Mobile Context-Based Framework for Monitoring Threats in Urban Environment*

Szymon Bobek, Grzegorz J. Nalepa, and Weronika T. Adrian

AGH University of Science and Technology
al. Mickiewicza 30, 30-059 Krakow, Poland
{szymon.bobek,gjn,wta}@agh.edu.pl

Abstract. With a rapid evolution of mobile devices, the idea of context awareness has gained a remarkable popularity in recent years. Modern smartphones and tablets are equipped with a variety of sensors including accelerometers, gyroscopes, pressure gauges, light and GPS sensors. Additionally, the devices become computationally powerful which allows real-time processing of data gathered by their sensors. Universal access to the Internet via WiFi hot-spots and GSM network makes mobile devices perfect platforms for ubiquitous computing. Although there exist numerous frameworks for context-aware systems, they are usually dedicated to static, centralized, client-server architectures. There is still space for research in a field of context modeling and reasoning for mobile devices. In this paper, we propose a lightweight context-aware framework for mobile devices that uses data gathered by mobile device sensors and perform online reasoning about possible threats, based on the information provided by the Social Threat Monitor system developed in the INDECT project.

Keywords: context-awareness, mobile computing, GIS, knowledge management, INDECT.

1 Introduction

Distributed reporting and notification systems for citizen security have become common and widely expected and adopted in recent years. Within the scope of the INDECT[1] project such solutions are being developed and evaluated. Principal objectives of the project include engaging citizens into active participation in the authorities efforts to provide instant notification for a number of security threats in a given neighborhood or a wider location. The threats can be considered in different categories, such as crime, natural disasters, accidents or traffic related events.

A system called *Social Threat Monitor* (STM) that meets the above mentioned needs was developed. The STM is a GIS-based solution that assists citizens in

* The research presented in this paper is carried out within the EU FP7 INDECT Project: "Intelligent information system supporting observation, searching and detection for security of citizens in urban environment" (http://indect-project.eu).

[1] See http://indect-project.eu

A. Dziech and A. Czyżewski (Eds.): MCSS 2013, CCIS 368, pp. 25–35, 2013.

reporting security threats together with their severity and location. The threats are classified using a general top-level ontology, with domain ontologies supporting the detailed specification of threats. The information about the threats is stored in a knowledge base of the system which allows lightweight reasoning with the gathered facts. All the threats can be located on a web-accessible map that can be analyzed by a group of users, e.g., police officials, regular citizens, etc.

The current version of the system is a web-based solution, composed of a server-side GIS-based service providing access to the knowledge base and a web client. Therefore, a standard-compliant web browser is expected to be used as the main user interface. Another method for interfacing with the system on the application level is provided by a dedicated API that allows posing queries and making updates of the knowledge base.

An apparent limitation of the current system is related to the use of mobile devices on the client side. In the first generation of the system, an implicit assumption was made, that the user has a standard web browser available. Moreover, this browser should be used with a standard (for regular desktop and laptop computers) point-and-click user interface. However, currently the most common use case scenario includes the use of a mobile handheld device, such as a smartphone or a tablet, with a number of multimodal interfaces and sensors. Therefore, a need for a new front-end for the system became apparent. The principal objective of this paper is to propose and discuss a design of a prototype of such a system. It uses the context-aware application paradigm that improves usability from the user perspective, and simplifies the use of multi sensor data available on the mobile devices.

The rest of the paper is organized as follows. In Section 2, issues of context modeling and reasoning in mobile devices are discussed. This leads to presenting the motivation of the development of a new STM front-end in Section 3. The architecture of the system is presented in Section 4. A practical use case scenario is briefly introduced in Section 6. Finally, summary and directions for future work are given in Section 7.

2 Context in Mobile Computing

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 way. Dey [10] defines context as *any information that can be used to characterize the situation of an entity*, where *an entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves.*

The *information* in Dey's definition may be:

- location of the user (spatial context),
- presence or absence of other devices and users (social context),
- time (temporal context),
- user *behavior* or *activity* (activity recognition, behavior modeling), and
- other environmental data gathered by microphones, light sensors, etc.

Raw information captured by the device sensors is usually useless without further preprocessing and interpretation. Thus, the main challenges in context-aware systems are context modeling and context-based reasoning. There have been done a lot of research regarding context modeling. Various methods of knowledge representation were used, e.g., rules and logic [8,19,25], ontologies [7,30], object-oriented languages (CML, ORM) [12], context lattices [32] or processes [14].

