MidSHM: A Flexible Middleware for SHM Application Based on Service-Oriented Architecture

Yuvraj Sahni, Jiannong Cao, Xuefeng Liu
Department of Computing
The Hong Kong Polytechnic University
Hong Kong
{*csysahni, csjcao, csxfliu*}@*comp.polyu.edu.hk*

***Abstract*—Wireless Sensor Network (WSN) is often used for developing Structural Health Monitoring (SHM) application by civil researchers but they do not have much expertise on hardware and network related issues. By providing programming abstractions and hiding low level network issues middleware layer makes it easier to develop an efficient WSN-based SHM application. Service-oriented architecture (SOA) is a popular approach for designing middleware for WSN as it provides flexibility in developing WSN applications by using loosely coupled services. SOA can overcome issues like adaptation, reliability which are usually difficult to deal using other middleware approaches applied for WSN. This paper surveys various middleware approaches for WSN focusing mainly on SOA-based approach. It discusses drawbacks in various middleware approaches and points out design issues that not completely addressed by existing middleware architectures designed for SHM application. An easy-to-use SOA-based middleware, named MidSHM, has been proposed to deal with various SHM application issues such as resource optimization, in-network processing, quality of service, and fault tolerance. Two different application examples enabled by MidSHM are also shown to illustrate its flexibility and usability.**

***Keywords*-middleware; service-oriented architecture; wireless sensor network; structural health monitoring**

I. Introduction

Structural health monitoring (SHM) of civil infrastructures using Wireless Sensor Network (WSN) system has been the focus of civil researchers for a long time. Fig. 1 shows two kinds of structures demonstrating SHM system based on WSN. The reason behind very few practical deployment of SHM systems for buildings/bridges like [8], [20], [21] is lack of expertise of civil researchers on low level hardware and network issues. The main focus of civil researchers is to the reliably send the vibration data of buildings/bridges to a central server for further monitoring but this approach is not optimal [27]. Techniques like distributed computing and in-network processing can be utilized to develop a robust and optimized system but applying these techniques on WSN is complicated. The sensor nodes used in WSN suffer from low network bandwidth, low memory storage, low battery capacity, low computation power and short communication range. SHM application requirements like high sampling rate, use of computation intensive algorithms for damage detection, and dynamic and scalable network topology make the task of civil researchers even more difficult.

Figure 1. Two SHM systems based on WSN

Middleware layer can compensate for the huge gap between high level application requirements and low level hardware and network issues. Middleware layer lies between the application layer at top and WSN system layers at bottom. Traditional application development approach requires application developer to learn about hardware, software being used, operating system and application specific issues in order to develop WSN application. WSN application development becomes easier by using middleware. Middleware provides programming abstraction to hide the complexity of low lying hardware, provides techniques to optimally manage the network resources and also satisfy various application requirements [34].

Various approaches have been tried for developing a middleware for WSN that can deal with all the challenges but it is difficult to deal with all the design issues at once. There is still scope for more research on middleware for WSN. Service-oriented architecture (SOA) approach for designing middleware layer is becoming very popular among researchers due to nature of services used for application development. SOA approach can be used to develop middleware that can handle heterogeneity, dynamicity, run-time network configuration, fault tolerance, quality of service and other issues easily using loosely coupled services. The services used are independent from each other and can be integrated in customized manner to deal with requirements of different applications while hiding low level hardware and network issues from application developer. New services can also be added at run-time to satisfy any new application requirement. SOA can also be used for integration of WSN with internet and with other networks too.

This paper surveys various approaches for designing middleware for WSN such as database approach, virtual machine approach, modular approach, message-oriented approach and more specifically service-oriented architecture approach. This paper also points out shortcomings of different middleware approaches surveyed. Survey of middleware approaches for WSN has been done previously in [18], [29], [30], [36], [39]. This approach of surveying different approaches and concentrating on SOA-based middleware gives a more broader view of middleware layer for WSN and helps in understanding the benefits of SOA-based middleware over other middleware approaches. A comparative study of different middleware approaches has been done previously in [32] but they did not consider service-oriented architecture approach for comparison.

This paper also gives an overview of some unaddressed design issues for middleware architecture for WSN-based SHM application. Based on the limitations discussed, a new middleware architecture, MidSHM, based on service-oriented architecture approach has been proposed for SHM application. MidSHM consists of three layers with each layer catering to different issues by using different services. The motivation behind MidSHM is that SOA has been used previously for SHM applications but previous designs only deal with some specific issues such as heterogeneity or energy efficiency. There are various issues such as fault tolerance, quality of service provisioning, support for multiple applications, integration of different networks that have not been explored in much detail. MidSHM overcomes the limitations of previous architectures by using layer level management and integrating the services in those layers to deal with both envisioned and new application requirements.

The main contributions of this paper are:

1) This paper gives an overview of various middleware approaches applied for WSN. This paper discusses in detail about SOA-based middleware architectures and also classifies them on the basis of applications targeted by these middleware architectures.
2) This paper identifies drawbacks in various middleware approaches and discusses some design issues that are yet to be addressed in designing middleware architecture for SHM application.
3) This paper proposes MidSHM, a new flexible middleware architecture based on SOA for SHM application. A comparison study has been done that shows MidSHM is better than other SOA-based middleware architectures. It can deal with issues such as fault tolerance, dynamic application adaptation, resource optimization, quality of service provisioning etc. The flexibility of MidSHM is illustrated using application examples.

