Autonomous Configuration of Smart Nodes in Ubiquitous Environment

Kil Jeon*, Seong-Ju Chang** and Min-Soo Hahn***

*Digital Media Laboratory, Information and Communication University, Korea, shoot666@icu.ac.kr
**Digital Media Laboratory, Information and Communication University, Korea, seongju@icu.ac.kr
***Digital Media Laboratory, Information and Communication University, Korea, mshahn@icu.ac.kr

ABSTRACT

In a ubiquitous computing environment, identifying the locations of users, sensors, and actuators is indispensable to provide proper services to the users. Location can be configured either manually or automatically. However, manual configuration is not proper for large systems because of its difficulty and error-prone nature. For large-scaled systems, smart nodes need automatic location configuration mechanism. Although there are various such methods using GPS, or lateration technique, they do not effectively satisfy critical requirements. This paper proposes a fully distributed method through which smart nodes automatically identify and configure their locations by communicating with neighboring nodes placed within a grid surface. A short-range IR communication device is attached at each of four edges of a smart node to make communication paths in all four directions. All smart nodes in the grid could locate themselves by activating these devices. This novel approach could help to configure an autonomous sensor/actuator network in its initialization and relocation process.

Keywords: Ubiqitous Computing, Sensor Network, Smart Node, Autonomous Configuration

1. INTRODUCTION

Ubiquitous computing environment[1] can be facilitated by deploying huge number of sensors and actuators into the environment and weaving them together. Information on the environment and the user generated by the sensor nodes is processed to extract context from the sensor information, then is transferred to the actuator nodes to provide proper services to the user based on the identified context. To transfer and process necessary information, sensors and actuators should be supported by computational power as well as communication capability.

Identifying locations of sensors and actuators is indispensable to extract required information from sensor readings and to provide services through the actuators which [1]are within the valid distance from the user.[2] Therefore, sensors and actuators should be able to configure their relative locations throughout the entire grid of nodes in the system.

Location can be configured by either manual or automatic method. However, manual location configuration method cannot be applied to a large system because it requires excessive effort not to mention its error-prone nature. To be used for large-scaled systems, location configuration of smart nodes need to be done automatically.

On the other hand, location information of each sensor and actuator can be used to build a location map of the whole system. Constructing and maintaining the location map of all sensors and actuators are very useful to route data from one node to another and to find a specific sensor or an actuator.

Automatic configuration of smart node locations and location map establishment through a centralized approach might be simple. However, breakdown of the processing unit that is engaged in the centralized location identification could cause a system-wide failure. To prevent such a point of failure, location configuration and the construction of the system-wide location map should be performed in a distributed fashion.

In this paper, we introduce a fully distributed autonomous location configuration method utilizing computational and communicational capabilities of multiple sensor/actuator nodes arranged in a grid form.

2. RELATED WORKS

The idea of having a collection of intelligent nodes localize themselves in a distributed method is not a new one. A number of research groups propose many localization methodologies like using GPS, calculating the location of nodes by capturing a system-wide event like lightning, utilizing location infrastructure and referencing directional

The work reported in the paper was supported by the grant of the Digital Media Laboratory, Information and Communication University.

