MWM: A Map-based World Model for Wireless Sensor Networks*

Abdelmajid Khelil, Faisal Karim Shaikh, Brahim Ayari and Neeraj Suri
Technische Universität Darmstadt
Dependable, Embedded Systems and Software Group
Hochschulstr. 10, 64289 Darmstadt, Germany
Tel. +49 6151 16{3414|3711|7066|3513}, Fax. +49 6151 16 4310
{khelil|fkarim|brahim|suri}@informatik.tu-darmstadt.de

ABSTRACT

A prominent functionality of a Wireless Sensor Network (WSN) is environmental monitoring. For this purpose the WSN creates a model for the real world by using abstractions to parse the collected data. Being cross-layer and application-oriented, most of WSN research does not allow for a widely accepted abstraction. A few approaches such as database-oriented and publish/subscribe provide acceptable abstractions by reducing application dependency and hiding communication details. Unfortunately, these approaches ignore the spatial correlation of sensor readings and still address single sensor nodes. In this work we present a novel approach based on a "world model" that exploits the spatial correlation of sensor readings and represents them as a collection of regions called maps. Maps are a natural way for the presentation of the physical world and its physical phenomena over space and time. Our Map-based World Model (MWM) abstracts from low-level communication issues and supports general applications by allowing for efficient event detection, prediction and queries. In addition our MWM unifies the monitoring of physical phenomena with network monitoring which maximizes its generality. Using two case studies we highlight the simplicity and also the versatility of the proposed architecture. From our approach we deduce a general modeling and design methodology for WSNs.

Categories and Subject Descriptors

C.2.1 [**Computer**]: Communication Networks—*Network Architecture and Design* [network communications, wireless communication]; C.2.2 [**Computer**]: Communication Networks—*Network Protocols*; I.6.5 [**Simulation and Modeling**]: Model Development—*modeling methodologies*

General Terms

World model, design paradigm, global and local view

Keywords

Wireless Sensor Networks, Event Detection, Monitoring

*Research supported in part by DFG GRK 1362 (TUD GKmM), EC NoE ReSIST and MUET

AUTONOMICS '08, September 23-25, 2008, Turin, Italy

1. INTRODUCTION

Wireless Sensor Networks (WSNs) represent networked autonomous embedded systems. With diversity as a key hallmark, WSNs often comprise computing nodes with heterogeneous communication, sensing, processing and storage capabilities. WSNs can be embedded in varied environments with the desired goal of sensing, monitoring, detecting and predicting phenomena of interest in the physical world. The WSN is considered as a tool for observing states of the real physical world [1], a *bridge to the physical world* [2] or its most proximate instrumentation [3]. The designer's view on WSN augments this user-centric view with the technical details. From the literature, three main system-level design paradigms for a WSN arise, namely, considering a WSN as a database, an event service or a network.

(a) *WSN as a database*: Major research issues are in designing efficient query dissemination and in-network selection, projection, join and aggregation. An example being tinyDB [4] that treats sensor data as a single table (sensors) with one column per sensor type. Research focuses here on data-centric communication [5, 6, 7, 8, 9, 10].

(b) *WSN as an event service*: A WSN is usually deployed for missions providing services such as tracking targets or detecting events. The mission determines the overall operation of the WSN including data generation, processing, filtering and transport. To support multiple missions and augment the service flexibility, the publish/subscribe (pub/sub) service architecture has been advocated for WSNs [11, 12, 13, 14, 15, 16].

(c) *WSN as a network*: WSN can be viewed as a communication platform. The standard WSN communication architecture is layered in a similar way to the OSI layer model. However, a modification of this architecture called a cross-layer model is commonly adopted. The cross-layer design is an envelop of optimizations that benefit from the the co-operation of non-adjacent layers [17]. Furthermore, new communication patterns such as convergecast and event-based data transport have been proposed.

Most of WSN research is application-oriented and relies on cross-layer approaches. Therefore, current research does

not allow for a widely accepted abstraction. While the design paradigms mentioned above reduce application dependency and hide low-level communication details, they still address single sensor nodes (although redundancy of nodes and spatial correlation of sensor readings are inherent in WSN). Subsequently, there is a strong need for a *holistic* design methodology. Such a methodology should be flexible/abstract and systemize/simplify the design as well as the deployment phases and involve functional as well as extra-functional attributes. Obviously, the holistic approach should retain the advantages of the existing paradigms. Overall, rather than addressing single nodes, designers should address spatially-correlated and appropriately-grouped nodes. We refer to such groups as *regions*. A few efforts do exist to address regions instead of single nodes [1, 18, 19].

The *map paradigm* builds on the region principle and therefore, provides excellent modeling primitives for WSNs. Global maps are created for the sake of network monitoring (e.g., residual energy map [20]) or of event detection (e.g., oxygen map [21, 22]). A promising direction consists in using maps to optimize protocols [23, 24], detect [25] or track [26] event boundaries. These papers highlight the map-based methodology as a powerful and highly promising abstraction level. In this work, we propose that maps are the natural step towards a *holistic Map-based WSN design*.

The user is interested in monitoring a certain physical phenomenon or generally the *physical world*. The WSN is the dashboard of the user for that physical world. Accordingly, a WSN has to create an appropriate model of the physical world of interest, collect the required data and provide it to the user in the appropriate form. Similarly, an administrator requires a dashboard on the "*network world*", e.g., to show where the energy is suffering more. Therefore, we model a WSN as a (physical/network) World Model and develop a holistic *Map-based World Model (MWM)*.

Paper Contributions: We develop the MWM for generalized WSNs. We show how this model retains the advantages of existing design methodologies while augmenting their efficiency and level of abstraction. We also present a step-wise design methodology to build an appropriate MWM and the process driving its usage. We demonstrate the applicability of the MWM through two case studies which highlight the benefits of the MWM approach in predicting a network event (a network partitioning) and a physical world event (a fire). In particular, we emphasize the following qualities of our MWM:

(1) Our MWM can appropriately reflect both physical and network worlds while being aware of the strong constraints on network and node resources. This is achieved by providing a model of high modularity/composablility and abstraction/flexibility, which allows for an easy customization and evolvability to (a) the application and (b) to the deployment constraints.

(2) The MWM holistic approach provides a natural way to define and detect arbitrarily complex events. The extraction of patterns from a given map presents a powerful and generic approach to define simple events [27]. The matching of composed patterns across a stack of maps allows to define composite events. Event operations can be matched to geometric operations on geometric patterns.

(3) The MWM approach not only allows for an efficient event detection but also for a simple and lightweight prediction of event occurrence. The matching of patterns on a temporal stack of the maps allows the prediction. The prediction capability of the model significantly enhances its utility. The model can be then used for proactive options to enhance the functionality and dependability of the WSN.

