ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 *2009. No. 6(94)*
ELEKTRONIKA IR ELEKTROTECHNIKA

ELECTRONICS
T170
ELEKTRONIKA

Smart Dust Motes in Ubiquitous Computing Scenarios

J. Preden, R. Pahtma
Laboratory for Proactive Technologies, Department of Computer Control, Tallinn University of Technology
Ehitajat e tee 5, Tallinn 19086, Estonia, phone+3726202100; e-mail: jurgo.preden@dcc.ttu.ee

Introduction

Wireless sensor networks introduced in the beginning of the current century are networks of tiny embedded computers that are equipped with a wireless communication interface, some sensors and an autonomous power supply. These devices are also called smart dust motes, implying small size and "smartness" distinguishing them from mere sensing devices. The networks formed by smart dust motes are (mobile) multihop ad-hoc networks (also called MANET networks) where each node also performs the task of a router, forwarding messages from other network nodes. This approach allows to extend the network without increasing the communication range of individual network nodes. Small form factor, low price, simple installation procedure and the offered capabilities seem to call for large scale wireless sensor network installations and wireless sensor networks are used today in a range of monitoring applications both as research prototypes and also in industrial applications. However the current use of wireless sensor networks is not as widespread as predictions a few years ago foresaw.

Ubiquitous computing (also pervasive computing, invisible computing) concept was first introduced by Weiser in 1991 [1]. Ubiquitous computing encompasses computing devices in the environment connected (typically via wireless communication interfaces) with each other. Software intensive devices whose functionality is enabled only by the use of appropriate embedded computing software and hardware are already common in our everyday life. Although devices with similar functionality existed before a computer (or several computers) were integrated into them, the non-computerized versions of such devices usually exhibit reduced functionality when compared to their modern counterparts (compare for example modern cars or washing machines to their counterparts thirty years ago). It is envisioned that computing devices will be even more ubiquitous, go far beyond the software intensive devices and personal computing devices in use today. The ubiquitous computing vision foresees that tiny, even invisible computers, embedded into the environment and everyday objects such as tools, appliances, clothing, even the human body, are able to interact directly with the environment and each other, maintaining up to date information on the environment. These smart devices will determine the current situation based on the collected information about their locations, human users in the vicinity, their peers. The devices will use preprogrammed algorithms and information collected via past interactions to determine optimal behaviour for the current situation. A ubiquitous computing system is expected to anticipate the needs and expectations of a human user and act accordingly.

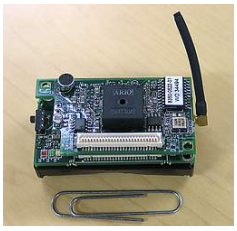

Fig. 1. Smart dust mote MicaZ from Crossbow Technology

Wireless sensor networks has been an emerging technology for almost a decade, however it has not been able to really take off. There are several reasons for this, but no single one can be called a major one. Using wireless sensor networks only for monitoring applications is promising and useful, however in most cases the advantages of smart dust do not outweigh the shortcomings when compared to established technologies. It is clear that it will take at least a few decades before ubiquitous computing environments will be available, so this field is not and will not be the driving force behind wireless sensor network development. Regardless of the slow development we can still look at the wireless sensor network technology and discuss what development is needed in order for this technology to enable ubiquitous computing development.

Distributed applications

Most current wireless sensor networks use a many to

one data flow model where all data collected by the sensor network nodes is transmitted to a single (or a few) sink nodes with little or no data preprocessing by the wireless sensor network nodes. Essentially these approaches use WSNs as mere data collection systems where wires have been replaced by wireless links. Such an approach is suitable for data collection systems but even in this application it is not very desirable in all cases since data is accessible only via the sink. Even some of the routing protocol implementations for wireless sensor networks are only able to transmit data to a single sink node. System architectures that rely heavily on the sink node are quite inefficient in terms of network bandwidth utilization as all (or a very large amount) the collected sensor data is transmitted from all the sources to a central data collection / data fusion centre via the sink node. The data could be interpreted locally instead, fusing data from nearby nodes and the only data sent out from the network would be application-level information.

