On trustworthy measurements when testing dependable systems: a discussion and experiences

Abstract
The scientific literature and the industrial practice agree since many years on the fundamental role of experimental evaluation (testing) of critical systems for the assessment of the dependability attributes, and consequently on the relevance of achieving trustworthy measurements. This paper discusses and motivates with the support of three case studies the possible role of the body of knowledge offered by measurement theory (metrology) to quantitatively assess the quality of measuring instruments, i.e., the tools, and the results collected. The paper first introduces notions of metrology and contextualises them for the dependability evaluation of systems. Successively it presents three case studies developed by the authors where attention to principles from measurement theory and the metrological assessment of tools and results are carried out. The paper ultimately reviews the main guidelines identified discussing their application in the case studies.

Keywords
Experimental evaluation, metrology, measurement theory, dependability

1 Introduction
Experimental evaluation (testing [1]) for the quantitative analysis of dependability [1] attributes is an attractive option to evaluate an existing critical system or prototype, because it allows to observe the real execution of such system to obtain (hopefully, highly accurate) information of its behaviour in a usage environment [7], [39].

A fundamental requirement at the basis of experimental evaluation is that the collected results can be claimed trustworthy [26], [28], [27]. A relevant support to achieve such goal can be offered by i) an accurate design of the measuring system, i.e., the instruments and features used to perform the measurements, and ii) an appropriate investigation of the quality of measurement results [2], [3]. This implies that the measuring system and all factors that may influence the results of the experiments, for example
the environment, need to be investigated and that possible sources of uncertainty or bias in the results need to be addressed [2], [3].

While there is generally a widespread consciousness of the relevance of such argumentations, few solutions or approaches exist to address it systematically in the field of dependable systems. This paper explores the topic of the experimental evaluation of dependable systems investigating a general approach composed of concepts and guidelines from measurement theory to offer a metrological assessment of tools and results. Three case studies developed by the authors are shown, that present different approaches to structure experimental campaign at the light of the concepts and guidelines previously introduced, and to perform a metrological assessment.

Ultimately, the paper concludes relating the case studies to the guidelines presented to highlight their application and give evidence on the insight they allowed to achieve regarding measurement instrument and results.

The rest of the paper is organized as follows. Section 2 introduces basic notions on measurement theory. Section 3 describes concepts and fundamental guidelines in the experimental evaluation of dependable systems at the light of principles of measurement theory. Section 4 to Section 6 present the three case studies, and Section 7 presents concluding remarks.

2 Measurement theory (metrology)

A few fundamental concepts related to characterize measurement systems and methods according to metrological criteria are introduced. A complete digest of metrological terms and concepts can be found in the standards GUM [2] and VIM [3], which includes relevant terms, fundamental measurement properties, and guidelines.

Measuring a quantity (namely the measurand) consists in quantitatively characterizing it. The procedure adopted to associate quantitative information to the measurand is called measurement. The measurement result is expressed in terms of a measured quantity value and a related (measurement) uncertainty.

Uncertainty provides quantitative information on the dispersion of the quantity values that could be reasonably attributed to the measurand. Uncertainty has to be included as part of the measurement result and represents an estimate of the degree of knowledge of the measurand. It is usually expressed in terms of a confidence interval, that is a range of values where the measurand value is likely to fall. The probability that the measurand value falls inside the confidence interval is named confidence
On repeatable measurements when testing dependable systems

level. Two different ways to compute the standard uncertainty (uncertainty expressed as a standard deviation) are described in [2] and are respectively called Type A and Type B uncertainty. Type A uncertainty is computed statistically, as an estimate of the standard deviation of the mean of a set of independent observations. Type B uncertainty is computed on the basis of a scientific judgement using all the relevant information available, as previous measurement data and knowledge of the behavior of relevant materials and instruments. Type B uncertainty does not depend on the amount of observations collected and consequently is especially relevant when the number of independent observations is low.

Resolution is the ability of a measuring system to resolve among different states of a measurand. It is the smallest variation of the measurand that can be appreciated, i.e., that determines a perceptible variation of the instrument’s output.

