Architecting Dynamic Situation Applications for Wireless Sensor Networks: an Active Rule-based Approach

Jinhui Chen[1, a]

[1]School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing, 210044, China

[a]cjh@nuist.edu.cn

Keywords: publish/subscribe; graph-oriented model; multimedia content

Abstract. This paper presents an approach, called DSECA, for modeling and implementing the architecture of dynamic situation applications for wireless sensor networks using an active rule-based model. The DSECA unites two of the application's aspects, namely the content aspect, and the process aspect, and provides means for the integration of situation events within the DSECA process model. A formal definition is proposed for the DSECA. Performance is evaluated. The results show that the DSECA can easily be constructed, while enabling an efficient management by the DSECA matching execution engine.

Introduction

Business environment architecture based on WSNs provides the foundation for open rich smart device data, while lowering the threshold for building intelligent and personalized services. However, the simple opening of these raw data can not well support intelligence services. We need to rely on a DS(DS) awareness technology. DS awareness is the perception of environmental elements with respect to time or space, the comprehension of their meaning, and the projection of their status after some variable has changed, such as time, or some other variable, such as a predetermined event. DS awareness technology is one of the major technologies for developing pervasive computing applications with flexibility, high adaptability, and self-management. DS applications for WSNs require technology support in modeling and reasoning DS. DS modeling effectively reduces the complexity of applications, while improving the sustainability and evolvability of applications. A well-designed reusable and sharable DS model help reduce the cost of acquisition, evaluation and maintenance of dynamic information.

Event-Condition-Active(ECA) rules is a short-cut for referring to the structure of active rules in event driven architecture and active database systems. Such a rule traditionally consisted of three parts: The event part specifies the signal that triggers the invocation of the rule; The condition part is a logical test that, if satisfied or evaluates to true, causes the action to be carried out; The action part consists of updates or invocations on the local data. The DS-oriented ECA rule(DSECA) is an active mechanism modeling method in the context aware technology on WSNs architecture.

A discussion on historical perspectives on cognitive engineering and situation awareness is included, along with future research needs for the construct. New models on sense-making, distributed situation awareness, and situated situation awareness are also discussed[1]. Z.Dörnyei(2014) summarizes some of the main challenges of dynamic systems research in general and then presents a concrete research template that can be applied to investigate instructed second language acquisition. This approach involves a special type of qualitative system modeling[2]. S.Hasan et al. (2011) discuss the synergy between information coming from semantic sensor networks together with existing information sources in enterprises to achieve high quality situational awareness to support decision making process, and also show how semantic sensor networks that respect linked data principles form a valid basis for dynamic and unified enrichment[3]. The dynamic response of temperature sensors within wireless and RFID nodes is dramatically influenced by the way they are housed as well as by the heat released by the node electronics itself; its characterization is basic to allow monitoring of high rate temperature changes and to certify the cold chain. Besides the time to rise and to recover is significantly different being mostly higher for the latter than for the former[4]. Activity recognition is a key component in identifying the context of a user for providing services based on the application. M. A. Awan et al. (2013) propose a context management model that is based on activity recognition. The model is composed of four components: a set of sensors, a set of activities, a backend server with machine learning algorithms, and a GUI application for the interaction with the user[5]. To achieve maximum resiliency against attacks, a dynamic intrusion detection protocol model is designed based on data flow for high error rate WSNs. By this protocol the attacks and compromised nodes can be effectively identified at runtime in high data rate static or dynamic WSNs. These models are only to some extent appropriate for architecting DS applications for WSNs[6].

In view of the above analysis, an approach, for modeling and implementing the architecture of DS applications for WSNs using the DSECA, is presented. The main characteristics of the DSECA include two aspects as follows: First, DSECA addresses the DS's content aspect. DSECA composes of the resources and the situation events, and describes dynamic relationships between situations. Second, DSECA addresses the DS's process aspect. DSECA provides a situation aware process model to adapt for the requirements of situation aware technology on WSNs computing environment.

A DS execution environment based on rule engine

ECA rules for DSs. All the things on WSNs can be abstracted into resources. Each resource corresponds to a unique resource identifier. Through generic connector interfaces, we carry out various operations on the resources. These operations don't change the resource identifiers. All resources are linked together to form a network of relationships. Situation event refers to an event behavior process which is triggered by the change of the context values under specific rules. DS refers to a collection of resources composed of the resources and the situation events. DS describes dynamic relationships between contexts. The relationships are established by the situation events. The fine-grained DSs can be aggregated into a greater granularity of DS by situation events.