(4) The MWM can model arbitrary real world situations besides events, which are modeled as situation transition like suggested in [1]. Furthermore, it aids unified modeling for both simple and complex scenarios.

The remainder of this paper is structured as follows. Section 2 gives an overview on models for sensing the real world, the system model that we consider in this paper and the main requirements on the MWM. In Section 3, we present our novel MWM-based architecture for sensor networks. Then, we define essential primitives such as event and query services and MWM management. Subsequently, we highlight the utility of the MWM for predictive monitoring and pro-active reconfiguration and provide a step-wise MWM-based design approach for WSN. In Section 4, we validate our approach using two case studies. In Section 5, we discuss related work. Section 6 concludes our work.

2. MODELS AND REQUIREMENTS

First we introduce important models for sensing the real world through a WSN, while arguing for the need of a holistic world model. We then present the considered system model and requirements for the proposed world model.

2.1 Models for Sensing the Real World

Usually, the user (represented by the sink) and the WSN agree on one or more models that should be kept consistent during deployment (Fig. 1). The WSN's task here is to create an appropriate model of the physical world, such as the ambient temperature map, as required by the user. The user observes the physical world through this *world model*. Upon deployment, these models are initialized. The WSN should report changes in the model agreed on (model update). The user can query the implemented model or trigger a model replacement if necessary, e.g., through code update techniques. An adaptive negotiable model is also possible in order to allow for evolvable WSN systems. The model update should be incremental in order to minimize communication overhead. Prediction models allow for predictability while minimizing the communication overhead.

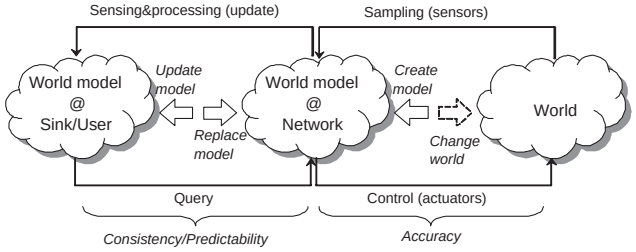


Figure 1: WSN models - overview

Each sensor node (SN) measures the physical signal of interest at a given sampling rate (with a certain noise level). The sampling rate should be tuned as a function of physical world dynamics, which impacts the dynamics of model

data. Subsequently, a SN produces a time series of the signal (*temporal sampling of the world*). *Spatial sampling of the world* is completed by the set of SNs. Both temporal and spatial sampling play an important role in the accuracy and the consistency of the world model. The physical world can be changed through deployed actuators and the network world through network maintenance or reconfiguration.

The design paradigms discussed in the introduction (database, event service and network) are the main existing approaches to realize the world model and its update or query. In order to provide for holistic abstractions, we provide, in Section 3, our Map-based World Model (MWM).

2.2 System Model

Our generalized WSN scenario consists of a network that is composed of stationary resource-constraint SNs, a static resource-rich sink and mobile resource-moderate assist nodes (ANs). The SNs generate the raw data necessary to create the desired world model. The ANs are usually involved in model management and less in its generation. Commonly, WSNs are built utilizing hundreds to thousands of cheap SNs.

A data sample is characterized by the ID and the location of the SN as well as the sensor reading and its timestamp. We assume that SNs know their own geographic position. The clocks of SNs are synchronized, e.g., via GPS or any efficient synchronization protocol [28].

2.3 Requirements on the MWM

We identify the following requirements on the MWM:

(a) The MWM should be *frugal and lightweight*, i.e., its creation, management and use require minimal resources with respect to storage, bandwidth and energy. Its management is crucial for its usability as it determines its efficiency. The more centralized (e.g., at the sink) is the MWM, the cheaper is its use but the more expensive is its management. This trade-off is of high importance and will be discussed in Section 3.3.

(b) We require the MWM to support different levels of details (i.e., zoom as required), and easy to *customize* to the mission. The modularity and composability of the model are important instruments to reach these goals, and to support diversified and multi-purpose applications.

(c) The MWM should also incorporate the "network world model" (the WSN) besides the "physical world model" (the science) in a *unified* way. The network world model reflects features of interest, such as indicators of network health, which may be used for detecting and predicting important network events. The MWM should also support pro-active reconfiguration of the network world. This unification would simplify the design of monitoring techniques for both network and physical world, e.g., by allowing for generic solutions that maximize data piggy-backing. To the best of our knowledge, this is the first elaboration of a unified model for both the physical world and network world. The MWM needs to be generic and independent from modeling of the physical or network world.

The quality of MWM can be indicated by means of the following metrics: (1) Accuracy (2) consistency, (3) efficiency and (4) scalability. There is always a trade-off between accuracy and scalability and between consistency and efficiency. These tradeoffs need to be investigated by the WSN designers while building the MWM.

3. THE MAP-BASED WORLD MODEL

We now define the MWM and provide a generalized architecture for WSNs relying on the MWM while retaining existing architectures. Next, we define core primitives such as MWM management, events and queries. Subsequently, we emphasize the viability of MWM for predictive monitoring and pro-active reconfiguration. Finally, we present a step-wise methodology to benefit from the MWM architecture for design and deployment of WSNs.

3.1 MWM Definition

WSNs on the one hand, are inherently embedded in the real world, the goal being to detect the spatial and temporal world's physical nature, such as temperature, air pressure, gas presence, or oxygen density. On the other hand, maps present a powerful/efficient tool to model the spatial and temporal behavior of the physical world being an intuitive aggregated view on it. As stated before, the main objective behind the deployment of WSNs is to create an appropriate model for the physical world. Therefore, without loss of generality, we model the world as a stack of maps presenting the spatial and temporal distributions of the sensed attributes of interest in the physical world (Fig. 2).

DEFINITION 1. *An* MWM *is a set of relevant maps from the real world of interest.*

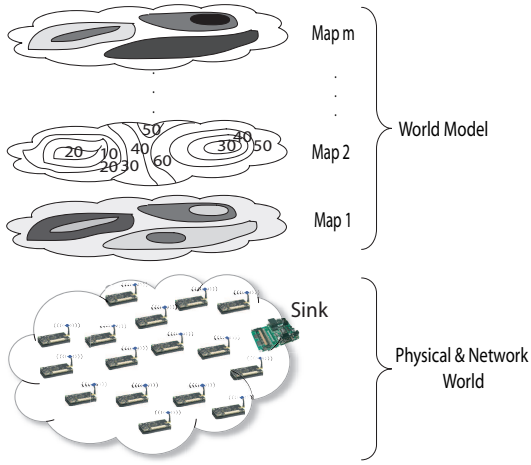


Figure 2: Map-based World Model (MWM)

A *Map* is an aggregated view on the spatial distribution of a certain attribute at a certain time. In the WSN literature we identify two main classes of maps, i.e., the choropleths (e.g., Map 1 in Fig. 2) and the isomaps (e.g., Map 2 in Fig. 2). A map is defined with respect to a certain attribute such as temperature. The map construction groups spatially correlated SNs with similar attribute's values to *regions*. In MWM we define a region by its border (a set of spatial points) and an aggregate (e.g., average) of the attribute's values obtained from all SNs located in the region's area. A map is then the collection of all regions of the WSN.