It is well known that any measurement system perturbs the measurand, determining a modification of its value. Minimizing such perturbation, that is minimizing the system’s intrusiveness, is therefore desirable when designing a measurement system.

Repeatability is the property of a measuring system to provide closely similar indications in the short period, for replicated measurements performed i) independently on the same measurand through the same measurement procedure, ii) by the same operator, and iii) in the same place and environmental conditions. In practice being able to achieve repeatability (and having trusted results which e.g., are not characterized by the same bias) requires to investigate on the intrusiveness and resolution of the measuring system, and on measurement uncertainty.

3 Metrological assessment of dependable systems

Issues with the way measurement is applied in assessing computer dependability, and the need for giving practice a better theoretical basis, were raised since many years, for example with respect to software reliability assessment [30], software metrics [31], fault-tolerant computing systems assessment [39] or measurement automation software [34].

In recent times special attention has been paid to the evaluation of dependability and QoS (Quality of Service, [40]) properties, with most of the attention devoted to the values output rather than the quantitative evaluation of the quality of measurement [33]. In fact, the investigations carried out in [5], [6], [29] showed that the approaches are sometimes intuitive and in general non-univocal. Thus, the body of knowledge of measurement theory, with the terms, fundamental properties and
Authors

guidelines reported in [2], [3], can act a relevant role to improve the confidence in both tools and results when performing dependability-related experiments.

However, it is to be clearly remarked that this does not mean tools, or experiments, are badly designed, nor that results are not correct; but the framework offered by metrology could improve the evidence on the quality of the results achieved.

Considering the measurement properties presented in Section 2, when experimenting dependable systems it is rarely identified a real effort to estimate uncertainty, and to determine solid bounds on the reliability and trustworthiness of the measures collected [29]. Although there are counter-examples (e.g., analysis of uncertainty and its sources is explicitly performed in [4] to improve the confidence in results of fault injection tests), an accurate characterization of measurement results explicitly addressing uncertainty is often missing [29].

Resolution is usually easy to compute and it is rarely cause of trouble in the experimental process; however it is worth mentioning that in some works, as shown in [5], [29], it is not discussed explicitly.

A deep awareness is in general present with regard to intrusiveness, which is not to be unexpected in dependability tools [38]. Intrusiveness can be investigated with different levels of emphasis depending on the characteristic of the measuring instruments. For example, [36] proposes a non-intrusive technique for software-implemented fault injection (SWIFI) to reduce the temporal and spatial overhead induced on the target system. Instead in [37] temporal intrusiveness is measured and analyzed for a SWIFI tool devised to assess the error detection mechanisms of the Time-Triggered Architecture (TTA) bus structure.

Repeatability is widely acknowledged as the most difficult property to achieve, especially when performing time measurements in distributed systems, due to limits on collecting accurate time values [41]: executions of the same run will probably not bring the same exact results [3]. Determinism of the target system is needed to ensure repeatability, including the starting state of the system: for example, in order to completely ensure repeatability of every experiment, a fault injection tool would have to copy the entire state of memory at start-up and restore it in each experiment [4]. Thus it is a matter of fact that repeatability is often not achievable when measurements are carried out on computer systems, and especially on highly distributed ones: the same environmental conditions can, in fact, hardly be guaranteed.
On repeatable measurements when testing dependable systems

It has to be noted that repeatability is a fundamental property in testing dependable systems. In fact, in dependable systems, faulty behaviour may often be not acceptable, and testing is a mean to check the proper behaviour of the system. Thus, repeatability of the experiments - having the same results for different repetition of an experiment - is easily acknowledgeable as fundamental. Approaches to address repeatability exist, for examples [42] proposes a methodology based on multi-criteria decision making to improve the repeatability and reproducibility of dependability benchmarks. However, typically each experimental campaign requires an ad-hoc discussion on experiments repeatability.

Concluding, common guidelines can be extracted from the measurement standards [2] and [3], to improve confidence in measuring dependable systems:

1. the measurand should be clearly and univocally defined;
2. all sources of uncertainty should be singled out and evaluated;
3. some attributes of major concern for dependability measurements, such as intrusiveness, resolution and repeatability should be evaluated;
4. measurement uncertainty should be evaluated.