In ubiquitous computing scenarios where the network nodes must realize a distributed application a sink-based data flow model is clearly not desirable. The configuration of an application in a ubiquitous computing system is not predefined – the nodes and their configurations are not known at design time, nor is this information available at the time of deployment. All the distributed applications realized by a given system may not be known at the time when the computing system is deployed. Applications are formed at runtime and the application components are also discovered at runtime.

A property of MANET type of networks (which is the foundation of ubiquitous computing systems) is that nodes can join and leave (or can be added to or removed from) the network and therefore the nodes (and the lower and higher level algorithms, including applications running in the network) must be also able to adapt to such changes. Such flexibility means that the configuration of the network is not known at design time but instead the configuration is formed (and may change) at run-time. A good example of a dynamically formed network is the deployment of motes from an airplane (which is realistic since motes are used in military monitoring applications) – the initial configuration of the network (the placement and density of nodes) is to a great extent determined by the way the motes fall to the ground, it can't be even expected that all the deployed motes will be operational or that the deployed network is fully connected. If the deployed nodes have different sensing modalities the specific sensor coverage of an area may also vary. As the configuration of the network is also dynamically changing (nodes may be destroyed, their power supplies depleted), the nodes must adapt to the changes in the configuration dynamically. No central coordinating entity or special nodes with special capabilities can be assumed (as all nodes are equally expendable) and therefore all the tasks in the network must be performed by the nodes themselves.

Such a dynamic system can be also envisioned in the context of a smart house – devices built into the house must interact with the devices that the residents of the house introduce to the house, devices are removed as they become obsolete or as the residents become bored with them. A smart house must also interact with computers embedded into the people that live in the house. When the ubiquitous computing system is deployed a resident of the house may not have a cardiac pacemaker but after the pacemaker has been implanted no manual configuration steps should be required. A pacemaker should, if required, be able to discover the coffeemaker in the kitchen and tell it to make weaker coffee because a person's heart rate is too high.

Data origin

The problem of data origin must also be addressed – if node deployment is not deterministic exact locations of nodes cannot be known until after the deployment (after a mote has fallen to the ground or after a device has been placed in the house by a resident). Node positioning can be quite complex in situations where no positioning infrastructure (such as GPS) exists, which is usually the case indoors and also outdoors in a hostile environment.

A common way to overcome these problems is to use some location aware anchor nodes (which location may be determined manually) that are able to position other nodes in the environment. Current positioning algorithms that rely on anchor nodes can be classified into three categories: algorithms that rely on the distance estimations, algorithms that use bearing information and algorithms that use both. The accuracy of the position estimate of all these approaches depends of the accuracy of the distance or bearing estimation. Given the fact that acquiring precise bearing or distance estimations is not possible with the limited hardware of current this is a problems that remains to be solved. We describe a study performed on evaluating the use of received signal strength indication in a companion paper "Utilization of Received Signal Strength Indication by Embedded Nodes."

Situation awareness of motes

MANET types of networks are open, unstructured, dynamic and heterogeneous and they present us fundamental problems which are not solvable by conventional computer science methods [2]. Due to the high number of devices and the unpredictable interactions between the devices themselves and the interactions between the devices and the physical world these networks exhibit emergent behaviour, which we can't predict or foresee but instead we can only watch the emergent behaviour develop as the systems operates [2].

It is clear that new theoretical approaches are required in order to be able to design and implement usable ubiquitous computing system prototypes using smart dust motes. The use of context and situation awareness has been suggested by several researchers [3] [4], however only the situation of the human user .is considered in most cases.

Situation awareness (SA) has been quite well studied
in the context of human factors research [5]. In [5] also a classification for situation awareness levels is suggested, which can be applied quite well also for situation awareness of computing systems. Level 1 SA – perception. Perception of information is vital to achieving any SA as without perception of relevant information it is very difficult, if not impossible to achieve a good level of SA. Level 2 SA – comprehension, it includes the process of combining, interpretation, storing and retaining information. Clearly the comprehension step goes beyond perception – comprehension of information means that multiple pieces of information (received from multiple sources) must be integrated and the relevance of the information must be evaluated within the scope of the current situation and the goal function. Level 3 SA is projection to the future. The highest level of SA is the ability to foresee the future state of the significant factors and the dynamics of these factors. Essentially it is the ability to project future events from the available past and present data, which experienced operators do on a regular basis.

