Meeting the challenges in the design and evaluation of a trackside real-time safety-critical system

Leonardo Montecchi, Andrea Ceccarelli, Paolo Lollini, Andrea Bondavalli
Department of Mathematics and Informatics
University of Firenze
Firenze, Italy
{leonardo.montecchi, paolo.lollini, andrea.ceccarelli, andrea.bondavalli}@unifi.it

Abstract—**Highly distributed, autonomous and self-powered systems operating in harsh, outdoors environments face several threats in terms of dependability, timeliness and security, due to the challenging operating conditions determined by the environment. Despite such difficulties, there is an increasing demand to deploy these systems to support critical services, thus calling for severe timeliness, safety, and security requirements. Several challenges need to be faced and overcome. First, the designed architecture must be able to cope with the environmental challenges and satisfy dependability, timeliness and security requirements. Second, the assessment of the system must be carried on despite potentially incomplete field-data, and complex cascading effects that small modifications in system properties and operating conditions may have on the targeted metrics. In this paper we present our experience from the EU-funded project ALARP (A railway automatic track warning system based on distributed personal mobile terminals), which aims to build and validate a distributed, real-time, safety-critical system that detects trains approaching a railway worksite and notifies their arrivals to railway trackside workers. The paper describes the challenges we faced, and the solutions we adopted, when architecting and evaluating the ALARP system.**

Keywords—**ALARP, railway, challenges, real-time, safety-critical, architecture, evaluation.**

I. Introduction

Recent years are witnessing an always increasing trend to deploy highly distributed, mobile and autonomous systems in outdoors environments, where they are subject to difficult environmental conditions and a large set of risks, as for example the exposure to cyber-attacks as well as physical tampering, limited and unstable connectivity, reduced power capability. An example of this category of systems is constituted by Wireless Sensor Networks (WSNs), whose applications include e.g., ocean and wildlife monitoring, manufacturing machinery performance monitoring, monitoring of building structures, military applications [24].

Introducing strict requirements on dependability, timeliness and security in this category of systems requires to face both architectural and evaluation challenges.

Architectural challenges are due to the intrinsic nature of this type of systems, as the intrinsic asynchrony of the network, the isolation which leads to potential physical tampering attacks, the environmental conditions which requires robust and fault tolerant systems.

Methodologies, tools and solutions for the evaluation of these systems are required to reliably quantify and assess their dependability, security and timeliness and ultimately prove that requirements are met. The complexity and heterogeneity of the system makes validation a critical task, since performing measurements on a real instance of the system, and exhaustively characterize all interactions in the full details is not feasible for complexity and costs reasons. The greatest challenge here comes from the interactions and cascading effects that modifications in system properties and environmental conditions have on the overall dependability metrics. When several system properties depend on each-other, a slight modification can have unpredictable results.

ALARP (A railway automatic track warning system based on distributed personal mobile terminals, [16]) is a three-year project started in 2009, and co-funded by the European Union close to 4 million euro of costs. The project objective is to design, develop, and validate an Automatic Track Warning System (ATWS) able to detect trains and other rolling stocks approaching a worksite, and notify their arrival to the workers, thus improving their safety. ALARP is a safety-critical, real-time system which consists of fixed nodes, installed in the worksite, and of mobile nodes worn by the railway workers operating in the worksite. All nodes are interconnected and exchange messages in real-time. Such a system clearly shares many of the challenges of highly distributed, mobile and self-organizing systems, in the typically harsh environment of a railway worksite.

In this paper we outline the challenges faced, and we depict the solutions applied, when architecting and evaluating the ALARP system. As ALARP is three-year project started in 2009, results on the many technical achievements have been published, or are being published, to a wide extent; this work does not aim to exhaustively discuss them but it presents proper references where necessary, where further details are available to an interested reader.

The rest of the paper is organized as follows. In Section II we present the ALARP requirements and the related main architectural challenges, and in Section III we show the architecture developed to answer such requirements and challenges. In Section IV we describe the problems related to the evaluation of the ALARP system, while in Section V we depict our evaluation approach. In Section VI conclusions and a summary of the main architectural and evaluation solutions with respect to the key challenges identified are reported.

II. Requirements and Architectural Challenges

A. ALARP requirements

The main task of the ALARP ATWS (Automatic Track Warning System) is to notify working gangs along train lines providing alerts or warnings related to approaching trains and/or on-track plants. This is intended to improve safety of the worksite by supporting or replacing human lookouts. ALARP reliably dispatches *risk events* to the workers; such risk event can be classified as an *alert*, if the worker is in a dangerous area (called *red zone*; that is, the worker is located close to the track on which the train is approaching and consequently is currently at risk), or it can result in a *warning* if the worker is located in a non-dangerous area (called *green zone*). According to the national regulations, the risk event should be delivered with sufficient time in advance to allow the workers to reach a green zone.

Using a network of wearable mobile devices, ALARP also monitors the health of the workers and maps their positions to identify the workers at risk i.e., those not responding due to health problems, or located close to the track while a train is approaching.

To be successfully applied in a railway worksite, the ALARP system must satisfy the following non-functional requirements: i) designed to be self-powered, so that it does not require external power supplies; ii) composed of portable and wearable devices communicating by wireless links; iii) adopt a user-friendly and trusted interface for workers; iv) able to operate in different working places such as open line, stations, tunnels, bridges; v) able to operate in different working conditions (e.g., night or daytime) and weather conditions (e.g., fog, snow, rain, high/low temperature); vi) composed of low-cost equipment and rely as much as possible on Off-the-Shelf (OTS) components to achieve low production costs.

As track-side workers are exposed to risk of harm, ALARP has strict requirements on safety: it is required to satisfy at least the Safety Integrity Level 2 (SIL 2) according to railway standards [21] (railway standards propose both qualitative and quantitative classes for the safety of equipments, and SIL 2 quantitatively means that the Tolerable Hazard Rate per hour THR is required to be between $10^{-7} \leq \text{THR} < 10^{-6}$).

Timeliness is also fundamental because ALARP is a real-time system, able to dispatch risk events within the expected time bounds. Security and privacy are also relevant, especially to authenticate safety-critical messages and to assure that private information, as the workers health status or its position, are not tracked or used beyond what is strictly necessary.

Ultimately, localization accuracy becomes an additional very relevant requirement, because it is mandatory to be able to accurately position a worker in the red or in the green zone, and dispatch alerts or warnings accordingly.

