Developing Advanced Context Aware Tools for Mobile Maintenance

P. Pistofidis * and C. Emmanouilidis **

* Democritus University of Thrace & ATHENA Research & Innovation Centre, Greece (e-mail: pistofid@ceti.gr)  
** ATHENA Research & Innovation Centre, Greece (e-mail: chrisem@ceti.gr)

Abstract: The rapid technological progress in the domains of wireless sensor networks, mobile application development and artificial intelligence has fueled the construction of “smart” environments in both domestic and industrial environments. Such environments are equipped with a wide range of sensors, identification tags and interaction devices, capable of sensing, recording, interpreting and reacting to human activity and presence. Context-awareness constitutes a major challenge for developers of smart services, since both design and implementation are increasingly supported by widely adopted frameworks and well-structured tools. The value adding features of context-based services find their way into many industrial systems through the adoption of mobile computing and its state-of-the-art devices. Acting as a constant context monitoring agent, a tablet or a smartphone can expand the capacity of a shop floor system actor to carry out asset management tasks. Context interpretation and system adaptation are highly specialized processes that essentially use predefined or dynamically build semantics to filter and configure the mobile actor’s access session. In this paper we discuss the context semantics for an IT infrastructure that serves engineering asset management and specifically supports maintenance practice and planning. Evaluating the available software methodologies that can drive the implementation of such an infrastructure, we perform a suitability study for the development of a maintenance management and a condition monitoring portable console. Our goal is to identify and assess the weighted significance for a domain-focused set of functional parameters, used to define a context-based mobile tool scaled for industrial shop-floors complexity.

Keywords: Context awareness, Condition Monitoring, Software Engineering, Mobile Maintenance.

1. INTRODUCTION

Context-aware mobile applications are able to adjust offered services in an innovative, event-driven way. In the functional domain, the concept of an event is tightly connected with the user-defined or, more commonly, the automated detection of a context change (de Reuver et al., 2009). Factors such as active personnel role, proximate industrial environment and currently assigned task constitute appropriate semantics that drive the events of a context aware service, employed to support engineering asset management.

Location awareness is the most easily conceivable dimension of context identification. Portable computing devices provide not only computation function and interaction interfaces, but also act as positioning beacons. Device localisation is just one constituent of the context vector that can also contain sensory fusion logic. To facilitate our study over context semantics in a maintenance system, we first pinpoint the unique features that mobile architectures possess:

Portability – Each portable device operates as an active monitoring agent that can be easily carried in different shop-floor environments. Late implementations allow autonomous operation that surpasses the 10 -12 hour limit of the most heavy-load working shifts. Combined with the fact that recent smartphones/tablets are able to reach such autonomy while powering a WiFi and a GPS module and a capable CPU, one can justify their description as “mobile workstations”. State-of-the-art devices employ sophisticated integration technologies to build hardware configuration that enable diverse context identification. Mobile operating systems (OS) and multi-core controllers are then able to support complex context-aware computation for innovative services.

Reachability – Mobile actors can couple client consoles and probe for or share field expertise in an on-demand and near real-time manner. Portable device implementations offer a wide range of methodologies that support collaborative work, through instant messaging clients, media rich shared spaces and network-adapted synced configuration dashboards.

Accessibility – Mobile networking has been a vital perquisite for high priority tasks that deal with on-site critical events. Wireless connectivity is upgraded with circuits and firmware that support newer versions of 802.11 (g/n/ac). Smartphones and tablets are early adopters of 3G/4G mobile telephony derivatives. Quality of Service is a concern that can be addressed by proper SLAs (Service Level Agreements) paving the way towards professional portable computing.

Localization – Localisation was the main pillar of context awareness in earlier implementations. Though context can be translated through different semantics, placing the mobile actor at its immediate surroundings and its close environment
provides any monitoring system with highly exploitable information for adapting its every aspect of functionality. Localization in maintenance can scale from proximity to single parts and/or machinery equipment; to identifying user’s position in distinct plant sectors, industrial sites or even metropolitan areas of asset presence.

