Proc. IEEE Percom Demos, San Diego, CA, March 22, 2013, pp. 312-314

Autonomous Configuration of Spatially Aware Sensor Services in Service Oriented WSNs

S. Yousaf Shah*, Boleslaw Szymanski*, Petros Zerfos†, Chatschik Bisdikian†, Christopher Gibson‡, Dominic Harries‡

*Rensselaer Polytechnic Institute, Department of Computer Science, Troy, NY, USA {shahs9@rpi.edu, szymab@rpi.edu}
†IBM T.J. Watson Research Center, Yorktown Heights, NY, USA {pzerfos@us.ibm.com, bisdik@us.ibm.com}
‡ IBM Hursley, UK {christopher.gibson@uk.ibm.com, dharries@uk.ibm.com}

***Abstract*—Service-oriented Architectures (SOA) for Wireless Sensor Networks (WSNs) are an active research topic. Yet, autonomous configuration of services for real life constraints (spatio-temporal, input/output interoperability, policies, security etc.) is still a challenging problem. In this demonstration we describe the results of our research into the automated and intelligent configuration and composition of services for complex tasks. We present a service-oriented system capable of performing service configuration under spatial and relevancy constraints. It can configure services in one of the three following modes: *distributed*, *centralized* and *hybrid*. It also supports automatic reconfiguration in the event of service failures. This system uses a generic cost representation for services that may include spatial coverage of the services in an area of interest along with other service configuration cost metrics. We demonstrate our system using state-of-the-art emulation frameworks with a real life scenario.**[1]

I. Introduction

Service configuration in pervasive wireless sensory systems (WSNs) is quite challenging as the requirements of the applications hosted on WSNs change over time and these changes must be reflected in the system configuration. As events in WSNs (e.g., a node fails, a service becomes unavailable on a node, etc.) happen over time the configuration mechanism should dynamically reconfigure the system according to the new requirements. An efficient configuration mechanism should be able to configure services in a way that their inputs and outputs are interoperable to perform a complex task. Moreover, the selection of services should be made by taking into account various constraints (expressed as policies) and different performance metrics. Typically, configuration mechanisms ignore the utility that a certain service brings to the overall system configuration and consider only the flat cost of using a service. Suppose a WSN is deployed as a support system for a disaster relief effort. A monitoring system configured in such a scenario might use audio and video feeds produced by other services to provide surveillance of the area to the effort coordinators. The service configuration in such a scenario should not only consider input/output portability [1] but also other factors, such as energy cost and spatial relevancy of services to the area of interest. In such a scenario, services that are more relevant (e.g., have a larger sensing range in the mission area) are more useful than services that provide the same outputs but with lower relevancy. In this demonstration we present the configuration of services incorporating such spatial relevancy [2] in the solution. Our system considers spatial constraints on service selection, and configures the system by choosing low-cost and spatially relevant services improving the spatial relevancy of the overall system. We show that our proposed mechanism is tolerant to failures, i.e., in the case of failures the system automatically reconfigures using the most relevant alternative services available. Following are the main features of the system:

- A novel self-recovering and fault tolerant mechanism for the configuration of services with spatial and other policy constraints.
- Centralized, distributed and hybrid service configuration mechanisms.
- A generic cost mechanism for services.
- Ability to ensure service configuration compliance with policy constraints.

II. Overview

A. Service Configuration with Spatial Relevancy

Relevancy of a service to the area of interest plays a major role in the configuration of WSN services. Therefore, we aim to configure services that are both low-cost and highly spatially relevant to the required composite service. However both of these problems are NP-hard ([2], [1]). Although one is a minimization problem and the other is a maximization problem, we model both as a single minimization problem and apply a Set Cover heuristic to find the minimum cost service composition. We incorporate the relevancy aspect in the configuration of a service via a generic cost function. In a service configuration, any use of a particular service incurs some cost to the hosting node. This cost can be flat, such as energy consumed, or a combination of factors such as edge delay, battery consumption and processing time costs;

[1]Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

we refer to such a cost as the *BaseCost*. Every sensor service has a *BaseCost* associated with it, which is incurred when the service is used. In case the user is not interested in the spatial relevancy of the service (which can be specified by the user in the request) our system will configure the system aiming at minimizing the *BaseCost*. We introduce a generic cost function that incorporates both the *BaseCost* and the *RelevancyCost*. The latter cost represents the irrelevancy of a sensor service to the users requested area of interest. The equation 1 shows the aggregated cost (*AggrCost*) incurred when a sensor service is used

$$AggrCost = \alpha \times BaseCost + \beta \times (1 - Relevancy(\Gamma_x)) \quad (1)$$

