

# Human-Computer Interfaces for Sensor/Actuator Networks

Lawrence Henschen<sup>(✉)</sup> and Julia Lee

Northwestern University, Evanston, IL, USA

henschen@eecs.northwestern.edu, julialee@agep.northwestern.edu

**Abstract.** In this paper we study the features needed for a generic but robust user interface for monitoring and controlling sensor/actuator networks. The need for such a generic interface will increase as the Internet of Things develops and reaches not only major industries but small enterprises that do not have the resources to develop such systems on their own. We propose a general mark-up language that can be used as the basis for a generic interface for sensor/actuator networks. We present an initial list of features that such an interface should have and then describe a set of tentative mark-ups which, with the use of XSLT and other XML-based technology, could lead to a generic interface that could be used for any sensor/actuator network. Because many sensor/actuator nodes and related devices may not have XML capability, we also address the technical issue of where in the pipeline the appropriate marked-up text should be inserted.

**Keywords:** Sensor networks · Sensor network interface · Mark-up language

## 1 Introduction

Sensor networks are becoming increasingly important in modern society. This trend will increase dramatically as the Internet of Things becomes a reality. While much attention has been paid to the underlying technologies, such as general communications and networking, relatively little attention has been devoted to the issue of a generic human-computer interface for sensor/actuator networks in general. There have been some interfaces for sensor/actuator networks described for specific applications, for example [1–3] and others. However, no generic solution that can be used directly as a convenient and comprehensive interface has been put forward. This is an important need that must be met in the future of the Internet of Things because more and more sensor/actuator networks will be used by small-to-medium enterprises that do not have resources to develop the networks and interfaces themselves and yet should not have to buy services from providers. The goal of our research is to develop a generic interface that any enterprise, large or small or even an individual, can download and easily use to describe and then use its own sensor/actuator network. Moreover, the deployment and use of the interface should not require technical expertise in HCI or networking. Ordinary users with no training in HCI or XML or networking should be able to describe, monitor, and control their sensor/actuator networks.

The development of a generic user interface depends on two key aspects - (1) the use of a standardized mark-up language to represent information about the sensors,

actuators, network, and the data transmitted and (2) a platform independent system using the mark-up language along with XSLT to extract relevant information and UIML to implement the actual interface. This paper introduces both of those topics and provides an initial list of features that such a generic interface should have.

There are, in fact, two separate interfaces that are needed to make a truly generic system that can be both configured and used by people with limited backgrounds in HCI, XML, computer networks, etc. The first interface would be used when the network was first created or when new nodes, sensors, or actuators were added. Ordinary users should be able to describe in human-oriented terms the nodes, sensors, and actuators in the network, how raw values from the sensors translate into meaningful values to a person monitoring the sensors, what kinds of control is available for the actuators, etc. In Sect. 3.5 we describe an interface for this. The second interface would be used by people monitoring the sensors in the network and issuing commands to the network and the actuators. Sections 3.1–3.4 describe the features we initially propose for this interface. Samples of pages from each of these interfaces are also given in the appropriate sections. Of course, these two types of interfaces can be built into one system with controlled accesses to different functionalities.

In Sect. 2 we mention some related prior work. In Sect. 3 we describe the features that should be present in any generic interface system. In Sect. 4 we describe how a suitable mark-up language could make the implementation of a generic interface system feasible. Finally, in Sect. 5 we make some concluding remarks.

## 2 Related Work

As noted in the Introduction, there are examples of interfaces made for specific application areas or even specific instances of sensor/actuator networks. In addition to those, there exist some examples of interfaces focusing on narrow aspects of sensor/actuator networks. For example, Representational State Transfer (REST) [4] provides some limited mechanism for accessing sensor/actuator nodes through GET/PUT/POST/DELETE, but this is hardly enough for a robust and useable interface. [5] provides a means of communication through their Binary Web Service, but they say nothing about the actual user interface. Moreover, in their system the interpretation of a command is left to the individual node, which leads one to believe the human user may not have had complete information about the functionalities in individual nodes. We believe the user interface should clearly specify exactly what functions are available in each node and what the consequences are of invoking that functionality. There are also interfaces designed for network structures, such as in [6]. These do not address issues like displaying information from the sensors themselves or allowing a user to control individual actuators in the network. Thus, there is a serious lack of well-designed but generic user interfaces focusing on the monitoring and control of the nodes in a sensor/actuator network.

Some work on standardized mark-up for sensor/actuator networks has been done. For example, SensorML [7] defines the standard for representing sensor nodes and networks as well as individual sensors and actuators. Observation and Measurement [8] defines the standard representation for data. Although these two could form a basis for a generic

interface system, important key components are missing that are necessary for a robust, usable, and learnable generic interface for using the sensors and actuators in a network.