In the rest of the paper we present three case studies in the field of dependability assessment, where we highlight the investigation of the measurement properties here discussed.

4 Case study 1: software clock

We describe the experimental evaluation of the Reliable and Self-Aware Clock (R&SAClock, [8], [24]), a software clock which is designed to be self-aware of its synchronization uncertainty [8], [24]: we define synchronization uncertainty as an adaptive and conservative evaluation of the distance of the local clock from the reference time (the offset). The offset is usually computed by the synchronization mechanism in use without offering guarantees on the accuracy of such computation. The synchronization uncertainty computed by the R&SAClock provides such guarantees: when asked to provide the time at any time instant \( t \), R&SAClock replies with an enriched time value \([\text{likelyTime}, \text{minTime}, \text{maxTime}]\), where \( \text{likelyTime} \) is the time value computed reading the local clock \( c \), and the interval \([\text{minTime}; \text{maxTime}]\) offers information on the confidence that can be associated to the time value \( \text{likelyTime} \). The computation of \( \text{minTime} \) and \( \text{maxTime} \) are based on the synchronization uncertainty computed by the internal mechanisms of R&SAClock [8].
It is required that the enriched time value includes the *true time* (i.e., the time provided by the *reference clock*): this means that given the enriched time value computed at any time instant $t$, such $t$ must fall within the interval $[\text{minTime}; \text{maxTime}]$. Our experimental evaluation aims to measure the coverage of such required condition for different configuration of R&SAClock.

The experimental plan covers a relevant set of cases, including i) different values of the software clock parameters, ii) different types of workload, and iii) the possible occurrence of faults in the system under test and/or in the underlying synchronization mechanism (we injected faults whose effects are failures of the NTP synchronization mechanism in disciplining the software clock).

### 4.1 The target system and the experiments

The target system consists of an R&SAClock prototype, which is installed as a software component on a computer, that we call PC_R&SAC (Figure 1). The local software clock is synchronized through the Network Time Protocol (NTP). An NTP client (a process daemon) running on PC_R&SAC synchronizes the local clock using information from the NTP server(s). PC_R&SAC is connected to one or more NTP servers through the Internet. In the implementation considered here, the R&SAClock is a C++ middleware service which interacts with the software clock and NTP.

![Figure 1. The measuring system developed and the target system [11].](image)

### 4.2 The measuring instrument

The design and implementation of the validation testbed follow two basic rules: i) grant a time resolution sufficiently lower than that of the system under test, ii) keep the software probes as simple as possible in order to reduce the intrusiveness on the system under test, and consequently reduce uncertainty of the measurement results and improve repeatability.
The measuring system is shown in Figure 1. PC_GPS represents the monitoring system, where a Controller triggers the Client to request the enriched time value to R&SAClock, and collects the returned values. The Controller can also access the reference clock (it is a clock synchronized using GPS).

The PC_R&SAC includes the target system, already described, and a software Client which can ask the enriched timestamp to the R&SAClock.

Both PC_R&SAC and PC_GPS log data relying on NetLogger [13], a tool for data logging in distributed systems that introduces minimal intrusiveness thanks to its binary format for data logging [13], [35].

The choice of keeping the monitoring system and R&SAClock on different nodes is justified by the need of minimizing the intrusiveness of the monitoring system on the operative system of the node the R&SAClock is installed on. For the same reason, the option of having the reference clock as a second clock on the same node of R&SAClock is not considered. Our choice is to have a node (PC_GPS) including the reference clock and the monitoring system.

Controller and Client interact periodically to collect the enriched timestamps as follows. The Controller sends a message containing a \texttt{getTime} request and an identifier ID to the Client at time instant $T_1$ ($T_1$ is collected reading the PC_GPS clock) to ask the enriched time value. When the Client receives the message (at time instant $T_2$, collected reading the PC_R&SAC clock), it forwards such request to R&SAClock. When the Client receives the enriched time value from R&SAClock at time instant $T_3$ (again $T_3$ is collected reading the PC_R&SAC clock), it notifies the Controller. The Controller finally receives such notification at time instant $T_4$ ($T_4$ is collected reading the PC_GPS clock).