**Identification** – Apart from the device itself, RFID technologies and data exchange frameworks such as near field communication (NFC), allow the identification and the contactless connection with nearby assets, especially for indoor environments asset identification. Thus, indoor identification is facilitated by asset or object tags that can best serve the context-based adaptation of mobile services (Kiritsis, 2011)(Harrison et al., 2010).

To ensure a qualitative result when designing a context-aware tool, it is important to determine the depth and range of the context parameters that characterise its application field. Exploring the semantics of these parameters and weighting their importance in the desired functionality may reveal new context perspectives and/or define strict specifications for implementation. The overall process will allow designers to narrow down the parameters that can leverage reliability in detecting rich informative contexts, and exclude unclear semantics that can mislead by re-arranging the priority and the decisions on the developing phase.

Focusing on mobile maintenance, portable applications must embody mechanisms that handle context in events. Context can then be translated into requests that profile, calibrate and dynamically invoke maintenance reporting, visualisation and data modelling services. A maintenance-oriented set of metrics and specifications have been selected to identify the suitability of well-established design patterns to achieve high availability and enhanced performance. Though following domain-generic context semantics may lead to an acceptable functionality, innovative tools are usually the result of insightful study of field-oriented requirements.

**2. MOBILE MAINTENANCE**

Mobile and portable technologies have already been employed to support the functional needs of maintenance tasks (Emmanouilidis et al., 2009)(Liyeanage et al., 2009). Industrial PDAs and Tablets leverage a condition monitoring system’s ability to interact with mobile actors (maintenance personnel). Though such systems provided the foundation for later adoption of similar technologies, early applications were based on limited hardware capacity and software that was lacking flexibility. Industrial portable devices, both in terms of hardware capabilities and software features, exhibited a lack of common ground (standards, prototypes, frameworks) for shared software platforms and firmware versions (for their circuit modules). The resulting compilations entered the emerging market with all-in-one suites, offering no integration with legacy systems, rare software updates and almost none interoperability with third party technologies.

Recently, mobile operating systems (Android, iOS, WindowsPhone) and portable device vendors have coupled their strengths to introduce a new breed of enabling devices. Yet, application development entered an implementation burst mode, largely addressing single-aspect and non-professional needs. Design patterns for client and server-side software complexity have largely been left unexplored in the name of rapid market domination. Even with the support of potent software frameworks and workstation-level hardware, mobile development still progresses around small scale applications. Only lately the introduction of mobile web applications, along with vendor’s intention to tightly connect the concept of “personal computing” with mobile computing (rather than desktop computing), have managed to target portable complexity at each true scope and depth.

Engineering asset management involves multi-disciplinary teams and decision making processes, with application-focused and field-derived specifications. The corresponding data models and maintenance services require a standardized decentralization of layered and interconnected components. This is a demanding task that web/grid services and rich internet applications have recently efficiently took upon and adequately served. Furthermore, it is a task that portable applications and services gradually come to serve, even more efficiently, due to the native advantage of a context changing user and hosting environment. The question at hand is which design pattern and technology blend can effectively and innovatively address the shop-floor functionality of context-aware engineering asset management.

In the next sections we first examine the context semantics that emerge from the problem space and across the system scope of maintenance IT infrastructure. Next, we survey and analyse design patterns that couple mobile clients with remote servers for delivering context-aware services. The strengths and weaknesses of each approach is mapped into a vector of adequacy values for system specifications termed as crucial for a context aware mobile system. To support our software engineering study we identify the engineering asset management service categories that can benefit from such services and we select a refined set that will participate in our pilot implementation and evaluation phase. Exploring the specifications for each category services, we compute and cross-examine the functional requirements vector with the analysis of design patterns. The results drive our conclusions on the decided architecture and implementation options.

