Advanced Engineering Informatics 27 (2013) 625–635

Contents lists available at ScienceDirect

Advanced Engineering Informatics

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

A migration-based approach towards resource-efficient wireless structural health monitoring

Kay Smarsly [a,*], Kincho H. Law [b]

[a] Department of Civil Engineering, Bauhaus University Weimar, Weimar, Germany
[b] Department of Civil and Environmental Engineering, Stanford University, Stanford, CA, USA

ARTICLE INFO

Article history:
Received 29 January 2013
Received in revised form 24 July 2013
Accepted 19 August 2013
Available online 14 September 2013

Keywords:
Structural health monitoring
Wireless sensor networks
Smart structures
Distributed artificial intelligence
Mobile multi-agent systems
Dynamic wireless code migration

ABSTRACT

Wireless sensor networks have emerged as a complementary technology to conventional, cable-based systems for structural health monitoring. However, the wireless transmission of sensor data and the on-board execution of engineering analyses directly on the sensor nodes can consume a significant amount of the inherently restricted node resources. This paper presents an agent migration approach towards resource-efficient wireless sensor networks. Autonomous software agents, referred to as "on-board agents", are embedded into the wireless sensor nodes employed for structural health monitoring performing simple resource-efficient routines to continuously analyze, aggregate, and communicate the sensor data to a central server. Once potential anomalies are detected in the observed structural system, the on-board agents autonomously request for specialized software programs ("migrating agents") that physically migrate to the sensor nodes to analyze the suspected anomaly on demand. In addition to the localized data analyses, a central information pool available on the central server is accessible by the software agents (and by human users), facilitating a distributed-cooperative assessment of the global condition of the monitored structure. As a result of this study, a 95% reduction of memory utilization and a 96% reduction of power consumption of the wireless sensor nodes have been achieved as compared with traditional approaches.

1. Introduction

According to the American Society of Civil Engineers (ASCE), deficient and deteriorating surface transportation infrastructure in the United States is expected to cost $912.0 billion by 2020 and more than $2.9 trillion by 2040 [1]. As the Urban Land Institute (ULI) reveals in its "Infrastructure 2012" report [2], the situation in other regions is similar, for example in China and India – countries that are rapidly urbanizing – or in Europe, where investments for infrastructure improvements of more than $2.6 trillion (€2.0 trillion) are needed. Other infrastructure systems, such as dams, buildings or wind turbines, face similar problems as they are subjected to ageing and other environmental factors. Therefore, future generations of civil engineering structures, termed "smart structures", are expected to be instrumented with structural health monitoring (SHM) systems so that the structures are capable of continuously monitoring and assessing their own structural conditions [3–5].

Structural health monitoring systems, consisting of sensors, data acquisition units, computer systems and connecting cables, are designed to detect structural changes before they reach critical states. By analyzing the sensor data recorded from the structures, SHM systems provide the opportunities to enhance the safety and reliability of engineering structures and to reduce the costs for management, maintenance and repair throughout the structures' life cycles [6]. However, in conventional SHM systems the installation of cables can be expensive, time-consuming and labor-intensive, entailing high maintenance costs for the SHM systems. Eliminating the need for connecting cables, wireless sensor networks have emerged as a cost-effective and reliable alternative to conventional, cable-based SHM systems [7–11]. Composed of numerous wirelessly connected sensor nodes, wireless sensor networks are installed in the structure to automatically collect, analyze, aggregate and communicate vast amounts of sensor data. The sensor nodes, integrating advanced embedded systems technologies, are capable of self-interrogating collected sensor data for signs of structural changes [12,13]. In essence, the sensor data is first analyzed and aggregated on the nodes, from high-bandwidth raw sensor data to low-bandwidth streams of processed results. The analysis results are then transferred to centralized computer systems, or to adjacent sensor nodes, for further processing.

By first analyzing the data sets locally and then communicating the results to the connected computer systems, transmissions of

* Corresponding author. Address: Department of Civil Engineering, Bauhaus University Weimar, Coudraystr. 7, 99423 Weimar, Germany. Tel.: +49 3643 58 4215.

E-mail address: kay@smarsly.de (K. Smarsly).

1474-0346/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.aei.2013.08.003

large records of raw sensor data can be avoided. As a result, the energy consumption for wireless data transmission is substantially reduced. However, considerable computational power is needed for the local execution of complex engineering analyses. Therefore, there have been active research efforts in the past several years towards reducing the power consumption of wireless sensor nodes by optimizing the sensor node hardware as well as the software embedded into the nodes. For example, new software approaches, such as energy-efficient source coding and resource-efficient network protocols, and new concepts on hardware circuitry and transmitter modules for improving energy-consuming node components have been proposed [14–16].