DSECA uses ECA rules to describe the DS process model. Also DSECA does some modifications to the situation aware process model to adapt for the requirements of situation aware technology on WSNs computing environment. DSECA shall meet the following requirements. First, it can describe both the DS resources and the rule resources. Second, it can describe the relationships between the DSs and the DSECA rules. Based on the above requirements, DSECA defines two types of files, such as situation file and rule file. Situation file is used to represent the composition and state of DS, and rule file is used to represent association rules between the different DSs.

DS execution environment. We turn DSECA into a language which can be recognized by ECA rule engine, and propose a DS execution environment. The execution environment is a rule policy system, and includes four components: event bus, situation management, policy service and rule management. Fig.1 shows a DS execution environment based on ECA rule.

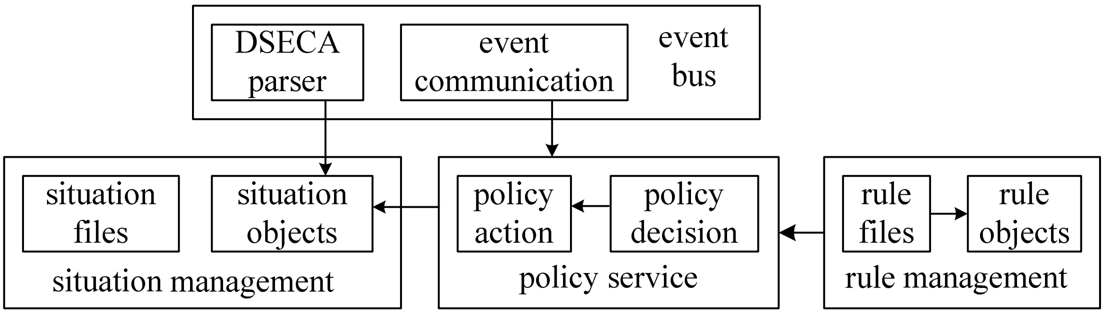

Fig.1. a DS execution environment based on ECA rule

According to the current event type, event bus monitors situational events generated by situation objects, and transmits the situational events to the policy service module to make a decision. DSECA parser analyzes situation files, and extract target situation files. Situation objects are the DS resources which are expressed by DSECA, they represent the managed policy objects. When mapped to WSNs applications, situation objects represent various sensor objects. Policy decision is responsible for parsing incoming events from the event bus, and matches the events with the policies in rule files. If the match is successful, the corresponding rule is activated. Policy action is responsible for executing the actions of matching rules to change the status of the target situation objects. Rule management is responsible for storing the rule details, including identification, event, condition and action. Rule files are parsed for structured storages.

Formal definitions of the DSECA

We learn from the automata concept to define resource net(RNet), and DS are formally defined by using RNet.

Definition 1 Resource net(RNet). A RNet is a five-tuple: RNet=(Q, SE, δ, Q_0, F), consisting of a finite set of resource states Q, a finite set of input situation events SE, a transition function δ: Q×SE→Q, a set of start resource state $Q_0 \subseteq Q$, and a set of accept resource states $F \subseteq Q$. Where δ represents unidirectional associations between resources, and includes sequence rules, selection rules and circulation rules, parallel rules. The associations are built by situation events.

In the paper, Let Res denote resources, RL the identifier and address of a situation, and RL is unique for every situation. Let NameS denotes the name of a situation, and NameS correlates directly with RL. Further, let CS be a situation which consists of a set of situations, and Ops be a set of operations over a situation.

Definition 2 Resource situation(RS). A RS is a five-tuple RS=(RL, NameS, CS, RNet, Ops), where RL, NameS and Ops are not empty, CS={Res} indicates that the type of situation is composed of the independent resources without situation event rules, and RNet=({Res},{}, {}, {Res}, {Res}).

In definition 2, RS is an empty RNet, and contains only one resource without any transition. So, RS's Q_0 and F are {Res}.

Definition 3 Atomic situation(AS). AS is composed of an unique situation event and a set of resources. A AS is a five-tuple RS=(RL, NameS, CS, RNet, Ops), where RL, NameS and Ops are not empty, CS={Res_1, Res_2, ..., Res_n} indicates that the type of situation is composed of several RSs. And RNet=(Q, SE_i, δ, Q_0, F), where Q={Res_1, Res_2, ..., Res_n}, δ:Q×SE_i→Q.