B. Main Architectural Challenges

The above mentioned requirements imply the following architectural challenges for the ALARP system. First, in networks composed by mobile nodes, achieving the required level of reliability, timeliness and security despite the possible harsh conditions (e.g., noise and interferences, low light, bad weather) and despite accidental and malicious faults is particularly challenging, due to the intrinsic asynchrony of distributed and mobile systems. Several factors induce asynchrony in mobile wireless systems, as for example the unreliability of the communication, the nodes mobility, evolutionary changes in the network topology and the consequent absence of continuous connectivity to global resources. Furthermore, threats to resilience, safety, security and privacy are particularly severe: i) device lifetime and communication are severely limited by scarcity of power; ii) use of wireless links means susceptibility to link attacks ranging from passive eavesdropping to active impersonation, message replay, and message distortion; iii) poor physical protection of mobile devices (especially in a hostile environment like the one considered in ALARP) makes them susceptible to physical damage, and vulnerable to theft or subversion; iv) notification to the workers requires a safety-critical, trusted HMI whose outputs are perceivable in any weather and working condition [15].

Finally, outdoors positioning using OTS low-cost GPS devices is usually poorly accurate [23], and leads to a severe uncertainty in the estimation of the position, thus requiring ad-hoc solution for the augmentation of GPS performances.

III. The ALARP SIL 2 Architecture

An overview of the ALARP architecture, focusing on how the abovementioned challenges and requirements are addressed, is reported in this paper with the aid of Fig.1. Exhaustive descriptions on mechanisms and solutions adopted have been published separately and partially referred in the following of the text; for an interested reader, a starting point to achieve details on the ALARP components and related references lists are [17], [18]. The ALARP architecture can be subdivided in three parts: the trackside *Train-Presence Alert Devices* (TPADs), the wearable *Mobile Terminals* (MTs), and the *communication network*.

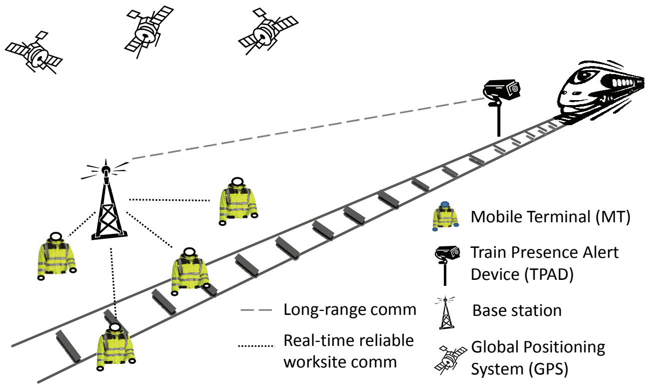
Mobile Terminal (MT)
Train Presence Alert Device (TPAD)
Long-range comm
Base station
Real-time reliable worksite comm
Global Positioning System (GPS)

Figure 1. The ALARP scenario.

A. The Train-Presence Alert Device (TPAD)

A Train-Presence Alert Device is able to sense approaching trains on the monitored track. From a functional perspective, the TPAD is composed of two main states. The first state, called *hibernate*, is designed to be a power-efficient state in which the TPAD is expected to spend most of its operational life. The TPAD consumes less power as possible during the hibernate state, but the drawback is that several false alarms are raised. In fact the sensors applied in this state are "extra sensitive" geophones and accelerometers i.e., they use very high gain detectors circuits, thus providing a high probability of detection, a very low missed detection rate, and a mid-to-high false alarm rate.

When the TPAD suspects a train is approaching, the TPAD moves to a full operational status, called the *triggered* state hereafter. The triggering is performed by a functional block which consists of two sensors typologies for train detection: a geophone and an accelerometer.

When the triggered state is activated, additional sensors (both geophones and accelerometers) are applied to support train detection; these additional sensors allow the TPAD to satisfy the required rate for both false alarms and missed detections. It is easy to note that such higher detection accuracy implies a higher power consumption level for this state. The combination of the detection outcomes of the sensors applied in the hibernate and the triggered state offers high probability of detection (sufficient to meet the SIL 2 requirements on Tolerable Hazard Rate) and a very low false alarm rate.

A TPAD is able to detect approaching trains in all weather conditions, it is self-powered, portable and easily movable in different working places, and equipped with an anti-theft system. The TPAD transmits *risk events* to all MTs operating in the worksite using a dedicated real-time wireless channel (see also Section III.C).

B. The Mobile Terminal (MT)

A Mobile Terminal is a distributed, low cost, wearable, context-aware, trustable and highly reliable wireless devices, which informs the workers about possible approaching trains and/or other events that could put at risk their safety. An MT is able to generate alarms and warnings, and to communicate and interact through wireless connections with other MTs and the trackside TPADs together with mechanisms to check validity and trust levels for this ad-hoc communication.

To improve its resilience, the MT architecture relies on the paradigm of *architectural hybridization*, which allows constructing systems introducing a *wormhole* i.e., a special component that presents improved characteristics with respect to the remaining system components, located in the *payload* [19]. In the MT, clock and clock synchronization services, that are devoted to timekeeping, as well as services with privacy and security requirements are assigned more urgent and trusted requirements than the other services: consequently these are located on the wormhole, whose behavior must be known and predictable, and provided to the rest of the services (in the payload) with certain guarantees at the wormhole interface. The other MT services, located in the payload, include services to realize the functional requirements of the MT and mechanisms for fault tolerance. Payload process uses the wormhole services through a *wormhole gateway*. Payload services do not need to know how wormhole services are implemented, and vice-versa.

Three software layers can be identified in the MT. These are: i) the application layer, ii) the middleware, and iii) the communication layer. The *application layer* has two macro-components: the *localization* component, in charge of locating the worker within the worksite, and the *application logic*, which contains the functional applications required by the MT (e.g., signaling approaching trains, monitoring the health and the position of the worker, showing information to the users through the HMI). The localization component computes the geographical coordinates of the worker, and places him on a pre-loaded map of the worksite; this allows to determine if the worker is in a green or in a red zone. This information is passed to the application logic which raises *warnings* or *alerts* whenever necessary. The localization component requires that the MT integrates a GPS (Global Positioning System) receiver which provides the geo-coordinates of the MT. As low-cost GPS receivers are not sufficiently accurate to reliably identify if a worker is in a red or a green zone, the GPS data is combined with information from *electronic fences* that are placed in the worksite area and separate red and green zones. In the localization component, the fence data is collected and fused with the GPS data: the final result allows determining with high accuracy if the worker is in a red or a green zone [5]. For privacy concerns, the application layer offers only short-term memorization of localization and health data.