The rest of the papers is as follows. Section II discusses different middleware approaches applied for WSN except SOA-based approach for middleware and gives pros and cons of those approaches. Service-oriented architecture approach is described in detail in section III using various middleware architecture examples based on SOA approach. Section III also discusses some unaddressed design issues for designing middleware architecture for SHM application. Section IV is about MidSHM and it is divided into three subparts: MidSHM description, application examples enabled by MidSHM, and evaluation consisting of comparison study with other SOA-based middleware. Section V concludes the paper with conclusion and future work.4.

II. Literature Review

Middleware layer has been used for long time for decreasing the complexity of WSN application development. There are many approaches for designing middleware for WSN like database approach, virtual machine approach, message oriented approach, data-centric approach, modular approach, application driven approach, and service-oriented architecture approach. Table I gives a list of different middleware approaches along with their corresponding examples. Description of different middleware approaches along with their limitations has been explained below.This section however does not discusses middleware based on SOA approach as it is discussed in detail in next section.

Database Approach: This approach considers WSN as a database system where applications can query the data from sensor nodes using structured query language (SQL) like queries. The major drawback with this approach is that it does not provides support for space-time relationship between events. This approach also does not allows rendering of data in real time and provides approximate results. *TinyDB* [28], *SINA* [37], *TinyLIME* [10], *DSWare* [24] are few examples of middleware architectures for WSN system based on database approach.

Message-oriented approach: Message-oriented approach uses publish/subscribe communication model to provide asynchronous communication between sender and receiver that are loosely coupled. This approach is very efficient for wireless sensor network systems which require event based monitoring. Some middleware architectures based on message-oriented approach are *TinyDDS* [7], *PS-QUASAR* [9], *Mires* [38].

Modular Approach: This approach is based on the fact that WSN have dynamic network topology, mobility and require application adaptation to deal with application failure. It is impossible for a single protocol to satisfy the demand of even a single application. A monolithic application software that adapts to various imagined scenarios cannot be used

Table I
VARIOUS MIDDLEWARE APPROACHES WITH EXAMPLES

Middleware Approaches	**Examples**
Database approach	TinyDB [28], SINA [37], TinyLIME [10], DSWare [24]
Message-oriented approach	TinyDDS [7], PS-QUASAR [9], Mires [38]
Modular approach	Impala [26], Agilla [15]
Application-oriented approach	MiLAN [19], MidFusion [1]
Virtual Machine approach	Mate [23], MagnetOS [5]
Service-oriented architecture approach	ISHMP Services Toolsuite [33], Servilla [16], OASiS [2], SensorsMW [3], WSN-SOA [22], MidCASE [4]

for application adaption as it is difficult to manage and more prone to errors due to large size and besides a future scenario cannot always be predicted. A modular approach is required where application is non-monolithic and updates are done on the fly. The drawback with approach is that it does not allows network heterogeneity because of the nature of its code instruction. *Impala* [26] and *Agilla* [15] are two common examples of middleware architectures based on modular approach.

Application-oriented approach: Application-oriented approach provides tight coupling between applications and sensor network. This approach allows applications to specify their requirement in form of QoS and adjust the network according to application need. The drawback with this approach is that the middleware designed is not generic. *MiLAN* [19] and *MidFusion* [1] are examples of middleware based on this approach.

Virtual machine approach: Virtual machine approach allows applications to be composed of small modules that are distributed through the network using special algorithms. The drawback with this approach is that because of large overhead it is not feasible to execute large number of executions. *Mate* [23] and *MagnetOS* [5] are two examples based on this approach.

III. MIDDLEWARE BASED ON SERVICE-ORIENTED ARCHITECTURE

Middleware approaches discussed in previous section make application development easier by providing programming abstractions and deal with some basic issues such as easier software installation, data aggregation etc. But this is not sufficient because different WSN applications require more advanced functions. WSN systems now use heterogeneous hardware and software for application development and deploy multiple applications simultaneously. This makes application development even more difficult and middleware must support flexibility, reusability, heterogeneity, and dynamicity among other features [29].

Service-oriented architecture provides loosely coupled services that are flexible and can be reused to deal with different functionalities of an application. An application is composed of group of services linked in a particular fashion. Same set of services can be linked in a different manner so as to satisfy the requirements of different application. Service-oriented architecture approach helps in development of dynamic, and highly adaptive applications by using services that can be linked in customized manner. It does not requires changing the inner details of services in order to deal with new application requirements [33]. The major drawback with this approach it that it is not very lightweight. Traditional web services mostly used for SOA have high memory requirement, and high computation and communication cost [31] and thus cannot be applied directly for wireless sensor networks.

A. Middleware approaches based on SOA

Each middleware architecture discussed in this section is very different from the other but they all use service-oriented architecture for middleware implementation. There are other approaches too based on SOA like MARINE ([12], [25]), [11], [35] but it is not possible to discuss every approach based on SOA. Middleware approaches presented below have been classified in terms of applications targeted. Fig. 2 shows middleware classification based on application targeted along with an example of a SOA-based middleware architecture corresponding to the application type.