We believe HCI issues should be a major driving factor in these standardization efforts. What do the humans sitting at a computer need in order to do what they need to do with the sensors and actuators? Our goal is to lay the foundation for an interface system that can simply be downloaded by anyone and used directly and immediately no matter what the application area or the nature of the sensors and actuators.

### 3 Features Necessary for Robust Interfaces

As noted in the Introduction, it is essential that the user interface be easy to use by people who know about sensor/actuator networks, or at least know about their own network, and who likely do not know about networking principles or HCI principles. Our own prior work in [9] was aimed at this goal; there we developed a user interface for programming sensor/actuator nodes that could be used by people who did not know how to program in traditional languages like C. The vast majority of sensor/actuator systems will be deployed and used by people who are not computer engineers. The systems they use should not require them to become computer engineers.

Based on our study of hundreds of applications described in the literature and our own experience, we have categorized the problem into the following layers – network layer, node layer, and sensor/actuator layer. The following sections describe features and capabilities that should be present at each of these layers in order to have a generic system that can be used by many different applications. Some special applications might need additional features, of course, but the system we propose based on a standardized mark-up language can easily be extended. What we describe here would be sufficient for most sensor/actuator network interfaces. Section 4, then, describes potential mark-ups that could be used to facilitate the features described here.

#### 3.1 Network Level Display and Control

The primary functionality at this layer is to define and view the set of all nodes in the network. A user should be able to add and delete nodes in the interface. For each node the user should be able to provide a name, a generic description of that node, the internet address of that node, etc. (Section 3.2 describes additional information that should be described in individual nodes. At the network layer the user is concerned with the structure of the network as a whole, not the details inside individual nodes.) A user should be able to specify the layout of the nodes, ranging from a simple two-dimensional grid to a three-dimensional distribution of nodes. Users may leave the node display as a simple grid but should also be able to drag nodes to position them on the display in correspondence to their actual physical distribution.

In addition to basic network structure the user should be able to identify and name groups of nodes and groups of sensors. A simple example of where this might be used is an environmental monitoring system. Users might need to focus on groups of nodes in different sections of a large building (e.g., on each floor of a high-rise).

Alternatively, users might want to view all the temperature information throughout the entire building, temporarily filtering out other sensor data such as humidity or the presence of motion. The forming of such groups transcends individual nodes and should therefore be implemented at the system-view level.

Users should be able to view the network as a whole, zooming out to get a more global picture and zooming in to get a more detailed picture of portions of the network. As the user zooms in, details of individual nodes can begin to be displayed. In addition, a user should be able to click on an individual node and perform all the operations described in the next section. Network layer information should include information such as whether or not the node is still functioning and communicating.

Users should be able to issue network-wide commands. These would include, at a minimum, global reset and synchronize/reset clocks. Commands that cause global data acquisition and processing, as opposed to data acquisition and processing in a single node, would also be useful.

### **3.2 Node Level Display and Control**

A user should be able to select any node in the network and view information about that node. Basic information about the node includes its name and general description, for example “Room 2-229” and “Temperature/Humidity Sensor Node”. The user should also be able to see similar information about each sensor and actuator in that node. Each of these should have a name and a general description. For example, an actuator could have name “Window Vent” and general description “Adjust opening of vent by window”. This kind of information helps the user understand the purpose and capabilities of the node as a whole.

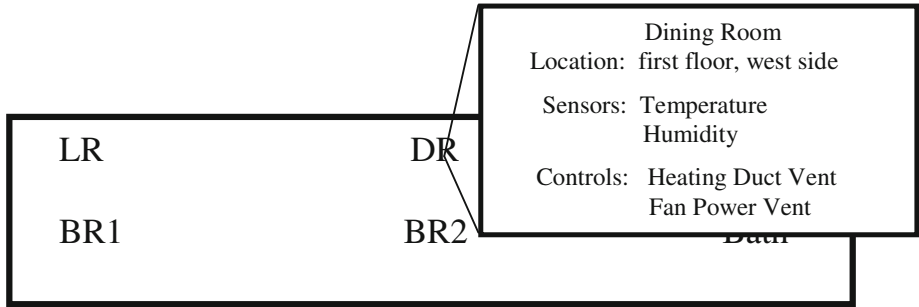
A user should also be able to instruct the node to perform any action of which it is capable. Basic node-level actions include reset and download new code. Actions relating to individual sensors or actuators are described in the next section.