The *middleware* provides time-keeping, security and fault tolerant services that guarantee the resilience of the MT. The finite state machine of the middleware includes two states, *low power* and *degraded* states, to specifically address issues on low power, time uncertainty and localization uncertainties. The low power state is entered when the MT has low battery power; in this state the MT provides a subset of its functionalities in order to save batteries. In particular, it minimizes the number of messages the MT exchanges; from low power state, an MT can only move to a safe state or turn off. Instead the degraded state is entered when temporal uncertainty (poor clock synchronization) and spatial uncertainty (low localization accuracy) exceed a given threshold. The MT activates specific procedures to mitigate such uncertainties, increasing the resources devoted to the execution of the algorithms for clock synchronization and localization [18].

The middleware is divided in a wormhole and a payload, and hides to the other layers the presence of these two different subsystems. The *timekeeping services* executing on the wormhole are a clock synchronization mechanism for external clock synchronization (the Network Time Protocol, NTP), which disciplines the software clock using the GPS signal for time reference, and a reliable and self-aware clock able to compute synchronization uncertainty, which is an adaptive and conservative estimation of the distance of the local clock from the reference time. The *authentication service*, again located on the wormhole, is in charge of i) maintaining private information of the worker as its private key, and ii) signing messages using such private key (all transmitted messages are signed with private-public key mechanism for authenticity).

The middleware services executing on the payload are instead mainly monitoring and fault tolerance mechanisms that provide the required SIL of the MT.

The *communication layer* of the MT implements a real-time and reliable communication protocol over the wireless transmission medium. It includes a multi-interface management, allowing worksite communication to take advantage of redundant communication interfaces. This provides physical redundancy of communication equipment, and optionally either transmission redundancy or balancing communication load between different interfaces. The primary deployed communication technology is IEEE 802.11.

The communication layer supports four different states. The *configuration state* is necessary to allow initialization by upper layers; the *normal state* performs all communication tasks including all nominal functionalities; the *degraded state* supports only a minimum level of communication, for example caused by degraded link quality (in this state, transmission load is limited to high-priority messages for safety critical events only); the *energy conserving state* also reduces the support of communication, but with the aim to only use communication interfaces and settings that can minimize the energy consumption in case of low battery situations.

Finally, an HMI has been designed to interact with the workers, especially tailored to provide reliable and safe notification of the *warning* and *alert* signals. Flashing lights (using diodes) installed on protective eyewear are used in order to capture the attention of the worker: a commercially available protective eyewear is fitted with LED lights for the transmission of visual warning/alert signals, where yellow LEDs are arranged above the eyes, and red LEDs are located below the eye. In case of an alert, the red LEDs are flashing, along a warning the yellow LEDs are flashing. As a second communication mean to improve guarantees of notification to the worker, an ear bone conductor [22] is adopted which transmits acoustic signals via vibrations through the skull bone; this allows to transmit acoustic signals that are hearable even in noisy environments. A self-powered watchdog installed in the MT is able to notify a safe state, activating both the visual and audio channels, in case of failure of the MT.

Regarding the hardware of the MT, the wormhole is a simple component which executes on a minimal hardware (a CPU, a RAM, a solid-state memory and an external memory), that ease monitoring and assessing its behavior. There is no watchdog timer: if the wormhole halts or crashes, the payload can notice its failure through its own timers (though in such a case a violation of the real-time requirements must be taken into account). The external memory of the wormhole is a pluggable memory (a smart card) which contains sensible information of the worker, as the worker's personal data and the private key for messages authentication.

The payload executes on a different hardware than the wormhole, and in addition to the fundamental hardware (CPU, RAM, Solid-State Disk, etc.) it requires the following hardware devices: a USB I/O interface, a GPS receiver (used for localization and for time synchronization), a hardware watchdog, an external device for heart-rate monitoring, and the visual and audio devices required to interface with the worker.

C. The Communication Network

The communication network of ALARP is based on a centralized approach, where all MTs use a two-hop communication via a coordinator, also called *base station*. The communication layer offers reliable broadcast and reliable unicast transmissions, which are both implemented by the Timed Reliable Communication (TRC, [4]) protocol, designed on the basis of the protocol presented in [20], and adapted to ALARP requirements and communication scenarios. The protocol is based on a time-slotting approach for polling MT nodes by the coordinator and implemented on 802.11b/g/n. The main protocol characteristics of worksite communication are summarized in the following:

- it is a synchronous protocol, using a round-based Time Division Multiple Access (TDMA) communication scheme.
- the TRC protocol is organized in three steps, corresponding to three different types of interactions between nodes, allowing a) the coordinator *polling* an MT node, b) the MT node responding using a message *request*, and c) the coordinator to *broadcast* the node message, leading to a poll-request-broadcast pattern executed for each MT node.
- the protocol design enforces bounded message dissemination times, by having a worst-case execution time for delivery.
- targeted to primarily disseminate safety-critical events, the protocol does not require (and therefore does not offer) agreement and validity properties. This also permits time savings in terms of a shorter overall worst case message delivery delay.

The communication between the base station and the TPADs is instead based on a bi-directional, Xbee 868Mhz channel and alternatively on cellular EDGE/UMTS; all exchanged messages are signed using public-private keys.

IV. Evaluation Challenges in ALARP

Evaluating dependability-related properties of safety-critical, dynamic, distributed systems in harsh environments poses several challenges. In such systems, performing measurements on a real instance of the system is typically too expensive or not even feasible; for this reason, modeling is extensively used to evaluate system-level dependability metrics.

State-space models (e.g., Stochastic Petri Nets, [25]) are commonly used for dependability modeling of computing systems: they are able to capture various functional and stochastic dependencies among components and allow the evaluation of various measures related to dependability and/or performance. The main problem raised by adopting state-based models is the exponential growth of the number of states at the increasing of system components ("state-space explosion" problem). Although several techniques have been developed to address the challenges raised by large models (e.g., see [26]), in dynamic distributed systems like ALARP, such problem is exacerbated, and state-space modeling methods alone rapidly become impractical.

With respect to model complexity, in such systems the greatest challenge comes from the interactions and cascading effects that modifications in system properties and environmental conditions have on the overall dependability metrics. When several system properties depend on each-other, a slight modification can have unpredictable results. In ALARP, such dependencies exist between the mobility of workers, the localization functionality, and network communication. Modeling such interactions in the full details is not feasible for complexity reasons; on the other hand, completely neglecting them will not provide accurate results.

