Proceedings of the 2013 Federated Conference on
Computer Science and Information Systems pp. 993–998

Learning sensors usage patterns in mobile context-aware systems

Szymon Bobek*, Krzysztof Porzycki†, Grzegorz J. Nalepa‡
AGH University of Science and Technology
Al. Mickiewicza 30, 30-059, Krakow, Poland
*szymon.bobek@agh.edu.pl, †kporzyck@student.agh.edu.pl, ‡gjn@agh.edu.pl

***Abstract*—Context-aware mobile systems have gained a remarkable popularity in recent years. Mobile devices are equipped with a variety of sensors and become computationally powerful, which allows for real-time fusion and processing of data gathered by them. However, most of existing frameworks for context-aware systems, are usually dedicated to static, centralized architectures, and those that were designed for mobile devices, focus mainly on limited resources in terms of CPU and memory, which in nowadays world is no longer a big issue. Mobile platforms require from the context modelling language and inference engine to be simple and lightweight, but on the other hand – to be powerful enough to allow not only solving simple context identification tasks but also more complex reasoning. These, with combination of a large number of sensors and CPU power available on mobile devices result in high energy consumption of a system. The original contribution of this paper is a proposal of an intelligent middleware for mobile context-aware frameworks, that is able to learn sensor usage habits, and minimize energy consumption of the system.**

I. Introduction

RESEARCH in the area of pervasive computing and ambient intelligence aims to make use of context information to allow devices or applications behave in a context-aware, thus "intelligent" way. Dey [1] defines context as *"any information that can be used to characterize the situation of an entity*. The *information* in Dey's definition may be: (1) location of the user (spatial context), (2) presence or absence of other devices and users nearby, or collaboration with other users (social context), (3) time (temporal context), (4) user *behavior* or *activity*, and possibly (5) any other environmental data gathered by microphones, light sensors, etc.

The variety of sensors available on nowadays mobile devices allow for complex contex-based reasoning, but at the same time requires a lot of resources and energy.

Although there are many frameworks and middlewares developed for context-aware systems [2], [3], [4], they do not provide full support for all of the challenges that we believe are crucial for mobile computing (e.g. smartphones or tablets), with respect to the context modelling and context-based reasoning. Those are:

Energy efficiency – most of the sensors, when turned on all the time, decrease the mobile device battery level very fast. This reflects on usability of the system and ecological aspects regarding energy saving.

Data privacy – most of the users do not want to send information about their location, activities, and other private data to external servers. Hence, the context reasoning should be performed by the mobile device itself.

Resource limitations – although mobile phones and tablets are becoming computationally powerful, the context aware system has to consume as low CPU and memory resources as possible in order to be transparent to the user and other applications.

System responsiveness – in mobile environment context changes very fast. Hence, no delays are admissible in processing contextual data.

Context data distribution – in mobile pervasive environments many devices produces huge amount of contextual information, hence the quality measures should be developed and distribution methods designed to fit characteristics of such unstable and dynamic network [5].

All of these require from the modelling language and inference engine to be simple and lightweight. On the other hand, the model should be powerful enough to allow not only solving simple context identification tasks but also more advanced context processing and reasoning.

This gives motivation for developing a solution that will allow for using advanced reasoning and modelling techniques, with as low energy cost as possible. The original contribution of the paper is a proposal of an intelligent middleware for mobile context aware frameworks, that is able to learn sensor usage habits, and minimize energy consumption of the system.

The rest of the article is organized as follows: In Section II an existing context aware systems and frameworks are presented, and the motivation of the paper is given. The architecture that can be used in combination with our approach is presented in Section III. The Section IV discusses the learning algorithm used for intelligent middleware and Section V presents an evaluation of the algorithm. Finally, summary and directions for future work are given in Section VI.

II. State of the art and motivation

In recent years, a lot of development was devoted to build applications that use mobile devices to monitor and analyse various user contexts. The availability of application distribution platforms for common mobile operating systems, e.g. Google Play for Android stimulated the popularity and adoption of such solutions. However, most of them focus only on a very narrow application area of context awareness. Most of them are end user applications, and not more generic

978-1-4673-4471-5/$25.00

frameworks. Some selected representative cases are briefly described below.

