Engineering Applications of Artificial Intelligence 43 (2015) 111–126

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Context-aware knowledge-based middleware for selective information delivery in data-intensive monitoring systems

Yulia Evchina*, Juha Puttonen, Aleksandra Dvoryanchikova, José Luis Martinez Lastra

Tampere University of Technology, FAST Laboratory, Korkeakoulunkatu 5L, FIN-33720 Tampere, Finland

ARTICLE INFO

Article history:
Received 16 January 2015
Received in revised form
10 April 2015
Accepted 13 April 2015

Keywords:
Context-aware systems
Information filtering
Semantic Web technologies
Ontologies
Building automation

ABSTRACT

Multiple embedded devices in modern control and monitoring systems are able to sense different aspects of the current context such as environmental conditions, current processes in the system and user state. The number of captured situations in the environment and quantity and variety of devices in the system produce considerable amounts of data, which should be processed, understood and followed by corresponding actions. However, fully delivered to the user regardless of their role in the system and needs, data flows cause cognitive overload and thus may compromise the safety of the system depending on the timely response of the operators. This paper addresses the problem of selective information delivery with respect to the user's role in the system, his needs and responsibilities, by proposing context-aware information management middleware. The system utilizes Semantic Web technologies by capturing relevant information in the knowledge model of the system, which decouples data from the application logics. A clear division of data and application logics enables context-awareness and facilitates the reconfiguration process, when new information should be added into the system. The chosen approach is justified with an analysis of main trends in context-aware solutions. The engineering principles of the knowledge model are described and illustrated with simple scenarios from the building automation domain. The prototype developed proves the feasibility of the approach via performance evaluation and demonstrates the reconfiguration capabilities of information flows in the system. Further work assumes the extension of the knowledge model and integration of the system with adaptive human–machine interfaces for multi-role and multi-user environments.

1. Introduction

The evolution of control and monitoring systems has enabled information accessibility for almost any aspect of the environment (Sauter et al., 2011). In modern systems it is possible to integrate large numbers of various sensors to capture environmental conditions (e.g. temperature, humidity, and light), particular features of the process of interest (e.g. the product status in the production line) and the state of the users involved in the process or system operation. Potentially, it could offer advantages for timely situational awareness of the all system users. In reality growing information complexity challenges the usability of the system, human performance, and safety overwhelming the cognitive capacities of a user. This paper addresses a problem of selective information delivery to different users of the system by proposing context-aware information management middleware (IMM), which is able to analyze the run-time data of the system and deliver the information which is relevant to the user given the current situation and user needs/responsibilities.

The aim is to bring the right information to the right user at the right time, thus information management middleware should be able to determine the "rightness" according to the situation. The introduction of context-awareness is claimed to enable the desired flexibility of system behavior (Baumgartner et al., 2010; Fischer, 2012). The context can be defined as "*any information that can be used to characterize the situation of an entity, where an entity can be a person, place, or physical or computational object*" (Abowd and Dey, 1999), and then the system is context-aware "*if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task*" (Dey, 2001). Both definitions are quite generic and allow different technological solutions for context-aware systems providing the right services (Emmanouilidis et al., 2013), actions (Feng et al., 2009) or system adaptation.

This paper proposes context-aware information management middleware for data intensive monitoring systems, where user is exposed to numerous sources of data coming from heterogeneous devices of the system and various situations of the environment.

* Corresponding author. Tel.: +358 408490281.
E-mail addresses: yulia.evchina@tut.fi (Y. Evchina), juha.puttonen@tut.fi (J. Puttonen), jose.lastra@tut.fi (J.L.M. Lastra).

http://dx.doi.org/10.1016/j.engappai.2015.04.008
0952-1976/

The middleware provides the user with the relevant information depending on the current situation, user role in the system, his needs and responsibilities. The main technological trends of achieving context-awareness in the systems are presented and analyzed, and the chosen approach using Semantic Web technologies (specifically OWL ontologies, SPARQL and SPARQL/Update[1] queries) is justified. The system utilizes a reconfigurable knowledge model (ontologies) which decouples data from the application logic. The clear division of data and application logic enables context-awareness and (1) facilitates the system reconfiguration process when new information is introduced into the system; (2) enables the implementation of domain independent tools which facilitate the process of adapting IMM in new domain. The proposed approach is supported by ontological engineering principles, a description of a prototype already implemented, and describing reference cases from the building automation domain.

The rest of the paper is organized as follows. Section 2 overviews the state of the art for context-aware systems and justifies the choice of Semantic Web technologies for the solution proposed. Section 3 describes the functional architecture of the system and the enabling technologies for the main architectural blocks. Section 4 states the main design principles of the knowledge model of the system, the query templates enabling reconfiguration and the domain independent capabilities of the solution. Section 5 describes the tools for implementing IMM in detail and is followed by section 6 with simple examples of scenarios from the building automation domain exemplifying the knowledge model and demonstrating in function the main principles of the proposed solution. Section 7 presents performance evaluation of developed prototype; and finally, Section 8 concludes the paper and highlights directions for future work.