Besides the resource consumption, a second major issue when deploying wireless sensor networks for structural health monitoring is the isolated, limited view of a wireless sensor node on a small area of the total structure. It is well known that changes in the global structural response and behavior (such as altered stiffnesses and structural stability) should also be considered in addition to the detection of local damages and deteriorations (e.g. corrosion, cracks, etc.). Since the sensor data is usually collected at critical locations, individual sensor information does not provide a global picture of the structural condition. In summary, besides making the hardware and software more resource-efficient, holistic (local/global) strategies are needed to assess local and global structural changes.

The goals of the research presented herein are twofold. First, the resource consumption of the sensor nodes is to be reduced with respect to memory utilization and power consumption. Second, a SHM system prototype, capable of holistically monitoring local as well as global structural phenomena, is to be implemented. To achieve these goals, this study integrates mobile multi-agent systems and dynamic wireless code migration into a wireless sensor network. The paper begins with a background on mobile multi-agent systems. Then, the migration-based monitoring concept is described in detail, and the architecture and prototype implementation of the agent-based SHM system are shown. Next, laboratory tests are presented validating the feasibility of the newly proposed concept as well as the performance of the prototype system. The paper concludes with a discussion of the test results and a comparison of the proposed migration-based concept with conventional approaches currently used in structural health monitoring.

2. Background on mobile multi-agent systems

Multi-agent technology, originating from distributed artificial intelligence research, is a rapidly developing research area that is of practical relevance since many years [17]. The broad range of application domains of multi-agent systems includes, e.g., process control, air traffic control, business process management, health care, water resource management, traffic and transportation engineering, building control, power engineering applications, and structural design [18–25]. More recently, multi-agent systems have also been applied in different branches of structural health monitoring, such as monitoring of bridges, dams, and wind turbines [26–29].

Although the term "agent" has often been misused as well as overused [30], one definition has been widely accepted in the artificial intelligence community; the "weak notion of agency", proposed by Wooldridge and Jennings [31], defines an agent as a computer program possessing four essential properties. An agent

- operates without the direct intervention of humans and, unlike a common software object, has control over its actions and internal states ("autonomy"),
- interacts with other agents through agent communication languages ("social ability"),
- perceives its environment, e.g. the physical world or a software environment, and responds in a timely fashion to environmental changes ("reactivity") and
- exhibits goal-directed behavior by taking initiatives ("proactiveness").

Multiple interacting software agents in association with the agent environment form a multi-agent system. Due to the above mentioned agent properties, multi-agent systems are characterized by a high degree of scalability, modularity, flexibility and extensibility, which makes multi-agent technology a suitable basis for solving distributed engineering problems as in structural health monitoring.

In the last decade, considerable success has been reported in porting multi-agent technology on mobile devices such as cell phones, smart phones, or wireless sensor nodes ("mobile multi-agent systems") [32–34]. The distinctive strengths of multi-agent systems – scalability, modularity, flexibility and extensibility – are utilized in mobile applications facilitating distributed-cooperative problem solving in highly dynamic environments. To adequately deal with the constraints associated with developing applications on small devices, the majority of mobile devices supports some form of the Java programming language [35]. Accordingly, most approaches towards mobile multi-agent systems are based on Java, typically using the Connected Limited Device Configuration (CLDC) [36]. CLDC, a fundamental part of the "Java Platform, Micro Edition" (Java ME), defines the most basic libraries and virtual machine features for resource-constrained devices. It is worth mentioning that CLDC, although offering all major advantages provided by the Java language such as object orientation, portability, robustness and security, in its current version 1.1 requires only 160 kB of non-volatile memory to be allocated for the CLDC libraries and for the Java virtual machine, and needs only 32 kB of volatile memory for the virtual machine runtime [36]. As can be seen from Table 1, the total memory budget needed by the CLDC specification, compared with the "Java Platform, Standard Edition" (Java SE) for desktop and server environments, is as little as 0.07% [37]

Several Java-based agent platforms for mobile devices, supporting the development of mobile multi-agent systems, are currently available. Examples include DARPA CougaarME [32], AFME [38], SPRINGS [39], 3APL-M [40], JADE-LEAP [41,42], and MAPS [43]. Agent platforms for mobile devices essentially provide lightweight subsets of Java classes supporting basic agent services for communication, for multitasking, or – if embedded into wireless sensor nodes – for accessing the node resources (e.g. sensors or memory). Detailed reviews as well as comparisons of architectures, programming models and performances of agent platforms for mobile devices can be found in [34,44,45].

It has been recognized in recent years that the performance and the dynamic behavior of mobile multi-agent systems can further be enhanced by wireless code migration [46]. Having demonstrated high effectiveness in conventional wired decentralized systems, code migration represents an emerging and powerful par-

Table 1
Minimum system requirements of Java SE and Java ME.

	Java Platform, Standard Edition (Java SE 7)	Java Platform, Mirco Edition (Java ME, CLDC 1.1)
Processor	266 MHz	16 MHz
Disk space	126 MB	32 kB
Memory	128 MB[a]	160 kB

[a] Windows 64-bit operating systems.