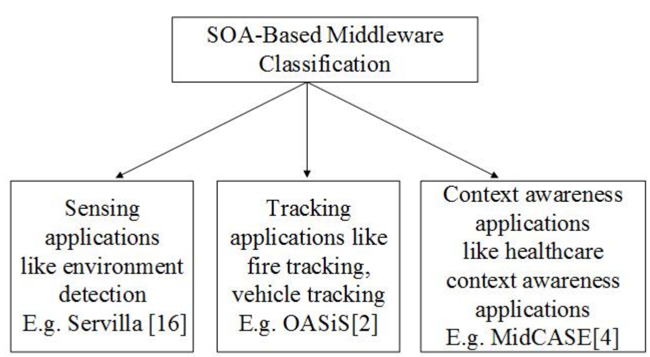

Figure 2. Application classification for SOA-based middleware

Sensing Applications

1. *Illinois Structural Health Monitoring Project (ISHMP) Services Tool suite* [33]: ISHMP services tool suite is an open source toolkit developed at University of Illinois at Urbana-Champaign for SHM application development using smart sensors. This toolkit provides set of services that can be used to implement SHM algorithms for modal analysis and damage detection. Services provided also help in implementation of middleware based on service-oriented architecture that facilitates in acquisition and reliable communication of sensor data across wireless sensor network. The services provided by ISHMP services toolkit can be divided into three categories:*a*) Foundation services, *b*) Numerical services, and *c*) Tools and utilities.

2. *Servilla* [16]: Servilla middleware architecture has been designed for heterogeneous WSN by using platform specific services to enable platform independent applications. This approach can be utilized to perform sophisticated operations over a large area for a long time by using benefits of both resource constrained and resource intense devices to perform various operations. Servilla provides energy efficiency by using novel binding semantics and different service invocation strategies. A new service specification language, ServillaSpec, has been used instead of traditional web services description language (WSDL) to support resource constrained sensor nodes.

Servilla middleware runs on individual nodes in WSN and it consists of virtual machine (VM) and a service provisioning framework (SPF). Servilla VM is provided by Agilla [15] but it has been modified to meet the needs of middleware architecture. SPF consists of a consumer (SPF-consumer) that discovers and accesses services, and provider (SPF-provider) that advertises and executes services. Servilla has been tested for structural health monitoring application.

3. *Adaptive Servilla* [17]: Adaptive Servilla improves upon Servilla by providing energy aware provider selection. Adaptive Servilla also increases energy efficiency by exploiting opportunities for sharing service executions and providing dynamic adaptation of provider. Servilla requires developers to manually adjust or specify the type of the bindings but Adaptive Servilla automatically adjusts the bindings based on the expected energy efficiency of the complete system. The process of rebinding to an alternative provider is done dynamically and internally within Adaptive Servilla so the application developer is unaware of this process.

Adaptive Servilla and Servilla do not support multi-hop network topology and QoS provisioning. While selecting provider, energy efficiency is considered but there are other parameters that can also be considered. Adaptive Servilla only provides passive adaption to network topology which leads to high latency so other techniques can be considered to overcome latency issue. Servilla and Adaptive Servilla are enhanced in [14] by including multi-dimensional QoS specifications from each stakeholder into a single value. The value obtained in [14] is then used to determine the best configuration of interactions. Efficacy of adaptive servilla has been demonstrated for medical patient monitoring and structural monitoring.

4. *SensorsMW* [3]: SensorsMW is a service-oriented middleware that allows applications to adapt and configure the low level hardware according to application requirements. It allows applications to specify their QoS requirements using WS-Agreement to monitor and manage Service Level Agreements (SLA) and dynamically configure the network at run-time. It provides independency from underlying WSN technology. It allows flexibility in delivering data and guaranteeing integration and inter-operability by acquiring data using Web Services. SensorsMW is comprised of four major components: Contracts Creator Component, WSN Gateway, Service Provider, and Data Registry. SensorsMW has been developed for network enterprises and this approach can be used either for periodic measurements or event monitoring. The middleware architecture has been tested for temperature measurement application.

Tracking Applications

1. *OASiS* [2]: OASiS is an Object-centric, Ambient-aware, Service-oriented programming framework for WSN application development. OASiS uses programming paradigm that provides separation of concern (SoC) by using multi-development process. OASiS deals with heterogeneity using service-oriented approach while the dynamic network configuration is provided by ambient-aware middleware. OASiS also supports real-world integration and specification of application specific and network QoS requirements. OASiS framework is very useful to develop dataflow applications such as vehicle tracking, fire detection, distributed gesture recognition etc. OASiS uses service graph to describe application dataflow. OASiS middleware consists of following services: Node Manager, Object Manager, Service Discovery Protocol, and Composer. This approach has been tested for heat source tracking application.

2. *WSN-SOA* [22]: This middleware architecture provides a multi-level approach that allows SOA to be used at low-capacity nodes without the overhead of XML-based technologies. This approach aims to enable auto-configuration feature at both network and service levels. This architecture classifies nodes in three classes: Full capacity nodes (uses Web services stack and Enterprise Service Buses), Limited capacity nodes (uses device profile for Web services(DPWS)), and Limited capacity nodes (uses WSN-SOA stack). WSN-SOA has been implemented using TinyOS. WSN-SOA architecture defines two kind of services: Management services, sensor/actuator services. The bridging between full capacity nodes and low capacity node is done using gateway or bridge. A publish/subscribe data dissemination mechanism using topic based message filtering system has been provided for more efficiency instead of just using one-one service translation between DPWS and WSN-

SOA. WSN-SOA has been tested for surveillance application which involved detecting the intrusions via seismic vibrations and tracking the intruder.

Context Awareness Applications

1. *MidCASE* [4] : Middleware Enabling Context-awareness for Smart Environment (MidCASE) is a distributed service-oriented middleware. It has been designed to support programmable application layer consisting of various scenarios with underlying heterogeneous hardware. The middleware achieves service in each awareness service domain and the aware process is done by applying rule based reasoning. Service-oriented architecture allows communication between different factors involved in context-awareness process which was not possible in traditional context awareness systems. MidCASE context awareness system under wireless sensor network consists of three layers: sensor device environment, middleware and application scenario. Middleware contains five layers: Hardware abstract, service registry, context-model, awareness and reason, and application presentation layer. MidCASE middleware has been designed for context awareness applications and has been tested for context awareness scenario in healthcare.