Following our requirements, the MWM should support both network and physical worlds. We fulfill this requirement by allowing the following two classes of maps: (a) User maps (uMAP) such as temperature, toxic gas and oxygen

maps, and (b) network maps (nMAP) such as residual energy and connectivity maps. This superposition allows for a unified management of both world models.

The main benefit from maps is to abstract from single SNs by addressing regions. While losing sampling details, the map abstraction usually sustains an acceptable accuracy concerning the spatial distribution of sensor readings. Accordingly, the map abstraction presents a natural step to increase efficiency in WSNs. The higher the number of SNs per region, the higher is the benefit. The map abstraction level of the MWM simplifies the design and deployment of WSNs as it can be easily accepted by different parties ranging from the user to network designer, programmer and administrator.

3.2 MWM Architecture

Fig. 3 illustrates the MWM architecture. This architecture retains established mechanisms, builds on top of basic communication protocols and provides valuable support for application design. Network primitives such as unicast, geocast, broadcast and convergecast are essential for query dissemination, event detection and MWM generation and management. The strength of the MWM is to hide communication and sensing details simplifying the design of queries and event detection and prediction even on resource-constraint nodes.

The application interests and requirements define the maps composing the MWM as well as the parameters needed for the construction of these maps. The user may use predicate-based or SQL-like language to specify, delete or modify interests and queries. An example of predicates is "the observed temperature is higher than a certain threshold". The event fire is then equal to the predicate value. Note that the event definition requires fixing the threshold value.

The query and event services (Fig. 3), provide powerful primitives for the application design and implementation. These services interact with each other as events can be realized through continuous queries and queries can be triggered by events. As mentioned in the system model, synchronized clocks and location information are important services for WSNs, which can be easily realized through existing protocols. The query and event service act then on MWM to answer the user's queries or to notify events of interest. MWM allows for addressing regions instead of single nodes, leading to the optimization of queries, event detection and prediction as we will detail later in this section.

From the WSN literature, we identify tinyDB [4] with its extensions as an established query service and pub/sub [11, 12, 13, 14, 15, 16] as an event service. As mentioned before, TinyDB treats sensor data as a single table (sensors) with one column per sensor type. TinyDB addresses single nodes. TinyDB queries are SQL-based and support selection, projection, and aggregation. The main reasoning behind pub/sub is the flexible grouping of nodes as publishers and subscribers by defining interest channels. The grouping of nodes is not fixed and depends on local decisions of nodes. The pub/sub architecture has been defined for WSN to support multiple sinks and to allow for multi-mission systems. Therefore, this architecture is orthogonal to the MWM approach and we suggest to deploy it for the management of distributed and/or replicated MWMs.

Unlike pub/sub and tinyDB the node grouping in MWM depends only on the spatial correlation of the sensor readings, which is desired for all WSN applications. In the MWM architecture, we allow for the integration of tinyDB and pub/sub (Fig. 3), allowing for an easy add-on of MWM in existing WSN deployments. MWM query service goes into tinyDB if each region is formed by one single SN. TinyDB is mainly a reactive service while pub/sub principally operates pro-actively. Our MWM is hybrid as the map construction can be completed in a pro-active way and the MWM query in a reactive way. There is always a tradeoff between pro-activeness and reactiveness that should be considered depending on the query frequency and MWM dynamics.

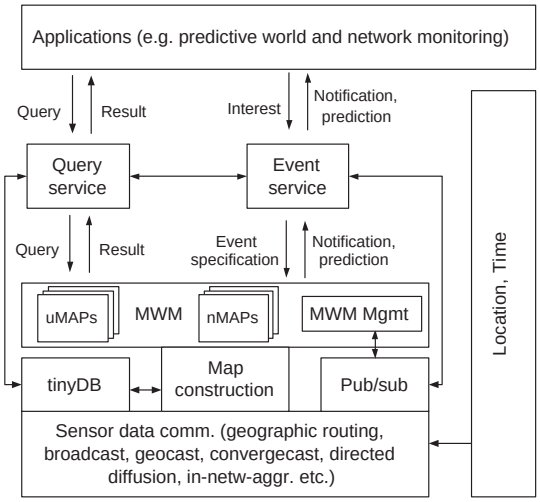


Figure 3: MWM architecture

3.3 MWM Management

The core question for the MWM management is to address (a) how much pro-activeness is needed in the MWM, (b) where to build and optionally replicate it (on sink or elsewhere within the network), and (c) how to maintain the required consistency level through updates. Necessary for investigating these issues is to fix the generators of the MWM and its users. As mentioned before, it is worth to investigate the pub/sub architecture for the management of the MWM and especially in the presence of mobile AN. Here the MWM generators will be publishers, its users the subscribers, and its managers the dispatchers.

Generally, the MWM management is the creation and maintenance of local, partial and global views on different nodes. A *local view* of a certain node is its knowledge about the MWM without the assistance of any other node, e.g., the values returned by its local sensors. A *partial view* is provided if only a part of the MWM is locally available at the node. This is for example the sensor readings of the neighboring nodes or the properties of the own map region. The *global view* is the knowledge of the entire MWM.

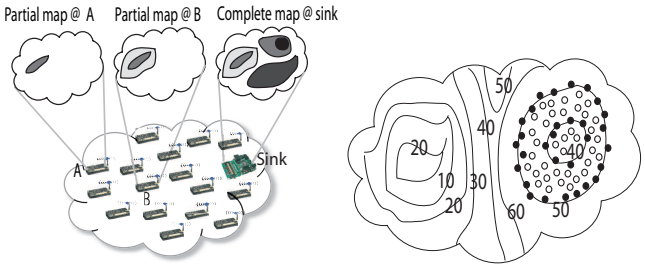
The MWM layer (Fig. 3) customizes the MWM by specifying the needed maps and the map knowledge needed by each node. This knowledge may range from simple local view to region view (partial view) to the map view (global view). The management of the MWM can be easily transformed into the management of a set of maps. The update of the MWM is completed by the update of the different maps/regions composing it. The MWM approach allows

for dynamically inserting and removing maps terming an MWM's re- and pro-activeness besides its reconfigurability and evolvability.