Hierarchical modeling (e.g., see [27]) is an established technique in reducing the largeness and complexity of models; however finding the proper decomposition of the system is not trivial. Moreover, the individual submodels can still be quite complex, and complementary approaches are needed.

Experimentation is widely recognized as a valuable support to modeling (and vice-versa), at least at the conceptual level, since it can be used to provide values for model parameters, and, most importantly, to reduce model complexity by allowing additional assumptions to be introduced and verified. Although modeling and experimentation have been used together (e.g., see [10], [11], [12]), the two techniques are rarely combined using to evaluate a complex real-life system. Profitably combining the two approaches in the evaluation of a dynamic distributes system requires a systematic approach, in which the interfaces and interactions between the two techniques are well-specified. Attempts in this direction were reported in the context of the DBench [13] and HIDENETS [14] projects.

Maintaining the right level of detail, while accurately modeling all the interaction between system components, is the main challenge in the evaluation of the ALARP system. Moreover, the evaluation approach should be able to adapt, without much effort, to different scenarios and configuration, in order to cope with possible changes in the environment or in system components themselves.

V. The ALARP Evaluation Framework

To summarize, the fundamental challenge is to master the complexity of the system, taking into account its architecture, the different types of components, its communication and localization aspects, and the mobility of workers. This requires focusing on a holistic approach which uses and combines different evaluation techniques applied to the different components and sub-systems, exploiting their potential interactions. The proposed approach combines different evaluation techniques, including analytical modeling and experimental measurements, which can be applied at different abstraction levels. Similar principles have been demonstrated to be necessary to master the complexity of large networked critical systems and to evaluate their dependability [1].

A. The Overall Framework

The essence of the ALARP evaluation approach is the application of several techniques at different abstraction and decomposition levels to solve sub-problems, and the exploitation of their interactions to support the system-level dependability evaluation. Interaction among different evaluation techniques can occur by means of:

- *Cross validation.* A partial solution (i.e., the solution of a specific aspect of the problem) validates some assumptions introduced to solve another sub-problem, or validates another partial solution. For example, a simulation model can be used to verify that the duration of an event in an analytical model follows an exponential distribution.
- *Solution feedback.* A partial solution, obtained by applying a solution technique to a sub-problem, is used as input to solve another sub-problem, possibly using a different method of analysis (e.g., a critical parameter in an analytical model is obtained using experimental evaluation).
- *Problem refinement.* A partial solution gives some additional knowledge that leads to a problem refinement (e.g., the architecture of a component changes because it is identified to create a system performance bottleneck).

Clearly, the decomposition of the ALARP system is not unique: we can identify different system decompositions corresponding to different levels of modeling abstractions. Choosing the particular system decomposition is of primary importance, since it determines the complexity of the identified submodels. In ALARP, we followed the approach of [1], where the decomposition of the system operates at two different abstraction levels:

- *System-level.* We first analyze the system from a functional point of view, and decompose it into a set of interacting sub-systems (i.e., the key ALARP components), each corresponding to some critical functionalities with respect to the validation objectives. Each component interacts with the other components through its interfaces.
- *Component-level.* The first system-level decomposition is then coupled with another decomposition operating at component-level, in which each system component is decomposed into three layers: *user*, *application* and *architecture*.

The *user* layer describes how the users interact with the application; mobility scenarios and application utilization profiles are just some examples of users' characteristics that can differentiate a user's class from another (e.g., distinguishing between the COSS[1] MT and regular MTs). The *application* layer describes the component behavior from the logical and functional point of view. The application level of a component consists of a set of functions, each relying on a set of middleware services offered by the architecture. The *architecture* layer is the part of the system capturing the dependability be-

[1] The Controller of Site Safety (COSS) is responsible for the safety of a worksite. In ALARP the MT can be configured to provide functionalities specific of the COSS role.

havior of the main hardware and software components that can affect the application-level behavior of the components. It describes how the functionalities and services of the application level are implemented on those resources.

B. Analysis Techniques and Their Interactions

Within such framework, several techniques have been applied to evaluate the dependability (mainly safety, reliability, and availability) of the ALARP system and its components. The ALARP evaluation framework supports both the analysis of system-level dependability properties, as well as the analysis of specific solutions.

The *analysis of the overall ALARP system* aims at: i) validating system-level dependability requirements, and ii) specifying and analyzing constraints on the dependability and performance of specific functionalities of the system e.g., identifying the range in which the loss probability of risk event messages should fall in order to fulfill system-level safety requirements. Such analysis is performed through a stochastic model of the system that implements the decomposition approach described above. Such model takes into account the interaction between system functionalities and components at a high-level of abstraction. Further details are provided in Section V-C.

Analyses targeting specific ALARP solutions (mechanism, protocols) aim at validating specific requirements and at providing parameter values for the overall system model. Within the project, several analyses targeting specific solutions have been performed (details have been published in separate papers, referenced in the text below):

- *Experimental analysis of wireless communication properties* [2]. Experimental evaluations concerning the TPAD to Worksite communications and the Worksite scenario have been conducted with the following objectives: i) create long-term measurement traces to improve characterization of link properties; ii) enable point-wise parameterizations with respect to the network models within the ALARP evaluation framework.
- *Experimental analysis of the TRC protocol* [4]. An experimental evaluation of the implemented TRC worksite protocol has been conducted to obtain: i) parameter estimation for model parameterization in non-congested and congested scenarios, and ii) protocol performance metrics to validate model results.
- *Experimental analysis of the localization solutions* [3]. Experimental evaluation of GPS devices from various manufactures and price ranges are tested to clarify their accuracy, their position update behavior and systematic errors. These results help to i) select the most suitable devices, ii) clarify the degree of systematic errors expected which can be mitigated, and iii) collect measurement traces for development and evaluation of localization techniques.
- *Simulation analysis of enhanced localization solutions* [5]. The ALARP localization solution is represented by the data fusion mechanism enabling the computation of the position estimates from the localization module. A simulation analysis of the data fusion mechanism is conducted based on experimental traces and simulated fences. This analysis enables to define a preliminary performance analysis of the localization solution as well as identify improvements. Localization and accuracy performance is included in the overall modeling framework to assess its system-wide impact.
- *Analytic analysis of the Timed Reliable Communication protocol* [6]. The goal of this analysis is to assess that the protocol allows satisfying the ALARP dependability requirements, to evaluate the existing tradeoffs, and to help in choosing parameter values for the final implementation. Results of this analysis can also be used to provide parameter values to the overall ALARP system model.
- *Simulation Analysis of the Timed Reliable Communication protocol* [4]. The timed reliable communication protocol in the worksite is implemented in a MATLAB simulation model. The simulation model allows assessing the detailed protocol behavior given a full implementation of the TRC algorithm. Thus, the MATLAB model enables a functional protocol analysis to i) validate the analytic model, ii) cross-validate the implementation in the experimental setup, and iii) serve as a tool in assessing different protocol channel execution settings by applying different theoretical channel models or using experimental measurements directly.