In the area of context-based reasoning, the following approaches were developed: machine learning and probabilistic inference [5,31], decision trees [18], rule-based and logic-based reasoning [25,19,8]. Although there are a lot of frameworks and middlewares developed for context-aware systems, they are usually limited to a specific domain and designed without taking into consideration mobile platforms. Examples include CoBrA [6] and SOUPA [7] for building smart meeting rooms, GAIA [26] for active spaces, Context Toolkit [8], etc.

There is still space for research in a field of lightweight context modeling and context reasoning targeted at mobile devices. Some attempts were made to develop such frameworks, like SOCAM [11], or Context Torrent [13]. However, these frameworks do not provide full support for all of the challenges that we believe are crucial for mobile computing, with respect to the context modeling and context-based reasoning:

1. **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.
2. **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.
3. **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.

All of these require from the modeling language and inference engine to be simple and lightweight. Aforementioned challenges were usually approached by the programmers at the very last phase of the development of context-aware application, or were not approached at all. We believe that solutions to these challenges should be provided by the framework architecture. This will oblige the programmer to build context-aware application in an efficient way, making the development easier and less error prone.

3 Motivation

The primary objective of research that lead to the development of STM was to build a semantically enriched environment for collaborative knowledge management. Using it, local communities should be able to share information about road traffic dangers and threats, i.e. closed roads, holes in the pavements and streets, dangerous districts or events that impede a normal traffic. STM was aimed to be a community portal that allows citizens to participate and cooperate in order to improve the security in the urban environment. Within the task several system prototypes have been developed [2,3,29]. The subsequent versions of the STM system have used intelligent processing to provide possibly most useful knowledge to the users. Categorization of threats and possibility of inferring new facts based on the ones entered by users was introduced.

The latest development of the system [16] included a redesign that emphasizes the social aspects of the system. It defined the application as a platform to exchange information about dangers and their location among citizens and with public services. The objective of the application is to help civilians and public services to improve safety in urban areas and detect dangers faster. Users are able to browse threats in selected area and submit information about a threat by adding comments and photos. Moreover, public services are able to inform people about dangers and to monitor threats and decide if an intervention is needed. The main limitation remained the user interface (UI) which should be intuitive and easy to use, potentially adaptable to various hardware platforms including desktop and mobiles. In fact, so far the system was not designed to use mobile devices on the client side. Nevertheless, currently the most common use case scenario includes the use of a mobile handheld device, such as a smartphone or tablet. Such a device has a number of multimodal interfaces and sensors, e.g. gesture based interface, GPS, etc. Therefore, a need for a new front-end for STM became apparent.

The idea comes down to propose a design of a prototype of such a front-end. It is based on the the context-aware application paradigm that improves usability from the user perspective. Moreover, it simplifies the user of multi sensor data available on the mobile devices. The principal idea is presented in Figure 1.

Being in a given situation, and location (i.e., context) the user can be automatically notified by their mobile device about the threats relevant to him and the situation. Relevance to the person may be related to their role defined in the STM system, as well as the context, e.g. a person who walks should not be bothered by warnings relevant only to drivers in the same location. The use of data fusion from the sensors and multimodal interfaces of the mobile device allows to limit the amount of data the user would have to provide to the system. In fact, we propose a major paradigm shift on the front-end side. Whereas the original interface of STM was mostly query-based, here we propose a push-based UI where the user is automatically notified only about the information relevant to him. The system automatically uses the context data from the mobile device, as well as the data acquired from the STM server to perform reasoning for the user. In the following section, the architecture of the system is discussed.

4 Context-Based STM Front-End Architecture

The proposed system is based on a service-oriented architecture (see Figure 2). It consists of three main elements:

1. sensors service – responsible for gathering data from sensors and performing initial preprocessing of them,