B. Design issues for SOA-based Middleware for SHM application

Structural health monitoring application involves monitoring the damage in civil infrastructures using hundreds of sensors nodes installed for a long a period of time. The data obtained from sensors nodes consists of mostly vibration data and this data is then passed through damage detection algorithms. This whole process must be reliable and robust to withstand various failures in sensor nodes and provide dynamic application adaptation. Middleware based on SOA has been utilized previously for SHM application development [16], [33] but there are lot of issues which still need to be addressed. Previous approaches generally focus on one of the issues among heterogeneity, dynamicity, in-network processing etc. and a middleware that can handle all the issues still needs to be developed. Some of the design issues for WSN based SHM system that have not been addressed in current SOA-based middleware approaches are:

1. Quality of Service provisioning in middleware architecture: QoS is very important feature to be considered while developing SHM application. There are very few middleware architectures in general which tackle QoS issue and there is still a lot of scope for further development in this area especially for SHM application. QoS factors like energy consumption, low latency, reliability, priority, and deadline can be included in middleware architecture to make the system more efficient.

2. Fault tolerance: Sensor nodes sometimes get damaged or fail and in some cases sensors attached to sensor nodes starts giving faulty sensor readings [6]. This issue is very important for SHM application as the device nodes are generally deployed in harsh environments where the possibility of such failure increases. Previous middleware architectures only consider damage node scenario but there is more to fault tolerance than that.

3. Dynamic service binding: Dynamic service binding has been used previously to adapt the WSN to network heterogeneity, dynamicity and other requirements but there is still a lot of scope for improvement [17]. For eg. only passive binding has been used in [17] which leads to high latency and new binding techniques can be tried to improve the overall system.

4. Provision to address emergency situation: Most of the time, nodes in WSN are either asleep or transmitting ambient data [40]. So, a provision must be there to wake up the nodes in emergency situation so that emergency event data is recorded and damage can be assessed accordingly.

5. Damage detection algorithm selection: Most of the middleware architectures that have been proposed implement a damage detection algorithm either centrally or distributed. Different algorithms can be used at different times according to the data collected to reduce the total energy consumption. System identification method technique has been swapped in [33] and this approach of swapping different techniques can be extended further. This kind of algorithm selection or swapping techniques can be helpful in minimizing the energy consumption.

6. Provision to deploy multiple applications simultaneously over heterogeneous network: Heterogeneous devices are being used to satisfy the demands of multiple applications running at one time. A network deployed for SHM can be used for other applications too such as home automation, smart lighting, temperature control. Such functionality is very essential for future scenarios.

7. Integration of multiple networks: A wireless sensor network meant for structural health monitoring application can be integrated with other networks such as Radio-frequency Identification (RFID), Internet etc. to fully exploit the resources available. WSN systems functionality will increase with support of such feature by middleware.

IV. MIDSHM - PROPOSED MIDDLEWARE ARCHITECTURE BASED ON SOA

The survey of different middleware architectures done in previous sections points out that SOA approach for middleware is suitable for applications like SHM but there is need for further improvement. MidSHM, a middleware architecture based on SOA approach, has been proposed to deal with WSN-based SHM application issues.

The most important feature for the middleware architecture is it should be easy-to-use. MidSHM provides a wide range of middleware services. These independent services can be easily linked together to develop any application as shown later with the help of two application examples.

The main challenge behind this approach is Web Services standards used such as SOAP, WSDL, UDDI are not suitable for resource-constraint devices used in WSN due to their bulky XMl-messaging format [2]. Many techniques have been applied previously to overcome this challenge. OASiS [2] used byte sequence messaging format, Servilla [16] developed a simpler service and task description language while WSN-SOA [22] developed a software architecture and protocol to deal with this challenge.

MidSHM can use any of the above techniques to overcome the challenge. The middleware architecture has not been implemented yet and for now only middleware architecture has been described. MidSHM has been designed to provide support for following features:

1) Resource Optimization: WSN system have resource-constraint environment. Resources like energy, power must be optimized.
2) Dynamic Network Topology: Self-configuration of network is required to deal with device failures in network.
3) In-Network Processing: It reduces the amount to raw data within the network and provides energy efficiency.
4) Quality of Service: QoS factors like reliability and deadline are important for SHM application.
5) Heterogeneity: Use of diverse device nodes provides energy efficiency [16] and is also important to deal with multiple applications in WSN system.
6) Fault Tolerance: It prevents faulty sensor readings which lead to incorrect damage detection result in SHM application.
7) Real World Awareness: WSN are used to measure real world phenomenon and thus must be aware of time and space associated with data measurement.
8) Run-time Reconfiguration: Dynamic application adaptation is required to provide new features in SHM application.