The interactions among those different techniques and with the overall ALARP system model are summarized in Fig. 2.

C. System-level Dependability Analysis

In this Section we describe the actual implementation of our framework to support the system-level dependability analysis of ALARP. The top-down decomposition approach described in Section V-A is applied to identify a set of submodels with precise interfaces that can be composed to model the overall ALARP system. The framework is then implemented using the Stochastic Activity Networks (SAN) formalism [15].

1) Modeled functionalities and metrics of interest

The overall ALARP model takes into account for the main functionalities of the system, their interactions, and the charac-

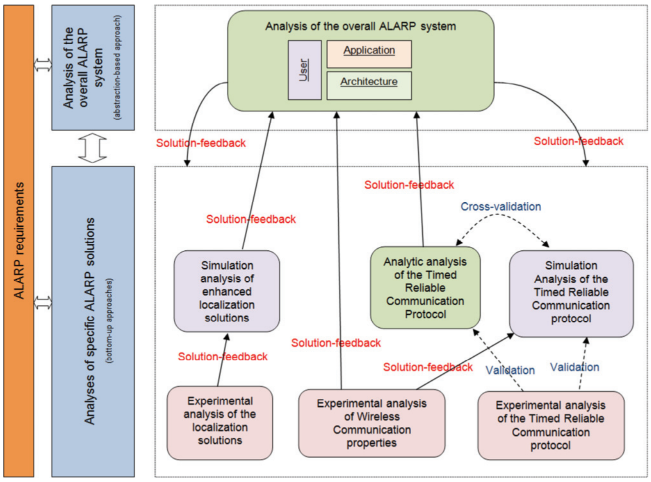
ALARP requirements
Analysis of the overall ALARP system
(abstraction-based approach)
Analyses of specific ALARP solutions
(bottom-up approaches)
Analysis of the overall ALARP system
User
Application
Architecture
Solution-feedback
Solution-feedback
Solution-feedback
Solution-feedback
Cross-validation
Simulation analysis of enhanced localization solutions
Analytic analysis of the Timed Reliable Communication Protocol
Simulation Analysis of the Timed Reliable Communication protocol
Solution-feedback
Solution-feedback
Solution-feedback
Validation
Validation
Experimental analysis of the localization solutions
Experimental analysis of Wireless Communication properties
Experimental analysis of the Timed Reliable Communication protocol

Figure 2. Interactions between the techniques composing the ALARP evaluation framework.

teristics of its environment. The main aspects of the system that are taken into account by the model are messages exchanged between the TPADs and the MTs; MTs functionalities (communication, localization, built-in tests, transition to safe state); TPADs functionalities (train detection, communication). Concerning the environment, the model supports the definition of different track layouts, with the possibility to associate different TPADs to different tracks, in order to accurately represent different worksite conditions.

The model allows the evaluation of safety, reliability and availability metrics. Examples of metrics that can be obtained by the evaluation of the overall system model are:

- $P_{catastrophic}(0,t)$: Probability that a catastrophic failure occurs in the time interval [0,t]. A catastrophic failure occurs when at least one worker is not correctly notified of the approaching train (and its MT is not in a safe state).
- $A_{MT}(0,t)$: Portion of time in which an MT is operational (i.e., not failed and not in safe state) within the interval of time [0,t]. Note that an MT can become not operational also due to failure in network communication.

2) Modeling approach

In performing the system decomposition, particular attention is devoted to the identification of the interfaces between the different submodels. Clearly defining the interfaces between submodels before their implementation improves the reusability, maintainability and modularity of the obtained submodels. Taking this concept to its highest level leads to a modeling paradigm that recalls object-oriented programming: the implementation of each submodel is independent from the other interacting submodels, and it only depends on the defined interfaces.

Submodels obtained in this way are modular i.e., they can be easily replaced or refined as needed, provided that the input and output interfaces remain the same. For example, the TPAD model could be refined in order to accurately model the internal behavior of the train detection mechanisms. This approach also eases the integration with external tools: a given submodel, implementing a specific function, may be replaced with an ad-hoc external tool, either directly or through a "wrapper" model. Successfully performing this integration of course requires that the defined input and output interfaces of the model are compatible with the input expected by the tool, and with the output it provides. An example of such integration is described in [8], where adopting this approach allowed SAN submodel implementing a mobility model to be replaced with the output produced by a vehicular mobility simulator.

Another dual aspect that enhances the modularity of submodels is the identification of their parameters. In complex systems like ALARP, different components may have a similar behavior, only differing by some numerical parameters that are specific of a particular instance of the component, depending on its role in the system, or on the environment in which it is operating. For example, within ALARP, two identical TPAD may experience different train detection probabilities, based on their individual location (e.g., in proximity of a turn, or in presence of obstacles). For this reason, a proper set of parameters should be defined for each submodel, and its implementation should be independent of the actual values of such parameters.

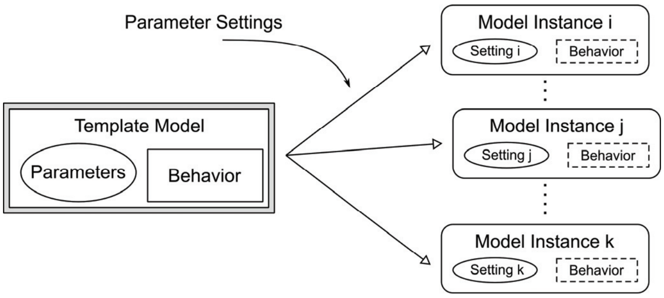

Figure 3. Template models and parameterization.

This process leads to the definition of "template submodels", which are composed of two parts: a part defining its behavior and a part defining its parameters (Fig. 3). In the construction of the overall model these templates are then instantiated multiple times, with different parameters settings. This approach saves the modeler from manually create (and maintain) multiple models for components having a similar behavior, which is a very time-consuming and error-prone task. Also, any change in a template model is automatically propagated to all the instances of that template.