information.
GPS[3] is a commercially successful solution which can be attached to a sensor/actuator node almost immediately. However, GPS cannot be guaranteed to operate in indoor environment not to mention the high cost to be used for acquiring the location of a sensor/actuator node. In addition to that, several meters of its resolution is too coarse for many ubiquitous computing applications.
Calculating the location of nodes by capturing global event is the method that is based on referencing two system-wide events such as light and sound which have different speeds. [4] Most nodes perceive fast event such as lightning at the same time. Then thunder arrives at different time depending on the distance between the point of lightning and the sensor/actuator node. If several events occur at different locations, all nodes can configure their locations automatically. Nevertheless, applying this methodology to an indoor environment is not simple, because the complex structure or interfering objects could block detecting global events at some locations.
Location service infrastructure built with ‘Cricket’[5] or ‘Active Bat’[6] can provide location information with an accuracy of a few *mm* to a few *cm*. These systems configure location based on time differences of arriving ultra sonic signals. These methods usually need manual configuration of location service unit which decreases scalability. Recently, automatic configuration methods using small number of anchor nodes are introduced. In these studies, however, nodes that are far from the anchor node tend to have inaccurate location information because of the accumulated error. In some location infrastructure, the strength of radio frequency signal is used to measure the distance from an adjacent node.[7] Using communication signal as the source of localization is good feature because additional device such as GPS or ultra sonic signal transceiver is not required. The problem of RF based location service infrastructure is the fact that it has low accuracy.
Directional information of neighboring nodes can be used to locate each node represented by the *x-th* element in row and the *y-th* element in column if all nodes are equal size and deployed in a grid pattern. ‘Z-Tile’[8][9] uses four wired communication paths (north, east, south and west) to identify four neighboring nodes and to communicate with them. In ‘Z-Tile’ system, there is a unique sink node which collects all data produced.
Our approach is similar to ‘Z-tile’ in the sense that the location of each node is represented by *(x, y)* coordinates. We also use four directions (top, bottom, left and right) to configure each node’s location. The similarity ends there because we use IR-based wireless communication device and build a system-wide location map for any point-to-point communications among all node in the system rather than just establishing a shortest routing path to a unique sink node.

3. SMART NODE GRID

In this research, we define smart node as a small device unit that has sensor/actuator, communication device transferring data to the nearby nodes and computational unit that processes data or controls an actuator.
We select four infra-red (IR) transceiver modules as the communication devices and one micro-controller for the computing unit. Figure 1 shows such components and their connections to a smart node.

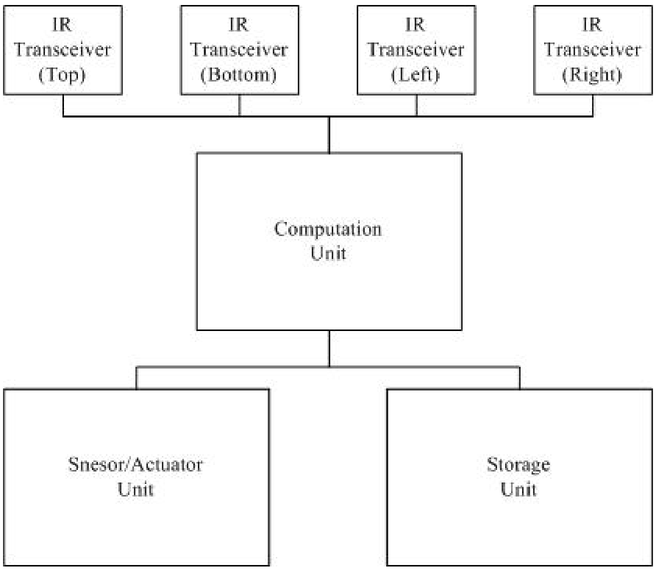

Figure 1: Smart node architecture.

As for four infra-red transceiver modules, we choose Vishay’s TFDU6120F that satisfies IRDA FIR specification. For the signal encoder/decoder, 4 PPM method is used. Due to the low operating speed of the microprocessor, actual data transfer rate is lower than expected 4Mbps. To control four communication modules with one controller, the communication channel is multiplexed.
For controlling communication module, processing data and storing location map, Microchip’s PIC18F252 is used. This microprocessor has 32 kilo bytes on-chip program memory, 1536 bytes on-chip RAM and 256 bytes on-chip EEPROM, and supports hardware USART, Capture, PWM and 5 10bit A/D Converting ports. For operating clock, we choose 20MHZ crystal.
There are several ports which interface a smart node with

other components. RS232C port is supported to communicate with other add-on component which supports RS232C interface. A smart node can be utilized to localize existing grid-pattern system such as 'Smart Architectural Surface'[10]. Our fabricated smart node also supports RS485 interface currently running on one computer which is used to verify location configured smart nodes. Figure 2 shows the actual implementation of a smart node utilized for this research.