A. MidSHM description

MidSHM consists of three layers as shown in Fig. 3: Networks management layer, task management layer, application management layer. Three-layer middleware architecture approach has been considered before in [13] [41] but those architectures did not use service-oriented architecture approach for middleware implementation. Functioning of each layers and the services used in those layer to tackle the issues is explained in detail below:

Network management layer: Network management layer handles all the network and hardware related issues in wireless sensor network. There are specific modules for catering to different issues. Data management module reduces the amount of data that flows within the networks by using techniques such as compression, filtering or data management using application specific algorithms. Sensor nodes have low battery life but the applications supported by WSN require device nodes to operate for a long time. A

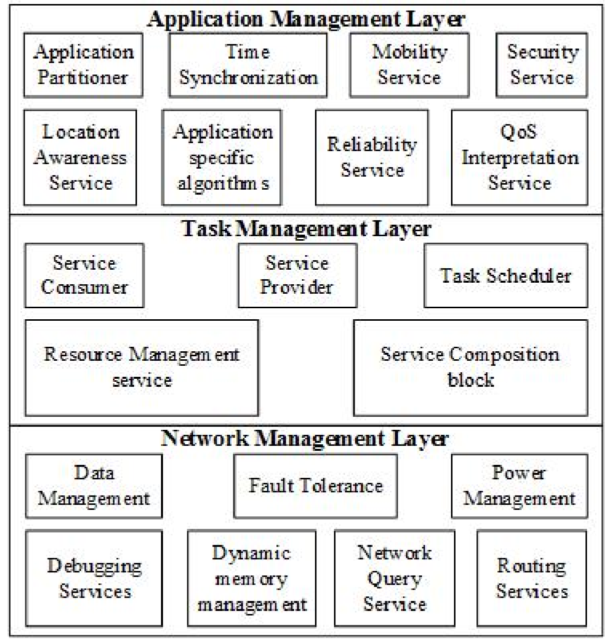

Figure 3. Proposed Middleware Architecture

power management module manages the power consumption of both individual node and the whole network in order to increase the network lifetime. Dynamic memory management module is used to deal with low memory issue of sensor nodes by providing dynamic memory for each task.

The data and control commands are sent from one node to other using routing protocol. Routing service provides routing protocol to reliably send the data for all three cases: unicasting, multicasting, broadcasting. Network query service queries the hardware below to determine the status of each node. The status report received consists of remaining battery power, channel and bandwidth utilization, memory consumption and other required parameters to optimize the resource consumption of each task. Fault tolerance module is responsible for two tasks: one is to provide error control service so that error can be identified and second one is to deal with the issue of faulty sensor readings which may result in faulty outcomes. Debugging service is used for testing sensor, radio and the network configuration before initiating tasks of an application.

Task Management layer: Task management layer is responsible for managing tasks by using various service modules. Two main components are service consumer which is responsible for discovering, matching and invoking services, and service provider which provides and executes services. The service binding is dynamic so in case of node failure or failure to meet QoS demands of an application the binding is changed dynamically. The task scheduler module determines the task scheduling according to resource consumption. There are three options available: on-demand, periodic, and event triggered. Sometimes a single service cannot meet the demands of a task so a service composition block is provided that combines multiple services to deal with the demands of a particular task. A resource management service takes parameters from network query service and QoS interpretation service to dynamically optimize the resource consumption of each task.

Application management layer: This layer is the top most layer of middleware layer and it lies just below application layer. This layer helps in management of multiple applications that are being run using WSN. The main function of this layer is to provide application specific services and in case a service is not available a new service is added. The middleware supports deployment of multiple applications simultaneously and each application is divided into multiple tasks for better handling. This layer helps in dynamic application adaptation.

Application partition service partitions each application into multiple tasks which are then passed on to task management layer for further handling. Synchronized sensing is responsible for collecting the synchronized data from sensors and time stamping it. Synchronized sensing is also responsible for setting the global and local clock on each node. Each node is aware of location of its neighbors with the help of location awareness service. Security service is for dealing with security related issues in WSN. Reliability service uses different techniques to ensure reliable transmission of data and control commands from source to destination.

QoS interpretation service considers QoS requirements from multiple applications and passes them down to task management layer for QoS consideration. Application specific algorithms module provides domain specific distributed algorithms required by WSN to calculate the final outcome or analyze the data obtained from sensor nodes. For e.g. SHM application requires damage detection algorithms like Eigensystem Realization Algorithm (ERA), Natural Excitation Technique (NeXT) etc. to be included in application specific algorithms block. New algorithms can be added at runtime according to application requirement.

B. Application Examples enabled by MidSHM

MidSHM presented in Fig. 2 has been developed considering structural health monitoring application but it is flexible and can be used for other applications too. The middleware has not been implemented yet but two application examples have been shown to demonstrate the flexibility nature of architecture. The application examples give an idea of how different services can be utilized to deal with different requirements of an application. Middleware architecture shown in both the examples has been truncated to show only those services that are used by application. Although services like resource management service, fault tolerance service etc. are very important for development of WSN system but they have not been used so as to simplify the application. The focus is on features like flexibility and usability of middleware architecture. In a practical scenario these services should not be ignored.

Fig. 4 shows a basic data collection application. This is a basic application which involves acquiring synchronized data from sensors, perform in-network processing and deliver the data to destination node. Network Initialization block configures the network and tests the radio and hardware using debugging service. Wake up block is responsible for setting the time during which the node is in active mode and carries out various tasks. These two blocks are basic blocks and common for most of the applications but internal settings may be changed as per the requirement of particular application. Service consumer and service producer block have been included in middleware architecture because they are main blocks that carry out calling and execution of services. This application is very basic and it is the building block for most of the applications developed using WSN system.