As said above, the objective of the validation is to check that given the enriched time value computed at any time instant $t$, such $t$ must fall within the interval $[\text{minTime}; \text{maxTime}]$. As the R&SAClock provides a synchronization uncertainty sometimes lower than 1 millisecond (ms), an accurate methodology is needed. Consequently, the validation test bed is based on a detailed analysis of which is the most suitable reference time instant to compare with R&SAClock output. The correct way to proceed is not to think at R&SAClock as a software device designed to answer the question “what time is it?”, regardless of the practical use of the output it gives. Performing a meaningful validation of R&SAClock means verifying if R&SAClock works properly and to what extent it is useful. This means taking the time from the reference clock when R&SAClock
provides its answer, rather than when the question is made. From the description above, this time instant corresponds to the reference time at which T3 is collected i.e., t(T3). Obviously T3 can be measured only on PC_R&SAC, thus on the node without the reference clock: consequently t(T3) can not be measured directly. The following solution has consequently being identified to assess that t(T3) is within \([\text{minTime}; \text{maxTime}]\).

The enriched time value should be compared to the reference time t(T3). For t(T3), the following relation holds (see also Figure 2):

\[t(T3) \in (t(T1) + \delta_1 + \Delta_{23}; t(T1) + \Delta_{14} - \delta_2 - \mu)\]

where t(Ti) is true time when Ti happens; \(\delta_1, \delta_2\) are the minimum transmission time (respectively, from PC_GPS to PC_R&SAC and vice-versa); \(\mu\) is the time elapsed between t(T3) and the beginning of the transmission plus the time elapsed between the reception of the ack at PC_GPS and the actual timestamping instant; \(\Delta_{xy}\) is the time interval \(|t(Ty)-t(Tx)|\).

Given the reasonable hypothesis that \(\delta_1 + \delta_2\) is much smaller than \(\Delta_{23}\), it is possible to reduce the uncertainty on t(T3) to a small interval. Thus, by comparing the \([\text{minTime}, \text{maxTime}]\) with the interval \((T1 + \delta_1 + \Delta_{23}; T4 - \delta_2 - \mu)\) shown in Figure 2, assuming that T3 is the median of this interval, we can verify if the interval \([\text{minTime}, \text{maxTime}]\) contains the true time.

The main contribution to the uncertainty on T3 is given by the resolution, that is the amplitude of the interval where T3 falls. In the experiments such interval has come out to be of the order of 100 microseconds (us).

A reasonable hypothesis underlying the above equation is true is that the delay between any Ti and the time its corresponding timestamp is taken, is the same for any i. Moreover, it should be noted that \(\Delta_{23}\) and \(\Delta_{14}\) are measured on different machines and, therefore, the interval in equation above could come out to be empty (due to severely different drifts). In such a case, the monitoring system can estimate t(T3) by subtracting \(\delta_2\) and
μ from the time provided by its clock (i.e., the reference clock) at T₄, when it receives the ack; i.e., t(T₃) = t(T₄) - δ₂ - μ.

4.3 Metrology assessment

As discussed above, a method to evaluate resolution has been derived, setting its value to 100 us. The main contribution to the uncertainty on T₃ is given by the resolution: according to [2], in such situations the true value is expected equally distributed in an interval given by the measured value and the measured value plus the resolution. For example, if 10 ms is the measured value and the resolution is 100 us, the expected true value is the midpoint of the interval [10; 10,1] ms, i.e., it is 10,5±0,5 ms, with confidence 1 and Type B uncertainty ±50 us. This is in fact the uncertainty we consider for the enriched time values collected in this case study.

Thanks to an accurate design, intrusiveness was deemed negligible as previously discussed.

When repeating each experiment, closely similar indications on the coverage were provided in the different runs. Thus, from the definition in Section 2, experiments can be declared repeatable.

4.4 A glimpse on results

From the results shown in [11] and [8], the R&SAClock is able to compute an interval [minTime; maxTime] which includes the true time most of the time, with a coverage which depends on its configuration parameters.