Definition 4 Composite situation(Cos). A composite situation is a resource situation, it can be composed of RS, AS or Cos. Cos is defined as (RL, NameS, CS, RNet, Ops), where RL, NameS, and Ops are the RL, NameS and Ops of the Cos, CS=$CS_1 \cup CS_2 \cup ... \cup CS_n$}, CS_i can be RS, AS or Cos. RNet=(Q, SE, δ, Q_0, F) are the results on the basis of sequence, selection, concurrent or circulation rules.

Implementation and evaluation

A distributed execution environment based on rule engine provides a basis of architecting DS applications for WSNs. For supporting platform independency and scalability, we have implemented the environment in Java and used the object-oriented DBMS MySQL for their persistent storage.

The infrastructure stores and processes the XML archives exported by the DSECA's situation files and rule files. The XML archives capture the three aspects of DSECA. The three aspects, namely the resource aspect, the situation event aspect, and the DS aspect, are reflected by the object 'resource', the object 'situation', and the object 'rule', respectively.

In order to evaluate the efficiency of the DSECA matching execution engine, we carried out some experiments to compare the costs of two selected operations, i.e. the cost of the rule resolution operation, the cost of the situation event matching operation. The rule resolution operation provides means for analyzing the semantic information of a DSECA rule. To any given DSECA condition, the situation event matching operation matches those situation events that satisfy the DSECA condition.

In all experiments, each has between 0 and 50 CSs and can contain up to 100 RSs or ASs. Fig. 2-3 depict response times for the two operations, where the columns ‘flat’, ‘normal’, and ‘deep’ indicate the maximum level of CS: in a flat hierarchy the maximum composite level is 1, in a normal hierarchy 2, and in a deep hierarchy 5.

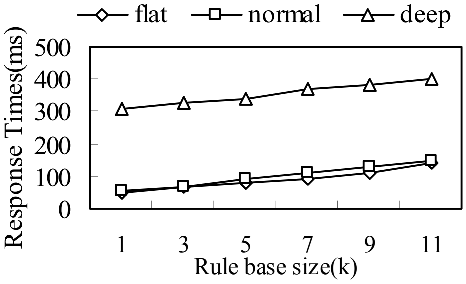

Fig.2 Response times of resolution operation

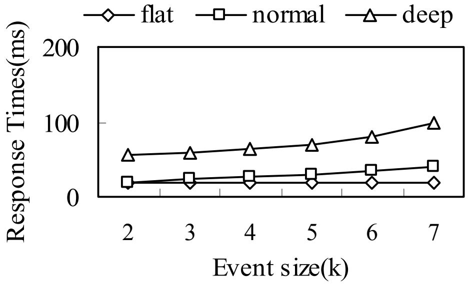

Fig.3 Response times of the matching operation

In Fig. 2, for the rule resolution operation type the response time grows at a more subdued pace. In Fig. 3, the response time of the situation event matching operation increases with the number of composite level contained in rules. The results show that the implementation of our prototype system is more efficient.

Conclusions

We propose DSECA for modeling and implementing the architecture of DS applications for WSNs. In our future work, we plan to establish a DSECA development environment by integrating the DSECA infrastructure, the DSECA rule matching execution environment, and the DSECA engineering tools to provide a profound basis for the seamless integration of DSECA rule to unleash its full potential for efficient awareness of WSNs content.

Acknowledgements

This work was sponsored by open fund project of Jiangsu provincial research and development center of intelligent sensor network engineering technology, China(ZK13-02-03, Software technology, platform and application for sensor network).

References

[1] C.D.Wickens, Situation Awareness Its Applications Value and Its Fuzzy Dichotomies, Journal of Cognitive Engineering and Decision Making. 9(1) (2015) 90-94.

[2] Z.Dörnyei, Researching complex dynamic systems: ‘Retrodictive qualitative modeling’ in the language classroom, Language Teaching. 47(01) (2014) 80-91.

[3] S.Hasan, E.Curry, M.Banduk, Toward Situation Awareness for the Semantic Sensor Web: Complex Event Processing with Dynamic Linked Data Enrichment, SSN. 839 (2011) 69-81.

[4] R.Badia-Melis, J.Garcia-Hierro, L.Ruiz-Garcia, Assessing the dynamic behavior of WSN motes and RFID semi-passive tags for temperature monitoring, Computers and Electronics in Agriculture. 103 (2014) 11-16.

[5] M.A.Awan, Z.Guangbin, S.D.Kim, A dynamic approach to recognize activities in WSN, International Journal of Distributed Sensor Networks. (2013).

[6] S.Janakiraman, S.Rajasoundaran, P.Narayanasamy, The Model—Dynamic and Flexible Intrusion Detection Protocol for high error rate Wireless Sensor Networks based on data flow, ICCCA2012, IEEE. (2012).