A. Context aware systems

The SocialCircuits platform [6] uses mobile phones to measure social ties between individuals, and uses long- and short-term surveys to measure the shifts in individual habits, opinions, health, and friendships influenced by these ties.

Jung [7] focused on discovering social relationships (e.g., family, friends, colleagues and so on) between people. He proposed an interactive approach to build meaningful social networks by interacting with human experts, and applied the proposed system to discover the social networks between mobile users by collecting a dataset from about two millions of users. Given a certain social relation (e.g., isFatherOf), the system can evaluate a set of conditions (which are represented as propositional axioms) asserted from the human experts, and show them a social network resulted from data mining tools.

Sociometric badge [8] has been designed to identify human activity patterns, analyse conversational prosody features and wirelessly communicate with radio base-stations and mobile phones. Sensor data from the badges has been used in various organizational contexts to automatically predict employee's self-assessment of job satisfaction and quality of interactions.

Eagle and Pentland [9] used mobile phone Bluetooth transceivers, phone communication logs and cellular tower identifiers to identify the social network structure, recognize social patterns in daily user activity, infer relationships, identify socially significant locations and model organizational rhythms.

Beside research projects, there exist also a variety of application that are used for gathering information about context from mobile devices, like SDCF [10], AWARE [1], JCAF [11], SCOUT [12], ContextDriod [13], Gimbal [2]. These are mostly concerned with low-level context data acquisition from sensors, suitable for further context identification. On the other hand, they do not provide support nor methodology for creating complex and fully customizable context-aware systems and do not provide any mechanisms for limiting energy consumption of the system.

What is more, all of the approaches described above use their own dedicated methods for gathering and maintaining context. These methods are mostly not applicable for reuse, or their functionality is limited to simple context matching. Some of the systems do not provide any support for context modelling nor context reasoning, limiting their functionality only to identifying and collecting contextual information.

B. Context aware frameworks

To solve the issue of reusability of the system, a lot of frameworks were designed. These frameworks are based on many different architecture paradigms which pros and cons in terms of energy efficiency, responsiveness, and privacy were presented in this Section.

[1]http://www.awareframework.com
[2]https://www.gimbal.com/

The system described in [14] uses *direct sensor access* architecture which is usually not very energy efficient, however it preserver privacy issues, since no communication with external servers is usually needed, and the interpretation of the sensor data as well as reasoning is performed directly on the host device

The CoBrA system [15] was build on *centralized context server* architecture. This approach is especially useful when a context-aware system is composed of many mobile devices with limited resources. The server relieves mobile agents from performing reasoning tasks. On the other hand, one has to consider privacy issues connected with sending private contextual data to remote server, quality of service issues, etc. This approach is also characterized with rather low responsiveness that stems from a possible lack of network connection or communication delays.

Service oriented architecture with combination of *distributed architecture* was used in SOCAM [16] system. In context-aware applications this architecture is used mainly in pervasive environment, where variety of context information from many different sources has to be processed. This architecture usually does not preserves privacy nor energy efficiency issues since usually it assumes communication over the web between each of its elements. Although SOCAM provides architecture for distributed mobile systems, it mostly solves problems of a low memory and CPU power of mobile agents, which nowadays is no longer a big issue for most of the mobile devices like smart-phones or tablets. On the other hand, energy efficiency issue is still a big problem, which was not addressed by none of the solutions described in this Section.

This gives motivation for developing an architecture that will allow for advanced context-based reasoning and modelling, but at the same time allow for minimizing energy usage costs of sensors that are needed in such reasoning. An overview of the proposed system is presented in following Section.

III. Intelligent middleware approach

The proposed solution incorporates an idea of a mobile device as an autonomous context-aware entity, equipped with intelligent middleware layer and context-based inference layer. The intelligent middleware act as a proxy between context sources and inference layer. It is able to learn sensor usage patterns and thus adjusting sampling rates to significantly improve energy consumption of the system (See Section IV).

The architecture of a system that may use intelligent middleware approach should consist of three main elements:

1) sensors layer – responsible for gathering data from sensors and performing initial preprocessing of them,
2) inference layer – responsible for context based reasoning and knowledge management, and
3) intelligent middleware layer – acting as an intelligent proxy between sensors layer and the inference layer.