5 Case Study 2: OTS GPS devices

The primary goal of the ALARP ATWS (Automatic Track-Warning System) [16] is to recall the attention of a working group operating on a railway worksite about the presence of an approaching train. ALARP keeps track of the status and position of the workers relying on low-cost GPS-based wearable devices, to identify the workers at risk (i.e., close to the track while a train is approaching) or to suggest escape routes. The localization requirements of ALARP demand that the railway worker is accurately localized for safety reasons.

The objective of the experimental evaluation here described is to assess if and to what extent cheap OTS (off-the-shelf) GPS devices can be successfully applied in the ALARP scenario [12], [25]. In particular, our work aimed to investigate the quality of low cost GPS receivers to provide feedbacks to the designer of the localization solution in ALARP.
5.1 Localization in ALARP
A typical railway worksite in which the workers need to be localized is an operation area of maximum 700 m length, in which workers typically move on foot (slow movement speed). The worksite can be located in place possibly surrounded by foliage, in canyons, or near buildings (i.e., there is limited visibility of GPS satellites). It is thus reasonable [17] to expect localization errors due to satellite clocks (errors in the synchronization of the different satellite clocks, typically in the order of 0.8 m to 4 m) and ephemeral satellite orbits (errors in precisely establishing the spacecraft location, typically on the order of 0.8 m) when receivers use the same satellite set. Also errors are expected due to ionospheric and tropospheric signal perturbation and delays (given by the transition of the signal through the troposphere and ionosphere), and due to the receiver’s design (errors due to the specific design of the receiver). All these enlisted errors can be considered as systematic. At the same time, we can expect that errors due the receiver’s thermal noise and external interferences exhibit negligible variations from a receiver to the others [17].

Conversely, we expect that multipath (reflection errors, one of the most significant and variable errors incurred in the receiver measurement process) affects randomly the localization error; this is expected as the major issue in localization measurements [17].

5.2 Target systems and measuring system
We selected for our tests two target systems: the ND-100S produced by Globalsat [18] and the Garmin 18 LVC [19]. The Garmin 18 LVC is a GPS device of a higher category of price and performance than the Globalsat (it costs around three times the Globalsat ND-100S). Using two receivers of different quality allows collecting information on how the localization error varies depending on the device used and on the tradeoff in performance and costs.

As measuring system, a reference system allows to compare the data collected using the GPS devices previously mentioned. The reference system is the Trimble system [20], [21] composed of a stationary reference station Trimble R7 [20] and a roving device Trimble R8 [21]. This Trimble system is able to calculate the position within few centimeters (it is by far more accurate than the other two GPS devices).

During the experiments, common laptops with OS Windows 7 are used to log the NMEA 0183 sentences (the protocol National Marine Electronics Association 0183 defines the information that GPS devices communicate) provided every second on USB ports by all devices involved (Globalsat...
ND-100S, Garmin 18 LVC, Trimble R7 and R8). Note that the NMEA sentences contain the time instant at which the sentence is generated, so they can be logged directly without the need to investigate possible delays or inaccuracies of the laptops in timestamping.

5.3 Experiments description
We previously discussed the localization error sources for a railway worksite; amongst those, some error sources are bounded to the characteristics of the environment in which the worksite is placed.

Here, we focus on the case when devices are close to the side of a high building, and are consequently subject to multipath and limited satellites visibility. Devices may have partial or no satellites visibility for a short period of time. Each time a receiver loses and re-acquires satellites visibility, it may need to execute a transient phase in which the computation of their position is particularly unreliable. Additionally, the characteristic of the environment may increase multipath and interference errors. We note that the systematic errors due to satellite visibility can potentially be observed by measuring the distance between the reference system and the two target systems, while random errors due to multipath are sources of uncertainty that cannot be mitigated.

The experimental campaign consists of experiments involving stationary and rovering measurement, at different distances from the side of the building, and in hybrid scenarios where the planned paths alternate proximity of building and empty areas.

5.4 Metrology assessment
Resolution depends on the GPS devices, and it can be extracted from their data sheets. For the three axis latitude, longitude and altitude, it resulted in the order of 1 meter for the Garmin and the Globalsat, and millimeter-level for the R7-R8.

