



Bloom filter-based workflow management to enable QoS guarantee in wireless sensor networks[☆]



Endong Tong^a, Wenjia Niu^{b,*}, Gang Li^{c,*}, Ding Tang^a, Liang Chang^d, Zhongzhi Shi^e, Song Ci^a

^a High-performance Network Laboratory, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, PR China

^b Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, PR China

^c School of Information Technology, Deakin University, 221 Burwood Highway, Vic 3125, Australia

^d Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, PR China

^e Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, PR China

ARTICLE INFO

Article history:

Received 15 July 2012

Received in revised form

2 March 2013

Accepted 3 April 2013

Available online 21 April 2013

Keywords:

Wireless sensor network

Hierarchical workflow management

QoS guarantee

Bloom filter

ABSTRACT

As a popular service composition technology, workflow has been successfully used in *wireless sensor networks* (WSNs) to compose a set of atomic services for service-oriented WSN applications. However, in a resource-constrained WSN, the sensed data is usually inaccurate or even missing, and this affects the normal execution of atomic services and may result in the non-guaranteed workflow QoS. Because the implementation of workflows in WSNs is usually hierarchical, effective workflow management in a WSN should consider both aspects of atomic services and sensor nodes, though this has largely been overlooked in existing research. Hence, a dynamic QoS-oriented, effective and efficient hierarchical workflow management mechanism is necessary. In this paper, we propose a *Bloom filter*-based hierarchical workflow management model, which coordinates both the *atomic service level* and the *node level* for guaranteed workflow QoS. Through constructing the service-level *counting Bloom filter* (CBF) to maintain the set of normal atomic services, and constructing the node-level *Bloom filter* (BF) to maintain the set of *attribute strings* of the current working nodes, an effective and efficient QoS degradation locating can be realized. Furthermore, the corresponding adaptation mechanism for guaranteed QoS is also developed. The case study and experimental evaluations demonstrate the capability of the proposed approach in WSNs.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

As the development of wireless technology, *wireless sensor networks* (WSNs) have been widely deployed in various applications. However, each WSN application usually has its own pre-designed implementation structure and this makes the interoperability among WSN applications difficult. Fortunately, *service-oriented architecture* (SOA) (Erl, 2005) provides the characteristic of supporting collaborations among distributed autonomous applications in an open dynamic environment. Hence, many attempts have been made to bring SOA into WSNs for effective realization of cooperative, interoperable and loose-coupled applications (Leguay et al., 2008; Gu et al., 2005; Delicato et al., 2003).

Due to the efficiency and the practicability, *workflow* has become a popular service composition technology, through which a sequence

of atomic services can be composed together to satisfy the specified functionality. However, practical SOA applications usually contain many atomic services with compatible functionalities. This provides the basis for the service composition to pay attention to the QoS requirements (Cardoso et al., 2004; Chen et al., 2011).

Take an example of *temperature control* application, in which the current temperature determines whether or not to turn the air conditioner on. If a user wants to achieve the *temperature control* functionality with two QoS requirements on $cost \leq 100$ and $deviation \leq 1^\circ C$ respectively, an abstract workflow can be built and composed of a set of abstract atomic services like

start → *data gathering* → *data reasoning* → *action execution* → *end*

Let us assume that two concrete atomic services exist to implement the *data reasoning* service:

- S_1 with QoS attribute definition ($cost = 60$, $deviation \leq 5^\circ C$) and
- S_2 with QoS attribute definition ($cost = 90$, $deviation \leq 0.5^\circ C$).

If S_1 is chosen, the user's expected functional can be achieved, though the *deviation* of S_1 ($5^\circ C$) is beyond the user's expectation ($1^\circ C$). This renders the composed workflow suboptimal for this user. Hence, a large amount of effort has been devoted to satisfy

[☆]This paper is an extended work for our preliminary work published in PRICAI 2012.

* Corresponding authors. Tel: +61 3 92517434; fax: +61 3 92517604.

E-mail addresses: Tonged@hpl.ac.cn (E. Tong), nwj6688@gmail.com (W. Niu), gang.li@deakin.edu.au, gangli@acm.org, gangli@gmail.com (G. Li), tangd@hpl.ac.cn (D. Tang), changl@guet.edu.cn (L. Chang), shizz@intsci.ac.cn (Z. Shi).

different QoS requirements simultaneously (Zeng et al., 2004; Yu et al., 2007).

However, due to factors such as the changes in environment, the QoS attributes of atomic services can change or even become unavailable (Rinderle et al., 2004) during the execution of corresponding workflows. Consequently, these changes may increase the risk of a composed service with poor QoS, which will further render the corresponding workflows incapable of meeting users' expected QoS. Accordingly, QoS-oriented dynamic workflow adaptation is necessary as this is a key issue in workflow management.

Workflow adaptation means that the workflow should adapt itself to the changing environment in order to satisfy both the functional and the QoS requirements. Methods for implementing the workflow adaptation can be divided into two categories: (Zeng et al., 2004; Ardagna and Pernici, 2007; Alrifai and Risse, 2009; Cardellini et al., 2009) *recomposing* and *replacing*. *Recomposing* reselects the sequence of appropriate atomic services to compose a new workflow with guaranteed QoS, and can achieve the *global* adaptation by considering the QoS of the whole workflow. While through *replacing*, abnormal atomic services will be replaced by those with the compatible functionalities. The *replacing* method belongs to the *local* adaptation, as it only considers the sanctification of the faulty service's functionality (e.g. *input* and *output*), while fails to guarantee the whole workflow QoS for lack of a global view.