2. State of the art

Context-aware systems are a fairly active research topic resulting in numerous solutions and application areas. In order to limit the scope during the literature analysis, the focus was placed on the solutions with use cases similar to data intensive monitoring systems (i.e. with active and numerous incoming data flows) and selective reaction to the present context (i.e. delivery of personalized services or performing particular actions rather than activity recognition). Thus, other features of context-aware systems, such as learning mechanisms and analyzing of historic data are beyond the scope of the paper. On the basis of these criteria three main trends in technological solutions of context-aware systems were identified: Complex Event Processing (CEP), Semantic Web technologies, and Linked Stream Data. This section analyzes the advantages and disadvantages of the approaches in depth, and justifies the choice of the Semantic Web approach for the development of the context-aware information management middleware.

2.1. CEP technologies

CEP is a method for analyzing information flows (events) and matching them with predefined patterns (rules). Context-aware systems with CEP capture relevant information in the rules, that is, they model the context via the rule condition part and provide context-sensitive reactions via the rule action part (Chung et al., 2013; Ottenwälder et al., 2014). Usually the matching is realized via standalone engines, which are configured via the engine specific rules and perform run-time event analysis (Gao and Bhiri, 2012). The existing engines are quite numerous, and many are freely available on the web (e.g. Drools Fusion,[2] Esper,[3] Storm,[4] Oracle CEP,[5] SASE,[6] etc.) and could be easily adapted to applications via provided API. The engines provide scalable infrastructure for analyzing streams of different events, and this ability helps to decouple the application logic from the business rules. The CEP context-aware applications are much used, especially in "sense and respond" systems (Zappia et al., 2012). For example, Drools was applied to understand the current context on the basis of readings from RFID marked devices in a hospital in (Yao et al., 2011). The data from the sensors were analyzed by the CEP engine to identify critical situations in health monitoring. CEP engines enable the collecting of the row and hard interpretable sensors data with RFID tags, and the interpretation and extraction of meaningful events pertaining to the patient. In automotive domain the CEP engine was used for data fusion from different sensors in a car (Terroso-Saenz et al., 2015). The approach made it possible to interpret the context of the route, driver, and travel conditions corresponding to the traveler's needs.

The CEP approach has several drawbacks. For example, the large number of rules in the applications requires supplementary rule consistency check (Weiss et al., 2011). The ontologies were used for consistency checking of rule databases (Gellrich et al., 2012), however the approach is still under investigation and the reported prototype was not yet clearly exemplified. Another problem is in the software interoperability, which impedes the reconfiguration process as far as the rules for CEP engines can be expressed in different languages (Eckert et al., 2011). A light-weight solution to CEP was applied to minimize the dependencies on external libraries and applications in order to focus on ease-of-use, extensibility and scalability; nevertheless the solution still uses engine-specific language for rule expressions (Zappia et al., 2012). There have been attempts to produce a description of CEP rules using ontologies. For example, CEP was enriched with an ontological description of events (Taylor and Leidinger, 2011), however the model does not consider other context information such as places and users and it is not clear how the system will react to changes. To summarize, the CEP approach is a powerful way of analyzing intensive streams of data, but it still has its difficulties: CEP engines have different languages for rule expressions, which require additional effort in engine adoption and result in complex reconfiguration processes during the system adjustment, especially if the knowledge base of rules is large and difficult to understand.

2.2. Semantic Web technologies

The matching of complex event patterns with CEP in an event cloud is able to anticipate only the context defined in the rule. If a pattern is not described with a rule, it is beyond further consideration. In other words, with CEP it is not possible to derive implicit knowledge from the explicit statements. Another breed of applications attempts to bridge this gap by capturing the current context in the knowledge model of the system, which helps to reason over the present situation in the environment. There are various technologies supporting context modeling such as key-value, markup, graphical, object-oriented and ontological models (Hoareau and Satoh, 2009; Bolchini et al., 2007). Context modeling with ontologies is seen as the most expressive way in terms of ability to capture knowledge of the environment,

[1] SPARQL Update: http://www.w3.org/Submission/SPARQL-Update/.

[2] Drools Fusion: http://www.jboss.org/drools/drools-fusion.html.

[3] Esper: http://esper.codehaus.org/.

[4] Storm: http://storm-project.net/.

[5] Oracle CEP: http://docs.oracle.com/cd/E13157_01/wlevs/docs30/get_started/overview.html.

[6] Stream-based And Shared Event processing (SASE): http://sase.cs.umass.edu/.