A Light Weight Service Oriented Architecture for the Internet of Things

Omar Aldabbas
Faculty of Engineering
AL Balqa' Applied University
Al-Salt, Jordan

***Abstract*—Internet of Things (IoT) is a ubiquitous embedded ecosystem known for its capability to perform common application functions through coordinating resources distributed on-object or on-network domains. As new applications evolve, the challenge is in the analysis and implementation of multimodal data streamed by diverse kinds of sensors. This paper presents a new service-centric approach for data collection and retrieval, considering objects as highly decentralized, composite and cost-sufficient services. Such services are constructed from objects located within close geographical proximity to retrieve spatiotemporal events from the gathered sensor data. To achieve this, we advocate coordination languages and models to fuse multimodal, heterogeneous services through interfacing with every service to accomplish the network objective according to the data they gather and analyze. In this paper we give an application scenario that illustrates the implementation of the coordination models to provision successful collaboration among IoT objects to retrieve information. The proposed solution reduced the communication delay before service composition by up to 43% and improved the target detection accuracy by up to 70% while maintaining energy consumption 20% lower than its best rivals in the literature.**

Keywords*—*Internet of Things; wireless sensor networks; sensing services; information extraction; data mining

I. INTRODUCTION

The key function of the Internet of Things (IoT) is to offer users access to information of interest from the big data gathered by smart devices connected over the Internet. Mining IoT data is a multi-phase procedure during which end users apply a structured approach to discover and retrieve the information encapsulated in data [1]. The authors of [2] reviewed IoT data mining approaches and categorized them into three classes based on the elements that motivate information requests; these are event-driven, periodic, and request-response interaction. Based on extensive literature review and investigation, it is clear that implementing one of these above approaches as a tool for mining IoT data is impractical due to its limited ability to extract a full picture of the current state of the ecosystem [2]. Based on this limitation, there is a requirement for a hybrid data mining approach that integrates two or more of the above mentioned data mining classes.

The heterogeneity of operating systems, hardware platforms, sensors and programming models in IoT devices makes data mining a challenging mission. The data streaming nature of some devices further raises the difficulty of implementing an integrated data mining approach. Methods of IoT data mining are also categorized by [3] into three classes. Resource-centric methods are constraint-based aiming at optimizing the usage of the restricted resources at single IoT objects. Nevertheless, the decentralized kind of IoT data makes it hard to extract network-level information by resource centric methods. The second data mining scheme is the network-centric approach. Such methods, unlike the one proposed in [4] [27], are characterized by constrained knowledge of the semantics of the network topology and the application. The third approach is the data-centric that relies on data identifiers and pre-specified object locations. The cost of obtaining data locations, particularly in large-scale applications, restricts the suitability of such approaches. Accordingly, there is a requirement for autonomous hybridization and optimization of data mining techniques for IoT applications. Particularly, there is a need for hybrid autonomous adaptation approach to provision inter-object coordination and communication among different data mining approaches.

The amalgamation of methods from different data mining classes offers a comprehensive solution to optimize information retrieval to current user's needs and adaptations initiating from the aforementioned discussed concerns. This aim is not merely a network self-configuration challenge; the core problem here is the heterogeneity of the data mining approaches and the data extraction and analysis methods. From the perspective of an integrated system behavior, theoretically rudimentary tasks, e.g., data extraction, need multifaceted programming skills, particularly when working with constraint object and network resources. For example, the data streamed by objects cannot be correctly analyzed, and the object cannot communicate with neighbors when the data mining model on each object is different.

Additionally, the majority of the current hybrid data mining methods have intrinsic limitations that constrain their deployment including tailoring for particular applications [5]; have weak spatiotemporal data correlation abilities [5, 6]; utilize high bandwidth and drain power resources [7]; several approaches compromise the quality and amount of extracted information for energy consumption [8]; some approaches do not provide high-level interfaces to the user to configure thresholds and generate queries [5]; and the high dependency among particular data mining approaches -i.e., static composition - applications, protocol stack and hardware platforms preventing making code reuse [5, 6]. The nonexistence of IoT data mining development platforms imposes handling information retrieval from the ground up for each new application. Such problems restrict the applicability of the implemented IoT hybrid data mining approaches, causing them to be complicated to use on anything other than the application developed for.

There is an increasing interest to modify the way for using the limited resources and capabilities in IoT ecosystems, to abstract and simplify them, converting them into communicating services of the network rather than capabilities of individual objects. Commonly, many objects interconnect to respond to a user request, linking their sensing, memory, and processing together. A decentralized query processing approach combines these resources and allocates jobs and data to participating object, so that the effectiveness of the query handling and the quality of the extracted data to be maximized. Consequently, different objects abilities together constitute the capability of the ecosystem as a whole. For instance, when an object equipped with sensors, memory, and processing resources is introduced to the ecosystem, it expands the data resources, on-board processing and memory capabilities to the ecosystem. This object is likely to improve the overall accuracy of the extracted data and reduce information retrieval. Alternatively, if an object leaves the ecosystem, it strips the ecosystem of its data, processing and memory resources.