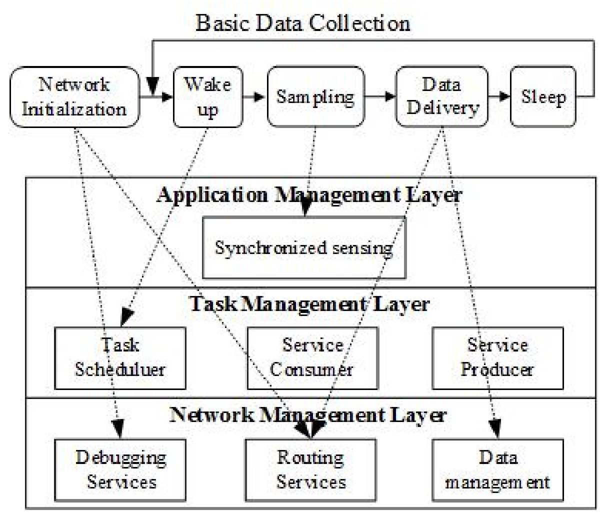

Figure 4. Basic Data collection application

Next application shown in Fig. 5 is Automatic SHM. This application detects the damage in structure and then automatically delivers the result to base station node for further action. Damage detection and data delivery have been combined in a single block. The damage detection is done within the network using distributed SHM damage detection algorithms and the data is transferred simultaneously from one node to other. After final computation the damage detection result is sent to base station node and this part is represented using Report delivery block in Fig. 5. Application specific algorithms block is generic and can be utilized for other applications too. SOA provides the flexibility of adding new services at run-time so new application specific algorithms as well as new services can be added to deal with different WSN applications.

Both of the applications presented are very different yet there is not much change required. MidSHM can be utilized for other applications too by using different services depending on the requirement of application. Context awareness service can be used for tracking application that requires knowledge of location of node and its neighbors. Mobility service can be used to track a mobile object. In case an application requires certain quality of service factors such as reliability, deadline etc. then QoS interpretation service comes into picture.

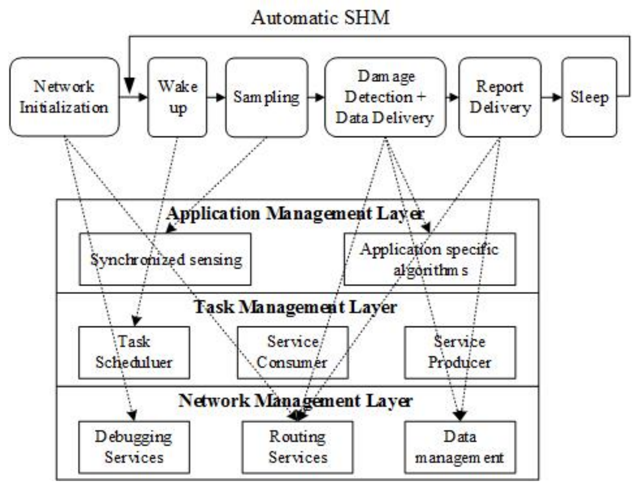

Figure 5. Automatic SHM application

C. Evaluation

The key to solving various issues using MidSHM is to separate the concerns [2]. Separation of issues approach is followed by partitioning the middleware architecture in three layers. Network management layer at bottom is mainly responsible for handling network and hardware issues. Task management layer calls different services and executes them while managing the resource consumption. The uppermost application management layer is specifically for handling different domain specific issues and provide different services as required by application.

All the functionalities are handled using middleware services. Each layer contains set of services to deal with different requirements. These services are managed by application developer using service consumer and service producer block in task management layer. Services used by the middleware architecture are located within the network which helps in in-network processing of data. Data management module contains different services such as data aggregation, filtering etc. for in-network processing. Issues such as Heterogeneity, scalability, run-time reconfiguration are resolved due to independent nature of services used in SOA. Services located on different hardware nodes can be composed together by virtue of being independent. This independent property also allows services to be introduced at run-time allowing dynamic application adaptation. Services can be used on any number of nodes and this allows network to be scalable.

Dynamic network topology is provided by using dynamic service binding which allows nodes to bind to different services in case of device failure. Routing service also helps by providing network routing protocols that help in self-configuring of network. Error control and tolerance against faulty sensor readings are provided by fault tolerance module present in network management layer. QoS interpretation service interprets QoS requirement by different applications while the Network Query Service queries the network status and requirements. These both services pass the requirements to task management layer for consideration.

Resource optimization is a broad term that includes optimization of energy, memory, computation and communication resources. These resources are centrally controlled by Resource management service which works together with other modules such as data management, power management, and dynamic memory management.

MidSHM provides awareness of time and space associated with data measurement by using location awareness service and synchronized sensing block. Taking real world awareness in consideration helps in efficient utilization of service for application development. Services can be deployed on multiple networks and can be used to integrate multiple networks. SOA has been used previously to integrate WSN with internet. Application specific issues can be dealt by providing services in application management layer and utilizing application specific algorithms.

A comparative study of MidSHM with other SOA-based middleware architectures studied in section III has been done and the results are summarized in Table II. We have used eight differetn characteristics to compare different middleware architectures. Characteristics of different middleware architecture have been analyzed and compared with MidSHM. The objective of this comparative study is to find out whether a middleware architecture supports a particular feature or not. *Checkmark* and *Xmark* are used for denoting whether a middleware architecture supports a particular feature or not and incase the feature is supported partially by middleware then *Partial* sign is used. Initial comparison study points out that previous middleware architectures have some drawbacks while MidSHM has the capability to support all design issues relevant to SHM application. The middleware has not been implemented yet so various practical issues that arise while implementing the middleware cannot be addressed at this initial stage.