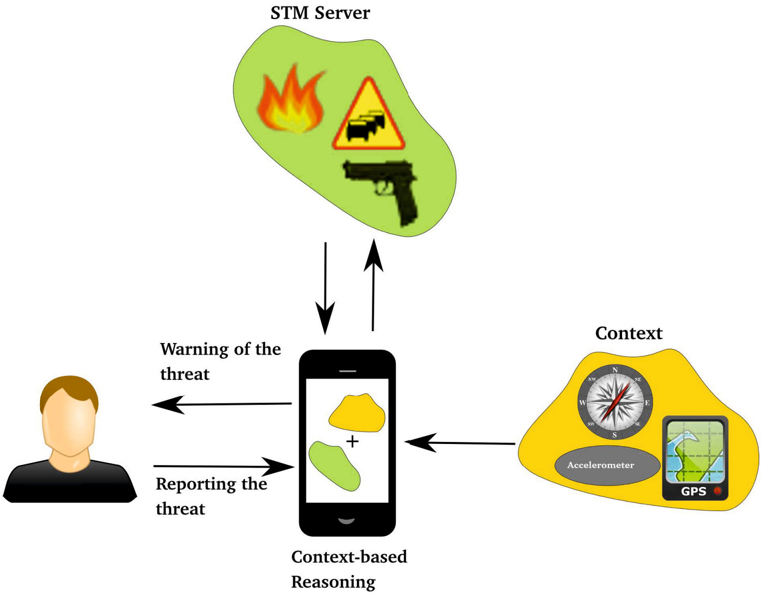

Fig. 1. Context-based front-end for Social Threat Monitor

2. inference service – responsible for context based reasoning and knowledge management. It provides TCP/IP API for context-aware applications,
3. working memory middleware – acting as an intelligent proxy between sensors service and the inference service.

Sensors Service
The Sensor Service gathers data directly form mobile device sensors. Due to the different possible sensor types (GPS, Accelerometer, Bluetooth), different methods for interpreting 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 Working Memory Middleware.

Inference Service
The inference service is responsible for performing reasoning, based on the model (knowledge base) and the working memory elements (facts). The service is capable of managing many models transparently switching between them. The reasoning task is performed by HeaRT [4,1]. It is a lightweight rule-based inference engine that uses XTT2 [24] notation for knowledge representation. It is written in Prolog and can be installed on mobile device together with tuProlog[2] interpreter, providing autonomous inference service. Moreover, the HeaRT inference engine, in contrary to other rule-based inference engines, provides custom

[2] See `http://alice.unibo.it/xwiki/bin/view/Main/`

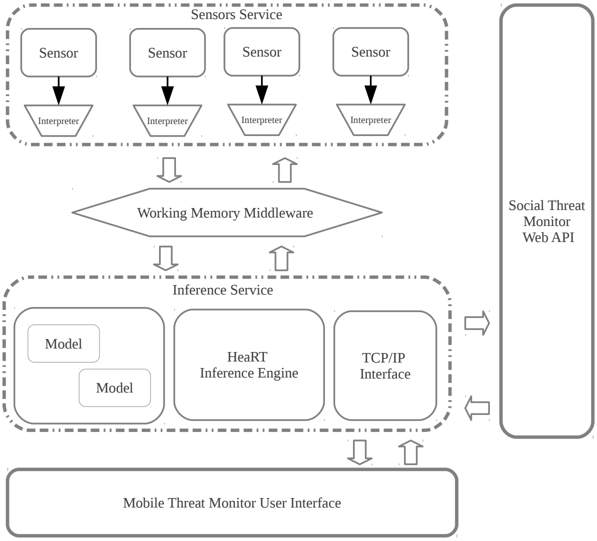

Fig. 2. Architecture of the mobile context aware framework for Social Threat Monitor system

verification module that can be used for automatic optimization of knowledge base (see Section 7 for details).

The inference service provides a TCP/IP interface for context-aware applications that may query HeaRT for important information. An exemplary query may concern listing all possible threats. This will require the inference engine to determine a context of the user (decide if the user is a driver, a cyclist, a pedestrian, decide where the user is, or where he or she will be in a nearest future). Based on this facts and on the data pulled from Social Threat Monitor system, the inference service will return the list of all the threats relevant for the user.

Working Memory Middleware

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

5 Context-Based Knowledge Management

The context aware framework presented in this paper uses XTT2 notation for knowledge representation. XTT2 [24] is a visual knowledge representation method for rule-based systems [17,23] where rules are stored in tables connected with each other creating a graph (see Figure 3).

The XTT2 has a textual representation called HMR. An example of a rule written in HMR language is presented below. The rule is referenced in Figure 3, in table *Today*.

```
xrule Today/1: [day in [sat,sun]] ==>[today set weekend].
```

The HMR representation is used by the HeaRT inference engine, which provides several inference modes, including:

Data-Driven which can be used to find all possible information that can be inferred from given data.

Goal-Driven which can be used to find only a specific information that can be inferred from a specific subset of XTT2 tables.

These inference modes allow the efficient reasoning in structured knowledge bases, like XTT2. Only the tables that lead to desired solution are fired, and no rules are fired without purpose, making the inference process less resource-consuming. Detailed description of the inference algorithms for XTT2 rule bases, can be found in [20]. For the purpuse of context processing a process-based description could also be used, see [15].