The *Sensor Layer* gathers data directly from mobile device sensors. Due to the different possible sensor types (GPS, Accelerometer, Bluetooth), different methods for interpreting

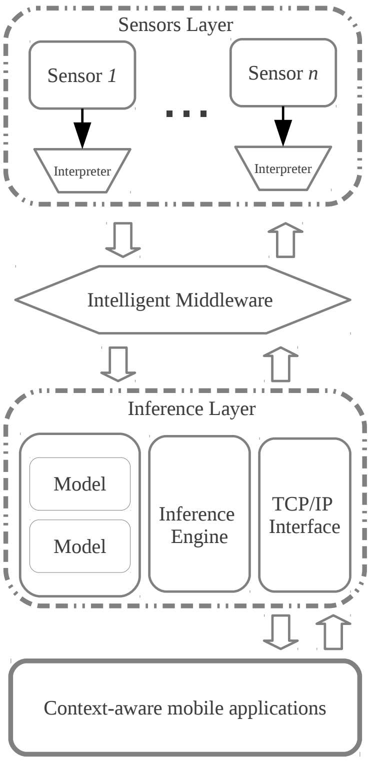

Figure 1. Architecture of the mobile context aware framework

these data are required. Hence, each sensor has its own interpreter module that is responsible for initial preprocessing of the raw data. Data preprocessing is triggered by the intelligent middleware.

The *Inference Layer* is responsible for performing reasoning, based on the model (knowledge base) and the working memory elements (facts). The inference engine may be a rule engine, first-order logic reasoner, probabilistic inference module, or any other custom approach. However, we argue, that to allow more complex reasoning tasks than just simple context classification, the best choice is lightweight rule engine [17].

The *Intelligent Middleware* is responsible for exchanging information between sensors layer and inference layer. The working memory is shared between all models stored within the inference layer, acting as a *knowledge cache*. Therefore, it minimizes the number of required requests to the sensors layer, improving power efficiency of the entire system.

The idea of separating Intelligent Middleware from inference layer is that it is able to learn sensors usage habits, and in consequence adapt itself to the individual device characteristics. It automatically generates a model of usage habits from historical data and based on that model data it adjusts the sampling rates for the sensors appropriately. It improves power efficiency of the system, since sampling rates are not fixed, but leaned from the usage patterns. On the other hand it may help in increasing responsiveness of the system, since the learned model allows predicting not only future sensor activity, but also context-aware application needs. Hence, it is possible to get the desired context in advance, before the application actually requests it. It can be especially useful in cases when context cannot be obtained by the middleware directly from the sensor layer, but has to be for example downloaded over the internet. However in this paper we focus only on the power efficiency advantage of the usage of the intelligent middleware approach.

The following sections describes in details the learning algorithm used, and provide an evaluation on a simple use case scenario.

IV. Learning algorithm

Input data. The algorithm takes as an input a vector of m percepts. Each percept is described by a pair (X_i, Y_i) interpreted respectively as time of percept and sensor activity state. Such a notation results in two vectors X, Y of size m such that $\forall_{i<m} 0 \leq X_i \leq 24 \wedge (Y_i = -1 \vee Y_i = 1)$. Time equals numbers of hours passed since last midnight and percept is:

$$Y_i = \{^{-1\ for\ inactive\ state}_{1\ for\ active\ state}$$

Learning objective. Sensor activity depends largely on its stochastic and inaccessible environment. Being so, it is impossible to predict it with absolute certainty, however, often some part of its variance can be explained by time of a day. The algorithm proposed aims to exploit that possibility by finding a function determining probability of sensor usage given time of a day $F(t) = P(X = 1|t)$. Problems of learning conditional probability are often addressed in Machine Learning by using logistic regression. The following paragraphs define necessary concepts and present the problem in terms of logistic regression with accordingly chosen parameters.

Hypotheses set. Finding objective function $F(t)$ is achieved by searching a hypotheses set H. Each function h in hypotheses set has to have the following properties:

1) be continuous,
2) be defined in range $< 0; 24 >$,
3) $h(0) = h(24)$,
4) return values in range $< 0; 1 >$ (probability).

To perform search it is necessary to represent each function in H in a general form parametrized by some vector w of length $2n+1$, such that every combination of parameters in w will yield in a proper hypothesis $h = H_w$. Such representation allows to transition from searching a set of functions to searching a linear space $\mathbb{R}^{2n+1}$.