**2.1 Maintenance Context Semantics**

Focusing more on technical implementation details, context is defined as a set of attributes and predetermined values, labelled in some meaningful way and associated with desirable semantics. When assessing context in the most common way, this is typically done by collecting information that describe the user’s location and identity, the time of the service request, as well as various aspects of the user environment and its state. Inside the scope of engineering asset management we adopt a comprehensive and balanced classification scheme, assigning context to one of the following categories (Emmanouilidis et al., 2012):

**User/Agent Context** – It is comprised by the semantics that describe the agent’s identity, system purpose, and a balanced
activity trace within the system’s domain. Maintenance personnel along with smart system units can function/operate in such a dynamic context with multiple transition from instance to instance. An agent context instance may be linked to a manually entered (profile) or system-inferred (history fusion) preferences set, system roles and access credentials. These context instances are most commonly annotated by a vector of attributes. Their values determine the agents’ position parameters (coordinates, orientation, differential distance), along with a valid timestamp. Differential distance and asset proximity are parameters that enable passive position through the identification of a context representing industrial entity (an asset/machine or an agent/sensor).

**System Context** – It holds the semantics that can define the condition state of any agent-supporting device. Such a context may either describe the operational state (i.e. computational load, level of service availability, currently processed job) of a maintenance diagnostics server that facilitates a monitoring multi-agent subsystem, or describe the operational state (battery level, network strength, free memory, number of cached maintenance apps) of a smartphone supporting the maintenance technician. When examining the transition between various context instances of the same device scale, static hardware and software features (display size, OS version, memory size) may well provide solid event semantics for the system adaptation mechanisms.

**Environment Context** – This context class addresses the condition variables that characterise an agent’s operational environment. Sensory data are collected by the same or another collaborating agent to populate a vector of parameters that annotates, along with a timestamp, the context instance of a shop-floor industrial environment. Context transitions from dark to more illuminated places can drive the automatic calibration of the portable device’s display contrast (power-aware operation of mobile units), while noise levels can trigger on and off the haptic (vibration) notifications for smartphone carrying maintenance personnel.

**Function/Service Context** – This context category may include various levels of instance and events complexity. Any agent-assigned function constitutes a context instance whose lifecycle is defined by its runtime duration. A maintenance engineer progressing from reporting an observation to requesting a diagnosis and then entering a work input introduces two service context events. A technician completing his corrective maintenance task and moving on to his scheduled visual inspection is also causing a context event. The system’s ability to capture such events can greatly enhance its ability to effectively prioritise (criticality) and arrange (sequencing, dependencies, constraints) such transition events in service/function workflows. Thus context awareness becomes a powerful enabling factor for maintenance planning, management and resource allocation.

**Collective/Social Context** – This context type is determined by dynamics of group participation and collaborative function of industrial agents (personnel or even sensor and smart systems). The semantics are related to coupling agents, whose linkage facilitates the flow of sampled data (wireless sensor network/WSN collective), structured information (WSN to portable, portable to engineer) and knowledge expertise (system to engineer, engineer to engineer). Each such exchange, along with any topology change (new participating agent, agent withdrawal) can be treated as a context event. Such context change can drive data fusion for system configuration. For example, a technician completing an assigned task can be tracked so that the system can annotate his status to “available” for the next scheduled task. Alternatively, a sensor agent can capture abnormal vibration patterns, invoking a collective operation of the proximate sensor group. This collaboration context change triggers a decision to assign the technician to inspect the specific part. The technician retrieves on-spot report of with the WSN state parameters and shares it with his colleagues in a virtual department dashboard via his portable console. Feedback on his post is recorded by the system which adapts by planning a preventive maintenance action for the monitored machinery.