The HeaRT inference engine provide a callback mechanism that allows to query external sources for information. The external source could be: database, user, or in our case working memory middleware and Social Threat Monitor system. Callbacks are associated with attributes, defined as e.g.:

```
xattr [ name: day,
class: simple,
type: day,
comm: in,
callback: [ask_working_memory,[day]]
  ].
```

The *comm* element in the attribute definition determines behavior of a callback. There are three different types of such behavior:

comm: in – the callback is obliged to pull the value of the attribute from external source,

comm: out – the callback is obliged to push the value of the attribute to external source,

comm: inter – the callback should not assert nor retract any information from external source.

More about callback mechanism can be found in [22].

6 Use Case Scenario

An exemplary XTT2 model (see Figure 3) presented in this section allows to alert users about threats in a context-aware way. The system takes into consideration spatial (localization of the user) and temporal (time of a day) contexts, as well

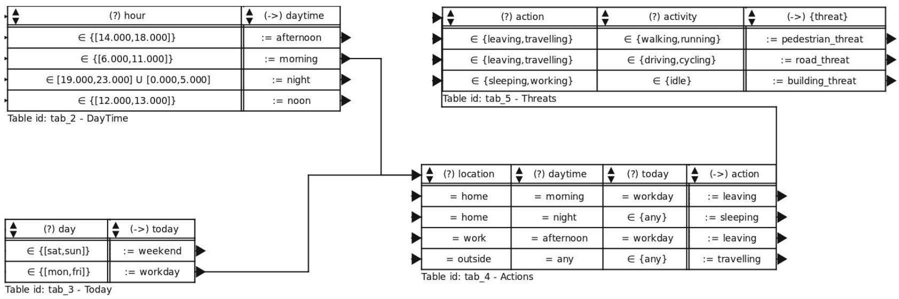

Fig. 3. Example of the model for a mobile threat monitor

as user activities. This allow the intelligent threats filtering. For instance, the model will prevent from warning a user who is driving a car about threats that are applicable to pedestrians only. This is achieved by selecting only these rules that are valid in current context.

Information about threats is fetched from Social Threat Monitor system via callbacks using the WEB API (see [2] for details). Information about user localization, time of a day, and user activities is pulled from a working memory middleware via callback mechanism. The working memory middleware obtains this information from sensors interpreters (for example: location from GPS sensor interpreter, activity like walking or running from accelerometer sensor interpreter, etc.).

Taking into consideration an example from Figure 3, and assuming that it is Monday, 8 o'clock am, and the user is driving a car, the system will process following rules: (1) rule 1 from *DayTime* table, (2) rule 2 from *Today* table, (3) rule 4 from *Actions* table, (4) and rule 2 from *Threats* table.

This inference chain will trigger several callbacks, including one that is assigned to `road_threats` in the *Threats* table. The callback will fetch all road threats from Social Threat Monitor system that are located near the user and assign it to `road_threats` attribute.

The application that implements Mobile Threat Monitor interface will be able to pull all information about threats from inference service via TCP/IP API and display it to the user.

7 Summary and Future Work

In this paper, we presented a mobile context-aware framework that uses data gathered by mobile device sensors and perform on-line reasoning about possible threats, based on the information provided by the Social Threat Monitor system. We argue that our solution addresses main challenges in mobile context-aware computing, that includes (1) *energy efficiency*, (2) *privacy*, and (3) *resource limitations*. The framework is designed in a service oriented architecture that includes:

1. *inference service*, that uses HeaRT inference engine to provide on-line efficient reasoning, preserving the (2) and (3);

2. *working memory middleware*, that works as knowledge cache minimizing the number of required requests to the sensors service, improving power efficiency of the entire system, preserving (1) and (2);
3. *sensor service*, that is responsible for gathering and initial preprocessing of the raw sensor data.

Future work includes more improvements towards power efficiency, and automatic optimization of knowledge bases, as well as intelligibility and usability of the system. The framework presented in this paper may be extended for additional functionalities, including:

Automatic threat reporting – The application could report anomalies in user behavior to the Social Threat Monitor system. When similar anomaly will be reported by many users, an alert will be raised. For instance, if a number of users start running from a building, there is probably a fire.

Learning user usage habits – The working memory middleware could learn user usage habits, to optimize sensors sampling. For instance, if there is no point in sampling accelerometer sensor at night with a high frequency, where there is high probability that the user will not move.

