Towards the Adaptive Integration of Multiple Context Reasoners in Pervasive Computing Environments

Daniele Riboni Linda Pareschi Claudio Bettini
Universita' degli Studi di Milano
D.I.Co., EveryWare Lab
via Comelico, 39, I-20135 Milano, Italy
Email: {riboni,pareschi,bettini}@dico.unimi.it

***Abstract*—The pervasive computing vision consists in realizing ubiquitous technologies to support the execution of people's everyday tasks by proactively providing appropriate information and services in a natural and transparent way based on the current context. Hence, a fundamental ingredient of pervasive computing is a mechanism to recognize the current high level context of users based on lower level context data provided, for instance, by body-worn and environmental sensors. Given the variability of encountered contextual conditions, the currently available data sources are highly dynamic; hence, context reasoning should continuously adapt to the change of available sources. In this paper we propose a technique to dynamically discover sources of context data, and to modularly integrate reasoners that use those data to infer higher level context information. Our proposal is corroborated by an implementation on mobile devices and sensors, and by an experimental evaluation showing its efficiency and effectiveness.**

I. Introduction

Automatic recognition of high level context information such as human activities is an enabling technology for many emerging ICT applications. Indeed, the capacity of automatically recognizing the current situation of users would greatly enhance the ability of a system to properly react and adapt itself to the actual circumstances. Currently, however, effective techniques to derive high level context data are available only for very confined domains (e.g., for the recognition of basic physical activities of users). On the contrary, effective recognition of more complex data (e.g., social interactions, working activities) out-of-the-lab is far from being achieved by techniques proposed so far.

In order to address the limitations of existing systems, we devised a context reasoning technique named *COSAR* [1] that combines symbolic and statistical reasoning methods. Experiments about activity recognition with COSAR have shown that the hybrid technique significantly improves recognition rates with respect to the use of solely statistical methods. However, like most state-of-the-art techniques proposed so far (e.g., [2], [3], [4], [5], [6], [7]), COSAR relied on the availability of a fixed set of sensor data.

We observe that, in pervasive computing environments, the number and kinds of currently available data sources may continuously change due to the variability of encountered contextual conditions. For instance, throughout the day a user can move from indoor environments in which a plethora of sensor devices is available, to outdoor environments in which sensor infrastructures are very limited. Similarly, depending on the current situation, the user can bring with her different portable devices having different sensing capabilities (consider, for instance, devices like mobile phones and smart watches that are equipped with accelerometers and GPS receivers, or garments like shoes equipped with motion sensors). Hence, the configuration of available sensors may continuously change depending on the users movements, on the devices she brings with her, and even on her outfit.

In this paper we extend COSAR to cope with dynamic configurations of sensors providing context data and, consequently, with the need to integrate dynamic sets of context reasoners that use those data to infer higher level context information. Different research challenges are involved in our work, including the design of a mechanism for dynamically retrieve context sources based on the requirements of context reasoners, and the definition of a hybrid intelligent system to integrate the different reasoning techniques. Our proposed solution adopts a publish/subscribe mechanism through which sensors advertise their presence, and context reasoners specify their requirements in terms of the context data they need. A lightweight software module, ideally hosted by a personal mobile device, is in charge of matching sensors specification and context reasoners requirements in order to dynamically activate reasoning. Since multiple reasoners may be available and running at the same time, the different predictions are integrated by a dedicated module, solving possible conflicts.

The rest of the paper is structured as follows. The overall system is shown in Section II. Section III illustrates the publish/subscribe mechanism. The technique to integrate the different predictions is presented in Section IV. Section V illustrates a prototype implementation of the system as well as experimental results. Section VI concludes the paper.