### 2.2 Client – Server Design Patterns

Software engineering practice has introduced new approaches in mobile development and in general portable computing. The primary motivation for building a mobile-supported system is the benefits resulting from a user experiencing a seamless adaptation of a constantly available system view. This view can optimally be loaded with options that adapt to a traced history, transparently acquired and fused, while tuning systems’ function to the apparent context. Such versatility introduces computational and presentation challenges. The need to balance implementation complexity between portable and (possible) server components is crucial and can determine a system prospects very early in its design phase. Tables 1 and 2 provide an analysis of state-of-the-art methodologies that power the development of mobile clients and server side components of context-aware systems. Each methodology is evaluated according to a set of widely valued features and performance metrics.

### 2.3 Maintenance Service Categories

Engineering asset management is a multidisciplinary domain, technically supported by IT systems that deliver diverse functionality and focus on differently prioritised tasks. These systems often deliver context-adaptive services, driven by context-aware mobile agents. Our study focuses on the following sub-domains and their corresponding tasks:

**Maintenance Management Systems:**
- (MM1) Track, issue and manage work orders
- (MM2) Manage assets, location sites, items and collections
- (MM3) Maintenance planning, scheduling and resource allocation

**Condition Monitoring Systems:**
- (CM1) Report, visualise and trend parameters history
- (CM2) Invoke and visualise feedback from novelty detection, diagnostics, prognostics and planning services
- (CM3) Configure and manage system/process parameters
When dealing with context, frequent updates covering news, highly
planning...piloting and fine tuning different hardware such as the GPS receiver or the sensor circuits. Despite such overhead, piloting and fine tuning different applications and services that request direct access to device almost necessary, w...developers, especially when updates and versioning come in one platform to another has been an exhaustive task for development strategy. Porting mobile implementation from small window of options for late modification is the...issue but an important one, since design decisions allow a...Balancing the traits of a mobile architecture is a challenging...manipulate data units...Server Services...Cloud based...Back...Asset...Management...Maintenance Management Console – The functional specifications of our mobile components indicate that the maintenance management console can be easily served by a hybrid application. HTML, CSS and Javascript technologies can lead to components easily packaged into application modules that operate over various platforms with medium-to-good control over device hardware for context identification (geo-tagging). Maintenance administration and logistics can benefit from the fast development cycles, the platform independence and the small complexity of thin hybrid clients. Furthermore, the small development cost and the non-existent approval overhead (through a market/app-store) of such clients will evidently allow for frequent updates covering increasingly more aspects of maintenance management and offering consistent versioning across all platforms.

On its server side components, the maintenance console can sufficiently support its services with a single remote database, and a backup site. However, maintenance planning and implementation approaches has proven to be the most effective way to assess true conformance to strict performance specification. In Tables 3 and 4 we analyse the importance of certain features with respect to the functional requirements of specific maintenance functions.

3. REQUIREMENTS ANALYSIS

Systems serving machinery sustainability can greatly benefit from a semantically-rich data schema and its corresponding handling mechanisms. A crucial system feature for any type of context awareness is the support from internal functions that efficiently manipulate data units with multiple levels of semantic annotation. Such a function set should be able to perform on-the-fly transformations of diverse maintenance data (from technical measurements to managerial KPIs) into customisable views, highly compressed storable objects and semantically enriched context descriptions.

Balancing the traits of a mobile architecture is a challenging issue but an important one, since design decisions allow a small window of options for late modifications in the development strategy. Porting mobile implementation from one platform to another has been an exhaustive task for developers, especially when updates and versioning come in play. Extensive refactoring of middleware components is almost necessary, when dealing with context-aware applications and services that request direct access to device hardware such as the GPS receiver or the sensor circuits. Despite such overhead, piloting and fine tuning different...
context participation introduce data fusion that can scale in computational complexity when performance and quality is sought. A dedicated server can serve as the secure hosting environment for such services. Even though modelling and context reasoning are challenges for the system capacity, federated cloud services that manipulate critical data on third party sites are not recommended if not essentially needed.