This research addresses the development of a scalable, adaptive, and energy efficient hybrid IoT data mining approach, which integrates the strengths of three different data mining classes to increase the quality and amount of the extracted information, while reducing resource consumption. This approach embraces a service oriented view for the implementation of modular and adaptive applications. We introduce a service-oriented and semantics layer to the current network stack. The IoT is constructed at various layers of abstraction. At every layer, a group of services is specified. This service-oriented approach implements heterogeneous data mining approaches as lightweight services. To enable uniform associations between different services, located on the same or different objects, the desired technique utilizes "coordination" models [9]. Coordination models offer methods to combine heterogeneous services by interfacing with each service to achieve the application goals using the data they gather and analyze. This integration platform provides the opportunity for quick and accurate information extraction as it allows parallel and asynchronous data processing.

The rest of the paper is organized as follows: Section 2 presents the related works focusing on their limitations. Section 3 gives the architectural components of the proposed service-oriented data mining approach. Section 4 shows the service abstractions on individual objects. Section 5 presents a case study to show how to use coordination rules to compose a hybrid data mining service. Section 6 concludes the paper.

II. Related Work

There has been continuous research into Service-Oriented Architecture (SOA) for highly decentralized systems, such as Internet of Things (IoT) over the last decade [10]. The authors of [12] proposed an SOA-based application development model, a standardized interface to retrieve network data, and a group of configurable service components to provision the implementation of applications and to manage the network behavior at runtime. Another SOA platform, called Atlas, with middle-ware designed around the theory of self-integrative and programmable ubiquitous space [11]. Atlas's service utilities are run on a centralized control server, and power consumption is not a high priority. Blumenthal and Timmermann introduced the Resource Aware Service Architecture (RASA) [13], which structures software modifications by injecting services at runtime. The regular transmission of code results in high communication overhead.

Recently, in [14], a vision of a future IoT system architecture, which is based on service discovery across each layer of IoT is presented. This architecture provides mechanisms for on-demand discovery and integration of devices, cloud storage and computing resources, as well as application integration services, which can be dynamically chosen and orchestrated to create IoT applications. More recently, the authors of [15] an SOA to address the scalability issues leveraging the Path Computation Element (PCE) model. PCE proved to be an efficient technology to separate the control tasks from the sending objects, which has a great impact on scalability growth. For a broad and recent state-of-the-art, we refer the interested to the reader to [16] and the references therein.

The reviewed SOA approaches concentrated on developing architectures for the inter-object collaboration and communication, particularly, remote service access and service orchestration. Nevertheless, there has been little focus on object local service composition.

III. The Service-Centric Framework Details

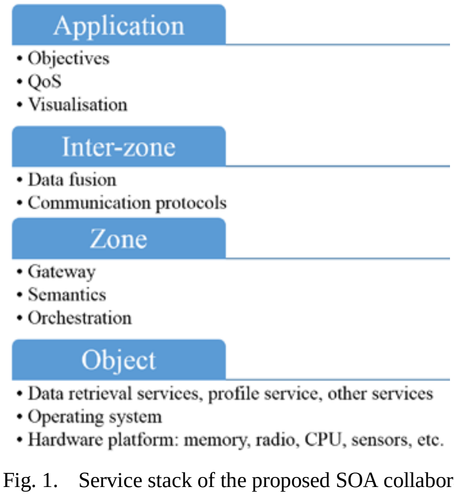
Application
• Objectives
• QoS
• Visualisation
Inter-zone
• Data fusion
• Communication protocols
Zone
• Gateway
• Semantics
• Orchestration
Object
• Data retrieval services, profile service, other services
• Operating system
• Hardware platform: memory, radio, CPU, sensors, etc.

Fig. 1. Service stack of the proposed SOA collaborative platform

In the design of our IoT service-oriented application platform, we build on notions from numerous SOA systems [3] in the domain of Wireless Sensor Networks (WSN) and current approaches for coordinating the functions of large numbers of parallel active objects [17]. In Figure 1, we give a component diagram that shows the service stack of the suggested collaborative platform.