II. System overview

The overall system is shown in Figure 1. The lower layer (SENSORS) includes sensors, worn by the user or spread

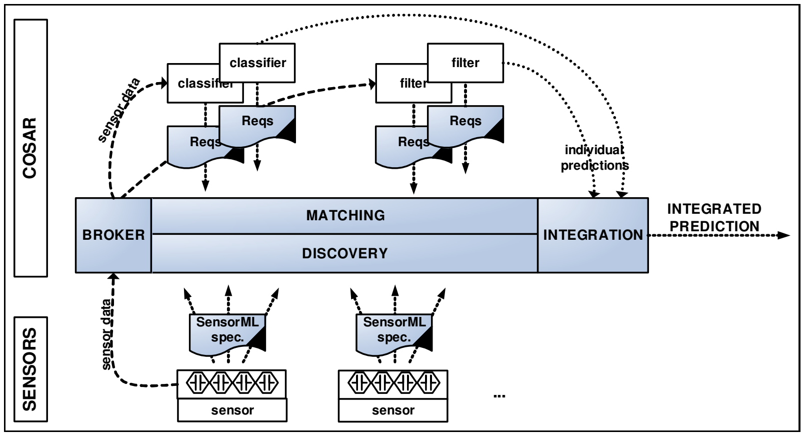

Figure 1. System overview

in the environment, which provide context data possibly useful to recognize higher level context data. Each sensor advertises its presence by broadcasting its specification, which is expressed through the OpenGIS Sensor Model Language (SensorML) [8]. Note that those sensors may be integrated into the user's mobile device.

The upper layer (COSAR) is hosted by the user's mobile device; it is in charge of performing hybrid reasoning to derive high level context data. COSAR supports multiple reasoners, including *classifiers* that provide weighted predictions about the recognized contexts, and *filters* that discriminate high level contexts as either possible or impossible with respect to lower level context data (for instance, activity "driving" is impossible if the user is in an indoor location, while it is possible if she is outdoor). Each reasoner specifies the sensor data it requires to perform its task in terms of needed data, sensor positioning, accuracy, and frequency of sampling. COSAR is in charge of discovering sensors that are in the communication range of the device. Reasoners' requirements are matched at run time with the discovered sensors, and when a matching is found, a stream communication is initiated from the sensors to the reasoner; then the reasoner becomes operational. When multiple reasoners operate at the same time, their individual predictions are combined to obtain an integrated prediction.

III. Dynamic activation of context reasoners

In this section we present the publish/subscribe mechanism used for activating context reasoners based on available sensor data. The mechanisms is based on standard description languages to enhance interoperability.

A. Sensor publishing

As explained in the introduction, in pervasive computing environments the set of available sensors may continuously change depending on the ever-changing context of users. Hence, a mechanism to discover the set of available sensors at run time is needed. Our solution adopts the OpenGIS Sensor Model Language (SensorML) [8] standard to describe the characteristics and placement of sensors. Through SensorML it is possible to specify a wide range of information, including sensor placement, kind of measured data, unit of measure, range, accuracy, frequency of sampling. For instance, consider the prototype of smart watch we are using for our experimentation (Figure 3). Part of its SensorML specification is reported in Table I (the XML syntax has been omitted to facilitate readability).

Each sensor advertises its presence by broadcasting through a wireless connection a short message containing a unique identifier of itself. Each time the DISCOVERY module of COSAR receives the message, it controls the unique identifier to check whether it has obtained the sensor description before. If the sensor description was not obtained before, that module asks for it, and the sensor responds with a SensorML *instance* document, which summarizes the set of measured data, and provides links to the detailed SensorML specifications of its components. Those specifications are published on the Web, and the sensor stores a copy of them on board. If COSAR has an Internet connection, it downloads them from the Web; otherwise it asks the sensor for them. Then, the MATCHING module queries the

description	Prototype of a smart watch worn on the right wrist
manufacturer	Sun Microsystems
sensorType	accelerometer
name	LIS3L02AQ Accelerometer
datum	The Z-axis is perpendicular to the Sun SPOT boards. The X-axis...
frequency	16Hz
quantity	accelerationX
unit	G
range	-6.0 6.0
quantity	tiltX
...	...

Table I
An excerpt of the smart watch specification

reasoners specifications to control if data provided by the sensor are required by a reasoner; in this case, it notifies the BROKER module, which initiates a stream communication from the sensor to that reasoner, and the reasoner becomes operational.

B. Reasoner subscription

Reasoners subscribe to those context data they need in order to operate. For instance, in our prototype system a statistical classifier is available that considers features extracted from 3-axis accelerations of an individual's right wrist. Hence, its specification states its requirements in terms of kind of data (3-axis accelerations), precise sensor placement (on the right wrist, with the X-axis perpendicular to the arm, ...), sampling frequency, etc. An excerpt of the specification of that classifier is shown in Table II (the actual XML syntax has been omitted for readability). The *model* field is specific to the reasoner; for instance, for the multiclass logistic regression [9] algorithm used by that classifier, it includes the encoded statistical model of the activities, automatically inferred from training data. When the needed data are available, the reasoner is activated, and its predictions are sent to the INTEGRATION module.