A representation that has the first three of required properties is given below:

$$S(\omega, t) = \omega_0 + \sum_{i=0}^{n-1} \left(\omega_{2i+1} \cos\left(\frac{i*t*12}{\Pi}\right)\right) + \sum_{i=0}^{n-1} \left(\omega_{2i+2} \sin\left(\frac{i*t*12}{\Pi}\right)\right)$$

It may be understood as a sum of some first terms of Trigonometric Fourier Series parametrized by vector ω. Using only low frequency components is desirable because they are most likely to describe habits of usage that usually occurs in long sequences of same actions. The only requirement left, that is – unbound return values, can be addressed by composing function $S(\omega, t)$ with sigmoid function:

$$\theta(x) = \frac{1}{1 + e^{-x}}$$

The resulting and correct hypotheses set parametrized by vector w is given by formula:

$$H_\omega(t) = \theta(S(\omega, t))$$

Interpreting hypothesis as probability

Assumed interpretation that $P(y = 1|x) = h(x)$ implies that $P(y = -1|x) = 1 - h(x)$. Because $h(x) = \theta(s(\omega, t))$, and the properties of θ: $\theta(-s) = 1 - \theta(s)$, the resulting probability formula is drawn:

$$P(y|x) = \theta(y * S(\omega, t))$$

This formula is based on the assumption made earlier, that y takes 1 for active and -1 for inactive state.

Learning input data Out of all possible functions in the hypotheses set one has to be chosen in terms of its lowest cost. In order to perform such a selection a cost measure has to be defined. The suggested measure is a combined probability of all the observations in a learning set. The higher the combined probability the better a hypothesis describes the user habit that gave rise to such sensor readings. Derivation of final formula to be optimized: $\max_\omega \prod_{i=0}^{m-1} P(Y_i|X_i)$ Maximizing an expression is equivalent to maximizing its logarithm:

$$\max_\omega \ln \prod_{i=0}^{m-1} \theta(Y_i * S(\omega, X_i))$$

$$\max_\omega \frac{1}{m} \ln \prod_{i=0}^{m-1} \theta(Y_i * S(\omega, X_i))$$

$$\min_\omega -\frac{1}{m} \ln \prod_{i=0}^{m-1} \theta(Y_i * S(\omega, X_i))$$

$$\min_\omega -\frac{1}{m} \sum_{i=0}^{m-1} ln(\frac{1}{\theta(Y_i * S(\omega, X_i))})$$

Formula in such a form is then subject to optimization. The optimization algorithm used in this case was gradient descend with initial ω coefficients set all to 0. Fast convergence to unique value is always achieved thanks to minimized formula being always convex.

V. Evaluation

We implemented a prototype of an intelligent middleware, that learns user habits based on the usage of device sensors (in this case a GPS sensor). We assumed that the GPS sensor is active if the speed of the device exceeds some fixed threshold, otherwise the sensor was assumed to be inactive. This reflects to the cases where someone was moving or not.

The learning and evaluation process is presented in Figure 2. We first performed an acquisition of the GPS sensor data and save it to SQLite database. We collected samples from 5 consecutive days, which later were preprocessed offline to be ready for the learning algorithm. The main aim of the preprocessing phase was to decide weather the GPS sensor was active or not, depending on the speed threshold. After the learning process was finished, we moved learned model back again to the mobile device and based on that, we were adjusting sampling rates of the sensors.