The top layer, Application Layer, designates the application aims or the high-level business processes of the IoT ecosystem. The goal of an IoT ecosystem specifies the information of interest from the user perspective and the kind of services required to retrieve it. The application aims are what defines the nature of data to be extracted, its frequency, scope, QoS, amongst other service or aim specific metrics. Even though the desired platform attempts to minimize the cost of retrieving data, some applications require specific end-to-end quality-of-service guarantees. For instance, for objects that are part of and self-driving car, it is critical that they respond in real-time to ensure the safety of passengers and other road users. Consequently, the desired platform must provide mechanisms to satisfy various QoS needs to ensure suitable utilization of object resources to satisfy the application objectives. The link between application aims and numerous layers and network components can be utilized to accomplish the required QoS. Eventually, data visualization is an essential component to deliver information in a way that end users can easily use and understand. It is also essential for applications to be able to show extracted information in various formats, e.g., maps or graphs, such that the format of information visualization strengthens its structure and content. Querying, executing, and responding to service requests between objects, potentially inter-zone communication. Finally, system managers utilize the profile-building interface as a tool to interact with the Profile services, i.e., to enter data mapping requests, configure coordination rules, and set other application configuration.

The Inter-zone Layer hosts the semantics of macro programs to deliver decentralized data flow. It fundamentally offers multi-hop communication with optional QoS feedback and control. Services at this layer utilize underlying protocols, which describe how services process and parse packets through description metadata instead of coding calls to each other in the source code. Communication protocols improve the interoperability among various services created by numerous providers by messages over specified message channels. This helps in reducing the complexity of the end application, hence, supporting application developers to concentrate on the application features. Most IoT systems utilize programming models that are application dependent, in-network abstractions, which are utilized in data processing. The systems published in [18] and [19] are instances of neighborhood-based abstractions, which deal with many objects cooperatively and a group of operations on it to allow the programmer to retrieve information on the state of the zone. Domain specific requirements, e.g., specific data semantic provision and data fusion, are held by the application layer to support the composition of application-specific information from raw data processing. Data fusion is a technique to enhance bandwidth utilization, power consumption, and information accuracy [20][21]. It integrates and merges inter-zone and multi-sourced data to produce comprehensive and higher accuracy information responses. To generate comprehensive information responses, the data fusion technique fuses data from different zones that are related to the same demand for information.

The Zone Layer is an abstraction layer to permit programmers to handle sensor objects as a spatially distributed entity rather than single isolated objects. This ultimately aims at reducing the amount of required processing and communication operations to retrieve the desired information. The cost of gathering raw data from the network is minimized by choosing a subgroup of sensor objects holding data related to the required information from a bigger group. Discounting extraneous data from an information request handling is also expected to increase the precision of the retrieved information. The Watershed procedure autonomously clusters sensor objects into uniform network clusters based on their topological associations and their soft-state, i.e., sensing-values. Clusters are then utilized as programming abstractions over which various information requests are processed. An information query could end up in creating a group of information retrieval services; everyone involve all active sensor objects in its zone. Such services are formed of a group of services, which fit in multiple zones. Service associated activities are considered as autonomic. At this layer, all zone semantic abstractions are given. Such abstractions are identified as contract expressions. The advantages of design-by-contract are voiced equally in software engineering [23] and programming languages [22] literature. Contracts are transparent to the object's operating system and are used on dataflow paths in a service composition. Composing a complex information retrieval service is regarded as the amalgamation of two distinct operations, the real processing operations with several operations that are part of extracting data and the coordination operations for the collaboration and communication between operations. Consequently, coordination is utilized to distinguish between the processing tasks of asynchronous and decentralized information retrieval from the communication ones, permitting the incorporation of these two key operations. The data extraction element in the Zone Level offers semantics of the sensor data, whereas the data computation elements offer service oriented access to data retrieval services.

A parametric orchestration, which incorporates a set of distinct services, is introduced to allow the system to adapt to various information requests. Every service is autonomously activated by allocating parameters values to meet the required information. For the desired system to function more effectively, a lower layer of access to physical devices is introduced. The lower level purpose is to stop the dependency of information extraction operations on the device operation and communication protocols. Classical layered protocol stack method limits the network software modularity. The majority of the current approaches comprise just the platform and application levels, leading to bounding the software to the attributes of the deployed platform. Therefore, sensor objects with different hardware/software platforms well as diverse methods for data sensing, storage, representations, filtering, processing, and routing cannot cooperate effortlessly. Eliminating the protocol layer helps to reduce the excessive overhead using its composite levels of abstraction and offers a simple method to alter and implement different software elements.

The Object Layer is concerned with the physical characteristics of sensor objects such as their OS, CPU power, memory size, power sources and wireless transceivers. It also includes the actual layout of sensors (physical topology) and network protocols. This layer is accountable for the interfacing with the operating systems and for the administration of resources of sensor objects. Moreover, this layer enables the coordination of physical resource allocation based on the application requirements as defined in the upper layers. Furthermore, services offered by the upper layers will potentially need some resource allocation provision. This level contains many operations related to information retrieval services. It specifies of three fundamental operations, i.e. trigger-based, periodic, and request-response service. Nevertheless, the overall performance, zone or inter-zone, which defines the organization of a hybrid information extraction services is platform-independent and is situated at higher layers of the defined stack. This level provision probing various sensor objects based on the dynamic topological, physical location and logical relationships between the services engaged in the application. The Profile service primary purpose is to log users of an information retrieval service. Generally, the Profile service stores the configuration regarding the utilization of data in a cooperative sensing manner. This configuration is essentially a set of coordination rules, which define the application business logic. Additionally, the Profile service stores other parameters, which are configured manually by system administrators to specify crucial and inherently static settings to help in the composition a complex-hybrid data retrieval service, which is suitable for a specific objective. Such settings cover service clients details (e.g., ID, type), data retrieval configuration (e.g., types of sensed data, parameters of gathered data, time stamp), environmental events, thresholds, etc. In addition to sensing the environment, IoT commonly has task-driven necessities related to hybrid services integration. Services integration necessitates information on the network topology, the state of objects, their position, and density, amongst others. All sensor objects host services to deliver such information.