IV. INTEGRATING MULTIPLE PREDICTIONS

The COSAR technique supports multiple kinds of statistical and symbolic context reasoners. In this section we illustrate the mechanism for integrating multiple predictions.

A. Classifiers

Based on low level context data, classifiers provide weighted predictions about recognized high level context data, which correspond to the classifiers' classes. A relevant use of classifiers is the recognition of human activities. Generally speaking, activity inference systems based on statistical learning methods rely on a set of preclassified instances that are used in a training phase to learn a statistical model of a given set of activities. The obtained model is then used to automatically classify new activity instances, by assigning a level of confidence to the recognition of each activity. Hence, each prediction is represented by a

reasoner	multiclass logistic regression
inferredData	writingOnBlackboard,brushingTeeth,jogging,...
frequency	1Hz
model	1.0337; 0.9903; -0.9349; 1.2073; ...
data	X-axis acceleration
statistics	kurtosis
frequency	16Hz
data	Z-axis acceleration
statistics	90percentile
frequency	16Hz
...	...

Table II
AN EXCERPT OF A REASONER SPECIFICATION

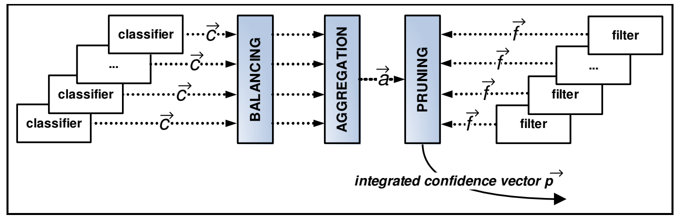

Figure 2. Integration of multiple context reasoners

confidence vector $\overrightarrow{c}$ in which the i^{th} element c_i corresponds to activity a_i, and its value corresponds to the confidence of the classifier regarding the recognition of a_i, such that $0 \leq c_i \leq 1$ and $\sum_{j=1}^{m} c_j = 1$.

B. Filters

In many cases, low level context data are insufficient to derive high level context information with enough confidence. However, low level data may be useful to restrict the set of possible high level contexts, by pruning some contexts that are clearly unfeasible. For instance, consider activity recognition, and suppose that the only available information is that the current user's location is a *classroom*. Without further information, it is impossible to draw conclusions about the actual user's activity. However, it is possible to understand that some activities are feasible in that context (e.g., "writing on a blackboard", "attending classes"), while many others are not. Hence, filters are in charge of pruning unfeasible high level contexts to restrict the candidate set. Each filter prediction is represented by a *filter vector* $\overrightarrow{f}$ in which the i^{th} element f_i corresponds to activity a_i, and its value corresponds to 1 if the activity is feasible given the low level context data; it is 0 otherwise. For instance, COSAR adopts a filter based on ontological reasoning to derive the set of feasible activities based on low level context data such as symbolic location, time, nearby persons, artifacts, and communication routes.

C. Integration

In general, different classifiers may recognize different sets of high level contexts. For the sake of this work, we assume that at most one high level context may be recognized at a time. Extensions to recognize concurrent contexts will be investigated in future work.

The integration mechanism is illustrated in Figure 2; it is composed of three phases. The first phase aims at *balancing* the individual predictions of each classifier by considering the number m of its classes. Indeed, since for each instance, each classifier distributes the same amount of confidence (i.e., 1) among its classes, the average confidence assigned by a classifier to each class is in inverse relation to the number of its classes. Hence, for each classifier, the values of the confidence vector are multiplied by the factor $\frac{m}{M}$, where M is the number of classes considered by at least one classifier.

After balancing, the confidence vectors C are *aggregated* to obtain an M-dimensional *aggregated confidence vector* $\overrightarrow{a}$. Aggregation is performed by averaging on the balanced predictions of the individual classifiers. Hence, for each $i \in \{1, 2, \ldots, M\}$, $a_i = \frac{\sum_{\overrightarrow{c}^{(j)} \in C} c_i^{(j)}}{|C_i|}$, where C_i is the set of classifiers for the class corresponding to c_i.