where α and β are the user defined weights balancing the impact of the *BaseCost* and *RelevancyCost*. Both the relevancy cost and base cost are normalized to the same range before the aggregated cost is calculated. We then use heuristics of the Set Cover problem to find the minimum cost composition. The cost function based on the specified weights maximizes the covered area while minimizing the base cost. In this work, we consider relevancy as the overlap between the area of interest and the area covered by the measurement of a sensor. The measurement from sensors can be any kind of information collected from the environment. For simplicity, in the demonstration we model the area covered by a sensor as disk of radius r around the sensor location, where the location of the sensor is defined by latitude, longitude and altitude. We also assume that the user specifies area of interest is disk of radius R defined in the request for configuration. Our system is not restricted to a disk coverage model; we have developed an extensive library for various coverage models (hexagonal, polygon etc.) that are easily pluggable into the system according to the formulation of coverage areas and overlapping regions. The relevancy of a service x is represented by equation 2.

$$\Gamma(x) = \frac{|C_R \cap c_{r,x}|}{|C_R|} \quad (2)$$

where C_R denotes area of interest with radius R and $c_{r,x}$ denotes the area of coverage of service x with radius r.

B. Modeling Configuration Constraints in Controlled English

The service composition process may be further constrained by requiring that the component services comply with a set of policies. We utilize policies written in ITA Controlled English (CE) [3]. CE is a controlled natural language defining a human-friendly (human-readable) language that is unambiguous for computers, whilst allowing the definition and expression of concepts, rules and relationships. We have created a CE domain model which allows service composition policies to be expressed as CE rules. These rules can make use of attributes that have been defined for services. For example, the following rule excludes services that have a UK affiliation from a composition:

```
if ( the service S is affiliated to
 the organization UK )
then ( the authorizer A produces
 the decision deny ).
```

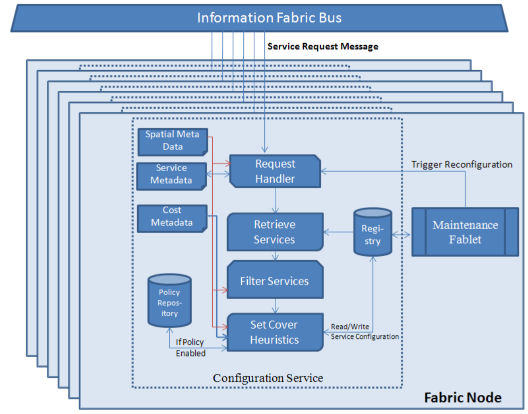

Fig. 1. Node level information flow

In addition to rules supporting authorization, the system also supports rules expressing obligations.

C. System Design and Implementation

The system has been developed in Java using the ITA Information Fabric [4], a SOA-based middleware for sensor networks. A sensor network built using the Information Fabric consists of a set of fabric nodes, each of which manages a set of assets and offers a set of services (including composite services). The fabric is a fully distributed infrastructure, and it is the federation of the fabric nodes that forms a service bus across the WSN. The fabrics registry of assets and services may be fully distributed, and a dynamic distributed federated database technology the Gaian Database [5] is used to provide a single consistent view of the full set of assets and services available. Service configuration in the system has been designed for extensibility. New components (e.g., a new coverage model, a more elaborate cost function, an additional system checker component) can be easily plugged into the system.

Figure 1 shows how each node handles a configuration request. After the request handler receives a request, the system retrieves services from the fabric registry. Services that do not provide coverage in the requested area of interest are filtered out. Then, a Set Cover heuristic is applied to the services based on the cost function in (1). If any policies are to be enforced (e.g., the user has requested hard configuration constraints) the corresponding policies are fetched from the *Policy Repository*. Only those services that fulfill the policy requirements are selected for composition. If the policies cannot be satisfied the service is not configured, otherwise the final service composition graph is written back to the registry.

A background service runs to perform periodic checks on the system. In the case of any service failing, a recomposition is triggered for all the services dependent on that service.

The system is capable of performing service configuration in three modes: centralized, distributed and hybrid. In the centralized mode of operation, a central node performs configuration of all the services hosted on fabric nodes. In distributed

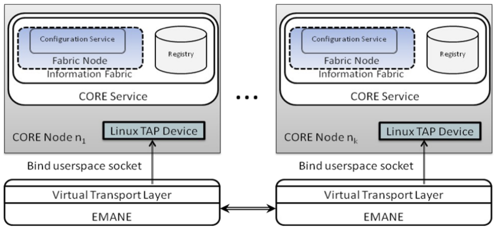

Fig. 2. Distributed mode of operation

mode, every service configures itself, so there is no centralized configuration node. In hybrid mode, a subset of nodes act as configuration nodes each of which configures a portion of the total services.