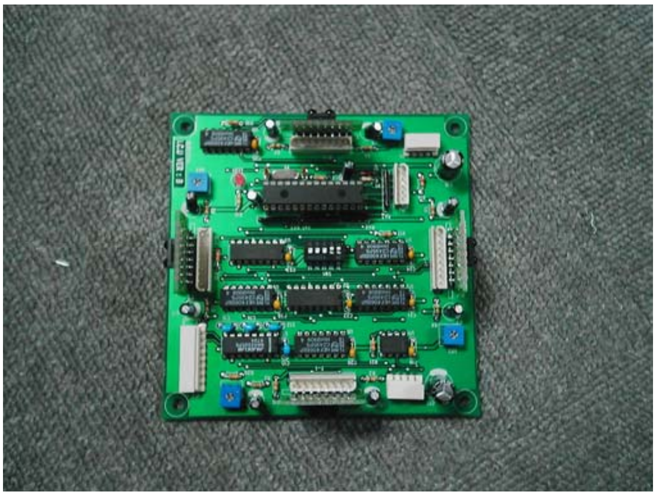

Figure 2: Fabricated smart node hardware.

4. LOCALIZATION

In this chapter, location configuration method for smart node by using explained hardware in previous chapter will be introduced. At first, initial configuration method is proposed. Then node addition rule and removal rule are explained. Figure 3 at the end of this chapter shows entire process for localization.

4.1 Initialization of Smart Nodes

Smart nodes fabricated for this research configure or reconfigure their locations through communicating with neighboring nodes when the power is initially turned on or comes back on after a power out event or a smart node(s) break down event.

When the power is turned on, all smart nodes initialize their own locations as (-1, -1), which represents unconfigured status. During normal operation, any smart node in a grid system periodically checks all four directions to know whether its neighboring nodes exist or not. If there is no neighbor in both top and left direction, the smart node configures its own location as (0, 0) and then starts to configure the locations of neighboring smart nodes. If a smart node is unconfigured and its neighboring node exists in either top or left direction, the node keeps its unconfigured status until a location data is transmitted from any neighboring node.

If the location of a smart node is newly configured, that node calculates its neighbors' locations and transfers their new locations. For example, if the location of a node is (x, y), neighboring node at the top direction will receive the new location (x, y-1).

There is a condition that decides whether or not a smart node that receives new location information updates its location. The rule is simple; if x or y coordinate of the newly received location information is larger than the original one, the node should update its location. This rule is necessary to prevent location collision caused by possible existence of multiple location configuration starting points within a smart node grid.