### **3.3 Sensor and Actuator Level Display and Control**

In addition to name and basic description, each device should have information specific to its operation. Each device should have meta-information about its values or units. For example, a temperature sensor should show its range (e.g., 0o–150o), units (e.g., Fahrenheit degrees), raw value (e.g., 47 out of range 0–255), and calibrated value (e.g., 32o F). An actuator should show its range, units, type (digital or analog), and current state (raw value and calibrated value). Additional meta-information could include manufacturer, model, energy consumption, etc. For sensors, the user should be able to see if the data is single-point or aggregated, for example averaged over a certain number of readings or a certain time period. For actuators, the user should be able to control the actuator to the extent possible in the node itself. For a digital actuator that means the user can turn the actuator on or off. For an analog actuator, the user should be able to adjust the output within the actuator’s range. We note that values from sensors and the current status of actuators is dynamic information. The user interface should always display the most recent values for these, although users should have the option to view historical values at the user’s discretion. Other information, such as names and ranges, is static.

### 3.4 Sample Scenario

We illustrate the features described in Sects. 3.1–3.3 with a sample scenario from a home monitoring system. Figure 1 shows a network view of the system with one of the nodes selected by the user and expanded for detailed viewing. Note that the information is displayed in terms that a non-technical, ordinary homeowner could easily understand. We will show in the next paragraph how such information is initially entered into the system. Figure 2 shows the display of a particular sensor that has been selected by the user. Now suppose the user decides that the vent needs to be turned on, and assume that the node is able to accept a command to that effect. Figure 3 shows the display in which the user can activate the vent and select the speed of the fan.



**Fig. 1.** View of a network with one node expanded

Dining Room Humidity Sensor

Description: Shows relative humidity in range  
10% to 90%.

Current value: raw=147 relative humidity=58%

**Fig. 2.** View of one sensor

Dining Room Fan Power Vent Control

Description: Controls the window fan that  
vents air to the outside.

Current setting: OFF

Click a new setting to change:   HIGH  
  MEDIUM  
  LOW  
  OFF

**Fig. 3.** View of one control

### 3.5 Inserting the Marked-up Text

In Sect. 4 we will describe some of the elements of a suitable mark-up language that could facilitate the implementation of features like the ones described in Sect. 3. Although not directly relevant to HCI, the question of how and where the marked-up text gets inserted into the data stream is important and must be answered in order to make a system like we are advocating possible. Many elements of a sensor network, such as RFID tags and battery operated motes, do not have the computing power to augment their sensor readings with suitable mark-up. There are several possible approaches. In [10], for example, the central system adds the mark-up to raw values that come from anywhere in the network. However, in principle the mark-up could be added at any point along the information path, for example at the lowest possible levels in the network, that is, at the point closest to the actual nodes where the element does have the computing power to annotate the data from sensors or extract command to the actuators from annotated text received from higher up in the network. This could be the node itself or the sink to which that node is connected. In a large system the managers at the central point of the system may not know all the details of the individual parts of the network. For example, in a large factory, the managers in the central office may not know details about the sensors and actuators in one small area of the manufacturing floor. However, the people who deploy and monitor those smallest parts of the network should know what their own local system has and should be able to describe the nodes, sensors, and actuators in their own small area.

Of course, there should be an interface at those points to allow a user to easily provide the descriptions needed. A system like our prior work in programming sensor nodes described in [9] can easily be implemented. In such a system the user will be prompted to enter the information about each node, sensor, and actuator in human terms. For example, the user will be prompted to type in the node name (e.g., “EV1”), the node location (e.g., “2-229 FORD”), and similar information for each sensor and actuator. The system then will automatically generate code for annotating information received from the sensors in the nodes and extracting and reformatting commands from the central system to the actuators in that local network. During operation, that element of the system will receive data from sensors and add the mark-up before transmitting further up the network structure. In the reverse direction, that element will extract commands

The figure shows a rectangular dialog box with a double-line border. At the top, it is titled "Specify Sensor/Actuator Node". Below the title, there are two labels: "Name:" and "Location:". The "Name:" label is followed by a single-line text input field. The "Location:" label is followed by a double-line text input field. At the bottom of the dialog box, there are two buttons side-by-side. The left button is labeled "Click to add sensors" and the right button is labeled "Click to add controls".

**Fig. 4.** Dialogue box to specify a node

received in marked-up form from the network and pass the raw commands to nodes underneath. Continuing the example from Sect. 3.4, Figs. 4, 5 and 6 show dialogue boxes in which a user can specify what information to display for the node, the sensor, and the actuator mentioned in that section.

