Dynamic Decision Support in Direct-Access Sensor Networks
A Demonstration

Joachim Hammer[1], Imran Hassan[1], Christine Julien[2], Sanem Kabadayı[2],
William J. O'Brien[3], and Jason Trujillo[2]
[1]University of Florida, Department of Computer and Information Science and Engineering
[2]University of Texas, Department of Electrical and Computer Engineering
[3]University of Texas, Department of Civil, Architectural, and Environmental Engineering

***Abstract*—This paper describes application demonstrations of a new middleware that supports dynamic decision support over networks of resource-constrained devices. For the purposes of our demonstrations, we tap into *intelligent job site* applications for the construction domain. The demonstrations are carefully constructed to highlight our middleware's ability to provide on-demand access to local data, aggregation of data across dynamically defined regions, fusion of heterogeneous sensor data, and intelligent application-sensitive sensor clustering. The paper briefly describes the middleware and the details of the demonstrations.**

I. Introduction

Several technological trends such as the miniaturization of computing power and the creation of small low-cost environmental sensors are driving the evolution of ubiquitous computing environments that are quickly becoming integral to existing application domains. These developments enable context-aware applications that allow users to directly access context information collected within their local environment. Application domains likely to be able to immediately benefit from such advances include intelligent construction sites, war-fighter coordination, and first responder deployments. All three domains have similar information needs (e.g., decision support for command and control, information fusion to integrate multiple data streams, simple visualization of information with drill-down capability, etc.), and participants in all three domains use small, resource-constrained devices that can function independent of a heavy-weight infrastructure.

We have developed a middleware framework and algorithms enabling ad hoc integration of sensing data for local decision support. Our framework connects users (e.g., war-fighters in the operational theater, responders in an emergency, construction workers in a dynamic and unsafe environment) directly to sensors in the local environment. Existing approaches, in contrast, commonly treat sensor networks as static arrays for data collection where the entire network has a common task, and sensing devices forward information to a single collection point where the data is accessed [4, 5]. While some processing can occur in the network [3], the goal is usually filtering and aggregation to improve efficiency. We focus on a significantly altered computational model in which environmental information is accessed locally in real time. As a result, our middleware can be deployed on small devices, operates independently of heavyweight infrastructure, and provides on-demand, real-time connection to information that enables users to complete their tasks quickly, safely, and with the best possible information.

The remainder of this paper is organized as follows. Section II describes the application domain we use for our demonstrations. Section III overviews the capabilities of our middleware, while Section IV describes the demonstrations themselves. Section V concludes.

II. An Application Domain

The *intelligent job site* [2] as envisioned for construction presents considerable challenges to pervasive computing, requiring both context-awareness and multiple decision support applications for physically distributed (and mobile) devices. The intelligent job site in Fig. 1 shows users, mobile devices, and sensors connected by a wireless network. A job site office in the upper left has access to archival data and enables centralized decision support applications (such as overall materials and trade coordination). Worker B carries a portable computing device that can interact with distributed sensors. For example, worker B can read an RFID tag with materials information (about the palette

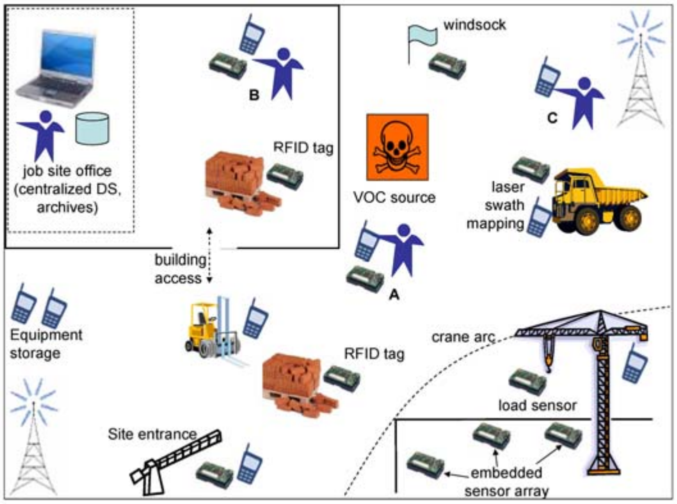

Fig. 1: The Intelligent Job Site

of bricks) and record its location. Outside the building, worker A nears a VOC (Volatile Organic Compound) source; worker A's sensor can monitor the level of the hazardous material and use the network to warn others if the concentration surpasses a threshold. A wind sock can provide wind information to augment this notification.

Also depicted is a dump truck with a laser swath mapping device used for collision detection. Other onboard devices provide the truck's speed, turning radius, etc. A similar application is shown in the lower right, where a crane has a load sensor that checks if the crane is in danger of exceeding its capacity. A computing device in the cab can also interact with devices on workers and vehicles to check if they are within the arc of the crane, significantly aiding safety maintenance. Also near the crane is an embedded array of sensors for structural health monitoring (these could be heat sensors that measure concrete cure rates or stress/strain sensors in walls or floors). Finally, the lower left shows a site entrance with various RFID readers to record materials deliveries (and on large sites, to provide directions to trucks as they enter the site). The entrance can also provide security, track worker entry and monitor tools and equipment.