4.2 Addition or removal of smart nodes

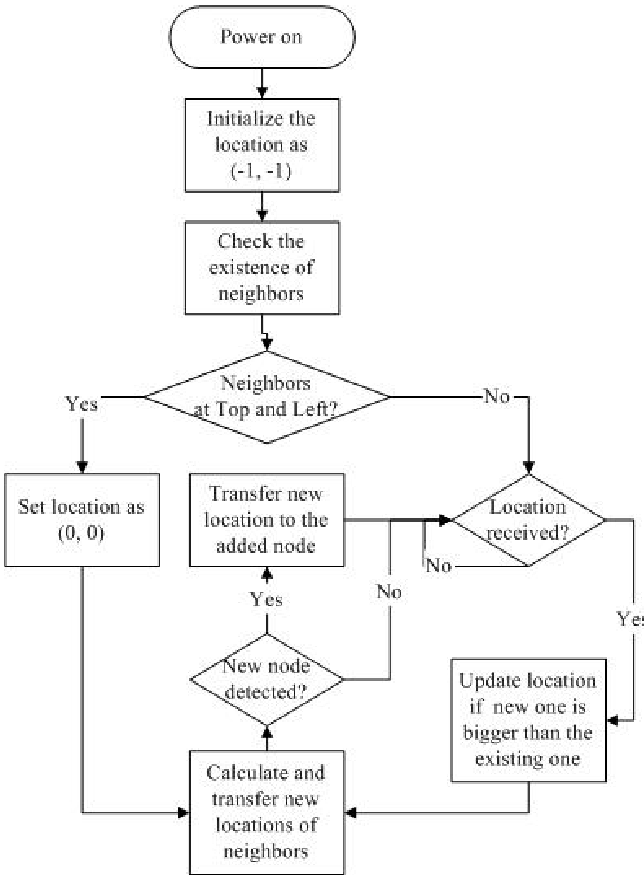

Figure 3: Location identification process.

While a smart node grid system is operating successfully after each node identifies its location, it is possible that some nodes belong to the system are removed or new nodes are added to the existing system. Especially, removal or addition occurs when there is the maintenance task needed for some smart nodes, additional installation of new nodes, abnormal behavior of some defective nodes or simply a communication error.

A new node added to the existing system should identify its location in consistent with the existing system. The smart nodes which completed their location identification tasks check their neighboring nodes periodically. If an occurrence of a new node is detected, the node which detects the existence of the new node calculates and transfers the location information of its new neighbor to the newly attached node.

Node removal does not affect the locations of other nodes in the system. Therefore, what has to be processed is to initialize the removed node. If a node removal occurs due to the breakdown of a node, the initialization of the node is done automatically. Otherwise, the removal of all existing neighboring nodes becomes the trigger which initializes node location.

5. ESTABLISHING LOCATION MAP

Location map is essential information to exchange data among smart nodes that are not adjacent each other and to find a proper actuator node in the current context. During the location identification process, a smart node additionally builds location map by communicating with neighboring nodes. Location map construction is based on the two rules depending on the situation: location allocation and location removal.

5.1 Rule for location allocation

Location allocation occurs when the location of a node is configured for the first time or a new node is added during operation. The same process happens when multiple-start point confliction occurs. Figure 4 shows the internal change of the location map when a location allocation process occurs.

The smart node whose location information is newly allocated receives a copy of the location map which belongs to the node transmitting the information. Received location map remains as the basis map of that node until the next new location is allocated. After the completion of storing location map, the smart node informs its neighboring nodes that it received a new address so that they can change their location map accordingly. This location map change event must be propagated throughout all nodes arranged in a grid form to have a system-widely consistent location map.