Type A uncertainty can be measured using data from repeated runs. Multipath was the main source of uncertainty, which was in the order of meters for the two target systems.

Due to multipath, we measured different indications on the behaviour of the devices in replicated runs, thus it negatively affected repeatability.

Considering that we log the NMEA sentences and elaborate data offline, we claim that our measuring system was not intrusive.

5.5 A glimpse on results
Relying on the results collected, it was identified that GPS alone is not sufficient for the purpose of ALARP. Thus GPS augmentation approaches
that combine the outputs of GPS with the outputs of electronic fences placed in proximity of the tracks [22] were deemed necessary and developed in the context of the ALARP project.

6 Case Study 3: safety-critical embedded railway system
We briefly present the experimental evaluation via fault injection of a safe train-borne Driver Machine Interface (SAFEDMI, [14]).

6.1 A brief description of SAFEDMI
The SAFEDMI project [14] aimed to design and validate a safety-critical Driver Machine Interface (SAFEDMI), with no hardware redundancy and using as much as possible hardware and software OTS.

In railway train-borne equipment, the SAFEDMI acts like an OTS safety-critical bridge between the operator (the train driver) and the EVC (European Vital Computer: it supervises the train movement). SAFEDMI communicates with the EVC as a slave; it acquires and manipulates driver’s commands (using a keyboard) from the EVC and it transforms EVC commands in graphical and audible information (using an LCD screen and audio devices). SAFEDMI target the requirements of Safety Integrity Level 2 (SIL 2 - railway standards [9] 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 $10^{-7} \leq \text{THR} \leq 10^{-6}$).

We consider in this work exclusively the Start-up and Normal operational modes of the SAFEDMI, which we have tested through software fault injection. In Start-up mode the initialization procedures and the thorough testing of all devices are performed. In Normal mode the SAFEDMI produces graphical and audio information to support train driving, as well as it acquires and processes driver’s commands; periodic testing activities are performed and diagnostic functionalities are available. A safe mode is entered when a malfunctioning is detected.

6.2 The measuring system
The measuring system built for performing software fault injection and its interactions with the target system (the SAFEDMI system) are shown in Figure 3 and explained in what follows.

We subdivide the measuring system in two functional blocks (the grey blocks of Figure 3). The first block, which is composed of the software components library, injector and workload generator, deals with the injections and the workload execution: its function is to execute the experiments. The functions of the second block are monitoring, data
collection and analysis: this block monitors the SAFEDMI, collects results and analyses them.

The workload generator is the tool EVC Packet Generator, a simulator of the EVC that runs on a PC connected to the SAFEDMI; the library and injection tool are instead both located on the SAFEDMI. The library is the mean to inject the available faultload: it enlists the available faults as well as the methods to inject them in the SAFEDMI. The identified faults are implemented adding extra code in genuine SAFEDMI functions or developing additional functions that are not part of the SAFEDMI genuine software.

The number of instructions needed to inject a fault is always small, and these instructions are fast to execute: the perturbation they introduce on system scheduling and the impact on the overall computational load can be considered negligible. The injection tool allows performing the runtime software injections in the SAFEDMI. It is a cyclic, light and low-priority thread active on the SAFEDMI. This thread executes cyclically once every 1000 ms with a deadline of 2000 ms. The injection tool reads from a configuration file the instructions about the experiment to execute (e.g., the fault to inject and the time instant at which it should be injected), and uses the library to select the faults. The injection tool can inject a single fault or a sequence of faults at specific time intervals one from the others.

Regarding the components of the second functional block, the data collector (or logger) is a diagnostic tool (called D360) located on the PC connected to the SAFEDMI. It receives, logs and organizes information

![Diagram](image-url)
received from the monitor, which executes on the SAFEDMI to timestamp events and to communicate events and related timestamps to the data collector. The data collector and the monitor communicate using a dedicated serial channel, different from the serial channel for the communication between the EVC Packet Generator and the SAFEDMI.