The last phase is the one of *filtering*. In this phase, the aggregated confidence of classes considered unfeasible by at least one filter is set to 0. The result of filtering is the *integrated confidence vector* $\overrightarrow{p}$, such that for each $i \in \{1, 2, \ldots, M\}$, $p_i = \begin{cases} 0 & \textit{if } \exists \overrightarrow{f} \in \overrightarrow{F} : f_i = 0 \\ a_i & \textit{otherwise} \end{cases}$, where $\overrightarrow{F}$ are the filter vectors.

V. Experimental evaluation

In this section we present a prototype implementation of our system, as well as a preliminary experimental evaluation of our approach.

A. Prototype implementation

In order to validate our approach, we have implemented a prototype of our proposed system. The upper layer (COSAR) has been implemented for the Android[1] platform; the graphical interface is shown in Figure 3. The prototype supports the recognition of physical and daily living activities such as walking, jogging, brushing teeth, etc. It includes two statistical classifiers and two filters. One classifier exploits acceleration-based features regarding the hip; the other one, regarding the right wrist. One filter, mentioned in Section IV-B, is based on ontological reasoning; the other one is rule-based, and considers the user's speed. The Android prototype implements the algorithms for the DISCOVERY, MATCHING, BROKER and INTEGRATION modules described in the previous sections.

The SENSOR layer has been implemented on *Small Programmable Object Technology (SPOT[2])* sensors by Sun® Microsystems, and on the Android platform. Sensors include a prototype of smart watch (shown in Figure 3) in charge of providing acceleration of the right wrist through the 3-axis accelerometer of the SPOT, and a SPOT widget to provide the current symbolic location. The Android device embeds sensors for accelerations of the hip (gathered from the embedded accelerometer, assuming that the user holds the device in her pocket), and speed, retrieved from the embedded GPS receiver. SPOT sensors broadcast their SensorML specification through wireless connections, while the specifications of the other sensors are stored on the Android device itself.

[1]http://www.android.com

[2]http://www.sunspotworld.com

Figure 3. COSAR implementation for Android and smart watch prototype

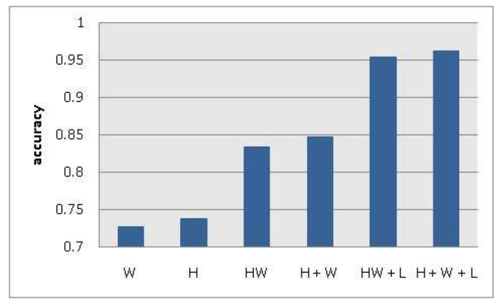

Figure 4. Accuracy (labels are explained in Section V-C)

B. Experimental setup

The goal of these experiments is to evaluate *i)* the effectiveness of our approach to combine multiple reasoners in a dynamic fashion, and *ii)* the feasibility of implementing our system on resource-constrained mobile devices.

Experiments concerned the recognition of 10 different activities performed both indoor and outdoor for a total of 5 hours by 6 volunteers wearing one Sun SPOT sensor on their left pocket and another one on their right wrist to collect accelerometer data, plus a GPS receiver to track their location. More details regarding our dataset may be found in [1].

C. Accuracy

In order to assess the effectiveness of our integration mechanism, we compared our technique with a monolithic one in which a single classifier exploits the whole set of features calculated from available context sources. We point out that, while in principle the monolithic approach could provide better recognition results with respect to a modular one, in pervasive computing environments the monolithic approach is unrealistic, since the set of context sources that will be available cannot be known in advance, and training the classifier based on every possible combination of context sources is unfeasible. Hence, the goal of these experiments is to show that the modular approach is as accurate as the monolithic one, while being applicable to real-world

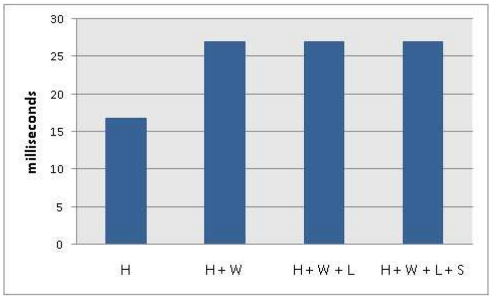

Figure 5. Computational cost (H classifier with hip data; H+W classifiers with hip and wrist data, respectively; H+W+L classifiers and location filter; H+W+L+S classifiers, location and speed filters)

scenarios.