In computing systems the same concepts of situation awareness levels can be used. Level 1 SA or perception is the acquisition and conditioning of data via sensors. Level 2 SA or comprehension is the interpretation of the data acquired in the perception step. The result of the comprehension step is assessment of some situation property values. The comprehension process could involve several data sources and also historic data. Comprehension can occur on several levels, i.e. lower-level situation property values can be used to assess higher-level situation property values.

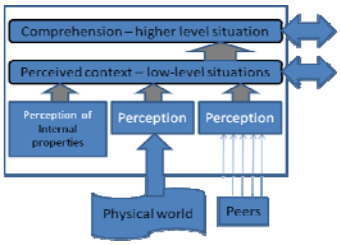

Fig. 2. Situation assessment

In the context of computing systems level 3 SA or projection may be trivial to achieve for simple situation properties, such as a temperature value, but much more complicated if not impossible for complex situations (such as the intent of a human). The hierarchy for achieving situation awareness and the data flows are depicted on Fig. 2.

In order for the data interpretation result to be usable all data used as input to the data interpretation algorithm must be accompanied with metadata that establishes the properties of the data, including data acquired in the perception step and also data generated as a result of the comprehension step. This metadata includes temporal and spatial validity intervals of data, i.e. where and for what period this sensor reading or situation property is valid.

In order to identify the higher level situation properties the lower level situations must be identified first. We can construct an example where climate of an area is assessed
from the human comfort perspective. The physical parameters used as input to the assessment algorithm are ambient temperature and relative humidity. For temperature the situation property tuple is the following: $S_T = (S_p, S_t, S_l)$ where S_p is the assessment of the temperature property (which may be temperature value in Celsius), S_t is the time period for which the temperature value holds and S_l is the area for which the temperature value holds. The time period for which the temperature value holds is derived based on the properties of the environment (in a room with an efficient climate control system the temperature may change quickly while in a cellar the temperature changes are slower) and possibly the trend in the temperature value. The area for which the temperature value holds is derived from the location of the temperature sensor. If the sensor is located outdoors the temperature value may hold for an area with a rather large radius while if the sensor is located in a room the temperature value may only hold for that specific room. A similar tuple is used for the humidity situation property: $S_H = (S_p, S_t, S_l)$. The evaluation of the humidity situation property is complex since in order to assess the humidity situation both the reading of the humidity sensor and the temperature situation are utilized – $S_H = f(S_T, S_{SH})$ - S_H is a function of S_T – temperature situation property and S_{SH} – the humidity sensor reading. The comfort situation property assessment is a function of temperature and humidity properties $S_C = f(S_T, S_H)$ and the situation property can be expressed with the following tuple $S_C = (S_p, S_t, S_l)$.
It is clear that the temporal and spatial validity interval are dependent on the validity interval of the data used as input by the situation parameter evaluation function. This imposes requirements on the data used as input for the evaluation function if situation parameter value is required for a specific region or time.

The sensor data received from other nodes may need to be conditioned utilizing the metadata of the data before it can be used as input to an interpretation function. Functionality similar to the channel function, as described in [6] can be used to manipulate the received data based on the requirements on the data – for example if data received from different nodes (incoming data from each node can be viewed as a different streams of data) has different validity intervals the data from different streams can be conditioned (using interpolation, averaging, normal distribution rules or methods similar to technical analysis) to achieve data properties that can be mapped to a required interval.

System component lifecycle

The lifecycle of embedded devices is another issue that must be considered in the design and implementation of ubiquitous computing systems. An embedded computer that functions as part of a physical object may have the same expected lifetime as the physical object which may be significantly longer than the lifetime of an average computer. However the distributed applications that an embedded computer is part of may change during the lifetime of the embedded computer. Since the embedded devices interact directly with the physical world their actions are partially also dependent on the physical world. Hence it is not possible to reason about the precise behaviour (schedulability of functions, duty cycles, bandwidth allocation, etc) of a device before the device is installed and becomes part of the physical world. It is not possible to predict all the possible combinations of applications that a device is going to be part of or the nature of the inputs from the physical world. We can consider a simple example from the everyday life – when we design a wrench we have no way of predicting what bolts and nuts that wrench will interact with during the lifetime of the wrench and by whom and how often the wrench will be used. The same way we have no way of predicting what interactions an embedded computing device will be part of during its lifetime. Instead the device must be designed and implemented in a way that makes it possible to cope and interact with the changing environment.