Intelligibility – Rule based system have a high capabilities of self-explaining their decisions. According to Dey [9], this is crucial factor of the system usability. The framework could provide mechanisms that will allow explaining its decision and asking user for corrections.

Automatic model optimization – HeaRT inference engine provides a verification plug-in that allows detecting anomalies such as: rules subsumption, redundancy and contradiction. Hence, a mechanism that will perform automatic optimization of an existing XTT2 model could be implemented [21].

There is also a possibility of a formal verification of systems developed with the presented approach. For example, a Petri net model for a context-aware application can be design. Some possibilities of including rule-based systems into Petri net models have been studied in [28] and [27]. It is possible to adopt these solutions to our needs.

References

1. Adrian, W.T., Bobek, S., Nalepa, G.J., Kaczor, K., Kluza, K.: How to reason by HeaRT in a semantic knowledge-based wiki. In: Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2011. pp. 438–441. Boca Raton, Florida, USA (November 2011)
2. Adrian, W.T., Ciężkowski, P., Kaczor, K., Ligęza, A., Nalepa, G.J.: Web-based knowledge acquisition and management system supporting collaboration for improving safety in urban environment. In: Dziech, A., Czyżewski, A. (eds.) MCSS 2012. CCIS, vol. 287, pp. 1–12. Springer, Heidelberg (2012), `http://link.springer.com/book/10.1007/978-3-642-30721-8/page/1`

3. Adrian, W.T., Waliszko, J., Ligęza, A., Nalepa, G.J., Kaczor, K.: Description logic reasoning in an ontology-based system for citizen safety in urban environment. In: Mercier-Laurent, E., et al. (eds.) AI4KM 2012: 1st International Workshop on Artificial Intelligence for Knowledge Management at the Biennial European Conference on Artificial Intelligence (ECAI 2012), Montpellier, France, August 28, pp. 63–67 (2012)
4. Bobek, S.: Heart rule inference engine in intelligent systems. PAR Pomiary Automatyka Robotyka 15(12), 226–228 (2011) ISSN 1427-9126
5. Bui, H.H., Venkatesh, S., West, G.A.W.: Tracking and surveillance in wide-area spatial environments using the abstract hidden markov model. IJPRAI 15(1), 177–195 (2001), http://dblp.uni-trier.de/db/journals/ijprai/ijprai15.html#BuiVW01
6. Chen, H., Finin, T.W., Joshi, A.: Semantic web in the context broker architecture. In: PerCom, pp. 277–286. IEEE Computer Society (2004)
7. Chen, H., Perich, F., Finin, T.W., Joshi, A.: Soupa: Standard ontology for ubiquitous and pervasive applications. In: 1st Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2004), Networking and Services, Cambridge, MA, USA, August 22-25, pp. 258–267. IEEE Computer Society (2004)
8. Dey, A.K.: Understanding and using context. Personal Ubiquitous Comput. 5(1), 4–7 (2001)
9. Dey, A.K.: Modeling and intelligibility in ambient environments. J. Ambient Intell. Smart Environ. 1(1), 57–62 (2009)
10. Dey, A.K.: Providing architectural support for building context-aware applications. Ph.D. thesis, Atlanta, GA, USA (2000), aAI9994400
11. Gu, T., Pung, H.K., Zhang, D.Q., Wang, X.H.: A middleware for building context-aware mobile services. In: Proceedings of IEEE Vehicular Technology Conference (VTC) (2004)
12. Henricksen, K., Indulska, J., Rakotonirainy, A.: Modeling context information in pervasive computing systems. In: Mattern, F., Naghshineh, M. (eds.) PERVASIVE 2002. LNCS, vol. 2414, pp. 167–180. Springer, Heidelberg (2002), http://dl.acm.org/citation.cfm?id=646867.706693
13. Hu, D.H., Dong, F., li Wang, C.: A semantic context management framework on mobile device âĹŮ
14. Jaroucheh, Z., Liu, X., Smith, S.: Recognize contextual situation in pervasive environments using process mining techniques. J. Ambient Intelligence and Humanized Computing 2(1), 53–69 (2011)
15. Kluza, K., Maślanka, T., Nalepa, G.J., Ligęza, A.: Proposal of representing BPMN diagrams with XTT2-based business rules. In: Brazier, F.M.T., Nieuwenhuis, K., Pavlin, G., Warnier, M., Badica, C. (eds.) Intelligent Distributed Computing V. SCI, vol. 382, pp. 243–248. Springer, Heidelberg (2011), http://www.springerlink.com/content/d44n334p05772263/
16. Ligęza, A., Adrian, W.T., Ciężkowski, P.: Towards collaborative knowledge engineering for improving local safety in urban environment. In: Canadas, J., Nalepa, G.J., Baumeister, J. (eds.) 8th Workshop on Knowledge Engineering and Software Engineering (KESE 2012) at the at the biennial European Conference on Artificial Intelligence (ECAI 2012), Montpellier, France, August 28, pp. 58–61 (2012), http://ceur-ws.org/Vol-949/
17. Ligęza, A., Nalepa, G.J.: 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 1(2), 117–137 (2011)