V. Conclusion and Future Work

This paper gives a comprehensive overview on middleware layer for WSN. Various middleware approaches for WSN have been surveyed and their drawbacks have

Table II
A COMPARATIVE STUDY OF VARIOUS SERVICE-ORIENTED ARCHITECTURE BASED MIDDLEWARE

	Resource Optimization	Dynamic Network Topology	In-Network Processing	Quality of Service	Heterogeneity	Fault Tolerance	Real World Awareness	Run-time Reconfiguration
ISHMP Services Toolsuite [33]	✓	✓	Partial	✗	✗	✓	✗	✗
Servilla [16]	Partial	✗	Partial	✗	✓	✗	✗	✓
Adaptive Servilla [17]	✓	✓	Partial	✗	✓	✗	✗	✓
OASiS [2]	Partial	✓	Partial	✓	✓	Partial	✓	✗
SensorsMW [3]	Partial	Partial	Partial	✓	✓	Partial	✓	✗
WSN-SOA [22]	Partial	✓	✗	✗	✓	✗	✓	✓
MidCASE [4]	✓	✗	✗	✗	✓	✗	✓	✓
MidSHM	✓	✓	✓	✓	✓	✓	✓	✓

been pointed out. This paper points out some of the design issues which are yet to be fully addressed by SOA-based middleware for SHM application. Middleware must be flexible and easy-to-use. These two characteristics are achievable using Service-oriented architecture approach for middleware. SOA approach uses loosely coupled services for middleware implementation which provides an abstraction layer for application developer and hides low level issues.

A SOA-based middleware architecture has been proposed to deal with SHM application issues. It has been found in initial evaluation that MidSHM can resolve issues such as Quality of service, fault tolerance, heterogeneity, and dynamicity among other issues discussed in the paper. Two application examples have also been discussed to give an understanding of the usability and flexibility of the proposed middleware architecture.

MidSHM is in its initial development stage and has not yet been implemented. A lot of unknown practical issues generally faced during implementation are yet to be addressed. The next step will be to design the middleware layer based on SOA and test its performance on real hardware. A thorough analysis including performance comparison MidSHM with other middleware approaches will be presented once the development of middleware layer is complete.

ACKNOWLEDGMENT

This research is financially supported in part with a grant from the Innovation and Technology Commission of the HKSAR Government to the Hong Kong Branch of National Rail Transit Electrification and Automation Engineering Technology Research Center (K-BBY1), the NSFC/RGC Joint Research Scheme (N_PolyU519/12), the Chinese National 973 project 2015CB352202, and NSFC grant of China 61572218.

REFERENCES

[1] H. Alex, M. Kumar, and B. Shirazi. Midfusion: An adaptive middleware for information fusion in sensor network applications. *Information Fusion*, 9(3):332–343, 2008.

[2] I. Amundson, M. Kushwaha, X. Koutsoukos, S. Neema, and J. Sztipanovits. Oasis: a service-oriented middleware for pervasive ambient-aware sensor networks. *Pervasive and mobile computing journal on middleware for pervasive computing*, 2006.

[3] G. F. Anastasi, E. Bini, A. Romano, and G. Lipari. A service-oriented architecture for qos configuration and management of wireless sensor networks. In *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, pages 1–8. IEEE, 2010.

[4] Y. Bai, H. Ji, Q. Han, J. Huang, and D. Qian. Midcase: a service oriented middleware enabling context awareness for smart environment. In *Multimedia and Ubiquitous Engineering, 2007. MUE'07. International Conference on*, pages 946–951. IEEE, 2007.

[5] R. Barr, J. C. Bicket, D. S. Dantas, B. Du, T. Kim, B. Zhou, and E. G. Sirer. On the need for system-level support for ad hoc and sensor networks. *ACM SIGOPS Operating Systems Review*, 36(2):1–5, 2002.

[6] M. Z. A. Bhuiyan, J. Cao, G. Wang, and X. Liu. Energy-efficient and fault-tolerant structural health monitoring in wireless sensor networks. In *Reliable Distributed Systems (SRDS), 2012 IEEE 31st Symposium on*, pages 301–310. IEEE, 2012.

[7] P. Boonma and J. Suzuki. Tinydds: An interoperable configurable publish/subscribe middleware for wireless sensor networks, 2009.

[8] M. Ceriotti, L. Mottola, G. P. Picco, A. L. Murphy, S. Guna, M. Corra, M. Pozzi, D. Zonta, and P. Zanon. Monitoring heritage buildings with wireless sensor networks: The torre aquila deployment. In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, pages 277–288. IEEE Computer Society, 2009.

[9] J. Chen, M. Díaz, B. Rubio, and J. M. Troya. Ps-quasar: A publish/subscribe qos aware middleware for wireless sensor and actor networks. *Journal of Systems and Software*, 86(6):1650–1662, 2013.

[10] C. Curino, M. Giani, M. Giorgetta, A. Giusti, A. L. Murphy, and G. P. Picco. Tinylime: Bridging mobile and sensor networks through middleware. In *Pervasive Computing and*