IV. Transforming the Sensor Object to a Service

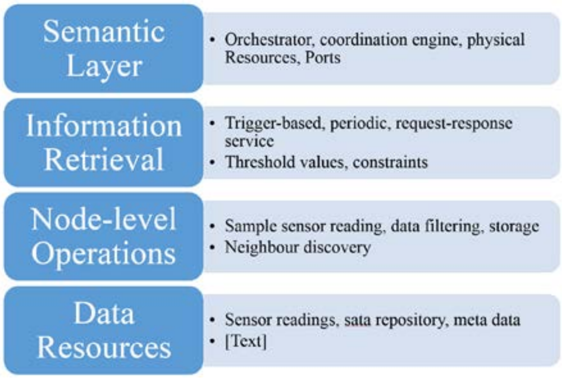
Semantic Layer
• Orchestrator, coordination engine, physical Resources, Ports
Information Retrieval
• Trigger-based, periodic, request-response service
• Threshold values, constraints
Node-level Operations
• Sample sensor reading, data filtering, storage
• Neighbour discovery
Data Resources
• Sensor readings, sata repository, meta data
• [Text]

Fig. 2. An abstract representation of a sensor object as service

Semantic Layer: This is the core layer to transform the sensor object into a service that abstracts its capabilities, resources, hardware and software platform details. At the Semantic Layer, the object retrieves the configured coordination rules, interprets, analyze and apply the coordination constraints to build a complex-hybrid information retrieval services at the individual object scale. At the Zone level, this layer also governs the provisioning of inter-object service composition, i.e., services located on different objects. Any object can advertise its services into one or more logical groups. In practice, service composition is implemented via ports. A port specifies events and commands offered by an object. This layer has parameters, which can be modified to change the behavior of the object. Furthermore, it applies non-blocking concurrency approaches that are investigated in [24]. To avoid any potential issues related to power consumption, clock distribution, and dense object distribution; the author implements asynchronous and asynchronous ports the local and inter-object exchange of information. The asynchronous ports permit remote running of commands on networked objects without disturbing the execution of local object processes, which initiated the remote call. The synchronous ports communications interrupt the execution of local process and pause the process termination on the remote host. The Orchestrator component provides a lightweight orchestration mechanism to provision the composition of services according to the defined coordination rules. Coordination rules define the resources to be assigned at the logical and physical levels and how to orchestrate them to ensure the desired service composition is achieved. The orchestration mechanism is designed to handle static-service-composition because the system administrator manually enters the coordination rules. The Orchestrator utilizes the object-defined services to discover the accessible resources at the local and inter-zone levels.

Information Retrieval: A service encapsulates some function, such as producing data, validating a transaction, or providing simple analytical services. The usability of any service is based on the functions offered through its interfaces, e.g. data extraction, logging, filtering, exchange, etc. Service interfaces are the primary tool that enables services to interact with each other. An information retrieval service wraps a nested functionality, which utilizes an ordered approach to find unstructured data and identify particular patterns encapsulated in the gathered data. In the presented solution, three types of information retrieval services, these are (1) Trigger-based: An object produces a notification message to report events, such as rising temperature or a moving object. At the basic level, a simple event is when a predefined threshold is exceeded. (2) Periodic: An object reports about the monitored environment on regular intervals. The system designer can set the reporting frequency, or it can adapt to application conditions. (3) Request-response: System users issue requests for information using a suitable query langue and application interfaces. Queries can be distributed to network zones that carry data that is relevant to the user query. The query can be issued at regular intervals to gather data in reactive mode about the monitored environment. Queries can provide an effective solution to retrieve data from particular sections of the network.

Object-level operations: In the model advocated in this paper, an object is made up of a set of services, which offer various functions including, an interaction service deals with the exchange of data messages, a fault-tolerance service attempts to heal software/hardware faults, a sensor calibration service to calibrate sensors, and a neighborhood service to discover neighbors and setup zones. The majority of object-based services offer applications and information retrieval services the capability to interact with the object hardware, namely sensors, actuators, battery level or the radio unit, by calling OS primitives.