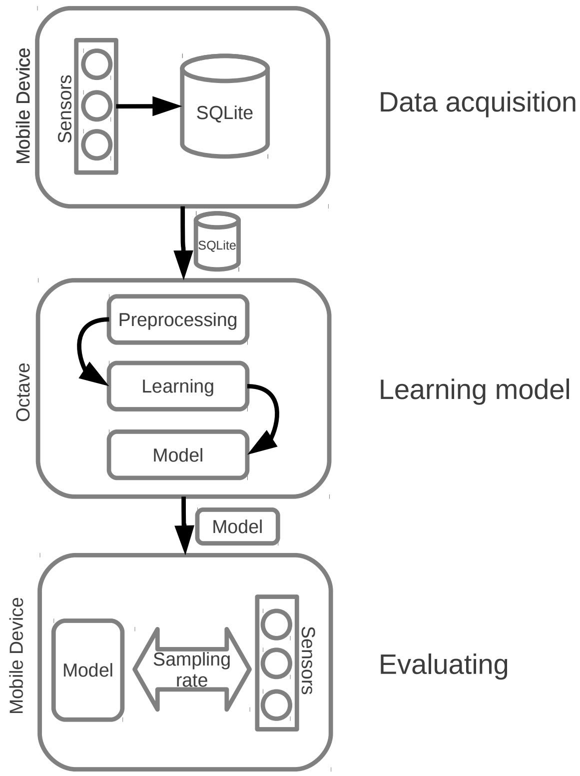

Figure 2. Learning and evaluation process of the intelligent middleware approach.

In the following Sections more details about implementation and evaluation results is presented.

A. Implementation

The prototype of the learning algorithm was written in Octave, and the evaluation of the learned model was performed on a Samsung Galaxy S II smartphone with Android 4.2 Jelly Bean installed.

The Octave learning phase was performed according to the learning algorithm described in Section IV. Fragment of a gradient descent source code responsible for learning parameters of the model is presented below.

```
for i=1:max_iterations,
  derivatives = zeros(nparams,1);
  for j=1:nsamples,
    product = -Y(j)*(X(j,:)*weights);
    error(i) += log(1+exp(product))/nsamples;
    sigm = sigmoid(product);
    for k=1:nparams,
       derivatives(k)=derivatives(k) +
```

```
          sigm*(Y(j)*X(j,k));
      end
    end
    derivatives = -derivatives / nsamples;
    weights -= learning_rate*derivatives;
end
```

During the evaluation phase, we used an Android device with a model of the sensor usage habits generated by the Octave algorithm. The algorithm that was adjusting sampling rates based on the learned model, performed following steps:

- Sample GPS sensor with a rate predicted by the intelligent middleware algorithm.
- When any movement is discovered, start sampling with the highest possible rate called `baseFreq` (we fixed this to be 1 second).
- After some fixed period of time called `continuityThreshold` (10 seconds in our approach), if no sensor activity was discovered, return to sampling rate predicted by the intelligent middleware algorithm.

The source code fragment responsible for this is presented below:

```
if(timeFromLastActivity < continuityThreshold){
   newFreq = baseFreq;
} else{
  float probability =
    middleware.getProbability(clockTime);
  int multiplier =
    (1.0f - probability) * scaleFactor + 1.0f;
  newFreq = baseFreq * multiplier;
}
if(newFreq != refreshFrequency)
  rescheduleUpdatesFromProviders(newFreq);
```

B. Results

We made experiments on two identical devices carried by the same person. One device was equipped with and intelligent middleware algorithm implemented and the other does not. Both devices were fully charged at the beginning of the experiment and was not recharged during it. We decided to use speed threshold equal to 5 km/h. With lower thresholds, the difference between intelligent middleware approach and the other one was hardly visible, because of the errors in GPS sensor readings which results in fake "active" states. As depicted in Figure 3, the intelligent middleware approach allowed for 50% battery saving than in case of the device without the algorithm implemented.

Figure 3 presents a proportion of the time that both devices worked on the battery. The right plot shows the time that the device without the intelligent middleware implemented worked, and the left plot presents a work time of the device with the intelligent middleware implemented.

The distance error which we define as a difference between the GPS samples generated in our approach and referenced samples generated by the approach without learning algorithm, is presented in Figure 4. The average distance error of the presented data equals 0.053 km. The high error in several

Figure 3. Difference in power consumption for device with and without learning algorithm implemented, with speed threshold set to 5km/h.

places on the plot is a result of noisy readings of GPS sensor rather than an algorithm fault.

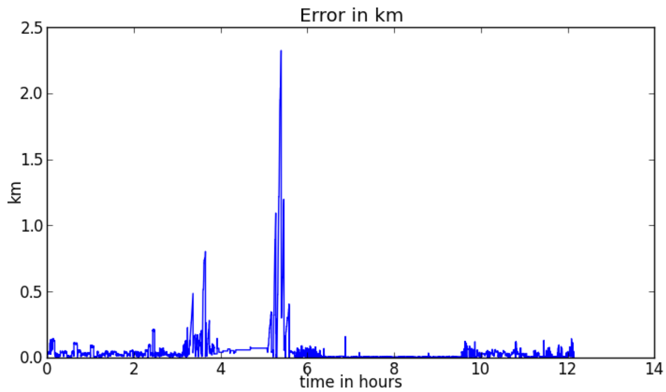

Figure 4. Error between position designated by the intelligent middleware approach and the reference data.