The underlying layer for map construction provides basic primitives for the region and map management at different nodes, i.e., their generation and update (Fig. 3). The management of the individual maps can be considered separately. This allows for a differentiated management for different maps composing the MWM: Some maps can be managed centrally on the sink, others fully distributed, etc. For the construction of the MWM, we should investigate opportunistic construction of maps (e.g. by maximizing message piggy-backing). Update of local views on the map may trigger updates to external view on the map (e.g. on the sink). If many nodes manage partial or global MWM views, pub/sub can be utilized for an efficient management of the replicated views or maps.

From the literature, we identify basic management functions that can be easily used independently of the map of interest (e.g., map construction on the sink and its update). The naive approach to construct a map on the sink would be if each node reports its value to the sink using multi-hop communication. This is obviously inefficient. Consequently, more efficient approaches have been developed using techniques such as in-network aggregation [20, 29, 30]. Other approaches use suppression mechanisms to reduce the number of nodes reporting their raw readings to the sink [31, 32, 27]. "Abstract Regions" [19, 33] is a family of spatial operators that allows nodes to form groups with the objective of data sharing and reduction within the groups by applying aggregate operators.

Fig. 4 (a) shows how the residual energy map is created by the eScan [20] approach and managed in the network. The eScan approach is incremental. Consequently, the closer a SN to the sink, the more global is its view on the map. If the Isolines approach [27] (Fig. 4 (b)) is implemented to create regions, then nodes located on the isolines (filled points) know that they are isoline nodes, and all other nodes know only their local views, i.e., own sensor readings (non-filled points).



(a) eScan (b) Isolines

Figure 4: MWM management examples

3.4 MWM Queries

A query is a user's request of data/information to the WSN. Most of existing querying languages in WSN are SQL-like [7, 8, 34, 4, 35]. For instance, queries in the established tinyDB approach [4], as in SQL, consist of a SELECT-FROM-WHERE clause supporting selection, projection, and aggregation. The main drawback of these approaches is that they address the WSN at the SN level, which limits the efficiency of query optimization, dissemination and processing. In order to define queries, a data model should be defined. Depending on this data model, the query should be transformed (optimized) and disseminated, the result generated and then reported to the querying instance. The data model in MWM is reflected by the stack of maps on different nodes. We define a query as follows:

DEFINITION 2. *A* query *is a request for a certain map, region or event in the physical or the network world.*

In MWM we continue using an SQL-like query language. However, we query maps and their regions instead of querying single SNs. Considering tinyDB, the user can query temperature using the following SQL query: `SELECT nodeid, temp FROM sensors WHERE temp > threshold`. In MWM we substitute nodes by regions. The SQL query above is then transformed to: `SELECT regionid, region_temp FROM tMAP WHERE region_temp > threshold`,
where tMAP is the temperature map.

It is worth mentioning that by abstracting and hiding information about the positions, capabilities and states of individual SNs, the querying of the WSN is simplified and made more natural. Acting on maps may significantly reduce the number of nodes involved in query processing. This is clear if the number of regions constituting the map is very low as compared to the total number of nodes. If the number of regions in the map is close to the total number of nodes then the MWM query processing converges to the tinyDB approach. Queries are in general a reactive service as they present a request-response operation pair which are timely correlated. The proactive map construction increases the efficiency if the applications frequently query the WSN with strict real-time requirements. Otherwise an on-demand map construction is also suitable. In addition, the result is more manageable for the application if it expresses a region and not a set of node IDs.

In addition, a query can be a long-term interest, i.e., a continuous query. An example of tinyDB continuous query for fire detection is: `SELECT nodeid, temp FROM sensors WHERE temp > threshold SAMPLE INTERVAL 1s`. In MWM a continuous query acts on regions and can be specified as follows: `SELECT regionid, region_temp FROM tMAP WHERE region_temp > threshold tMAP UPDATE INTERVAL 1s`.
We note here that for the temperature map we define only two classes for attribute values: Normal temperature and fire temperature. Therefore, the map is composed of only one non-event region and a very few event regions. Consequently, the benefits of applying the continuous query to the map is very high as compared to applying it to single nodes.

Map construction on the sink is also a continuous query. The sink disseminates the interest for a certain map. SNs reply with their current values immediately and later only with necessary updates (Fig. 3). This query should fix: (a) The map of interest and (b) the update model, i.e., when to send an update and to which node. For the eScan [20] approach the query is on the residual energy of SNs. The query is disseminated using flooding creating a tree having its root at the sink. The update model is aggregation-tree based, i.e., nodes send their updates to their parents which aggregate the updates of all leaves.

MWM supports the following broad classes of queries: (a)

Physical world queries that request user-centric situations (one or more map attributes) such as target location, and (b) network health queries which are queries over the network itself, such as in which regions the residual energy of nodes is below a certain threshold. On a centralized MWM the query should be directed to the sink and on a distributed MWM to the relevant regions.

3.5 Events and Situations in MWM

Real world events are generally identified by a spatially correlated value change of one or multiple attributes. This is an important step to transform physical world issues into the MWM. This step requires a deep understanding of the domain and may need to involve the corresponding domain and sensor experts. Real world events should be transformed into MWM events while considering the user requirements.

Usually, a user is interested in simple events of the real world (both physical and network) such as fire or network partitioning and its properties such as perimeter, progress gradient and epicenter. Additionally, a user might be interested in composite events such as explosion (fire + acoustic + light), and in situations in the world such as oxygen or residual energy distribution. Similarly, the user might require the prediction of events or situations with a certain accuracy. In the following we review the main existing approaches and define events and situations in the MWM.

Most of the existing approaches on event detection in WSNs are based on defining thresholds and are network-topology-based. Nodes transmit their local suspects to parent nodes, which apply data fusion to decide if an event has occurred, e.g., [36, 37, 35, 38]. Also a continuous query, as discussed earlier, allows for event detection. These works do not express the progress (trends) of the event after its occurrence, which is a feature of high interest to support post-incident operations. A few recent papers emphasize the event-region view such as the distributed event-region detection [39, 40, 41, 21, 22] or the detection of event-boundary [42, 25, 26, 43]. The map-based event detection using contour matching has been defined in [21, 22]. [26, 43] also rely on the map paradigm to track and query event region contours, which highlights the importance of our MWM. Our approach is inspired by the map-based event-region detection while generalizing and systemizing it for both physical and network world events.

In the following, we show that our map-based approach allows for the definition of simple and composite events, as well as arbitrary real world situations and therefore presents a holistic design paradigm for WSNs. As argued in [1] states of the real world (situations) generalize the event notion in WSNs as events describe only a state change. In the MWM we consider the generalized approach of real world situations, which are easily reflected by the map in a first level of detail and its regions for a finer level of detail.