The monitor is an extension of the SAFEDMI log manager thread, that is a SAFEDMI genuine thread used for diagnostic activities (so we do not introduce a new thread in the system). The log manager thread is the thread with the lowest priority in the SAFEDMI, and it has no deadlines: it executes only when other threads are not running. As a consequence, to provide precise timestamping of events it is necessary to collect each time instant (by invoking the SAFEDMI system call `getTime`) as an atomic action with the raising of the event.

6.3 Metrology assessment

The resolution of the measuring system is investigated only for time-related quantities. System resolution for time instants is 2 ms; it is the resolution of the SAFEDMI timer used as the base for the activities of the scheduler and of all threads.

Three components of the measuring system may be intrusive and perturb the SAFEDMI: the library, the injection tool and the monitor. To investigate intrusiveness we need to analyze perturbations in time and memory. Memory perturbation is negligible, since the executable files, the dedicated variables and the dedicated memory areas of library, injection tool and monitor are very small compared to the SAFEDMI memory.

Time perturbation needs a deeper investigation. The injections are performed through few, quick instructions that are executed at worst in few microseconds. The injection tool and the monitor are low priority threads that execute mainly when other threads are not running, to be as low intrusive as possible. The monitor sends data to the data collector using a completely dedicated communication channel: thus this communication does not alter the communication between the SAFEDMI and the EVC Packet Generator.

To further analyze intrusiveness, a schedulability analysis of SAFEDMI threads has been performed using the SchedAnalyzer tool (it provides a pessimistic estimation of the CPU computational load of the overall set of threads on the CPU): it resulted that the set of threads is schedulable (threads deadlines are guaranteed to be met, and there is enough CPU free time to guarantee that the injection tool and monitor threads will execute
On repeatable measurements when testing dependable systems

without influencing other threads execution). Thus, considering that resolution is 2 ms, we can state that intrusiveness is negligible.

Repeatability instead can be guaranteed only in specific cases, for the following limitations. First, there are no guarantees that the injections are performed exactly at due time instants because of the low priority of the injector thread. The EVC Packet Generator severely affects repeatability: despite it is supposed to generate always the same workload with the same exact timing, such exact timing is not guaranteed because of the non-real time OS (Microsoft Windows) in use. Running our experiments, we noted that experiments performed while the SAFEDMI is in start-up mode were highly repeatable, showing the same results in the various runs of the same experiment [15]. This is mainly due to the fact that in Start-up mode the EVC Packet Generator is not active, thus not introducing variability. Instead, because of the EVC Packet Generator, results differed significantly when the SAFEDMI is in Normal mode.

Due to the limited time period in which we had access to the SAFEDMI prototype, we collected only a limited number of observations. Consequently we compute a Type B uncertainty through an investigation of the system behavior instead of a Type A uncertainty computed through standard deviation. Type B uncertainty is estimated for time-related measurements as follows. When an event is raised, the \texttt{getTime} system call is invoked as an atomic action with the event: the contribution to uncertainty of this block of instructions is orders of magnitude smaller than 2 ms (it is at worst microseconds). For each event recorded, the resolution of the target system (2 ms) is the most significant contribution to uncertainty, while other contributions to uncertainty could be considered negligible. According to [2] and as already explained in Section 4.3, in such situations the true value is expected equally distributed in an interval given by the measured value and the measured value plus the resolution (e.g., if 10 ms is the measured value and the resolution is 2 ms, the true value is expected within the interval [10; 12] ms). The expected true value should be set as the midpoint of the identified interval with an uncertainty of at most half the interval (e.g., if the interval is [10; 12], the expected true value is 11±1 ms and confidence 1).

However, in this case study our purpose is to estimate the safety of a critical system: we preferred to differentiate from the approach proposed in [2] and to report an uncertainty that is conservative, meaning that it must never err on the side of being too small. Consequently, for each event, we pessimistically consider that the corresponding time instant is
collected with uncertainty of ±2 ms; thus, the uncertainty of time intervals is set to ±4 ms.

6.4 A glimpse on results
For all the injections performed, the safety mechanisms of the SAFEDMI were able to detect the error and correctly activate the safe state. Only in one case a slight violation of the requirement was observed, in which the reaction time (time to transit to safe state) was slightly exceeding the maximum allowed time of 100 milliseconds. In such case, it was suggested to system designers to shorten the period of the thread devoted to manage transition to safe state [15].