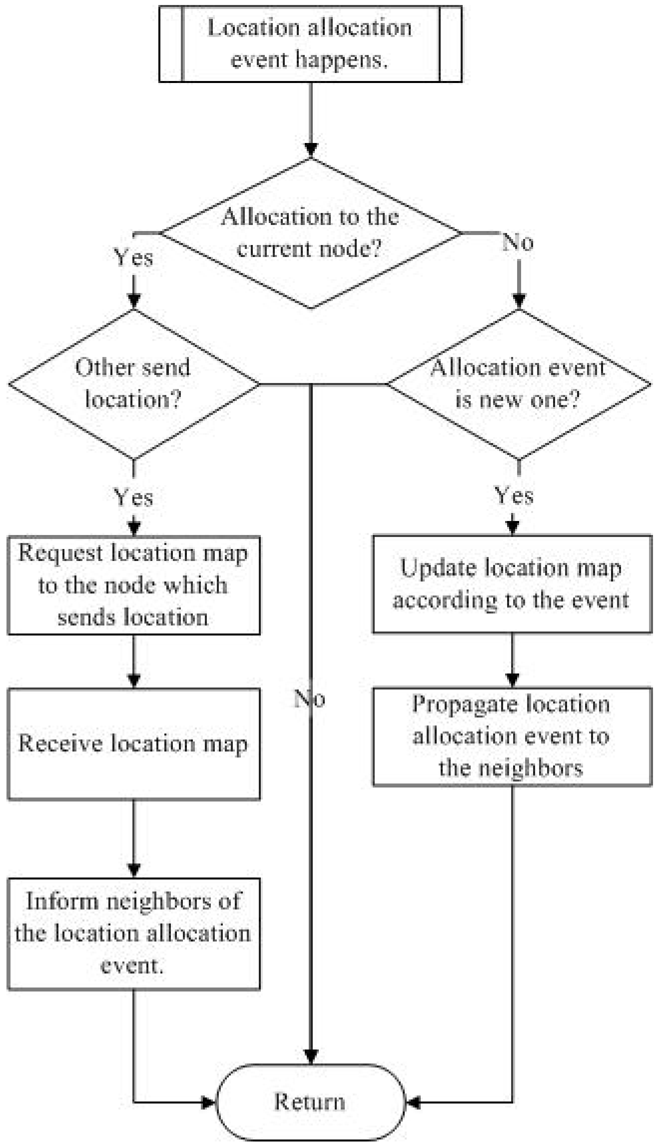

Figure 4: Location map building process (location allocation)

5.2 Rule for location removal

Location removal process occurs when a smart node is physically removed from the system or is broken down. In some cases, location removal event is critical because removal of a node could divide the system into two disjoint sub-systems. In this case, because there is no route from a sub-system to another, the location map of a sub-system should not contain the nodes in another disjointed sub-system. Figure 5 shows the update rule of location map when a location removal event happens.

The Nodes which detect a location removal event of neighboring node delete the removed node from their location map and check the connection between itself and every neighboring node of the removed node. If there is no connection, all nodes which are connected to that node are removed. Every deletion event must be notified to the neighbors. Any node that receives deletion event modifies its location map and retransmits that event to the neighbors.

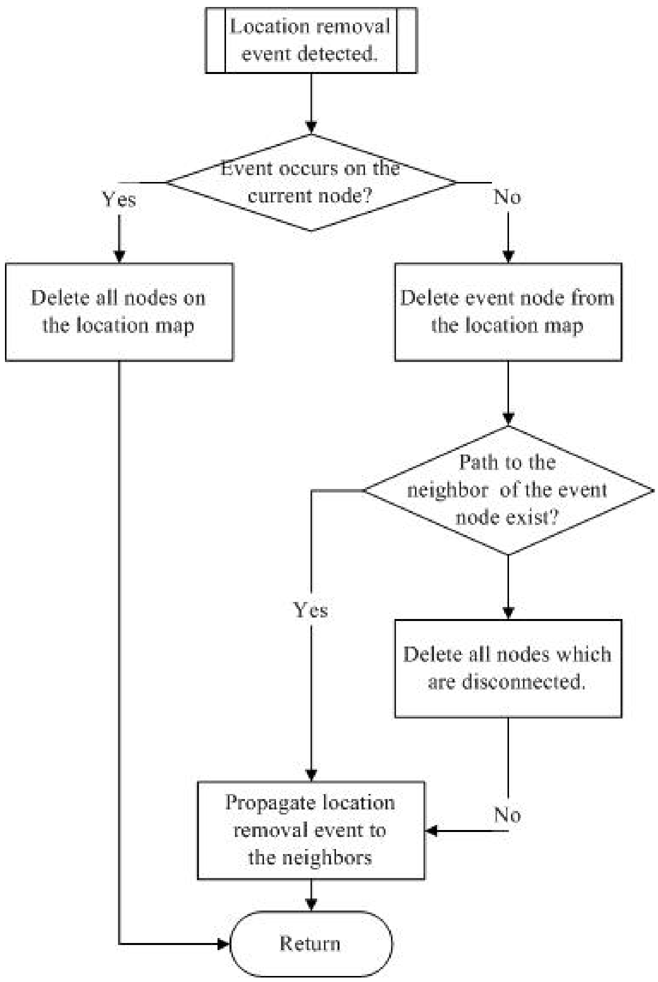

Figure 5: Location map building process (location removal)

6. FURTHER WORK

The autonomous location configuration method for a grid form smart nodes presented here has been tested with a simulator and has been proved to work successfully when there is no communication error and/or breakdown of nodes. Run-time node addition and deletion are also tested.