Definition 3. *A* real-world situation *is a snapshot of the real world of interest at a particular instant of time t. The corresponding situation in MWM is the collection of all relevant maps at time t.*

Definition 4. *An* event *is one atomic situation transition, i.e., a change of one single map of the MWM.*

In MWM we define an event as a predicate $P(attr_1, ..., attr_k)$, where $attr_i$ is the attribute of map_i of the MWM. The event definition is user-centric. Besides simple thresholding, geometric pattern matching can be easily supported by MWM. Event specification, detection and prediction as well as operations on events are then transformed into pattern matching problem. Event detection can be also realized using as a continuous query. Each event is well defined by a certain type, validity time, and validity region.

A *composite event* is a pattern for multiple events. Event composition is defined by a function of simple (or other composite) events and algebraic operators. Operators consist of conjunction, disjunction, concatenation, sequence, concurrency, iteration, negation, selection, spatial restriction, and temporal restriction [44]. One of the main benefits of MWM is to map several operations into temporal geometric operations. An example of composite events can be the following: An explosion is detected, when the temperature, acoustic, and light sensors detect excessive heat, a bang, and a light flash, respectively [1]. An explosion is detected if the three events occur with overlapping validity intervals. We note that the MWM approach allows for defining context-sensitive events. These are events that are of interest under a certain context, e.g., the dissemination of a chemical should be reported to the fire fighters only outside the normal manufacturing phase.

The requirements of the user on the accuracy of the event and situation detection should be considered during region and map construction. Degradation of map accuracy may lead to false positive or false negative detections. One well-known problem concerning accuracy is the outlier observations, which are false positive events. The outlier problem is easily handled during the map construction, where a few outliers can be tolerated within a single region. Furthermore, the event detection latency that can be tolerated might impact the design of the algorithms for constructing regions and maps.

3.6 Predictive Monitoring and Proactive Reconfiguration

We highlight the benefits of our developed MWM approach for predictive monitoring of both physical and network real world events and situations as well as for proactive reconfiguration of the network.

Predictive Monitoring: It is challenging to predictively monitor sensor fields over time, i.e., combining data from both the spatial and temporal domains. A history of the MWM may support the detection of events that persist over longer periods of time. Efficient long-term data collection from the WSN can enable fine-grained trend analysis and event prediction. Therefore, protocol designers should manage relevant MWM snapshots in a MWM history for an appropriate time window w. Future snapshots can then be predicted with a certain accuracy. There is only limited existing work on event prediction in WSNs. The dual prediction scheme (also called dual prediction reporting) has been shown to lead to potentially high communication and energy savings once adequate prediction models are used [45, 46, 47, 48]. Different approaches have been proposed to perform time series forecasting in WSNs [46, 47], ranging from simple heuristics to sophisticated modeling frameworks. A simple autoregressive model can be used for predictability. Typically, the model to use (e.g. constant, linear, etc.) is fixed a-priori, while model parameters are estimated on the basis of incoming data [46, 47]. In [48], the authors present

adaptive model selection instead of using a fixed model.

The main drawback of these techniques is that they act on single SNs, which limits their efficiency in terms of number of message and processing complexity. However, most of these approaches can be easily integrated into our MWM model acting on regions and maps instead of single nodes. The main benefit of MWM is the appropriate abstraction level for prediction provided by the map without sacrificing much the accuracy of predictability due to the sacrifice of time series accuracy. Especially, when the average number of nodes per region is high, the processing complexity for prediction and the storage overhead for the map history is minimized, allowing for an efficient in-network predictability on resource-limited devices such as ANs.

Proactive Reconfiguration: Sensing the physical or the network world constitutes a first step towards a reaction such as actuation back to this world. The maps of the MWM allow for an optimized and goal-oriented spatial intervention (e.g. network maintenance) and navigation (e.g. for mobile entities such as autonomous vehicles and users) in the sensor field. In particular, the pro-active reconfiguration and maintenance are of high interest for future WSN given the growing reconfigurability of entities with respect to hardware and mobility. The MWM allows for an efficient event-driven triggering of reconfiguration and for map-based assessment of reconfiguration options. Examples of MWM supported proactive reconfiguration are the following: (a) Map-based sensor re-tasking and re-programming, and (b) reactive and pro-active node placement to maximize data collection and to provide for self-healing and graceful degradation (e.g., by delaying network partitioning).

3.7 MWM-based WSN Design

The main design benefits arise from MWM being holistic and relying on an abstraction level that is accepted by users as well as application and network designers. WSN designers are usually overwhelmed by a vast number of primitive low-level design issues such as node level communication. It is invariably left to the designer to implement the details of communication, making it both complex and error-prone. However, the designers would benefit from a higher-level generic view. A widely accepted higher-level view is given by the maps, which allows for a simplified but holistic design of future WSNs. Maps also provide an intuitive abstraction level for debugging WSNs.

MWM provides an abstraction level that may simplify the interconnection of heterogeneous and autonomous WSN systems, which will play a major role in future WSN research (e.g., SensorGRID [49] and SensorWEB [50]). In general, the MWM provides a generalized method to deliver (event-driven) context for context-aware applications in ubiquitous computing. Moreover, maps provide for an appropriate way for the standardization of cooperative WSNs. Protocol and system design for WSN normally include a heuristic design phase that is followed by validation and optimization phase. Our MWM approach is well-suited for these phases.

We now present how one can benefit from MWM for system design and deployment. We describe a set of five necessary steps for building and configuring a typical MWM in general with the purpose to detect and predict events and situations in the physical and network worlds as well as proactive reconfiguration. We show the application of this step-wise design approach for the presented case studies.

`STEP 1`: Identify the problem, i.e., by specifying the situations and events of interest from the physical world (this phase involves domain/sensor experts and users) or the network world (involving network designers and administrators). This is also the appropriate time to fix the user requirements such as on the quality of observation/information (e.g., the event detectability, predictability and accuracy).

`STEP 2`: Identify the required maps and define events and their operations, queries etc according to the MWM specification.

`STEP 3`: Sketch a solution for the identified problem assuming an MWM global view given by all maps (e.g., at the sink). The general solution consists in event/situation specifications, and detection/prediction techniques.

`STEP 4`: Determine the required MWM knowledge of each node in the network while minimizing the needed global knowledge, the overhead for the MWM's creation, distribution and management. This MWM customization should retain the validity of the sketched solution. The globally sketched solution can now finally be designed for the fixed MWM specification. The main result is then a set of (mostly distributed) detection and prediction algorithms.

`STEP 5`: Select requisite primitives (broadcast, unicast etc.) for intra- and inter-region communication.

4. CASE STUDIES

We illustrate some of key aspects of the proposed MWM for WSNs by presenting two simple case studies. Each study covers an example of the broad classes of the physical and the network world events: Fire and network partitioning. For simplicity, we consider a WSN composed of static SNs and one single static sink. We assume that the WSN is connected at the time of deployment. We deliberately suppress details since our objective is to highlight the MWM concept.