Results of the comparison are shown in Figure 4. The label W denotes the classifier based on acceleration features from the wrist, while H from the hip. With the label H+W we denote the COSAR approach exploiting acceleration features from the hip and wrist, while HW denotes the monolithic one exploiting the same data. Similarly, H+W+L (resp. HW+L) denotes the COSAR (resp. monolithic) approach using the accelerations and the symbolic location. Results are positive, since they show that our modular technique slightly outperforms the monolithic one in terms of accuracy.

D. Efficiency

In order to evaluate the feasibility of implementing our system on resource-constrained mobile devices, we have performed experiments with the Android implementation of COSAR by varying the number and kinds of running reasoners. We did not perform experiments with sensors, since they are only in charge of communicating data to the COSAR layer. Experiments have been performed with an HTC Magic^{TM} device (528MHz processor, 288MB RAM) running the Android implementation of COSAR. Results are shown in Figure 5, and report the aggregated execution time of the most computationally expensive operations, i.e., extraction of features from the received data, and execution of the classification tasks. In this experiments, context reasoning was repeated every 1 second, and the number of features (after feature selection) for each classifier was around 20. We performed experiments with increasing number of running reasoners. Even using two classifiers and two filters, the aggregated execution time per second was below 30ms.

VI. Conclusions and future work

In this paper we proposed techniques for dynamic discovering context sources, and for modularly integrating context reasoners to infer high level context information. The efficiency and effectiveness of our techniques is shown by experimental results with a prototype implementation of the system on mobile devices and sensors. Future work includes an extension of our techniques to consider the temporal characterization of activities (e.g., duration), as well as their temporal relationships (i.e., the probability that a given activity a_i is followed by an activity a_j). To this aim, we plan to design a temporal extension of our ontology, and to investigate the use of a probabilistic framework such as Hidden Markov Models.

Acknowledgments

This work has been partially supported by a grant from Sun® Microsystems. The authors would like to thank Francesco Noya for his excellent programming work.

References

[1] D. Riboni and C. Bettini, “Context-Aware Activity Recognition through a Combination of Ontological and Statistical Reasoning,” in *Proceedings of the 6th International Conference on Ubiquitous Intelligence and Computing (UIC 2009)*, ser. Lecture Notes in Computer Science, vol. 5585. Springer, 2009, pp. 39–53.

[2] O. Amft and G. Tröster, “Recognition of dietary activity events using on-body sensors,” *Artificial Intelligence in Medicine*, vol. 42, no. 2, pp. 121–136, 2008.

[3] O. Brdiczka, J. L. Crowley, and P. Reignier, “Learning Situation Models for Providing Context-Aware Services,” in *Proceedings of HCI 2007*, ser. Lecture Notes in Computer Science, vol. 4555. Springer, 2007, pp. 23–32.

[4] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford, “A Hybrid Discriminative/Generative Approach for Modeling Human Activities,” in *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, L. P. Kaelbling and A. Saffiotti, Eds. Professional Book Center, 2005, pp. 766–772.

[5] L. Liao, D. Fox, and H. A. Kautz, “Location-Based Activity Recognition using Relational Markov Networks,” in *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*. Professional Book Center, 2005, pp. 773–778.

[6] S. Wang, W. Pentney, A.-M. Popescu, T. Choudhury, and M. Philipose, “Common Sense Based Joint Training of Human Activity Recognizers,” in *Proceedings of IJCAI 2007*, 2007, pp. 2237–2242.

[7] M. Stikic, T. Huynh, K. V. Laerhoven, and B. Schiele, “ADL Recognition Based on the Combination of RFID and Accelerometer Sensing,” in *Proceedings of Pervasive Health 2008*. IEEE Computer Society, 2008, pp. 2237–2242.

[8] M. Botts and A. Robin, “OpenGIS Sensor Model Language (SensorML) Implementation Specification,” Open Geospatial Consortium Inc., Tech. Rep. OGC 07-000, 07 2007.

[9] S. le Cessie and J. van Houwelingen, “Ridge Estimators in Logistic Regression,” *Applied Statistics*, vol. 41, no. 1, pp. 191–201, 1992.