Specify a sensor for: **Dining Room**

Name:

Description:

Values: Minimum  Maximum

**Fig. 5.** Dialogue box to specify one sensor

Specify a control for: **Dining Room**

Name:

Description:

Control type:

Enter control speeds in order:

**Fig. 6.** Dialogue box to specify one control - partially filled in

## 4 Mark-up Language for Sensor/Actuator Networks

We are developing a mark-up language suitable for the application and features described in Sect. 3. Existing mark-up languages, such as [5, 6], can easily be adapted and augmented. Space precludes a complete presentation, so we illustrate with a few examples to show the general nature of the language.

For the network layer the items of interest are the list of nodes in the network, the display of the network structure, and groups. Marked-up text might look like this.

```

<network name="sample network">
  <nodelist>
    <node id="..." > </node>
    ...
  </nodelist>
  <group id="GP1" name="First Floor">
    <groupnode id="GN1" nodemember="N3"/>
    ...
  </group>
  ...
</network>

```

The marked-up text for an individual node might look like the following.

```

<node id="N1" name="Room 2-229" nodeaddress="..." ... >
  <sensorlist ... >
    <nodesensor sensor="S1"/>
    ...
  </sensorlist>
  <actuatorlist ... >
    <nodeactuator actuator="A42"/>
    ...
  </actuatorlist>

```

The string “Room 2-229” is part of what a user would see when he/she selected this node from the network display. The mark-up for an individual device, such as an actuator, might look like the following.

```

<actuator id="A42" name="Ventilator control" ... >
  <description>
    This control turns the vent fan motor off or to one of
    three speed levels.
  </description>
  <control controltype="discrete" numberoflevels="4">
    <controllevel level="0" name="OFF"/>
    <controllevel level="1" name="LOW"/>
    <controllevel level="2" name="MEDIUM"/>
    <controllevel level="3" name="HIGH"/>
  </control>
</actuator>

```

A user viewing this actuator in the interface would see the name and description and would be presented with four choices for control - OFF, LOW, MEDIUM, and HIGH.

As can be seen, the element and attribute names are quite generic, and specific details are relegated to attribute values or element text.



## 5 Conclusion and Future Work

We have proposed a generic system that would allow users to easily configure the interaction for sensor/actuator networks. The system consists of two separate interfaces. The first allows a user to describe in human terms the nodes, sensors, and actuators in the system and to provide all the information needed for the second part of the interface system. That second interface allows a user to monitor the sensors and control the actuators in the network also in human terms that the user can easily understand. Many features, such as alarms or trends or history as in [3], can be added on top of the basic system, but these will not require additions to the mark-up language. Similarly, standards, such as the ISO standard form smart grids [11], can be built on top of a system like we have proposed. The availability of a system like we have described will bring the usage of sensor/actuator networks and in the future the Internet of Things within the realm of users who are not trained in HCI but who nevertheless need to monitor and control their networks.

**Acknowledgements.** This work was supported in part by Northwestern University.

## References

1. Kosnik, D., Henschen, L.: A web-enabled data management interface for health monitoring of civil infrastructure. In: Proceedings of the 15th HCII International Conference, vol. 2, pp. 107–116 (2013)
2. Keller, I., Lehmann, A., Franke, M., Schlegel, T.: Towards an interaction concept for efficient control of cyber-physical systems. In: Proceedings of the 14th HCII International Conference, vol. 17, pp. 149–158 (2014)
3. SCADA System for Industrial and Utilities Automation. <http://www.scada-system.net>
4. REST. [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)
5. Castellani, A., Bui, N., Casari, P., Rossi, M., Shelby, Z., Zorzi, M.: Architecture and protocols for the internet of things: a case study. In: Eight Annual IEEE International Conference on Pervasive Computing and Communications, pp. 678–683 (2010)
6. SwissEx. <http://www.swiss-experiment.ch/index.php/SwissEx:Infrastructure>
7. OGC.SensorML. [http://www.ogcnetwork.net/SensorML\\_Spec](http://www.ogcnetwork.net/SensorML_Spec)
8. ISO 19156:2011. Geographic Information – Observation and Measurements. [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=32574](http://www.iso.org/iso/catalogue_detail.htm?csnumber=32574)
9. Henschen, L., Lee, J.: A web-based interface for a systems that designs sensor networks. In: Proceedings of the 15th HCII International Conference, vol. 4, pp. 688–697 (2013)
10. Week 5, OpenIoT paper. ISO/IEC 30101:2014. Sensor network and its interfaces for smart grid system: [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=53221](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=53221)
11. ISO/IEC 30101:2014. Sensor network and its interfaces for smart grid system: [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=53221](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=53221)