4.1 Detecting and Predicting Fires

We briefly describe how the MWM approach simplifies fire detection process and compare it to the existing WSN-based approaches [51][52]. Beyond fire detection, we also sketch a method for prediction.

`STEP 1`: Fire event detection is safety-critical and consequently should be responsive, i.e., reliable and timely. Additionally, the perimeter of the fire is valuable and should be reported to the user. We assume that the user also requires receiving an early fire warning.

`STEP 2`: For detecting fire, various maps are of interest such as temperature, humidity, barometric pressure and wind maps. For the sake of simplicity we consider the MWM is composed of tMAP only. Using this MWM, we define the event fire as follows. A fire event occurs, if a region is formed with a temperature value higher than a certain threshold $T_{th-detect}$. To allow for prediction we proceed as follows. If the temperature approaches the $T_{th-detect}$ but is still below this value, then an early warning should be delivered to the user.

`STEP 3`: Assuming the global view, i.e., the complete tMAP at the sink, now we sketch a solution for the detection and prediction of fire. First of all, we have to derive the parameters for the map construction from the application requirements. The key idea will be to detect the no-event, pre-event and event regions of the tMAP. We suggest to use a simple thresholding method. We divide the temperature range in three bands, i.e., no-event if $region\_temp < T_{th-predict}$,

early warning, if $T_{th-predict} \leq region\_temp < T_{th-detect}$, and event report, if $region\_temp \geq T_{th-detect}$. A more fine-grained division of the band $[T_{th-predict}, T_{th-detect}]$ is also possible for more accurate predictability. We assume an isomap according to this thresholding at the sink. In the normal case the map is composed of one single no-event region. When a fire breaks out, then an event region is created and the event (region and value) is reported to the user. Around this region a pre-event region may also be present and reported to the user. Generally, a pre-event region is created which is followed by a event region. For this scenario the minimal local knowledge required by SNs is only their own temperature values. For this configuration the storage overhead is optimal. However, to construct the tMAP the sink requires the local view of each SN. For this purpose one can use the different schemes available in the literature such as [20][27][31].

`STEP 4`: For the solution presented in Step 3, the sink has the MWM=tMAP global view. Now, we improve the solution by minimizing the needed global view and maximizing the localization of algorithms. Assuming an isomap construction following the Isolines approach [31] and the thresholding method described above, SNs exchange their readings with their one-hop neighbors. A node starts this sharing phase only if it changes its temperature band. Only nodes on a region border keep track of being border nodes, all other nodes have only local views. Accordingly, we can design a simple solution, where border nodes of pre-event or event region report their positions and the values to the sink. That all border nodes report their values to the sink is important to know the region and it allows also for avoidance of false positives and false negatives.

`STEP 5`: In this step, the designers should fix the communication primitives needed to implement the algorithms sketched in Step 4. They include broadcast for neighbor discovery and border node identification, and convergecast routing protocol for sending information from border nodes to the sink. Additionally, some existing suppression techniques can be deployed to reduce the number of reporting border nodes.

In non-map based architectures all nodes whose temperature is above threshold will report to the sink [52]. In [51], the authors use sensors for temperature, precipitation, relative humidity and wind speed in order to provide early warning of a potential forest fire, and estimate the scale and intensity of the fire if it materializes. They deploy cluster-based aggregation of all the sensed values (calculation of the so-called Fire Weather Index (FWI)) and report them to the sink where the detection and prediction is achieved. The aggregation is simplistically completed independent from the values of sensor readings. However, the MWM approach accounts for current sensor readings and subsequently enhances the efficiency of event detection. Further sophisticated approaches such as directed diffusion [53] usually aggregate the data but they explicitly do not define and report the perimeter of the event region. This clearly shows the benefits of utilizing maps for event detection.

4.2 Predicting Network Partitioning

Now we qualitatively investigate the detection and prediction of network partitioning. We assume that energy depletion leading to a node crash is the reason for network partitioning.

`STEP 1`: The problem we consider here is the network partitioning, i.e., if either a set of nodes run out of energy, or a set of nodes can not reach the sink anymore. Network partitioning limits the sensing coverage of the WSN and requires its maintenance. For safety and maintainability reasons, we require the prediction of network partitioning.

`STEP 2`: The map of interest for detecting network partitioning is the connectivity map (cMAP). In order to allow for prediction, the energy map (eMAP) as well as the connectivity map are required. For simplicity, we start with a connected WSN and focus only on the prediction of network partitioning. Subsequently, the MWM is composed of the eMAP. We observe the energy level of the different regions of the eMAP and as this energy level approaches a predefined low level (E_{th}), an early warning is reported. A fine-grained strategy for early warning reports is also possible. For instance we can adapt the reporting frequency to the energy level of the region.

`STEP 3`: Now we sketch a solution assuming the knowledge of the eMAP at the sink. Recording the different eMAP regions that run out of energy allows to predict the isolation of some energy-rich regions. The key idea is to monitor the weak energy regions of the eMAP and to use a regression algorithm to predict the time the regions run out of energy and the time when the "dead" regions isolate other energy-rich regions (Fig. 5).

`STEP 4`: For prediction we do not need global map knowledge at the sink and not during the complete WSN lifetime. It would be sufficient if only the regions of weak energy report this state to the sink that predicts the network partitioning (temporal suppression). Furthermore, only a few node of each region (e.g., border nodes) has to report their value to the sink (spatial suppression). For eMAP, we suggest using the Isolines approach [31] to construct the map and to consider the following global classes of residual energy: 0-E_{th}% and E_{th}-100%.

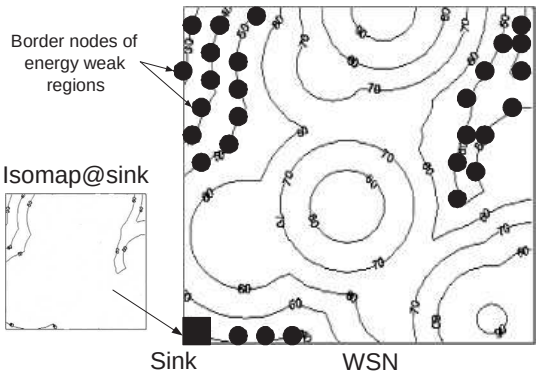


Figure 5: Prediction of network partitioning

Having the incremental reports (locations and energy values) of border nodes (isoline nodes), the sink can now depict the relevant isolines and predict the coverage drop of the corresponding region. This also allows to predict if some other regions in the network will be isolated as a result of the coverage drops of some regions.

`STEP 5`: This step is similar to Step 5 of the previous study.