Template model instances are then composed according to precisely defined rules (derived from the component-level and system-level decompositions), in order to obtain the overall model for the desired scenario. The ability to easily create different instances of the same model makes it also easier to evaluate the system under different conditions and different scenarios, which requires only adding or removing model instances or changing their parameters.

3) Implementation using Stochastic Activity Networks

The two-level decomposition approach based on template models is implemented using the Stochastic Activity Networks (SAN) formalism [7]. SAN models can be composed by means of the Rep/Join state-sharing formalism, which allows composing models using the *join* and *replicate* operators [9]. Both operators support multiple levels of composition, thus allowing combining the system-level decomposition with the component-level decomposition, as described in Section V-A.

The support for the definition of model interfaces and parameters is provided by special kinds of places that can be added to SAN models, called "Extended Places". Such places are not limited to hold an integer number of tokens, but instead can hold an instance of a given datatype. Supported datatypes for extended places include most of C++ basic types, arrays, as well as data structures. Types can be nested as well, thus allowing extended places to hold multi-dimensional arrays, arrays of structures, or any other possible combination of data types. Thanks to extended places, interfaces between submodels can be defined in a very detailed way, supporting a very abstract implementation of model behavior. The template submodels are then instantiated multiple times in the overall model, and their interfaces combined using the Rep/Join formalism.

At component level, submodels representing the user, architecture, and application layers are composed, to obtain the

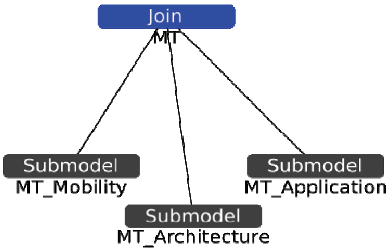

Figure 4. Composed model for the “MT” submodel.

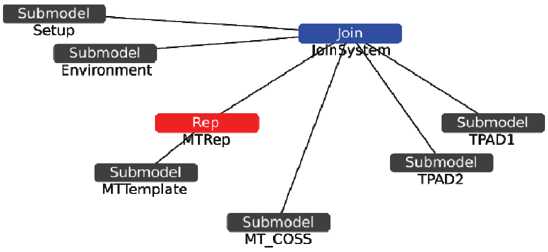

Figure 5. Composed model for a scenario including 2 TPADs, the COSS MT, and an arbitrary number of identical “regular” MTs.

model of the system component. Fig. 4 depicts the composition of an MT model; in this case the user layer is represented by the “MT_Mobility” atomic model. Once models for individual system components have been obtained, they are assembled, based on the system scenario to be modeled. Fig. 5 depicts the composition of an overall ALARP system model for a scenario including 2 TPADs, the COSS MT, and an arbitrary number of identical “regular” MTs. Following this approach, the model can be easily adapted to changes in the system scenario. In principle, this approach could be taken to a further level by adopting a Model-Driven Engineering (MDE) approach [28], in which the dependability model is generated by automated transformation from a (semi-)formal description of the scenario of interest. The clear interfaces of submodels, and the precise composition rules allow the composition of the overall system model to be automatized.

4) Modeling workers mobility

Mobility of workers is a fundamental aspects in the modeling of the overall ALARP system, since it has a direct impact on many aspects of the system, including localization and networking properties, or simply the possibility that a catastrophic failure actually results in the injury of the worker. In this Section we take mobility as an example, to show how the framework supports the refinement of submodels in order to adapt to different level of details in the analysis of the system.

Thanks to the modularity of the framework, the “MT_Mobility” model of Fig. 4 interfaces with the other submodels by means of the “IsInRedZoneOf” interface. Such interface, implemented as an extended place, contains an array of Boolean values. Each of these values is associated with a TPAD, and it is set to *true* if the user is within the red zone distance with respect to the track which is protected by that particular TPAD. Therefore, the only constraint on the internal implementation of the “MT_Mobility” submodel is that it correctly handles the “IsInRedZoneOf” interface.

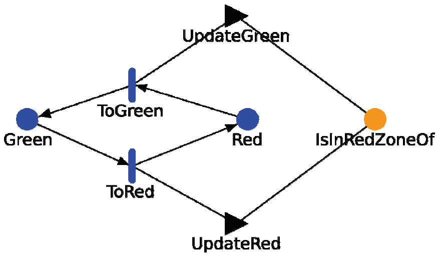

Figure 6. Simple probabilistic mobility model.

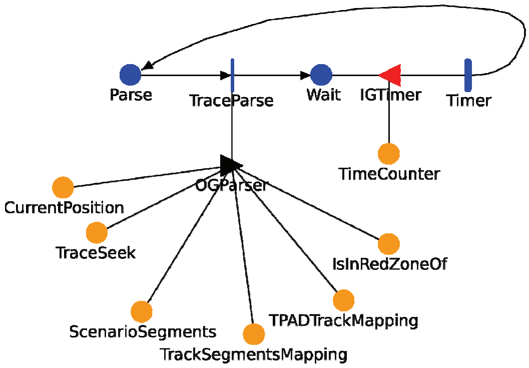

Figure 7. Mobility model based on the import of real GPS traces.

A very simple implementation of such model considers only two locations for the worker: a “generic” green zone, and a “generic” red zone (Fig. 6). The worker moves between the two zones, staying in each of them for a given amount of time that is given by a certain probability distribution. When the user moves from a zone to the other, the interface place “IsInRedZoneOf” gets updated. Obviously, this model only provides very basic information about workers’ positions, e.g., accurately evaluating the impact of different track layouts and worksite shapes is not possible. On the other hand, such model can be used from the early phases of system development and has the advantage of being extremely simple.

While leaving the rest of the system model unchanged, the “MT_Mobility” submodel can be easily refined to directly import real-world GPS traces (e.g., those obtained from the experiments described in Section V-B) into the overall SAN model. Fig. 7 shows a sample SAN implementation of a mobility model periodically reading external GPS traces.

The actual reading of the GPS trace is performed by the code of the “OGParser” output gate, which is executed every time the “TraceParse” activity fires. The gate also updates the interface place “IsInRedZoneOf”, thus providing fresh positions for the workers to the rest of the model. The “Timer” activity introduces a delay between two consecutive readings of the trace. Place “TraceSeek” is used to maintain the last position (in bytes) that has been reached in reading the trace file. This approach accurately represents the mobility of workers within the system model, at the price of much higher resource demands for model evaluation.

Thanks to the flexibility of the framework, in which submodels can be easily refined, both the two different mobility models described above can be used in the evaluation of the system, based on the actual scenario, target measures of interest, and required level of detail.