Smart space project

The Research Laboratory for Proactive Technologies is involved in a smart space project, implementing a ubiquitous computing environment prototype. The MicaZ smart dust motes from Crossbow Technology are embedded into the environment at fixed positions. The environment must provide human users with information in a context sensitive manner, as specified by the user. The smart dust motes are equipped with sensors, which allow them to collect information on the physical properties of the environment. The situation information (which includes the environmental conditions) collected by the nodes is stored in the network in a distributed manner. The network of motes also functions as a positioning system utilizing a modified distributed version of the CABP positioning algorithm presented in [7]. Human users are provided situation information of interest by the mote network based on the situation of the user, which is in turn determined by the mote network.

Acknowledgements

The work presented in this paper was partially supported by the Eliko competence center and the Estonian Information Technology Foundation.

References

1. **Weiser M.** The computer of the 21st century, Scientific American. – Scientific American Inc, 1991. – Vol. 265. – P. 66–75.
2. **Milner R., Stepney S.** “Nanotechnology: Computer Science opportunities and challenges”, Submission by the UK Computing Research Committee to the Nanotechnology Working Group of the Royal Society and the Royal Academy of Engineering, 2003.
3. **Yau S.** Situation–Aware Contract Specification Language for Middleware for Ubiquitous Computing, The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems. – 2003.
4. **Matheus C., Baclawski K. and Kokar M.** Derivation of ontological relations using formal methods in a situation awareness scenario, Proc of SPIE Conference on Multisensor, Multisource Information Fusion. – 2003. – P. 298–309
5. **Endsley M. R. Garland D. J.** Situation awareness analysis and measurement – Lawrence Erlbum Associates Publishers, 2000.
6. **Motus L. Rodd M. G.** Timing Analysis of Real–Time Software, Oxford: Elsevier, 1994.
7. **Preden J.** "Communication area based positioning", Proc. The 3rd IEEE International Conference on Mobile Ad–hoc and Sensor Systems. – 2006. – P. 336–347.

Received 2009 02 27

J. Preden, R. Pahtma. Smart Dust Motes in Ubiquitous Computing Scenarios // Electronics and Electrical Engineering. – Kaunas: Technologija, 2009. – No. 6(94). – P. 19–22.

The ubiquitous computing vision foresees that tiny computers, embedded into the environment and everyday objects are able to interact directly with the environment and each other, maintaining up to date information on the environment. Wireless sensor networks that are networks of tiny embedded computers can be viewed in a sense as a foundation for developing ubiquitous computing. In ubiquitous computing scenarios where the network nodes must realize a distributed application a sink-based data flow model, which is currently prevailing in the sensor networks field, is clearly not desirable. A service-based application model where the computing agents are situation aware is seen as a solution for realizing such applications. The article introduces the situation awareness concept for computing devices and describes some case studies. Ill. 2, bibl. 7 (in English; abstracts in English, Russian and Lithuanian).

И. Преден, Р. Пахтма. Исследовония влияния количества пыли на работу компьютерного управления // Электроника и электротехника. – Каунас: Технология, 2009. – № 6(94). – С. 19–22.

Представляются результаты исследования количества пыли при передачи информации. Предлагается применять бес проводные датчики сетей. Описывается концепция ситуации в разных условиях компьютерного управления. Ил. 2, библ. 7 (на английском языке; рефераты на английском, русском и литовском яз.).

J. Preden, R. Pahtma. Išmaniųjų dulkių krislelių įtaka kompiuterizuotam valdymui // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2009. – Nr. 6(94). – P. 19–22.

Kompiuteriniam sistemų valdymui gali būti taikomi išmaniieji dulkių krisleliai. Integruoti į aplinką tokie krisleliai gali nuolatos stebėti, perduoti informaciją vieni kitiems, sekti ir kaupti dienos įvykius. Jiems realizuoti reikalingi bevieliai jutiklių tinklai, kurie sudaryti iš smulkių integruotų kompiuterių. Pateikiama situacijos koncepcija ir trumpa galimybių studija. Il. 2, bibl. 7 (anglų kalba; santraukos anglų, rusų ir lietuvių k.).

DOI: 10.5755/j02.eie.10067