III. The Middleware

To support the local data needs of ubiquitous computing, we have created a distributed software architecture that enables domain experts to create customizable decision support for dynamic networks involving resource-constrained sensing devices. Specifically, our middleware provides the following four capabilities not found in today's sensor network coordination frameworks:

- *On-demand local data access*: Contrary to existing applications, the "clients" in our targeted scenarios are immersed in the sensor network. As such, our middleware includes new communication and query protocols that allow users to *opportunistically* interact with locally available sensors.
- *Regional aggregation*: Emerging applications must intelligently aggregate data from a particular (dynamically changing) region. This has required our middleware to include novel algorithms for group communication and aggregation.
- *Expressive sensor fusion*: Sensor network aggregation commonly operates on single data types and generates measures like average, minimum, etc. When users are immersed in a network, their data needs include requests for abstracted information, and we have developed novel ways to place some of this abstraction burden in the sensor network using lightweight *sensor fusion*.
- *Intelligent, application-dependent sensor clustering*: Finally, to support users' queries, an application-dependent set of sensors may, for a particular application, need to dynamically organize themselves to ensure that all related sensors are included. Our middleware includes algorithms that assist applications in creating these dynamic groups.

To address these requirements, our middleware is separated into three layers of abstraction, as depicted in Fig. 2.

In the uppermost layer, the Decision Support Layer (DSL), applications define chunks that specify application-level behaviors that require interaction with

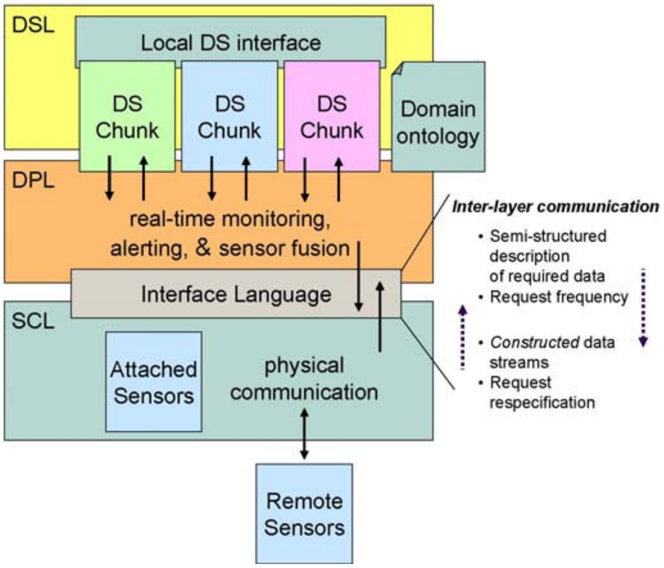

Fig. 2. A three-tiered architecture

the pervasive sensor network. Each of the three applications described in Section IV is represented by a single chunk. Chunks may use information about data types from within the domain by accessing the *domain ontology*, which is shared with the middle layer, the Data Processing Layer (DPL).

In the DPL, applications' chunk specifications are decomposed into one or more *query plans* that translate from types available in the domain ontology to types available from the underlying physical sensors. This may include dynamically creating sensor fusion software components to, for example, create a crane load sensor out of multiple physical sensors attached to a crane, as described in Section IV. In addition to containing knowledge about the types of data required to satisfy a request, the query plan also extracts from the chunk information about constraints on that data and information about simple aggregation to be performed on the data. Constraints on the data may include, for example, that the data is within a certain distance of requestor's device, the data is from a particular area, the data is no older than a specified time, etc. Also, queries from the DPL can be either *one-time* (requiring just one reply from the lower layer) or *persistent* (requiring periodic responses from the lower layer). All of this information is packaged together into simple queries that are handed to the lowest layer, the Sensor Communication Layer (SCL). When a response to a query is returned from the SCL, depending on the result and the particular chunk, the query plan may dispatch additional queries.

While the DPL provides the sensor fusion and intelligent clustering described above, the SCL has no access to domain knowledge. This is beneficial because it means the SCL is completely reusable across domains, but it restricts the types of functions for which the SCL can be responsible. Even so, given an appropriate amount of information from the DPL, the SCL can perform efficient on-demand local data access using the regional or logical constraints on where the data should come from. In addition, the SCL can provide filtering and aggregation in non-domain specific ways. Ultimately, responses from queries in the SCL find their way back through the DPL to the DSL and the application. In the next section, we provide specific descriptions of the application chunks that demonstrate different portions of the middleware's capabilities.