From the literature, we identify approaches to detect [54]

and suspect [55] network partitioning but no work to its prediction. [54] detects linear partitions,i.e., cuts. Partitioning is detected by monitoring a small subset of sensor nodes, called sentinels. The position of sentinels is calculated on the sink, given the position of all sensor nodes. The sentinel nodes periodically send a heart beacon. When the sentinel nodes are not reachable anymore, the sink concludes that the part of the network where this sentinel node is located is partitioned. The main drawbacks of this scheme are (1) the frequent blind reports to the sink, (2) not all forms of partitions can be detected by this approach (only linear cuts but not holes (dead regions) nor islands (isolated regions)) (3) the poor accuracy of this detection approach as if a sentinel node crashes, it is not always a sign of network partitioning. Furthermore, this approach does not provide information about the region really affected by partitioning, which complicates the maintenance options. The Partition Avoidance Lazy Movement (PALM) protocol for mobile sensor networks [55] is a decentralized approach, where a SN can locally suspect network partitioning and move to avoid it. The PALM scheme assumes that each SN knows its own position and the position of the sink. The SN periodically collects the position of all its neighbors and checks if some neighbors are located in a small angle towards the sink. If no neighbor is located in this "promising zone", the SN suspects network partitioning. The main drawback of this scheme is that every SN has to periodically broadcast its location and to blindly check if network partitioning is suspected.

5. RELATED WORK

Modeling Techniques in WSN: Modeling is the common approach to master the complexity of WSN. Examples are network models, simulation models, mobility models and analytical models to mention a few. Existing models are purpose-oriented models but provide important models that are orthogonal to the world modeling and therefore complementary to our MWM. Additionally, further modeling techniques from the Geographic Information Sytems (GIS) literature and from the spatial temporal databases can provide important techniques such as modeling languages and standards for our MWM. The several existing tools such as the Space Time Toolkit [56] can be used for map data analysis. In particular, we cite the existing modeling languages such as SensorML, REACTIVEML and LUSSENSOR, which simplify the specification of our MWM. In this work we do not focus on the MWM specification and formalization and keep it for future work.

Modeling the Real World: We are not the first to define world models. Such a modeling technique is well established for Augmented Reality or Virtual Reality [57]. The most related models to us are those for ubiquitous computing and sentient computing. (a) The ubiquitous computing [58][59][60] models are optimized for complex physical worlds and rely on powerful infrastructure that creates and manages the model for mobile clients. In these projects real world events have been defined. In [61], the author investigated the observation of complex high-level physical world events through a world model managed on internet servers. Many related concepts can be adopted if our MWM is centrally stored and managed by the sink. However, in this paper we argued for a distributed MWM, i.e., only partially centralized at the sink, given the strong resource constraints for the WSN prohibiting a full data collection at the sink. (b) Sentient computing deploys sensors to perceive the environment and react accordingly. One use of the sensors is to construct a world model in the infrastructure, supporting location-aware or context-aware applications [62][63]. Sentient computing focuses only on indoor scenarios and ambient intelligence applications. All these world models do not fulfil vital requirements for a world model for WSN. In particular, they are not frugal and do not support a network world model in a unified way.

6. CONCLUSIONS

As maps provide a widely acceptable abstraction, circumventing the problem of addressing single nodes in a WSN, we have developed a map-based system architecture. The proposed intuitive and lightweight map-based world model (MWM) uniformly models both the physical and the network world using maps. Besides predictive monitoring the MWM provides an important decision base for pro-active WSN reconfiguration to enhance WSN functionality, dependability or security. Our architecture is flexible and presents a powerful tool for both designing and deploying WSNs. This has been elucidated by two case studies.

In future work, we are planing to refine MWM by formalizing the definition of the querying language and developing efficient/frugal algorithms for replication and consistency of the MWM on the sensor, assist and sink nodes. In order to simplify the access to MWM for the community, we aim to develop extensions to common simulators such a Tossim and platforms such as TinyOS. In particular we aim to provide add-ons and tools for map-based logging and map-based visualization of simulation and real data.

7. REFERENCES

[1] K. Römer and F. Mattern. Event-Based Systems for Detecting Real-World States with Sensor Networks: A Critical Analysis. In *DEST at ISSNIP*, 2004.

[2] J. Elson and D. Estrin. *Sensor Networks: A Bridge to the Physical World*. Chapter in the book Wireless Sensor Networks, Kluwer Academic Publishers, 2004.

[3] D. Estrin et al. Instrumenting the World with Wireless Sensor Networks. In *ICASSP*, 2001.

[4] S. Madden et al. TinyDB: an Acquisitional Query Processing System for Sensor Networks. *ACM Trans. on Database Systems*, 30(1), March 2005.

[5] P. Bonnet et al. Towards Sensor Database Systems. In *MDM*, 2001.

[6] Y. Yao and J. Gehrke. Query Processing for Sensor Networks. In *CIDR*, 2003.

[7] S. Madden et al. The Design of an Acquisitional Query Processor for Sensor Networks. In *SIGMOD*, 2003.

[8] H. Wu et al. Distributed Cross-Layer Scheduling for In-Network Sensor Query Processing. In *PerCom*, 2006.

[9] M. Sharifzadeh and C. Shahabi. Supporting Spatial Aggregation in Sensor Network Databases. In *GIS*, 2004.

[10] D. Chu et al. The Design and Implementation of A Declarative Sensor Network System. In *SenSys*, 2007.

[11] P. R. Pietzuch et al. Composite Event Detection as a Generic Middleware Extension. *IEEE Network*, 18(1), 2004.

[12] P. Costa et al. Publish-subscribe on Sensor Networks: A Semi-probabilistic Approach. In *MASS*, 2005.

[13] J. Steffan et al. Towards Multi-Purpose Wireless Sensor Networks. In *SENET*, 2005.

[14] E. Souto et al. Mires: A Publish/subscribe Middleware for Sensor Networks. *Personal Ubiquitous Comput.*, 10(1), 2005.

[15] M. Sharifi et al. A Publish-Subscribe Middleware for Real-Time Wireless Sensor Networks. In *International Conference on Computational Science*, 2006.

[16] J.H. Hauer et al. A Component Framework for Content-based Publish/Subscribe in Sensor Networks. In *EWSN*, 2008.

[17] V. Srivastava and M. Motani. Cross-layer Design: A Survey and the Road Ahead. *IEEE Communications Magazine*, 43(12), Dec. 2005.

[18] D. Gracanin et al. On Modeling Wireless Sensor Networks. In *IPDPS*, 2004.

[19] M. Welsh and G. Mainland. Programming Sensor Networks Using Abstract Regions. In *NSDI*, 2004.

[20] Y. Zhao et al. Residual energy scan for monitoring sensor networks. In *IEEE WCNC*, 2002.