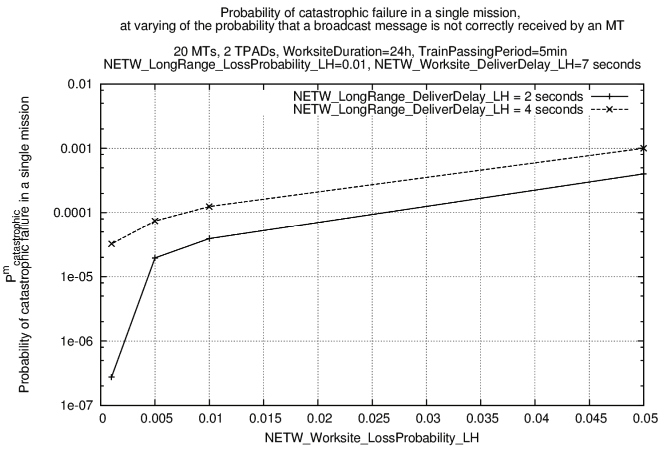

Figure 8. Probability of catastrophic failure in a single mission at varying of the probability that a broadcast message is not correctly received by an MT.

D. Evaluation

While the overall quantitative evaluation of the ALASP system cannot be summarized here, in this Section we provide an overview on how the results provided by the overall system model interact with more specific analyses of ALARP functionalities, and vice-versa.

Fig. 8 depicts the relation existing between the quality of network communication and the probability of catastrophic failure. In particular, it is shown how the probability of catastrophic failure in case of a train approaching the worksite varies at varying of two specific parameters related to network communication: the probability of failure of a message broadcast using the TRC protocol[2], and the average time required for a message transmitted by a TPAD to reach the access point at the worksite[3].

The results show that the probability of catastrophic failure is heavily affected by both these quantities, since each of them characterizes one of the two "hops" that a message has to perform in order to reach the MTs. The probability of catastrophic failure increases almost exponentially as the probability of not receiving the broadcasted message increases (logarithmic scale is used for y-axis values). The time required for the message to reach the worksite also affects the probability of catastrophic failure, and its effect becomes more evident as the probability of not receiving the broadcasted messages decreases. In such cases, in fact, the contribution of losses in the communication between the TPADs and the worksite becomes much more important, and lower transmission delays increase the probability for the message to reach all the workers within the required time bound.

From such results, constraints on network communication can be derived. In the evaluated scenario, for example, a probability of catastrophic failure lower than 10^{-6} can be achieved only if the communication from the TPADs to the worksite requires 2 seconds, and the loss probability of messages at the worksite is 10^{-3} or lower. Such constraints can then be evaluated by means of more specific analysis on the ALARP communication functionality e.g., the experimental analysis of wireless communication properties (see Section V-B) can be applied in order to verify that the constraint is met in the considered environment.

The evaluation framework presented in this paper contributes to a more general assessment process of the ALARP system, aiming to verify its dependability and real-time requirements, and provide guidelines to support its certification according to relevant standards. The final assessment of ALARP, as well as guidelines to support its certification, are provided in deliverable D6.6 of the project [29], which is going to be released as a public document.

VI. CONCLUDING REMARKS

This paper described our experience in architecting and evaluating the ALARP system, a distributed, real-time, safety-critical system that notifies trackside workers of approaching trains. The combination of the harsh environment in which ALARP operates, its strict requirements, and its dynamicity and mobility, raised several challenges, both in the definition of the system architecture and in the evaluation of its dependability properties. We identified and discussed such challenges, and described how they have been addressed within the project for the design and evaluation of the ALARP system.

Table I and Table II summarize the challenges identified in ALARP, and the strategies that have been adopted to overcome them. The authors of this work are well-aware that different approaches may be successfully applied when designing and evaluating critical systems, and peculiarities which vary from one case to the other are often at the basis of specific decisions. Despite this, given that a wide class of dynamic, distributed, real-time, safety-critical systems are characterized by similar challenges, we believe that system architects and Verification and Validation experts can benefit from the analyses and experiences described in this paper.

TABLE I. ARCHITECTURAL CHALLENGES AND HOW THEY HAVE BEEN ADDRESSED IN ALARP

Challenges	Main approaches applied
Safety critical system	TPAD sensors have very-low missed detection rate. Real-time and reliable communication network is available. MT payload executes fault-tolerant mechanisms.
Real-time mobile system in outdoor, harsh environment	Starting from [20], the real-time communication protocol has been tailored for ALARP. Time requirements are monitored in the MT wormhole.
Safe and trusted HMI	Very perceivable audio and visual signals are applied, based on eyewears and ear bone conductors.
Scarcity of power (batteries)	TPAD mostly uses low power consumption sensors. The MT executes modalities for energy-saving when power is low, still not affecting safety.
Reliable network	Real-time and reliable communication protocol designed starting from [20].
Security	Anti-tampering TPAD. Authentication of all messages using private-public keys.
Privacy	Protection of private information in the wormhole. Low-term memorization of sensible data as position and health data.
Precise localization	GPS-augmentation approach that merges inputs from GPS data and electronic fences.

[2] Parameter "NETW_Worksite_LossProbability_LH"

[3] Parameter "NETW_LongRange_DeliverDelay_LH"

TABLE II. EVALUATION CHALLENGES AND HOW THEY HAVE BEEN ADDRESSED IN ALARP

Challenges	Main approaches applied
Model largeness	Two-levels (system- and component-) system decomposition. Support for different abstraction levels via submodels refinement/simplification.
Capturing details of specific functionalities	Interaction between multiple techniques (e.g., experimentation) to validate assumptions and provide parameter values. Support for different abstraction levels via submodels refinement/simplification.
Variability of scenarios and environment	Template parametric models and composition rules.

ACKNOWLEDGMENTS

This work has been partially supported by the European Project FP7-234088 ALARP (A railway automatic track warning system based on distributed personal mobile terminals). The authors would like to thank the many partners and the collaborators involved in the project for their precious contributions, and in particular Francesco Brancati, Carlo Dambra, Fabio Duchi, Jesper Grønbæk, Lorenzo Falai, João Figueiras, Mick Pappas, Boris Malinowsky, Michael Schultheis, Hans-Peter Schwefel, Andrea Seminatore, Jurij Wakula and Avraham Zruya.