Although location identification and the establishment of location map are implemented, actual data routing has not been implemented yet in this research. A simple algorithm of routing data in the shortest path on the location map would work. If a bottle neck phenomenon occurs at one node, however, data packet may be lost by the buffer full status. Therefore, routing method has to provide a flexible data path according to the current traffic status.

Because we select using one microprocessor for a smart node, limitation of communication bandwidth and synchronization problem happened. Allocating one micro processor per a communication device would help to increase the communication bandwidth.

At this point, our smart node system is tested only with simulator based on a small size system of having 25 nodes. Larger smart node system must be implemented and tested to be used in real world applications.

7. CONCLUSIONS

Although many research groups are working in the field of autonomous localization of sensor/actuator nodes, previous works are almost exclusively based on broadcast signal like radio frequency, ultra sound and ultra wide band and does not take the benefit of grid formation constructed by multiple smart nodes as is often founded in the artificial structures like a building. We have presented an approach using directional wireless communication device such as infrared transceiver as a localization device. In most applications, location of a node is sufficiently represented based on x and y indices depending on the node's horizontal and vertical positions on the entire grid system. The fact that all nodes in the system have consistent location map enables all nodes to communicate with other nodes in a system without the help of dedicated external system. Thus smart node system proposed in this paper has a great potential to make a novel yet efficient infrastructure for a ubiquitous computing environment.

ACKNOWLEDGEMENTS

The work reported in the paper was supported by the grant of the Digital Media Laboratory, Information and Communication University

REFERENCES

[1] M. Weiser, and J. Seely-Brown, The Coming Age of Calm Technology, Beyond Calculation: The Next Fifty Years of Computing, P.J. Denning and R.M. Metcalfe, eds., Copernicus, Heidelberg, Germany (1998).

[2] J. Hightower, and G. Borriello, Location systems for Ubiquitous Computing, IEEE Computer, vol. 34, pp. 57–66 (2001).

[3] B. Hoffmann-Wllenhof, H. Lichtenegger, and J. Collins, Global Positioning System: Theory and Practice, Fourth

Edition, Spinger-Verlag (1997).
[4] M. Broxton, J. Lifton, and J. Paradiso. Localizing a Sensor Network via Collaborative Processing of Global Stimuli, Proc. of the European Conference on Wireless Sensor Networks (EWSN2005) (2005).
[5] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan, The Cricket Location-Support System, Proc. 6th Ann. Int'l Conf. Mobile Computing and Networking (Mobicom00), ACM Press, New York, pp. 32-43 (2000).
[6] A. Harter et al., The Anatomy of a Context-Aware Application, Proc. 5th Ann. Int'l Conf. Mobile Computing and Networking (Mobicom 99), ACM Press, New York, pp. 59-68 (1999).
[7] J. Hightower, R. Want, and G. Borriello, SpotON: An Indoor 3d Location Sensing Technology Based on RF Signal Strength, UW CSE 2000-02-02, Univ. Washington, Seattle (2000).
[8] B. Richardson, K. Leydon, M. Fernstrom, and J.A. Paradiso, Z-Tiles: Building Blocks for Modular, Pressure-Sensing Floorspaces, Proc. of the ACM Conference on Human Factors and Computing Systems (CHI 2004), Extended Abstracts, Vienna, Austria, pp. 1529-1532 (2004).
[9] B. Richardson, and M. Fernström, Network and Control Protocol for a Wired Self-Organising Sensor Network. Proc. CCCT 2003 (2003).
[10] S. Chang, J. Cung, H. Kim, Y. Kim, M. Lee, Smart Architectural Surface: Modularized Platform for Polymorphic Functional Changes and Multimodal Interactions, Proc. International Conference on Artificial Reality and Telexistence (ICAT2004) (2004).