VI. SUMMARY AND FUTURE WORK

In this paper we presented a prototype of an intelligent middleware approach that is able to learn sensor usage habits and adjust sensor sampling rates to minimize energy consumption of the context-aware system. The middleware was presented as a part of a context reasoning platform tailored to the needs of such intelligent distributed mobile computing devices. We argue that most of the existing solutions are not fully applicable to mobile architectures, and does not fulfil energy efficiency needs of context-aware distributed systems. The presented approach was designed to solve that issue, however we believe that it is suitable for predicting not only future sensor activity, but also context-aware application needs. Hence, it is possible to get the desired context in advance, before the application actually requests it. It can be especially useful in cases when context cannot be obtained by the middleware directly from the sensor layer, but has to be for example downloaded over the internet.

We used a logistic regression algorithm to learn sensor usage model form historical data. This allowed for adjusting sampling rates of the sensors according to usage probability.

Evaluation on a real device showed that we can gain up to 50% of energy saving using this algorithm.

As a future work the implementation of the algorithm for an Android device is planned to allow real-time online learning and full evaluation of the intelligent middleware approach, not only for a GPS, but also other sensor like accelerometer, gyroscope, etc.

It is also planned to implement the learning algorithm that uses Markov chains, and compare it to existing implementation. We plan to design and develop an architecture dedicated for mobile context aware applications equipped with an intelligent middleware layer and rule based inference layer provided by the HeaRT [18] inference engine, which is a lightweight rule-based engine that uses XTT2 [19] notation for knowledge representation. This will allow for lightweight reasoning [20] and also verification of context models [21]. We plan to incorporate and evalueate the middleware in the context-aware system for monitoring threats in urban environment proposed in [22]. We also believe that it would be valuable to compare challenges and problems in mobile context-aware computing with an area connected with research about wireless sensor networks [23]. This two fields of science can possibly benefit from exchanging solutions and ideas especially regarding energy efficiency and resource limitations issues.

Acknowledgment

The paper is supported by the AGH University of Science and Technology Grant 11.11.120.859.

References

[1] A. K. Dey, "Providing architectural support for building context-aware applications," Ph.D. dissertation, Atlanta, GA, USA, 2000, aAI9994400.

[2] F. Sahafipour and R. Javidan, "A comparative study of context modeling approches and applying in an infrustructure," *Canadian Journal on Data Information and Knowledge Engineering*, vol. 3, no. 1, 2012.

[3] T. Strang and C. Linnhoff-Popien, "A Context Modeling Survey," in *In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England*, 2004. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.2060

[4] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of context modelling and reasoning techniques," *Pervasive Mob. Comput.*, vol. 6, no. 2, pp. 161–180, Apr. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.pmcj.2009.06.002

[5] P. Bellavista, A. Corradi, M. Fanelli, and L. Foschini, "A survey of context data distribution for mobile ubiquitous systems," *ACM Comput. Surv.*, vol. 44, no. 4, pp. 24:1–24:45, Sep. 2012. [Online]. Available: http://doi.acm.org/10.1145/2333112.2333119

[6] I. Chronis, A. Madan, and A. S. Pentland, "Socialcircuits: the art of using mobile phones for modeling personal interactions," in *Proceedings of the ICMI-MLMI '09 Workshop on Multimodal Sensor-Based Systems and Mobile Phones for Social Computing*, ser. ICMI-MLMI '09. New York, NY, USA: ACM, 2009, pp. 1:1–1:4.

[7] J. J. Jung, "Contextualized mobile recommendation service based on interactive social network discovered from mobile users," *Expert Syst. Appl.*, vol. 36, no. 9, pp. 11 950–11 956, Nov. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2009.03.067

[8] D. Olguin, B. N. Waber, T. Kim, A. Mohan, K. Ara, and A. Pentland, "Sensible organizations: Technology and methodology for automatically measuring organizational behavior," *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS-PART B: CYBERNETICS*, pp. 43–55, 2009.