18. Logan, B., Healey, J., Philipose, M., Tapia, E.M., Intille, S.S.: A long-term evaluation of sensing modalities for activity recognition. In: Krumm, J., Abowd, G.D., Seneviratne, A., Strang, T. (eds.) UbiComp 2007. LNCS, vol. 4717, pp. 483–500. Springer, Heidelberg (2007), http://dl.acm.org/citation.cfm?id=1771592.1771620
19. Loke, S.W.: Representing and reasoning with situations for context-aware pervasive computing: a logic programming perspective. Knowl. Eng. Rev. 19(3), 213–233 (2004), http://dx.doi.org/10.1017/S0269888905000263
20. Nalepa, G.J., Bobek, S., Ligęza, A., Kaczor, K.: Algorithms for rule inference in modularized rule bases. In: Bassiliades, N., Governatori, G., Paschke, A. (eds.) RuleML 2011 - Europe. LNCS, vol. 6826, pp. 305–312. Springer, Heidelberg (2011)
21. Nalepa, G.J., Bobek, S., Ligęza, A., Kaczor, K.: HalVA - rule analysis framework for XTT2 rules. In: Bassiliades, N., Governatori, G., Paschke, A. (eds.) RuleML 2011 - Europe. LNCS, vol. 6826, pp. 337–344. Springer, Heidelberg (2011), http://www.springerlink.com/content/c276374nh9682jm6/
22. Nalepa, G.J.: Architecture of the HeaRT hybrid rule engine. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part II. LNCS, vol. 6114, pp. 598–605. Springer, Heidelberg (2010)
23. Nalepa, G.J., Kluza, K.: UML representation for rule-based application models with XTT2-based business rules. International Journal of Software Engineering and Knowledge Engineering (IJSEKE) 22(4), 485–524 (2012), http://www.worldscientific.com/doi/abs/10.1142/S021819401250012X
24. Nalepa, G.J., Ligęza, A., Kaczor, K.: Formalization and modeling of rules using the XTT2 method. International Journal on Artificial Intelligence Tools 20(6), 1107–1125 (2011)
25. Ranganathan, A., Campbell, R.H.: An infrastructure for context-awareness based on first order logic. Personal Ubiquitous Comput. 7(6), 353–364 (2003), http://dx.doi.org/10.1007/s00779-003-0251-x
26. Ranganathan, A., McGrath, R.E., Campbell, R.H., Mickunas, M.D.: Use of ontologies in a pervasive computing environment. Knowl. Eng. Rev. 18(3), 209–220 (2003)
27. Szpyrka, M.: Design and analysis of rule-based systems with adder designer. In: Cotta, C., Reich, S., Schaefer, R., Ligęza, A. (eds.) Knowledge-Driven Computing: knowledge engineering and intelligent computations. SCI, vol. 102, pp. 255–271. Springer, Heidelberg (2008)
28. Szpyrka, M., Szmuc, T.: Decision tables in petri net models. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 648–657. Springer, Heidelberg (2007)
29. Waliszko, J., Adrian, W.T., Ligęza, A.: Traffic danger ontology for citizen safety web system. In: Dziech, A., Czyżewski, A. (eds.) MCSS 2011. CCIS, vol. 149, pp. 165–173. Springer, Heidelberg (2011)
30. Wang, X., Zhang, D., Gu, T., Pung, H.K.: Ontology based context modeling and reasoning using owl. In: PerCom Workshops, pp. 18–22 (2004)
31. Ye, J., Coyle, L., Dobson, S.A., Nixon, P.: Representing and manipulating situation hierarchies using situation lattices. Revue d Intelligence Artificielle 22(5), 647–667 (2008)
32. Ye, J., Dobson, S.: Exploring semantics in activity recognition using context lattices. J. Ambient Intell. Smart Environ. 2(4), 389–407 (2010)