Data Resources: The object Data Resources layer is a crucial layer of the object as a service stack. At this layer, the utilization of shared data resources is managed and tracked, a log and a register for available resources are upheld, and buffer pools are grouped. Logs and reports are produced at this layer and offered to the whole system. The logs contain events read/write, in sequential order and specify the services which are detecting the event, and the notification messages presents detected events in a concise method to the whole system. IoT ecosystems are characterized by composite data structures because of the multimodal sensing and decentralized applications implemented on heterogeneous objects. To construct an effective service, this layer offers a data abstraction service, which interprets the imperfect data setting. Information abstraction advances the capability to influence data analysis, irrespective of its structure, because of logical and new schema's, which is present in middleware. For instance, the field characteristics of sensing data generated by an object at this layer are ID, Object, Sensor Samples, Time Stamp and Status.

V. Application Scenario of Coordinated Information Retrieval

This section of the paper presents how the primitives of a rule-based grammar can be applied to provide effective coordination between IoT objects to retrieve information. Such primitives are the primary phase on the way to the incorporation of rule-based grammar in IoT objects information retrieval tasks. Moreover, it shows an application scenario, which demonstrates how various information retrieval services can be coordinated. Coordination languages offer means of integrating two or more information retrieval services through interfacing with every service to create a unified service, which can run on a highly decentralized large-scale IoT ecosystem. The author addresses events, which are triggered by several units targets, e.g. a group of soldiers in a battlefield surveillance application. In this IoT ecosystem, an event-based information retrieval service is utilized to notify the periodic-based information retrieval service-driven service to enhance the gathered data accuracy. Objects send their sensor samples periodically to the end user to specify the position of the soldiers. In addition to periodic transmission of data, some objects run a trigger-based service. Once an object senses an event in the form of movement in the monitored environment, it adjusts its periodic information extraction service to escalate the information frequency. Then, the object informs its neighbors in the same logical zone about the event in their region to adapt their periodic data collection services.

In this scenario, the Coordination Language Facility (CLF) [25] is chosen as the coordination layer on top of the IoT ecosystem infrastructure. In CLF, the utilization of rules to manage the operation of different or single vendor services is established on a proactive approach. A CLF system dynamically attempts to effect its environment, instead of just replying to external activities. This characteristic compliments the autonomy of every active IoT object. An IoT object listens for activities to happen in the environment and creates new reports based on the defined rules and their interpretation by its rules engine.

Listing 1: Integrating two information retrieval services using CLF

```
waitForNext @ sample (Motion ) < >- send (Motion ) @ check_activity (Motion )
activity ( normal ) @ interval ( normal ) < >- #b
activity ( normal ) @ i n t e r v a l ( high ) < >- adjust_interval ( normal )
activity ( high ) < >- adjust_interval ( f a s t ) @ inform_neighbour
(Motion , Position , Interval )
received (Motion , position , I n t e r v a l ) @ interval ( normal )
< >- sample (Motion ) @
send (Motion ) @ adjust_interval ( f a s t )
received (Motion , Position , Interval ) @interval ( high ) < >- #b
```

When implementing the above described scenario, the coordination rules are located on and maintained by the individual IoT objects to enable several objects to collaborate autonomously, i.e., such rules apply to individual objects. The waitForNext primitive is a read-only token, which returns a value at the start of every sensing interval; it acts as a trigger to begin the search for an instance of the rule. The sample primitive stores the current motion readings from the motion sensor. The send primitive accepts the sensor sample and transmits it to the higher layer objects (e.g., cluster head or directly to the sink). The check_activity primitive examines the level of soldiers movement or activity within the object's sensing coverage area. If the level of activity, activity(normal), is within normal bounds and the transmission interval is also normal, interval(normal), then the object remains idle and continuous listening. Once the object's present level of activity is normal and the transmission interval is high, interval(high), the object sets the interval back to normal to save its energy, change_interval(normal). If the object detects a high activity within its area of coverage, then it sets its time-driven information retrieval service frequency to quick and sends a notification message to its neighbors informing them about the event in their region notify_neighbour(Motion; Location; Interval). The object includes in this notification message its sensor reading, position, and a new interval value. After an object receives a notification message, received(), it sets its periodic information retrieval service interval to quick and it sends its present sensor reading.

This application scenario demonstrates that coordination rules offer a practical and intuitive approach for integrating services from hybrid information retrieval classes. The coordination engine can manage the objects resources needed for composing a hybrid and distributed information retrieval service. What exactly is being coordinated, how the coordination is achieved, and what are the relevant grammar primitives that must be implemented and utilized, are all questions that we plan to address in the future.