[9] N. Eagle and A. (Sandy) Pentland, "Reality mining: sensing complex social systems," *Personal Ubiquitous Comput.*, vol. 10, no. 4, pp. 255–268, Mar. 2006.

[10] M. Atzmueller and K. Hilgenberg, "Towards capturing social interactions with sdcf: An extensible framework for mobile sensing and ubiquitous data collection." in *Proc. 4th International Workshop on Modeling Social Media*. ACM Press, 2013.

[11] J. E. Bardram, "The java context awareness framework (jcaf) – a service infrastructure and programming framework for context-aware applications," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, H.-W. Gellersen, R. Want, and A. Schmidt, Eds. Springer Berlin Heidelberg, 2005, vol. 3468, pp. 98–115. [Online]. Available: http://dx.doi.org/10.1007/11428572_7

[12] W. V. Woensel, S. Casteleyn, and O. D. Troyer, *A Framework for Decentralized, Context-Aware Mobile Applications Using Semantic Web Technology*, 2009.

[13] B. van Wissen, N. Palmer, R. Kemp, T. Kielmann, and H. Bal, "ContextDroid: an expression-based context framework for Android," in *Proceedings of PhoneSense 2010*, Nov. 2010. [Online]. Available: http://sensorlab.cs.dartmouth.edu/phonesense/papers/Wissen-ContextDroid.pdf

[14] E. Elnahrawy and B. Nath, "Context-aware sensors," in *EWSN*, ser. Lecture Notes in Computer Science, H. Karl, A. Willig, and A. Wolisz, Eds., vol. 2920. Springer, 2004, pp. 77–93.

[15] H. Chen, T. W. Finin, and A. Joshi, "Semantic web in the context broker architecture." in *PerCom*. IEEE Computer Society, 2004, pp. 277–286.

[16] T. Gu, H. K. Pung, D. Q. Zhang, and X. H. Wang, "A middleware for building context-aware mobile services," in *In Proceedings of IEEE Vehicular Technology Conference (VTC*, 2004.

[17] A. Ligęza and G. J. Nalepa, "A study of methodological issues in design and development of rule-based systems: proposal of a new approach," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 2, pp. 117–137, 2011.

[18] G. J. Nalepa, "Architecture of the HeaRT hybrid rule engine," in *Artificial Intelligence and Soft Computing: 10th International Conference, ICAISC 2010: Zakopane, Poland, June 13–17, 2010, Pt. II*, ser. Lecture Notes in Artificial Intelligence, L. Rutkowski and [et al.], Eds., vol. 6114. Springer, 2010, pp. 598–605.

[19] G. J. Nalepa, A. Ligęza, and K. Kaczor, "Formalization and modeling of rules using the XTT2 method," *International Journal on Artificial Intelligence Tools*, vol. 20, no. 6, pp. 1107–1125, 2011.

[20] G. Nalepa, S. Bobek, A. Ligęza, and K. Kaczor, "Algorithms for rule inference in modularized rule bases," in *Rule-Based Reasoning, Programming, and Applications*, ser. Lecture Notes in Computer Science, N. Bassiliades, G. Governatori, and A. Paschke, Eds., vol. 6826. Springer Berlin / Heidelberg, 2011, pp. 305–312.

[21] G. Nalepa, S. Bobek, A. Ligęza, and K. Kaczor, "HalVA – rule analysis framework for XTT2 rules," in *Rule-Based Reasoning, Programming, and Applications*, ser. Lecture Notes in Computer Science, N. Bassiliades, G. Governatori, and A. Paschke, Eds., vol. 6826. Springer Berlin / Heidelberg, 2011, pp. 337–344. [Online]. Available: http://www.springerlink.com/content/c276374nh9682jm6/

[22] S. Bobek, G. J. Nalepa, and W. T. Adrian, "Mobile context-based framework for monitoring threats in urban environment," in *Multimedia Communications, Services and Security: 6th International Conference, MCSS 2013: Kraków, Poland. June 6-7, 2013. Proceedings*, 2013.

[23] C. S. Raghavendra, K. M. Sivalingam, and T. Znati, *Wireless sensor networks*. Springer, 2006.