Unlike traditional SOA, workflow management in a WSN is much more flexible. Due to the data-centric characteristic and the strong data gathering capability of a WSN, atomic services are inevitably redundant. Hence, the simple adoption of SOA into WSN may bring issues such as the *waste of application resources* or *inefficient service management*. In order to alleviate this problem, Tong et al. (2011) proposed a *reasoning-based context-aware workflow management model* (Recow) for WSNs. In this model, a rule-based reasoning module extracts semantic information so that the lower-level sensor data will have a loose-coupled connection with the upper-level logic process. By deploying these *semantic atomic services* whose inputs are characterized with semantic information, Recow can build a flexible workflow by taking advantage of its loose-coupled connection with the sensor level.

However, as the implementation of workflows in a WSN is usually hierarchical, faulty atomic services or sensor nodes will affect the correct execution of workflows. Effective workflow management in WSNs should consider both the aspects of atomic services and the aspects of sensor nodes. Accordingly, this calls for an effective and efficient hierarchical workflow management mechanism. More specifically, the following two challenges should be addressed:

- As WSNs are limited in energy and computation resources, the gathered sensor data can be inaccurate because of *energy exhaustion*, *device fault* or *environmental obstruction*. However, as WSN applications are data-centric, the inaccuracy in the lower-level sensed data will affect the correct execution of upper-level atomic services, thus further deteriorate the corresponding workflow QoS. This forms the bottom-to-up dynamical characteristic of WSN applications.
- In the hierarchical WSN workflow management, the lower level hides the implementation details for the upper level. While most WSN applications have a requirement of low latency, once the workflow QoS deteriorates to an unacceptable threshold, the hierarchical management should locate and fix the problem promptly.

Due to these challenges, an efficient management mechanism in high demand. As the Bloom filter is efficient in representing the huge number of elements, it can be utilized to improve the efficiency of workflow management in a WSN. More specifically,

hierarchical Bloom filters can be constructed to represent atomic services and sensor nodes. However, the effective interaction among different levels of Bloom filters remains an open and challenging issue, which is also the research objective of this paper. Hence, we will firstly exploit the Bloom filter at each level, and then through flexible interaction between the hierarchical *Bloom filters*, an efficient hierarchical *workflow management* in WSNs can be achieved. Accordingly, this paper will further focus on the following research issues:

- A *hierarchical workflow management model*. We will propose the hierarchical workflow management model, which coordinates both the atomic service level and the node level for guaranteed workflow QoS so that it is suitable for resource-constrained WSNs.
- *Bloom filter-based QoS-oriented workflow adaptation mechanism*. Firstly, dynamic frequency is utilized in the QoS monitoring for effective detection of the QoS degradation. Then, if the QoS degradation occurs, Bloom filters on the service level will interact with Bloom filters on the node level to locate the source of the QoS degradation. Finally, workflow adaptation can be activated to guarantee the workflow QoS again.

The rest of this paper is organized as follows. Section 2 reviews the related work, followed by the proposed hierarchical workflow management model in Section 3. Section 4 presents the workflow adaptation mechanism for both the *service level* and the *node level*. Section 5 provides experiments that illustrate the benefits of the proposed approach. Finally, Section 6 concludes this paper.

2. Preliminaries and related work

2.1. Preliminaries

2.1.1. QoS for workflows

The atomic service QoS refers to the *non-functional* characteristics of atomic services, which can be divided into the *general QoS* and the *domain-dependent QoS* (Kritikos and Plexousakis, 2009). The *general QoS* are the common characteristics (e.g. *the execution time*, *cost* and *availability*) that are independent of their application domains, while the *domain-dependent QoS* are characteristics that strongly rely on their specific application domains (e.g. *the search accuracy* for a *search engine*, and the *voice quality* for *IP telephony*).

Similarly, the workflow QoS represents the quantitative and qualitative characteristics of a workflow necessary to achieve a set of initial requirements. Here, two concepts should be noted, the *expected QoS* and the *actual QoS*. The *expected QoS* represents the user's requirements which the workflow should achieve, while the *actual QoS* refers to the actual QoS value the workflow achieved. As a workflow is composed of a set of atomic services, the workflow QoS is associated with the QoS of its component atomic services.

For the QoS monitoring, Zeng et al. (2007) argued that most QoS metrics can be observed or computed based on service operations. They presented a QoS observation model that defined the business-level metrics together with their associated evaluation formulas, and implemented a QoS monitoring system. Fei et al. (2008) presented a policy-driven distributed monitoring framework for QoS information collection, in which some QoS can be observed by capturing SOAP messages transmitting among atomic services. For example, the *response time* can be calculated by the time stamp of the first and last message of an atomic service. Based on these obtained atomic service QoS, the workflow QoS can be computed by aggregating the QoS of its component atomic services. Jaeger et al. (2004) indicated that a workflow can have different composing patterns (Jaeger et al., 2004) such as *sequential*, *parallel*, *conditional* and *loop* etc. Different

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات