Towards techniques and methodologies for collecting trusted observation results

Andrea Ceccarelli
University of Florence, Viale Morgagni 65, I-50134 Firenze, Italy
andrea.ceccarelli@unifi.it

Abstract

In monitoring, data collection and analysis, especially for pervasive and dynamic systems, the data collected (the results of observations) can be affected by non-negligible errors and consequently they can lead to misleading or non-optimal decisions. Although guidelines and good practises to properly collect the results exist and are often applied, the uncertainty of results and the quality of the measuring system is rarely discussed. Written by a 1*st year PhD student at the University of Florence, this paper constitutes a research direction for the development of the PhD program: the paper presents a preliminary investigation and two examples towards the assessment of measuring systems and results to provide trusted observations.*

1. Introduction

Monitoring, data collection and analysis [1] are widespread activities in which results are collected and analyzed to compute quantities and to execute services. Especially in pervasive, highly distributed and dynamic systems, the results of the observations may be affected by errors: for example, typical sources of errors for time observations are clock synchronization errors, unpredictable network delays or changes in system dynamics [2]. If decisions are taken using results affected by errors, such decisions could be wrong or not optimal.

Information about the quality of results could help preventing such mistakes; monitoring systems and testing activities could take advantage of the possibility to use such information to increase the trustworthiness in the results and the consequent decisions [3].

Nevertheless, current situation is typically that, even if the measuring system (the monitoring system in use to execute services or for testing) is carefully designed and actually provides trusted results, there is seldom attention to quantify how well the measuring system performs and what is the uncertainty of the results collected [3].

We present three examples that point out such concepts. In Wireless Sensor Networks (WSNs), real-time and adaptive sensing applications require a specific time synchronization quality in order to collect reliable data (e.g. loose synchronization reduces the ability to precisely locate the source of a noise in applications for acoustic sources location [4]), but keeping clocks continuously synchronized brings relevant energy consumption: to estimate the quality of clock synchronization can help to determine an optimal trade-off between the energy costs of transmitting synchronization beacons and the necessity of highly synchronized clocks to collect reliable data [5].

As a second example, we consider measurements (and especially time measurements) performed on a system by a testing tool. If the tool perturbs the system in use and the extent of such perturbation is unknown, then the results collected may suggest a misleading representation of the actual system behaviour without the operator being able to notice this as such.

As a third example, we consider localization and tracking [6] algorithms in WSNs. The coordinates that identify the location of a subject may be erroneous due to several causes as obstacles (static or mobile), low quality of clock synchronization, non-constant RSSI (Received Signal Strength Indication), sensor nodes distribution and orientation, or environmental conditions [6]. The uncertainty in the localization, if available, allows to circumscribe the area in which the subject is surely located (e.g. limit the research area for avalanche victims).

In this paper we investigate the possibility to assess the measuring system and the results achieved for monitoring, data collection and analysis activities. The objective of the research direction identified is to define and exploit mechanisms, good practices and possibly methodologies that guarantee observations results that are trusted, providing benefits both for the design of systems and for the testing activities.

The paper is organized as follows. In Section 2 we identify guidelines and good practices for trusted observations and we identify the objectives for our research, in Section 3 we present two preliminary results where we exploited the previous argumentations, and in Section 4 we report conclusions.

2. Monitoring, data collection and analysis

In Section 2.1 we observe guidelines and examples for trusted observations, and in Section 2.2 we discuss the quantities that are mostly troublesome for monitoring and data collection in (especially dependable [1]) systems.

2.1. Guidelines and good practises to monitoring and data collection

In literature, there are several quantities that are recognized as representative of the measuring system and of the quality of measurements. For example, in the field of fault injection [7], we can identify quantities as perturbation, repeatability, reproducibility, monitoring-time resolution and controllability [7], [8]. Although such quantities are well-known and often addressed, there are no widespread rules and practices to drive in the assessment of measuring systems [3]. We identify metrology (measurement theory, [9]), a science that proposes standards and good practices for the assessment of measuring systems and results, as a possible candidate to provide such rules and practices.

In the following we present basic concepts of metrology; we note that these concepts are neither new nor revolutionary, but they introduce a more rigorous approach to the definition and achievement of trustworthiness in observations and results [3].

For a measuring system, the ability to identify the *uncertainty* of the measurement results is important. Uncertainty provides quantitative information on the dispersion of the quantity values that could be reasonably attributed to the measurand (the quantity that is measured); it represents an estimate of the degree of knowledge of the measurand. As an example of the importance of uncertainty in dependable systems, let us consider a hard real-time system. Because of possible errors in measurements, an indicator (the result of a measurement) may be recognized as below a given threshold, while actually it is not. Computing the uncertainty of the result could point out that there is the potential risk that the indicator is actually above the threshold.

Any measuring system perturbs the measurand, determining a modification of its value. Estimating and possibly minimizing such perturbation, that is estimating and minimizing the system's *intrusiveness*, is therefore desirable when designing a measuring system. For example, a tool for testing that collects sufficiently reliable data in a non real-time environment, may behave very differently in a hard real-time environment where timing predictability may be affected by the additional overhead of its monitoring task or by other mechanisms (e.g. a mechanism that performs fault injection).

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. Even if it is often significantly smaller than uncertainty, resolution is important in tools for the testing of real-time systems, since it needs to be much lower than the imposed time deadlines to achieve useful quantitative evaluations of the dependability quantities.

Finally, *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 location and operating conditions. Repeatability is crucial for testing: especially in distributed systems where recreating the same exact conditions is typically difficult (e.g. it is almost impossible to guarantee the same measurement conditions of a Wide Area Network through time) [10]. The challenge to achieve repeatability is minimizing the impact of factors which contribute to the measuring system but that are not controlled by the operator.

2.2. Measurements that require computation of their quality

We classify measurements that can be performed on computing systems in two classes: measurements with negligible uncertainty and measurements with non-negligible uncertainty.

The first class includes static quantities which depend on the static characteristics of the system (e.g. software quality measurements as number of source code lines), as well as countable dynamic quantities which depend on a particular execution of the system (e.g. number of packets re-transmissions, or number of queuing operations). Measurements belonging to this class are typically characterized by very low uncertainty.

The second class is identified by measurements with non-negligible uncertainty, which generally refer to the dynamic behavior of the system and involve the estimation of continuous quantities. Few examples are: end-to-end communication delays, quality of clock synchronization, Mean Time To Failure, Mean Time Between Failures.

It is easy to note that this latter class includes quantities whose measurement presents more challenges than those belonging to the former one. Looking closer at this class, we identified the crucial role of *time measurements* [3], that are typically the most critical ones to face when designing and testing systems and services. However we acknowledge that the analysis should not restrict only to time measurements: depending on the kind of system and service, different kind of measurements which suffer of non-negligible uncertainty may be involved and influence critical aspects of the system. For example the uncertainty in spatial measurements is a critical issue in algorithms for reliable localization and tracking.

We believe that many pervasive and adaptive systems and the testing activities performed on them can take advantage of the ability to use information on the quality of collected data: systems' services and testing tools that provide trusted and reliable results can increase trustworthiness in their decisions. As a research direction we propose to rely on guidelines and good practises from metrology, monitoring and data collection to identify and exploit indicators, mechanisms and methodologies that improve the trustworthiness of measurements performed.

3. Two examples to achieve trusted results

We show two examples in which previous observations are put into practice: in Section 3.1 we present an instrument for testing, and in Section 3.2 we show a mechanism (a software clock) that provides reliable information on the quality of clock synchronization.

3.1. A fault injection instrument for a safe Driver Machine Interface

In railway train-borne equipment, the Safe Driver Machine Interface for ERTMS train (SAFEDMI, [11]) acts like a safety-critical bridge between the train driver and the onboard automatic train control system (European Vital Computer, EVC [11]). We present the testing activities by software-implemented fault injection [7] performed on the SAFEDMI to evaluate the coverage offered by its safety mechanisms [11].

The measuring system build (an instrument for software-implemented fault injection) is composed of (i) a simple thread that performs run-time injections, (ii) a fault library that enlists the faults to inject and (iii) a low-intrusive monitor that collects events and allows off-line analysis of the results. The workload is generated by means of the EVC Packet Generator [11], a software executing on a PC with Microsoft Windows® OS connected to the SAFEDMI and that simulates the behaviour of a real EVC.

The measuring system and the achieved results have been assessed applying principles of metrology. In fact we investigated resolution, intrusiveness, repeatability and uncertainty. The resolution of the measuring system for time instants is 2 ms (milliseconds); it is the resolution of the SAFEDMI timer used as base for the activities of the scheduler and of all threads. Thanks to an attentive design of the measuring system that makes it extremely low-intrusive (the measuring system is basically composed of two low-priority threads, a monitor and an injector, that execute few and quick instructions), the intrusiveness has been evaluated negligible (it is orders of magnitude smaller than resolution). Repeatability of the experiments instead cannot be guaranteed. In fact i) the injector is a low priority thread (to provide low intrusiveness), as a consequence there are no guarantees that the injections are performed timely, and ii) the EVC Packet Generator does not guarantee the exact timing of the workload execution in different runs of the experiments because of the non-real time OS (Microsoft Windows®) in use. Finally, a type B uncertainty according to [12] for the time intervals was computed to be ±4 milliseconds.

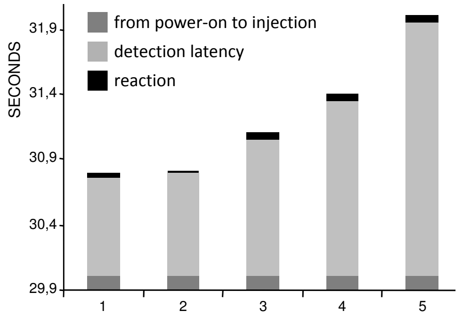

Figure 1. Five executions of a fault injection experiment on the SAFEDMI

The main quantities identified for the evaluation are two time intervals: (i) detection latency (time elapsed from the injection of an error and its detection) and (ii) reaction (after detection, time needed to react properly, i.e. entering a safe mode that prevents possible hazardous operations). As example, Figure 1 shows five independent executions of a fault injection experiment in which an erroneous execution flow is forced in the function that controls the LCD lamp. In the figure, the time interval elapsed from the power-on of the SAFEDMI to the injection is in dark grey, detection latency is in light grey and reaction is in black. The errors are always detected by the safety mechanisms of the SAFEDMI and they never result in a violation of the safety requirements.

Further details on the fault injection activity performed on the SAFEDMI can be found in [11].

3.2. A clock aware of synchronization uncertainty

Many pervasive and distributed systems require synchronized clocks to correctly execute their services. Some services could take advantage from the awareness of actual distance from global time (the *offset*); unfortunately the offset is typically a variable factor very hard to compute because of causes as changes in system dynamics, faults, or environmental changes [5], [13]. Synchronization mechanisms typically compute an estimated offset, but they usually offer no guarantees of closeness of this value to offset. Instead worst case bounds on offset are usually available (the *accuracy*), since they are imposed to systems as requirements, but unfortunately these bounds are usually far from typical execution scenarios and consequently are of little use in practice.

To this purpose the notion of uncertainty of the time measurement as used in metrology [12], [9] can provide an estimation of synchronization quality that in practice can result more useful than accuracy and offset. We call *synchronization uncertainty* the ability to provide an adaptive and conservative estimation on distance of local clock from global time: at any time t, synchronization uncertainty is required to be between offset and accuracy [14]. The first of the three examples shown in Section 1 is a sample case in which synchronization uncertainty is useful; a more detailed discussion is in [14].

The Reliable and Self-Aware clock (R&SAClock, [14]) is a software clock for external clock synchronization that provides to users (e.g. system processes) both the time value and the synchronization uncertainty associated to the time value.

When a user asks the current time to R&SAClock (by invoking the function *getTime*), R&SAClock provides an enriched time value *[likelyTime, minTime, maxTime, FLAG]*. *LikelyTime* is the time value computed reading the local clock. *MinTime* and *maxTime* are collected using the synchronization uncertainty provided by the R&SAClock internal mechanisms: *minTime* and *maxTime* are respectively a left and a right bound of the reasonable values that can be attributed to the actual time. The user that exploits the R&SAClock can impose an *accuracy requirement*, that is the worst synchronization uncertainty that the user can accept in order to work correctly. Correspondingly, R&SAClock can give value to its output *FLAG*, which is a boolean value indicating whether the current synchronization uncertainty is within the accuracy requirement or not.

Further details on the R&SAClock can be found in [14].

4. Conclusions

When observing systems, and especially complex and distributed ones, results affected by (measurement) errors may be collected by means of measuring systems; consequently the decisions taken exploiting such results may be not correct or optimal. The awareness of the quality of results and of the measuring system in use allows to discard or to opportunely weight results affected by a relevant error: in other words, trustworthiness in results and in the decisions taken exploiting them is increased.

In this paper we investigated guidelines and good practices to assess the quality of measuring systems and collected results; basing on such observations, we developed i) a trusted instrument for testing a safe DMI and ii) a clock that provides quantitative information on the quality of time measurements. As future work, we plan to identify mechanisms and methodologies for systems and for testing tools that increase trustworthiness in the observations collected and consequently in the decisions taken. The aim of our research is to improve both the quality of results provided by the assessment activities and the quality of data provided by systems and services.

Acknowledgment

This work has been partially supported by the European Community through the projects IST-FP6-STREP-26979 (HIDENETS) and IST-FP6-031413 (SAFEDMI). The author is indebted to Dr. Lorenzo Falai and Dr. Michele Vadursi, co-authors of works whose results proved useful to define the work here presented.

References

[1] A. Avizienis, J. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE TDSC*, vol. 1, no. 1, pp. Page(s): 11–33, 2004.

[2] P. Verissimo and L. Rodriguez, *Distributed Systems for System Architects*. Kluwer Academic Publisher, 2001.

[3] A. Bondavalli, A. Ceccarelli, L. Falai, and M. Vadursi, "Foundations of measurement theory applied to the evaluation of dependability attributes," in *DSN '07: Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 522–533.

[4] D. Ganesan, S. Ratnasamy, H. Wang, and D. Estrin, "Coping with irregular spatio-temporal sampling in sensor networks," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, pp. 125–130, 2004.

[5] K. Römer, P. Blum, and L. Meier, "Time synchronization and calibration in wireless sensor networks," in *Handbook of Sensor Networks: Algorithms and Architectures*, I. Stojmenovic, Ed. John Wiley & Sons, Sep. 2005, pp. 199–237.

[6] L. Girod and D. Estrin, "Robust range estimation using acoustic and multimodal sensing," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)*, Maui, Hawaii, October 2001.

[7] M.-C. Hsueh, T. Tsai, and R. Iyer, "Fault injection techniques and tools," *Computer*, vol. 30, no. 4, pp. 75–82, Apr 1997.

[8] "AMBER Consortium, Deliverable 2.1- State of the Art. April 2008, http://www.amber-project.eu."

[9] *ISO International Vocabulary of Basic and General Terms in Metrology (VIM)*, 3rd ed., BIMP, IEC, IFCC, ISO, IUPAC, and OIML, 2008.

[10] 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," *Software Engineering, IEEE Transactions on*, vol. 16, no. 2, pp. 166–182, Feb 1990.

[11] A. Ceccarelli, D. Iovino, and A. Bondavalli, "A fault injection tool for trustworthy, comparative measurements and analysis," in *IEEE SRDS 2008 Workshop*, October 5 2008.

[12] *Guide to the expression of uncertainty in measurement*, BIMP, IEC, IFCC, ISO, IUPAC, and OIML, 1993.

[13] M. Satyanarayanan and et al., "Pervasive computing: Vision and challenges," *Personal Communications, IEEE*, vol. 8, no. 4, pp. pp. 10–17, August 2001.

[14] A. Bondavalli, A. Ceccarelli, and L. Falai, "Assuring resilient time synchronization," in *SRDS 2008: Proceedings of the 27th IEEE Symposium on Reliable Distributed Systems*. Washington, DC, USA: IEEE Computer Society, 2008.