To improve the fidelity of the demonstration in distributed mode two technologies are used to emulate a WSN: the Common Open Research Emulator (CORE [6]) and the Extendable Mobile Ad-hoc Network Emulator (EMANE [7]). This enables physical network nodes to be emulated such that they run as separate processes with independent network stacks. Figure 2 shows the abstract architecture of the distributed version of the service configuration system. As shown, the fabric node runs as a CORE service inside a CORE node. Each CORE node runs inside an independent Linux container and is integrated with the EMANE. Thus, all the data link and physical layers are emulated by EMANE.

III. Demonstration

In the demonstration, we present the autonomous system configuration described above. Complex configuration details are abstracted by a graphical user interface (GUI) that is accessible through a desktop system or mobile device (e.g., an iPad, iPhone or Android phone). We demonstrate our system by simulating a real life scenario described below.

A. Scenario

A user is interested in a composite service that is capable of monitoring an area for activity and visualizing the location of events. The composite service uses component services such as a camera service, a set of acoustic detection services and a localization service to geolocate the event.

B. Service Configuration for the Scenario

Our demonstration shows the overall capability of the system and its resilience. The user is provided with a map interface showing the area of interest with services hosted at different locations (all connected via the fabric). Using the GUI, the user encircles the area of which activities (s)he wishes to monitor. The system automatically selects services that maximize coverage in the area of interest and filters out any unrelated services. After selecting appropriate services, the system configures those services and links them together to create a composite service that performs the task. The CE policies that govern the service configuration can also be displayed, edited and deployed, potentially resulting in different recomposition. Figure 3 shows a GUI displaying

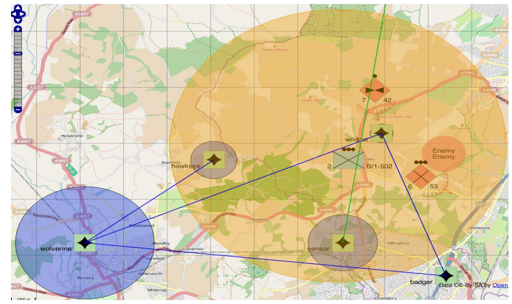

Fig. 3. GUI of service configuration showing services and their interaction

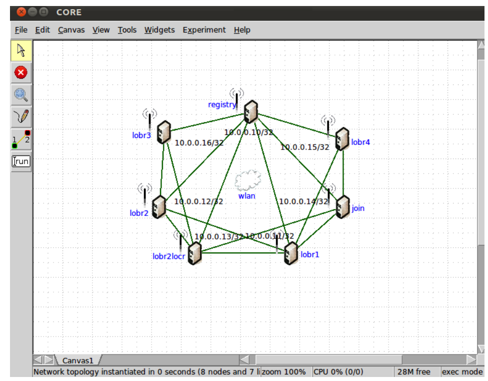

Fig. 4. Network level view of the emulated distributed service configuration

services; the links between services denote the wiring between services.

Distributed operation of the system uses *CORE* to visualize the emulated WSN, and a browser-based visualization utility is used to show the service level graph of the configured services. Using the *CORE* GUI (as shown in Figure 4), the user can disconnect any node from the rest of the system to introduce failure. The system detects the failed node and reconfigures itself to provide an alternative composite service that meets the needs of the user.

C. Technical Requirements

The demonstration will require 2 external monitors, space for 2 laptops and monitors and 5 power sockets.

References

[1] S. Geyik, B. Szymanski, and P. Zerfos, "Robust dynamic service composition in sensor networks," *Services Computing, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2012.

[2] G. Tychogiorgos and C. Bisdikian, "Selecting relevant sensor providers for meeting your quality information needs," in *Mobile Data Management, 2011 12th IEEE Int'l Conf. on*, 2011.

[3] D. Harries, D. Braines, and C. Gibson, "Towards an expression of policy in controlled english," in *5th Ann. Conf. Int'l Techn. Alliance*, 2012.

[4] J. Wright, C. Gibson, F. Bergamaschi, K. Marcus, R. Pressley, G. Verma, and G. Whipps, "A dynamic infrastructure for interconnecting disparate isr/istar assets (the ita sensor fabric)," in *Information Fusion, 2009. FUSION'09. 12th Int'l Conf. on*, 2009.

[5] G. Bent, P. Dantressangle, D. Vyvyan, A. Mowshowitz, and V. Mitsou, "A dynamic distributed federated database," in *2nd Ann. Conf. Int'l Techn. Alliance*, 2008.

[6] J. Ahrenholz, "Comparison of core network emulation platforms," in *MILCOM 2010*, 2010.

[7] [Online]. Available: http://http://labs.cengen.com/emane/index.html