Communications, 2005. PerCom 2005. Third IEEE International Conference on, pages 61–72. IEEE, 2005.
[11] F. C. Delicato, P. F. Pires, L. Pinnez, L. Fernando, and L. da Costa. A flexible web service based architecture for wireless sensor networks. In *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on*, pages 730–735. IEEE, 2003.
[12] F. C. Delicato, J. M. Portocarrero, J. R. Silva, P. F. Pires, R. P. de Araújo, and T. Batista. Marine: Middleware for resource and mission-oriented sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 17(1):40–54, 2013.
[13] B. Elen, S. Michiels, W. Joosen, and P. Verbaeten. A middleware pattern to support complex sensor network applications. *status: published*, 2006.
[14] C.-L. Fok, C. Julien, G.-C. Roman, and C. Lu. Challenges of satisfying multiple stakeholders: quality of service in the internet of things. In *Proceedings of the 2nd Workshop on Software Engineering for Sensor Network Applications*, pages 55–60. ACM, 2011.
[15] C.-L. Fok, G.-C. Roman, and C. Lu. Agilla: A mobile agent middleware for self-adaptive wireless sensor networks. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 4(3):16, 2009.
[16] C.-L. Fok, G.-C. Roman, and C. Lu. Servilla: a flexible service provisioning middleware for heterogeneous sensor networks. *Science of Computer Programming*, 77(6):663–684, 2012.
[17] C.-L. Fok, G.-C. Roman, and C. Lu. Adaptive service provisioning for enhanced energy efficiency and flexibility in wireless sensor networks. *Science of Computer Programming*, 78(2):195–217, 2013.
[18] S. Hadim and N. Mohamed. Middleware: Middleware challenges and approaches for wireless sensor networks. *IEEE distributed systems online*, (3):1, 2006.
[19] W. B. Heinzelman, A. L. Murphy, H. S. Carvalho, M. Perillo, et al. Middleware to support sensor network applications. *Network, IEEE*, 18(1):6–14, 2004.
[20] S. Jang and B. F. Spencer Jr. Structural health monitoring for bridge structures using smart sensors. Technical report, Newmark Structural Engineering Laboratory. University of Illinois at Urbana-Champaign., 2015.
[21] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, pages 254–263. IEEE, 2007.
[22] J. Leguay, M. Lopez-Ramos, K. Jean-Marie, and V. Conan. An efficient service oriented architecture for heterogeneous and dynamic wireless sensor networks. In *Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on*, pages 740–747. IEEE, 2008.
[23] P. Levis and D. Culler. Maté: A tiny virtual machine for sensor networks. In *ACM Sigplan Notices*, volume 37, pages 85–95. ACM, 2002.
[24] S. Li, S. H. Son, and J. A. Stankovic. Event detection services using data service middleware in distributed sensor networks. In *Information Processing in Sensor Networks*, pages 502–517. Springer, 2003.
[25] W. Li, F. C. Delicato, P. F. Pires, Y. C. Lee, A. Y. Zomaya, C. Miceli, and L. Pirmez. Efficient allocation of resources in multiple heterogeneous wireless sensor networks. *Journal of Parallel and Distributed Computing*, 74(1):1775–1788, 2014.
[26] T. Liu and M. Martonosi. Impala: A middleware system for managing autonomic, parallel sensor systems. In *ACM SIGPLAN Notices*, volume 38, pages 107–118. ACM, 2003.
[27] X. Liu, J. Cao, S. Lai, C. Yang, H. Wu, and Y. L. Xu. Energy efficient clustering for wsn-based structural health monitoring. In *INFOCOM, 2011 Proceedings IEEE*, pages 2768–2776. IEEE, 2011.
[28] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tinydb: an acquisitional query processing system for sensor networks. *ACM Transactions on database systems (TODS)*, 30(1):122–173, 2005.
[29] N. Mohamed and J. Al-Jaroodi. A survey on service-oriented middleware for wireless sensor networks. *Service Oriented Computing and Applications*, 5(2):71–85, 2011.
[30] L. Mottola and G. P. Picco. Programming wireless sensor networks: Fundamental concepts and state of the art. *ACM Computing Surveys (CSUR)*, 43(3):19, 2011.
[31] N. B. Priyantha, A. Kansal, M. Goraczko, and F. Zhao. Tiny web services: design and implementation of interoperable and evolvable sensor networks. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 253–266. ACM, 2008.
[32] J. Radhika and S. Malarvizhi. Middleware approaches for wireless sensor networks: an overview. *Int J Comput Sci*, (9):6, 2012.
[33] J. Rice, K. Mechitov, B. Spencer Jr, and G. Agha. A service-oriented architecture for structural health monitoring using smart sensors. In *Proceedings of the 14th World Conference on Earthquake Engineering, Beijing, China*, 2008.
[34] K. Römer, O. Kasten, and F. Mattern. Middleware challenges for wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(4):59–61, 2002.
[35] I. K. Samaras, J. V. Gialelis, and G. D. Hassapis. Integrating wireless sensor networks into enterprise information systems by using web services. In *Sensor Technologies and Applications, 2009. SENSORCOMM'09. Third International Conference on*, pages 580–587. IEEE, 2009.
[36] T. Sheltami, A. Al-Roubaiey, A. Mahmoud, and E. Shakshuki. A publish/subscribe middleware cost in wireless sensor networks: A review and case study. In *Electrical and Computer Engineering (CCECE), 2015 IEEE 28th Canadian Conference on*, pages 1356–1363. IEEE, 2015.
[37] C.-C. Shen, C. Srisathapornphat, and C. Jaikaeo. Sensor information networking architecture and applications. *Personal communications, IEEE*, 8(4):52–59, 2001.
[38] E. Souto, G. Guimarães, G. Vasconcelos, M. Vieira, N. Rosa, C. Ferraz, and J. Kelner. Mires: a publish/subscribe middleware for sensor networks. *Personal and Ubiquitous Computing*, 10(1):37–44, 2006.
[39] M.-M. Wang, J.-N. Cao, J. Li, and S. K. Dasi. Middleware for wireless sensor networks: A survey. *Journal of computer science and technology*, 23(3):305–326, 2008.
[40] Z. Wang, S. Pakzad, and L. Cheng. Sandwich node architecture for agile wireless sensor networks for real-time structural health monitoring applications. In *SPIE Smart Structures and Materials+ Nondestructive Evaluation and Health Monitoring*, pages 83450L–83450L. International Society for Optics and Photonics, 2012.
[41] Y. Yu, B. Krishnamachari, and V. K. Prasanna. Issues in designing middleware for wireless sensor networks. *Network, IEEE*, 18(1):15–21, 2004.