VI. Performance Evaluation

The application scenario described in the previous section has been implemented on a small-scale testbed. The built testbed is used for testing and evaluating the efficiency of hybrid information retrieval service composition. The testbed is made up of six Sun SPOT (Sun Small Programmable Object Technology) nodes and two base stations acting like gateways to the Internet. Sun SPOT is a wireless sensor node for a WSN application prototyping, which was developed by Sun Microsystems. This node uses the IEEE 802.15.4 protocol for its communication, and different from other available hardware platforms, it runs the Squawk Java Virtual Machine. Sun SPOT processor board has ARM architecture 32 bit CPU with ARM920T core running at 180 MHz. It has 512 KB RAM and 4 MB flash memory. A 2.4 GHz IEEE 802.15.4 radio had an integrated antenna. The sensor board comes with an integrated a 3-axis accelerometer, light sensor and temperature. An abstract representation of the testbed is given in Figure 3.

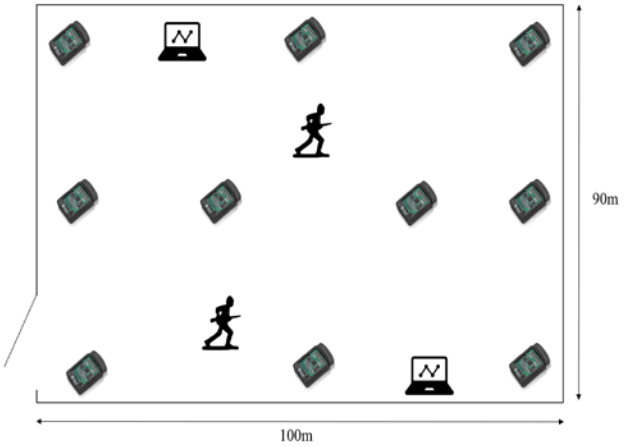

Fig. 3. Testbed layout showing Sun SPOT devices distributions and two laptops connecting the base stations. The running soldier icons determines the location of activity to be captured by the deployed nodes

Due to lack of resources to install motion sensors to detect the location of moving objects, the sensor devices were placed on students tests in a large lecture theater. To generate activity, the lights in the room were switched off and two students were asked to wander in the room using powerful LED torches. The light sensor readings were then used to indicate the presence of a target in the room.

The goal of the experiment was to compose a hybrid service on the run time. This service combines both periodic and event-driven information retrieval approaches. Initially, sensor nodes were operating in the periodic mode with large sensing intervals, i.e., every 3 seconds. When an object is detected, sensor nodes will switch the event driven service to continuously monitor the environment. The event driven service will check any sampled readings against preset thresholds, which were calculated based on the ambient light level in the room.

We compare our proposed approach to EDSOA [26] from the literature as they both offer similar services for the development of service-oriented IoT applications. EDSOA is an information-centric session approach to describe service behavior working upon distributed events, called event session. We define two benchmarks service communication delay and sensing accuracy. Service communication delay is the communication time for two services running on different nodes before a composite service is built.

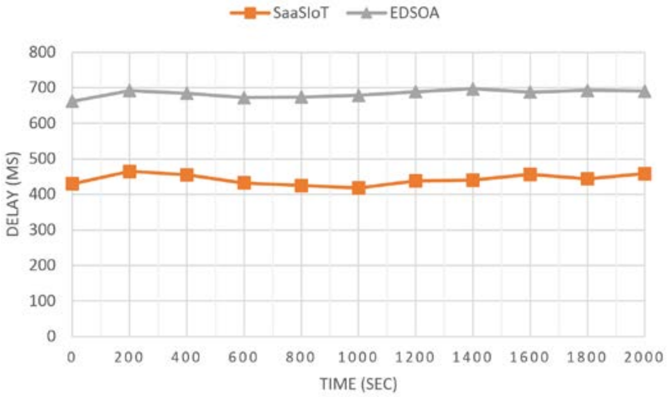

Fig. 4. Communication delay before service composition

Figure 4 shows the communication delay time comparison results. The delay time starts from when an objected is detected by the first node up to the complete service composition. Compared to EDSOA, the communication delay before a service is composed has been significantly reduced. This is mainly due to the fact that the composition rules have been predefined. Even though these rules are static, their application is dynamic. The implementation of the coordination rules was executed in parallel on multiple devices. The devices that are in the same coordination state will be ready to collaborate with neighboring nodes without any delays.

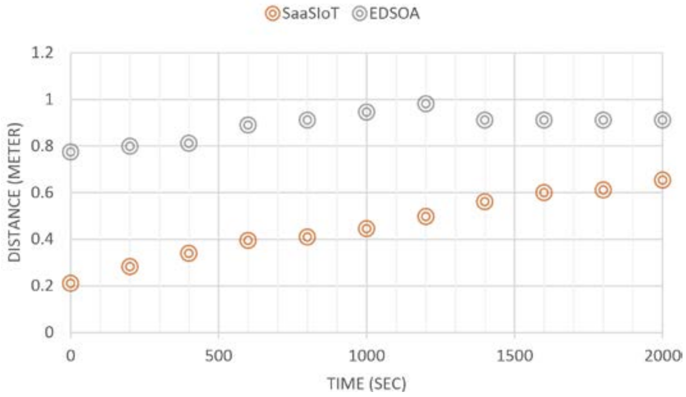

Fig. 5. Target detection accuracy

Figure 5 shows the detection accuracy rate. The coordination based approach has again performed significantly better than EDSOA due to the high levels of coordination between sensing objects. The ability of the composite information retrieval service to switch from one sensing mode to another and dynamically adjust the sensing interval has also contributed to the reduction of target detection localization delay and consequently increases the accuracy of the returned information.