REFERENCES

[1] A. Bondavalli, O. Hamouda, M. Kaâniche, P. Lollini, I. Majzik, and H.-P. Schwefel, "The HIDENETS Holistic Approach for the Analysis of Large Critical Mobile Systems," IEEE Transactions on Mobile Computing, vol. 10(6), pp. 783-796, 2011.

[2] B. Malinowsky, J. Grønbæk, and H.-P. Schwefel, "Realization of Timed Reliable Communication over Off-The-Shelf Wireless Technologies", in press (to appear at IEEE WNCN), 2013.

[3] A. Bondavalli, A. Ceccarelli, F. Gogaj, A. Seminatore, and M. Vadursi, "Experimental assessment of low-cost GPS-based localization in railway worksite-like scenarios," Measurement, vol. 46(1), Elsevier, pp. 456-466, 2013.

[4] B. Malinowsky, J. Gronbaek, H.P. Schwefel, A. Ceccarelli, A. Bondavalli, and E. Nett, "Timed Broadcast via Off-the-Shelf WLAN Distributed Coordination Function for Safety-Critical Systems," In Proc. of the 9th European Dependable Computing Conference (EDCC), pp.144-155, 2012.

[5] J. Figueiras, J. Gronbaek, A. Ceccarelli, H.P. Schwefel, "GPS and Electronic Fence Data Fusion for Positioning within Railway Worksite Scenarios", In Proc. of the IEEE 14th International Symposium on High-Assurance Systems Engineering (HASE), pp.17-23, 2012.

[6] L. Montecchi, P. Lollini, B. Malinowsky, J. Grønbæk, and A. Bondavalli, "Model-based analysis of a protocol for reliable communication in railway worksites," In the 15th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM) Paphos, Cyprus Island, pp. 23-32, 2012.

[7] W.H. Sanders, and J.F. Meyer, "Stochastic activity networks: formal definitions and concepts," Lectures on formal methods and performance analysis, Springer-Verlag New York, Inc., pp. 315-343, 2002.

[8] A. Bondavalli, P. Lollini, and L. Montecchi, "QoS Perceived by Users of Ubiquitous UMTS: Compositional Models and Thorough Analysis," Journal of Software, 4, pp. 675-685, 2009.

[9] W. Sanders, and J. Meyer, "Reduced base model construction methods for stochastic activity networks," IEEE Journal on Selected Areas in Communications, vol. 9, pp. 25-36, 1991.

[10] T. Israr, M. Woodside, and G. Franks, "Interaction Tree Algorithms to Extract Effective Architecture and Layered Performance Models from Traces," Journal of Systems and Software, vol. 80(4), pp. 474-492, 2007.

[11] K. Nagaraja, G. Gama, R. Bianchini, R.P. Martin, W. Meira, and T.D. Nguyen, "Quantifying the performability of cluster-based services," IEEE Transactions on Parallel and Distributed Systems, vol.16(5), pp. 456-467, 2005.

[12] J. Arlat, M. Aguera, L. Amat, Y. Crouzet, J.-C. Fabre, J.-C. Laprie, E. Martins, and D. Powell, "Fault Injection for Dependability Validation - A Methodology and Some Applications," IEEE Transactions on Software Engineering, vol. 16 (2), pp.166-182, February 1990.

[13] DBench – Dependability Benchmarking (Project IST-2000-25425). http://www.laas.fr/DBench/ [last accessed 29 January 2013], 2001.

[14] HIDENETS - HIghly DEpendable ip-based NETworks and Services (Project IST-FP6-STREP-26979). http://www.hidenets.aau.dk/ [last accessed 29 January 2013], 2006.

[15] A. Bondavalli, A. Ceccarelli, and P. Lollini, "Architecting and validating dependable systems: experiences and visions," In *Architecting dependable systems VII*, A. Casimiro, R. de Lemos, and C. Gacek (Eds.). Springer-Verlag, Berlin, Heidelberg, pp. 297-321, 2010.

[16] ALARP - A railway automatic track warning system based on distributed personal mobile terminals (Project FP7-IST-2010-234088). http://www.alarp.eu [last accessed 29 January 2013], 2010.

[17] A. Seminatore, L. Ghelardoni, A. Ceccarelli, L. Falai, M. Schultheis, and B. Malinowsky, "ALARP (A Railway Automatic Track Warning System Based on Distributed Personal Mobile Terminals)," Procedia - Social and Behavioral Sciences, vol. 48, pp. 2081-2090, 2012.

[18] A. Ceccarelli, A. Bondavalli. J. Figueiras, B. Malinowsky, J. Wakula, F. Brancati, C. Dambra, and A. Seminatore, "Design and Implementation of Real-Time Wearable Devices for a Safety-Critical Track Warning System," In Proc. of High-Assurance Systems Engineering (HASE), pp.147-154, 2012.

[19] P. Verissimo, "Travelling through wormholes: a new look at distributed systems models," SIGACT News 37, 1 (March 2006), pp. 66-81, 2006.

[20] M. Mock, E. Nett, and S. Schemmer, "Efficient reliable real-time group communication for wireless local area networks," Proc. European Dependable Computing Conference (EDCC), J. Hlavicka, E. Maehle, and A. Pataricza (Eds.), Springer-Verlag, pp. 380-400, 1999.

[21] EN 50129, "Communication, signalling and processing systems – Safety-related electronic systems for signalling," 2003.

[22] R.M. Stanley, and B.N. Walker, "Intelligibility of bone-conducted speech at different locations compared to air-conducted speech," Proc. HFES 2009, San Antonio, TX, 2009.

[23] N.D. Weston, and V. Schwieger, "Cost effective GNSS positioning techniques," Technical Report 49, FIG, Copenhagen, Denmark, 46 pp., 2010.

[24] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," Commun. ACM, vol. 47(6), pp. 53-57, 2004.

[25] M. Malhotra, and K.S. Trivedi, "Dependability modeling using Petri-nets," IEEE Transactions on Reliability, 44, pp. 428-440, 1995.

[26] D.M. Nicol, W.H. Sanders, and K.S. Trivedi, "Model-based evaluation: from dependability to security," IEEE Transactions on Dependable and Secure Computing, 1, pp. 48-65, 2004.

[27] Y.-H. Dai, Y. Pan, and X.A. Zou, "Hierarchical Modeling and Analysis for Grid Service Reliability," IEEE Transactions on Computers, 56, pp. 681-691, 2007.

[28] D.C. Schmidt, "Guest Editor's Introduction: Model-Driven Engineering" IEEE Computer, 39, pp. 25-31, 2006.

[29] ALARP Deliverable D6.6 "Final assessment and guidelines", To be released.