IV. Application Scenarios: The Demonstrations

To demonstrate the novel features of our middleware, we have created three application examples from the domain of the *intelligent construction site*. The following application demonstrations connect a set of Crossbow mica2 motes [1] to a handheld Windows Mobile device. We do not demonstrate these applications on a full-blown construction site. Instead, we simulate the situation using the same hardware (sensors and handhelds) intended for use on real-world construction sites. Some of the data driving the demonstrations (e.g., changing strain on a trench wall or wind) is pre-generated. Other data (e.g., gas concentration or crane movement) is simulated in real time. This dynamic data can be changed in the demo to show different situations. Since the middleware instead of the application is the real contribution of the work, our demonstrations highlight specific middleware functions at different points. The application examples are:

A. *Gas cloud detection*

Hazardous materials on job sites can cause significant danger if leaks are not properly detected. In this scenario, the construction worker's handheld device has a software chunk whose behavior is specified as monitoring the concentration of hazardous gas in the nearby area, and, if a threshold is surpassed, triggering a warning to the user and retrieving wind information to calculate the gas's potential spread. The query plan devised by the DPL first registers a *persistent query* for hazardous gas concentrations exceeding a certain threshold, and this persistent query is passed to the SCL which handles all of the filtering in this example. When the sensor detecting gas concentration (which, for the purposes of the demonstration, we simulate with a temperature sensor and a hair dryer) senses a value over the specified threshold, a reply is automatically sent to the DPL on the handheld. The query plan indicates that, under such a condition, a second query should be generated to retrieve the nearest wind data (to calculate the gas cloud's potential movement). The worker also receives a warning to evacuate. In comparison to existing solutions, this example demonstrates the feasibility of easily expressing temporally dependent queries in our middleware, and pushing filtering tasks to the sensors themselves, reducing the communication overhead associated with persistent monitoring.

B. *Crane safety circle*

In this scenario, the user of the handheld device requests a picture of the region around the base of a crane where it is unsafe to walk or drive. In this case, the formulation of the query plan within the DPL is complicated due to the fact that it must use information from the domain ontology to understand what kinds of sensors are necessary to construct the requested abstracted information. The DPL then passes simple queries to the SCL to dynamically discover sensors attached to components of nearby cranes (e.g., the base of the crane, the arm of the crane, the load on the crane, etc.). The middleware uses information from the domain ontology to determine how to combine the information collected from these distributed sensors to calculate the requested region. The DPL subsequently registers persistent queries on these particular sensors and remains connected to them. As the SCL generates and sends updates, the DPL refreshes the presentation of this region and displays the changes to the application. This example performs on-the-fly heterogeneous data fusion from multiple sensor streams *in the field*, which has not been performed in existing solutions.

C. *Trench strain*

Another safety situation that arises on construction sites results from the collapse of trenches as they are dug. In this scenario, wireless sensors are placed in a trench wall as workers dig the trench. Workers with handheld devices can specify behaviors that create a map of the strain within the trench at different locations relative to the workers. In this case, the chunk first requests a global map of the strain values in the trench. The DPL creates a simple query plan that sends a one-time query to all strain sensors in the region to retrieve their strain values, which are mapped for the user to view. The chunk then allows the user to select a subset of these strain sensors to monitor more persistently, and the DPL takes this information and registers persistent queries on specific sensors as selected in the chunk. This enables significant energy savings by not requiring every sensor to communicate all of the time. Future work within our middleware will attempt to generalize this style of interaction that uses only some fraction of sensors that satisfy a certain type while trying to distribute the sensors selected evenly across some region. When an anomalous condition is detected by one of the persistent queries, the strain chunk automatically creates new queries that wake up other strain sensors in the same area to collect a complete view of the situation before issuing the necessary warnings. This demonstration specifically highlights our middleware's ability to provide dynamic regional aggregation *in the network* using our novel communication protocol.

V. Conclusions

This paper has presented a new approach to creating decision support applications for sensor networks in which immediate access to local data is a fundamental requirement. We have developed a middleware that embodies a suite of abstractions that simplifies programming of such applications. This paper briefly introduced the middeware and then focused on how a set of application demonstrations designed for an intelligent construction site that emphasize the novel characteristics of our middleware which include on-demand access to local data, the ability to aggregate sensor readings based on regions, fusion capabilities defined on-the-fly in an application-dependent manner, and the definition of application-specific clusters of sensor nodes. The middleware has the potential to enable domain programmers to rapidly create expressive applications without having to directly handle the complexities associated with distributed programming.

References

[1] Crossbow Technologies, Inc., http://www.xbow.com, 2006.

[2] Fiatech, "Strategic Overview: Capital Projects Technology Roadmap Initiative (version 1)," http://www.fiatech.org/projects/roadmap/cptri.html, 2003.

[3] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed Diffusion for Wireless Sensor Networking," *IEEE/ACM Transactions on Networking*, 11(1):2–16, 2003.

[4] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An Analysis of a Large-Scale Habitat Monitoring Application," in *Proceedings of SenSys*, pages 214–226, 2004.

[5] N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A Wireless Sensor Network for Structural Monitoring," in *Proceedings of SenSys*, pages 13–24, 2004.