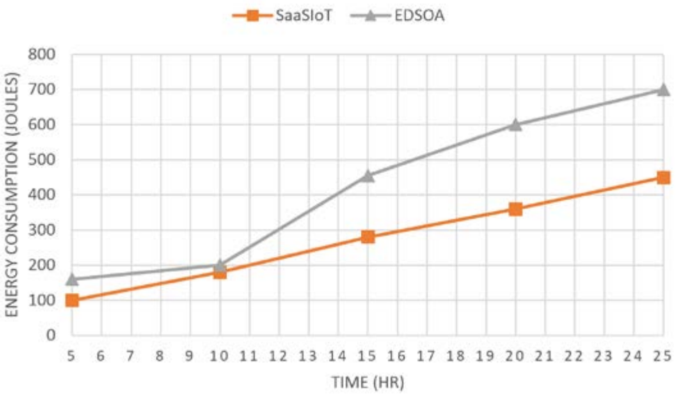

Fig. 6. Energy consumption in Joules

Figure 6 shows the cost of the service composition in Joules. The number of exchanged messages is significantly less than that of EDSOA due to the execution of the coordination rules independently on each device. The outcome of the coordination rules, which is based on the current environmental condition imposes that only the nodes that share knowledge about the target needs to communication and perform collaborative sensing. Due to the nature of the sensed environment, such nodes are normally physically located close to each other. This reduces the bridging distance between devices and results in smaller energy consumption.

VII. Conclusion

This paper attempted to address the challenge of utilizing heterogeneous sensor data to allow collaboration and information extraction from a group of objects at various levels of abstractions. In the presented approach, the logical grouping of objects is a crucial notion for service composition and coordination. Coordination languages can provide a hardware/software platform independent solution for the development of run-time hybrid and effective information retrieval services. We advocate adding a new level of abstraction to group objects with mutual mission restrictions. The proposed solution permits objects to collaborate and support network-level connectivity with a shared state of data and applications.

The contributions of this paper are three folds. First, we present a service stack of the proposed SOA collaborative platform and detail its individual components. Second, we present a practical approach to transforming the IoT object to a service. Finally, we demonstrated the applicability of such an approach using a scenario application implementation in the CLF coordination language. We have shown that the coordination engine can manage the objects resources needed for composing a hybrid and distributed information retrieval service. What exactly is being coordinated, how the coordination is achieved, and what are the relevant grammar primitives that must be implemented and utilized, are all questions that we plan to address in the future.

Acknowledgment

I am thankful for Al-Balqa' Applied University (BAU) for supporting me to do this research.

References

[1] Abuarqoub, A., M. Hammoudeh, and T. Alsboui, An Overview of Information Extraction from Mobile Wireless Sensor Networks, in Internet of Things, Smart Spaces, and Next Generation Networking. 2012, Springer Berlin / Heidelberg. p. 95-106.

[2] Alsboui, T., et al., Information Extraction from Wireless Sensor Networks: System and Approaches. Sensors and Transducers, 2012. 14-2: p. 1-17.

[3] Gracanin, D., et al., A service-centric model for wireless sensor networks. IEEE J.Sel. A. Commun., 2006. 23(6): p. 1159-1166.

[4] M. Hammoudeh, O. Aldabbas, S. Mount, S. Abuzour, M. Alfawair and S. Alratrout, "Algorithmic construction of optimal and load balanced clusters in Wireless Sensor Networks," Systems Signals and Devices (SSD), 2010 7th International Multi-Conference on, Amman, 2010, pp. 1-5.

[5] Lee, B.-D., Adaptive Data Dissemination Protocol for Wireless Sensor Networks, in Security-Enriched Urban Computing and Smart Grid. 2010. p. 188-195.

[6] Lee, C.-H., C.-W. Chung, and S.-J. Chun, Effective processing of continuous group-by aggregate queries in sensor networks. J. Syst. Softw., 2010. 83: p. 2627-2641.

[7] Bhargavi, R., et al., Complex Event Processing for object tracking and intrusion detection in Wireless Sensor Networks, in Control Automation Robotics Vision (ICARCV), 2010 11th International Conference on. 2010. p. 848 -853.