[21] W. Xue et al. Contour Map Matching For Event Detection in Sensor Networks. In *ACM SIGMOD*, 2006.

[22] M. Li et al. Non-Threshold based Event Detection for 3D Environment Monitoring in Sensor Networks. In *ICDCS*, 2007.

[23] M.V. Machado et al. Data Dissemination in Autonomic Wireless Sensor Networks. *IEEE Journal on Selected Areas in Comm.*, 23(12), 2005.

[24] O. Goussevskaia et al. Data Dissemination Based on the Energy Map. *IEEE Comm. Magazine*, 43(7), July 2005.

[25] K. Ren et al. Secure and Fault-Tolerant Event Boundary Detection in Wireless Sensor Networks. *IEEE Trans. on Wireless Comm.*, 7(1), Jan. 2008.

[26] X. Zhu et al. Light-weight Contour Tracking in Wireless Sensor Networks. In *INFOCOM*, 2008.

[27] Y. Liu and M. Li. Iso-Map: Energy-Efficient Contour Mapping in Wireless Sensor Networks. In *ICDCS*, 2007.

[28] B. Sundararaman et al. Clock Synchronization for Wireless Sensor Networks: A Survey. *Ad-Hoc Networks*, 3(3), May 2005.

[29] J.M. Hellerstein et al. Beyond Average: Toward Sophisticated Sensing with Queries. In *IPSN*, 2003.

[30] W. Xue et al. Contour Map Matching For Event Detection in Sensor Networks. In *SIGMOD*, 2006.

[31] I. Solis and K. Obraczka. Isolines: Energy-efficient Mapping in Sensor Networks. In *ISCC*, 2005.

[32] X. Meng et al. Contour Maps: Monitoring and Diagnosis in Sensor Networks. *Computer Networks*, 50(15), 2006.

[33] M. Welsh. Exposing Resource Tradeoffs in Region-based Communication Abstractions for Sensor Networks. *SIGCOMM Comput. Comm. Rev.*, 34(1), 2004.

[34] P. Bonnet et al. Querying the Physical World. *IEEE Personal Comm.*, 7(10), 2000.

[35] D. Abadi et al. REED: Robust, Efficient Filtering and Event Detection in Sensor Networks. In *VLDB*, 2005.

[36] C.T. Vu et al. Composite Event Detection in Wireless Sensor Networks. In *IPCCC*, 2007.

[37] A.V.U.P. Kumar et al. Distributed Collaboration for Event Detection in Wireless Sensor Networks. In *MPAC*, 2005.

[38] J. Heidemann et al. Diffusion Filters as a Flexible Architecture for Event Notification in Wireless Sensor Networks. In *USC/Information Sciences Institute (ISI-TR-556)*, 2002.

[39] B. Krishnamachari and S. Iyengar. Distributed Bayesian Algorithms for Fault-tolerant Event Region Detection in Wireless Sensor Networks. *IEEE Trans. on Computers*, 53(3), 2004.

[40] A. Dogandzic and B. Zhang. Distributed Estimation and Detection for Sensor Networks Using Hidden Markov Random Field Models. *IEEE Trans. on Signal Processing*, 54(8), 2006.

[41] J. Fang and H. Li. Distributed Event Region Detection in Wireless Sensor Networks. *EURASIP Journal on Advances in Signal Processing, Special Issue on Distributed Signal Processing Techniques for Wireless Sensor Networks*, 2008.

[42] K.-P. Shih et al. CollECT: Collaborative Event deteCtion and Tracking in Wireless Heterogeneous Sensor Networks. In *ISCC*, 2006.

[43] R. Sarkar et al. Iso-Contour Queries and Gradient Routing with Guaranteed Delivery in Sensor Networks. In *INFOCOM*, 2008.

[44] E. Yoneki and J. Bacon. Unified Semantics for Event Correlation over Time and Space in Hybrid Network Environments. In *CoopIS*, 2005.

[45] I. Lazaridis and S. Mehrotra. Capturing Sensor-generated Time Series with Quality Guarantee. In *ICDE*, 2003.

[46] D. Tulone and S. Madden. PAQ: Time Series Forecasting for Approximate Query Answering in Sensor Networks. In *EWSN*, 2006.

[47] S. Santini and K. Römer. An Adaptive Strategy for Quality-based Data Reduction in Wireless Sensor Networks. In *INSS*, 2006.

[48] Y.A.L. Borgne et al. Adaptive Model Selection for Time Series Prediction in Wireless Sensor Networks. *International Journal for Signal Processing*, dec. 2007.

[49] C.K. Tham. *Sensor-Grid Computing and SensorGrid architecture for Event Detection, Classification and Decision-Making*. book chapter in Sensor Network and Configuration: Fundamentals, Techniques, Platforms, and Experiments, 2006.

[50] S. Chien et al. Lights Out Autonomous Operation of an Earth Observing SensorWEB. In *RCSGSO*, 2007.

[51] M. Hefeeda and M. Bagheri. Wireless Sensor Networks for Early Detection of Forest Fires. In *MASS*, 2007.

[52] D. M. Doolin and N. Sitar. Wireless Sensors for Wildfire Monitoring. In *Proc. of SPIE Symposium on Smart Structures and Materials*, 2005.

[53] C. Intanagonwiwat et al. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *MOBICOM*, 2000.

[54] N. Shrivastava et al. Detecting Cuts in Sensor Networks. In *IPSN*, 2005.

[55] K.P. Shih et al. PALM: A Partition Avoidance Lazy Movement Protocol for Mobile Sensor Networks. In *WCNC*, 2007.

[56] vast. *Space Time Toolkit*. http://vast.uah.edu/.

[57] G. Reitmayr and S. Dieter. Semantic World Models for Ubiquitous Augmented Reality. In *SVE*, 2005.

[58] D. Nicklas and B. Mitschang. The NEXUS Augmented World Model: An Extensible Approach for Mobile, Spatially Aware Applications. In *OOIS*, 2001.

[59] D. Dudkowski et al. Efficient Algorithms for Probabilistic Spatial Queries in Mobile Ad Hoc Networks. In *COMSWARE*, 2006.

[60] F. Kawsar. Prithibi: An Open Platform for Digitizing Real World through Sentient Artefact Model. In *Joint Workshop on Next-Generation Computing Infrastructure*, 2007.

[61] M.P. Bauer. *Observing Physical World Events through a Distributed World Model*. University of Stuttgart, Germany, http://elib.uni-stuttgart.de/opus/volltexte/2007/3043/, 2007.

[62] C. Town and Z. Zhu. Sensor Fusion and Environmental Modelling for Multimodal Sentient Computing. In *IEEE CVPR*, 2007.

[63] M. Addlesee et al. Implementing a Sentient Computing System. *IEEE Computer*, 34(8), Aug. 2001.