10]http://www.gumstix.com

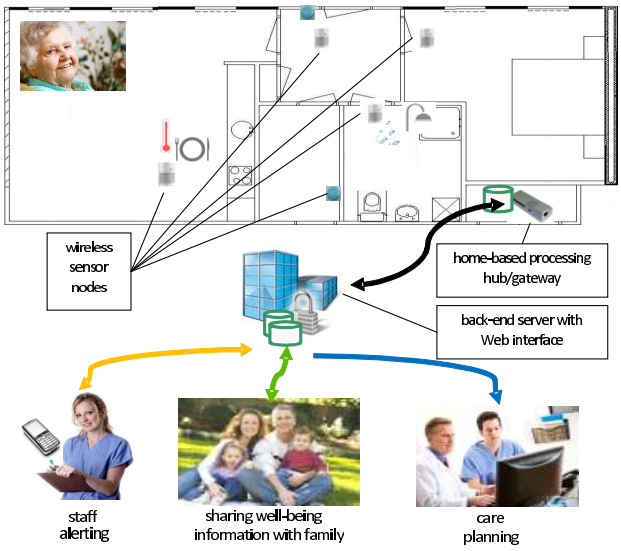

Fig. 5. Distribution of data in the system.

system, with currently a PC added for the reasoner. A backend system is used to archive high-level events coming from the home-based systems, perform long-term pattern analysis, and present information to various clients, e.g. family members and care specialists. This forms a hierarchical data distribution system, which is attractive from a scalability and fault tolerance perspective. We envision that simple abstractors which do not involve intensive data processing could run directly on sensor nodes. Moreover, local reasoners could run on hub nodes. We also envision a distributed database management system to be used to further abstract from the underlying sensor infrastructure and make distributed querying transparent to the rest of the system.

V. Conclusions

We have demonstrated that generalized events can be used to structure knowledge and reasoning for high-level interpretation of sensor data in independent living applications. Using R-entailment, a formalism for modeling knowledge is obtained which is easy to use in practice, which provides a flexible means to record knowledge and configuration data for different contexts, and which enables feasible reasoning using rules and ontologies at the time of service request. A database system can be used to efficiently link system components by managing and making available generalized events.

Acknowledgment

The authors would like to thank Stan Baggen for discussions in which the notions of value event and transition event were formulated, Richard Doornbos for useful discussions on deployment, and Simon Abernethy for help in implementation.

References

[1] A. Agostini, C. Bettini, D. Riboni: Hybrid reasoning in the CARE middleware for context awareness. Int. J. Web Engineering and Technology, 5, 3-23 (2009).

[2] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, D. Riboni: A survey of context modelling and reasoning techniques. Pervasive and Mobile Computing, 6, 161-180 (2010).

[3] S. Dalal, M. Alwan, R. Seifrafi, S. Kell, D. Brown: A rule-based approach to the analysis of elders' activity data. In: AAAI 2005. pp. 9-13

[4] A.K. Dey: Understanding and using context. Personal and Ubiquitous Computing 5, 4-7 (2001).

[5] M. ElHelw, J. Pansiot, D. McIlwraith, R. Ali, B. Lo, L. Atallah: An integrated multi-sensing framework for pervasive healthare monitoring. In: Pervasive Health 2009.

[6] P. Hayes: Naive Physics I: Ontology for Liquids. In: J.R. Hobbs, R.C. Moore (Eds.): Formal Theories of the Commonsense World pp. 71-107 (1985)

[7] P. Hayes: RDF Semantics, W3C Recommendation, http://www.w3.org/TR/2004/REC-rdf-mt-20040210

[8] K. Henricksen, J. Indulska: Developing context-aware pervasive computing applications: Models and approach. Pervasive and Mobile Computing 2, 37-64 (2006)

[9] I. Horrocks, P.F. Patel-Schneider: Reducing OWL entailment to description logic satisfiability. Journal of Web Semantics 1, 345-357 (2004)

[10] H.J. ter Horst, M. van Doorn, N. Kravtsova, W. ten Kate, D. Siahaan: Context-aware music selection using knowledge on the semantic web. Proc. 14th Belgium-Netherlands Conference on Artificial Intelligence pp. 131–138 (2002)

[11] H.J. ter Horst: Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. Journal of Web Semantics 3, 79–115 (2005)

[12] H.J. ter Horst: Combining RDF and part of OWL with rules: Semantics, decidability, complexity. In: Proceedings 4th Int. Semantic Web Conference. vol. 3729, pp. 668–684. Springer LNCS (2005)

[13] P. Hu, J. Indulska, R. Robinson: An autonomic context management system for pervasive computing. In: PERCOM 2008. pp. 213-223

[14] A. Katz, A.B. Ford, R.W. Moskowitz, B.A. Jackson, M.W. Jaffee: Studies of illness in the aged. The index of ADL: A standardized measure of biological and psychosocial function. Journal of the American Medical Association, 185, pp. 94-101 (1963)

[15] M.P. Lawton, E.M. Brody: Assessment of older people: Self-monitoring and instrumental activities of daily living. The Gerontologist, 9, pp. 179-186 (1969)

[16] U. Naeem, J. Bigham: Recognising activities of daily life through the usage of everyday objects around the home. In: Pervasive Health 2009.

[17] D. Nicklas, M. Grossmann, J. Mínguez, M. Wieland, M.: Adding high-level reasoning to efficient low-level context management: A hybrid approach. In: PERCOM 2008. pp. 447–452

[18] A. Ranganathan, R.E. McGrath, R.H. Campbell, M.D. Mickunas: Use of ontologies in a pervasive computing environment. Knowledge Engineering Review, 18, 209-220 (2004)

[19] A. Ranganathan, J. Al-Muhtadi, R.H. Campbell: Reasoning about uncertain contexts in pervasive computing environments. IEEE Pervasive Computing 3(2), 62-70 (2004)

[20] D. Riboni, C. Bettini: COSAR: Hybrid reasoning for context-aware activity recognition. Personal and Ubiquitous Computing (2010)

[21] R. Robinson, K. Henricksen, J. Indulska: XCML: A runtime representation for the Context Modelling Language. In: PERCOMW 2007. pp. 20-26

[22] S. Russell, P. Norvig: Artificial Intelligence. Prentice Hall, Englewood Cliffs (1995)

[23] D. Salber, A.K. Dey, G.D. Abowd: The context toolkit: Aiding the development of context-enabled applications. In: CHI 1999. pp. 434-441.

[24] H. Storf, M. Becker, M. Riedl: Rule-based activity recognition framework: Challenges, technique and learning. In: Pervasive Health 2009.

[25] E.M. Tapia, S.S. Intille, K. Larson: Activity recognition in the home using simple and ubiquitous sensors. In: PERVASIVE 2004. pp. 158-175

[26] D.H. Wilson, C.G. Atkeson: Simultaneous tracking and activity recognition (STAR) using many anonymous, binary sensors. In PERVASIVE 2005. pp. 62-79