[8] Bahrepour, M., Meratnia, N. and Havinga, P.J., 2009, July. Sensor fusion-based event detection in wireless sensor networks. In Mobile and Ubiquitous Systems: Networking & Services, MobiQuitous, 2009. MobiQuitous' 09. 6th Annual International (pp. 1-8). IEEE.

[9] Abreu, J.a. and J.L. Fiadeiro, A coordination model for service-oriented interactions, in Proceedings of the 10th international conference on Coordination models and languages. 2008, Springer-Verlag: Oslo, Norway. p. 1-16.

[10] Hammoudeh, M.; Newman, R.; Dennett, C.; Mount, S.; Aldabbas, O. Map as a Service: A Framework for Visualising and Maximising Information Return from Multi-ModalWireless Sensor Networks. Sensors 2015, 15, 22970-23003.

[11] King, J.C., Atlas: a service-oriented sensor and actuator network platform to enable programmable pervasive computing spaces. 2007, University of Florida: Gainesville, FL, USA.

[12] Delicato, F., et al., Exploiting Web Technologies to Build Autonomic Wireless Sensor Networks, in Mobile and Wireless Communication Networks. 2006, Springer Boston. p. 99-114.

[13] 13.Blumenthal, J. and D. Timmermann, Resource-Aware Service Architecture for Mobile Services in Wireless Sensor Networks, in Wireless and Mobile Communications, 2006. ICWMC '06. International Conference on. 2006. p. 34.

[14] D. Georgakopoulos, P. P. Jayaraman, M. Zhang and R. Ranjan, "Discovery-Driven Service Oriented IoT Architecture," 2015 IEEE Conference on Collaboration and Internet Computing (CIC), Hangzhou, 2015, pp. 142-149.

[15] V. B. C. Souza, X. Masip-Bruin, E. Marin-Tordera, W. Ramírez and S. Sánchez-López, "Towards the scalability of a service-oriented PCE architecture for IoT scenarios," Networks and Optical Communications - (NOC), 2015 20th European Conference on, London, 2015, pp. 1-6.

[16] Vanitha, V., V. Palanisamy, and K. Baskaran, Automatic Service Graph Generation for Service Composition in Wireless Sensor Networks. Procedia Engineering, 2012. 30(0): p. 591 - 597.

[17] Hammoudeh, M., Newman, R., Dennett, C. and Mount, S. (2013), Interpolation techniques for building a continuous map from discrete wireless sensor network data. Wirel. Commun. Mob. Comput., 13: 809–827. doi: 10.1002/wcm.1139

[18] Newton, R., G. Morrisett, and M. Welsh, The regiment macroprogramming system, in Proceedings of the 6th international conference on Information processing in sensor networks. 2007: Cambridge, Massachusetts, USA. p. 489-498.

[19] Mottola, L. and G.P. Picco, Programming wireless sensor networks with logical neighborhoods, in Proceedings of the first international

conference on Integrated internet ad hoc and sensor networks. 2006, ACM: Nice, France.

[20] Hammoudeh, M., R. Newman, and S. Mount, An Approach to Data Extraction and Visualisation for Wireless Sensor Networks, in Proceedings of the 2009 Eighth International Conference on Networks. 2009, IEEE Computer Society. p. 156-161.

[21] Mohammad Hammoudeh, Robert Newman, Information extraction from sensor networks using the Watershed transform algorithm, Information Fusion, Volume 22, March 2015, Pages 39-49, ISSN 1566-2535, http://dx.doi.org/10.1016/j.inffus.2013.07.001.

[22] Findler, R.B. and M. Felleisen, Contracts for higher-order functions, in Proceedings of the seventh ACM SIGPLAN international conference on Functional programming. 2002, ACM: Pittsburgh, PA, USA. p. 48-59.

[23] Meyer, B., Applying "Design by Contract". Computer, 1992. 25(10): p. 40-51.

[24] Mount, S., et al., CSP as a Domain-Specific Language Embedded in Python and Jython, in CPA'09. 2009. p. 293-309.

[25] Andreoli, J.-M., S. Freeman, and R. Pareschi, The Coordination Language Facility: coordination of distributed objects. JOURNAL OF THEORY AND PRACTICE OF OBJECT SYSTEMS (TAPOS, 1996. 2: p. 1-18.

[26] Yang Zhang, Li Duan, and Jun Liang Chen. 2014. Event-Driven SOA for IoT Services. In Proceedings of the 2014 IEEE International Conference on Services Computing (SCC '14). IEEE Computer Society, Washington, DC, USA, 629-636. DOI=http://dx.doi.org/10.1109/SCC.2014.88

[27] M. Hammoudeh, J. Shuttleworth, R. Newman and S. Mount, "Experimental Applications of Hierarchical Mapping Services in Wireless Sensor Networks," Sensor Technologies and Applications, 2008. SENSORCOMM '08. Second International Conference on, Cap Esterel, 2008, pp. 36-43.