**Condition Monitoring Console** – Moving to more site-oriented mobile components, the condition monitoring console demands higher responsiveness and intuitive interfaces. Operating this client in a shop-floor environment and under the pressure of on-spot practice, indicate the significance of highly adaptive and responsive tools. Context awareness is on higher demand and at greater accuracy (asset-based). Native applications allow better communication/polling of GPS modules, while also engaging the instant exploitation of every sensor available on the device (voice, light, vibration). Though deciding the proper implementation OS is not an easy choice, Android presents itself as the best option, since it supports a wider range of devices (over iOS), and exhibits greater framework maturity for tablet devices (over WindowsPhone).

On the server side a condition monitoring console handles larger volumes of data with respect to more demanding trending and localization functions. Even small scale data analytics and context identification require more federated implementations, whose replication/migration/load balancing can ensure the availability of condition state/diagnosis data of proximate assets, along with the reliability of the corresponding computation services (Sensor Cloud). It is important to denote that though cloud services can optimally map our specifications, a dedicated server can very easily suffice with single site scalable supporting services.

Our research constitutes part of the WelCOM project (Pistofidis et al. 2012). As we progress in our scheduled workplan, we will deliver results that derive from joint efforts and collaborations with three (3) SMEs, one large industrial manufacturing enterprise and two research Centres. One of the project’s deliverables is an intelligent (context aware) mobile maintenance advisor that serves the sustainable asset operation of an industrial end user. The piloting will take place at KLEEMANN Lifts, a large enterprise with global presence in the lifts industry, currently holding more than 3% of the global market. The WelCOM system constitutes a layered platform architecture operating on various levels of the maintenance function (Fig. 1).
Towards these goals we design and develop mobile components that facilitate context awareness to adapt their content in terms of filtered functions and information flows. These mechanisms provide both a system-automated (clustering and classification of proper history) and a user-declared (settings) control on what arrives at the immediate operation console of each shop-floor (technicians/engineers) or executive (managers) personnel. The developed applications will follow a profiled behaviour that prompts, records and identifies the user’s context semantic dimensions. According to the proximate sensed environment, the activity history and the cross user collaboration pattern, the mobile operation console will adapt its interface and content to the perceived context (Fig. 2). The maintenance environment context identification can itself be driven by increasing asset self-awareness, delivered by novelty detection functionality embedded in wireless sensor network solutions (Emmanouilidis and Pistofidis, 2011).

5. CONCLUSIONS

Sustainable asset operation can greatly benefit from context-adaptive tools residing in mobile devices. Wireless condition monitoring is currently one of the most flexible and dynamic paradigms for maintenance in sustainable manufacturing and mobile devices present an increasing popular and powerful hosting environment for services that augment the role of shop-floor personnel. Within an e-maintenance framework (Pistofidis et al., 2012), we propose a two-sided mobile system that attributes users with control over managerial and engineering maintenance functions. Such a wireless monitoring tool can bring the following knowledge benefits:

- Adoption by managers, engineers and technicians. Their interaction patterns, their preferences and the utilization level of context based output (reports, alarms, trends, stats) and input (observation, diagnosis, assignment).

- Integration of client implementation with server components and their connection dependencies. Using a context reporting device constitutes an efficient method to map network coverage, and bandwidth sufficiency for the proposed communication and data exchange (synchronization) pattern.

- Suitability of technologies to serve maintenance management and condition monitoring. Currently the first is mainly addressed by Computerized Maintenance Management Systems, while the second has not decoupled all of its implementations from the domain of solid application clients. The progress of our application development will reveal the extent at which hybrid applications and cloud components are suitable constituents for our system. Their market dominance and wide adoption, alone, dictates their candidacy for participating in the WelCOM maintenance platform.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the enthusiasm and contribution by the WelCOM project partner’s staff, namely KLEEMANN Lifts, National Research Foundation, Prisma Electronics, GDT SA and ATLANTIS Engineering. The financial support received through GSRT (grant 09SYN-71-856, project WelCOM) is gratefully acknowledged.

REFERENCES