7 Concluding Remarks
This paper discussed the possibility of improving trustworthiness in measuring instruments and measurement results when assessing dependable systems by applying principles of measurement theory. The paper first reviewed principles and guidelines from measurement theory that are deemed more relevant for the experimental assessment of dependable system, and then it presented three case studies where such principles and guidelines were applied.

In Table 1, the guidelines summarized in four points presented at the end of Section 3 are reported, and it is described how these guidelines are addressed in the three case studies.

Reporting main observations, the first case study required the definition of an ad-hoc methodology, to build a measuring system that was adequate to measure a software clock. This methodology allowed to: i) mitigate uncertainty sources and intrusiveness, thus increasing our confidence in the measuring system and results, ii) achieve (and compute) a resolution sufficiently low, which is compatible with the objective of testing the software clock R&SAClock, iii) compute uncertainty, for each of the time value that we extracted from R&SAClock, and ultimately iv) achieve repeatable indications on the coverage of R&SAClock, i.e., results are repeatable. Ultimately, this case study is an example in which resolution is not intuitive to compute, differently to what usually happens as stated Section 2.
On repeatable measurements when testing dependable systems

Table 1 – Case studies at the light of the guidelines of Section 3.

<table>
<thead>
<tr>
<th>Guidelines</th>
<th>Case Study 1 R&amp;SAClock</th>
<th>Case Study 2 GPS devices</th>
<th>Case Study 3 SAFEDMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. measurand clearly defined</td>
<td>coverage of “true time is contained within the enriched time-stamp” for different configurations of R&amp;SAClock</td>
<td>position distance w.r.t. the reference device</td>
<td>latency and reaction</td>
</tr>
<tr>
<td>2. sources of uncertainty singled out and evaluated</td>
<td>an ad-hoc methodology allowed to evaluate sources of uncertainty, compute resolution and Type B uncertainty</td>
<td>main source is multipath, not mitigated due to the considered target systems</td>
<td>main source are small differences in the timings of the EVC Packet Generator</td>
</tr>
<tr>
<td>3. evaluate</td>
<td>deemed negligible thanks to attentive construction of the measuring system</td>
<td>absent by construction of the measuring system</td>
<td>deemed negligible thanks to attentive construction of the measuring system</td>
</tr>
<tr>
<td>intrusiveness</td>
<td>computed using an ad-hoc methodology</td>
<td>extracted from the data sheets of the devices</td>
<td>it is the DMI timer</td>
</tr>
<tr>
<td>resolution</td>
<td>similar indications on coverage are achieved in replicated runs</td>
<td>due to sources of uncertainty, repeatability could not be achieved.</td>
<td>achieved only for experiments in Start-up mode thanks to the absence of communication with the EVC Packet Generator</td>
</tr>
<tr>
<td>repeatability</td>
<td>enriched time values are measured with Type B uncertainty set to ±50 us</td>
<td>Type A, in the order of meters</td>
<td>Type B uncertainty set to ± 4 ms</td>
</tr>
</tbody>
</table>

In the second case study, the multipath was a relevant source of uncertainty that could not be mitigated, and it affected repeatability (as from Section 2, repeatability requires to investigate on [...] measurement uncertainty). Type A uncertainty could be computed for the replicated runs, however it resulted in several meters for each point of the paths due to the abovementioned source of uncertainty. The investigation of sources of uncertainty allowed us to rapidly understand the cause of variations in the results that we were collecting, i.e., to explain the large uncertainty we were measuring.

In the third case study we showed a measuring system designed to minimize intrusiveness, and an evaluation of Type B uncertainty which favored a conservative approach even at the cost of being a higher value (±4 ms). It is relevant to note that sources of uncertainty were identified
for a specific configuration of the target system (Normal mode), while in a different configuration (Start-up mode) such sources of uncertainty were eliminated, and as a consequence experiments were highly repeatable.

8 ACKNOWLEDGMENTS
Left blank for double blind review.

9 REFERENCES


[16] ALARP - A railway automatic track warning system based on distributed personal mobile terminals - FP7-SST-2010-234088.


