On Embeddable Passive Testing ⋆

Krzysztof M. Brzeziński *

* *Institute of Telecommunications, Warsaw University of Technology, Warsaw, Poland (e-mail: kb@tele.pw.edu.pl)*

Abstract: Despite fruitful research on passive testing algorithms and techniques, the concept of passive testing remains elusive, and is even dismissed as inherently inconsistent or a mere *façon de parler*. Consequently, passive testing still does not belong to the repository of conceptual, technological, and linguistic devices of computer science and telecommunications research communities. This work aims at "homing in" on the concept of passive testing. To this end, we concentrate on one aspect of this concept, namely on how a passive tester can be represented within the overall architecture of a distributed, reactive Discrete Event System. It is shown that such models in current use are deficient: internally inconsistent, or even "unimplementable". A new, very simple model is proposed, which allows a passive tester to be treated as any other system entity. This allows the modelling of systems in which a passive tester is embedded. The potential uses of such systems are also identified.

Keywords: Testing, model, architecture, behaviour, correctness.

1. INTRODUCTION

We consider the empirical, but formalized assessment of the logical behaviour of reactive systems of the DES (Discrete Event System) class. A classical example of such systems are distributed control structures, in which the individual system entities coordinate their actions by exchanging signalling information (messages, signals), according to the rules of a protocol. Distribution is not qualified by the geographical size of a system, which may be global, as in public telecommunications systems, or very localized (e.g., for on-chip structures).

The activity characterized above amounts to *testing*. In further discussion we refer to the selected elements of formal testing, established as part of the umbrella concept of *formal methods* (Woodcock et al., 2009), and, more specifically, to *model-based testing* (Broy et al., 2005). Model-based testing uses a formal model of the "correct" behaviour (also called a *reference speci cation* **Ref**), and a suitable algorithm (i.e., a finite sequence of logical steps) acting on this model, to assess the actual behaviour of a *Thing under test* (**Tut** – a generic, but non-standard term) and to automatically produce a *verdict* as to the "behavioural correctness" w.r.t. the model.

The methods, or the most general *kinds* of testing, may be divided into *active* and *passive*. This division is controversial. Passive testing is regarded by many researchers as not responding to the concept of testing at all, and it still remains the *niche* research subject, less developed (despite more than 25 years of research) than mainstream active testing. We advocate the uniform treatment of active and passive testing, as it can be shown (Brzeziński, 2009b) that there are no fundamental methodological reasons (other that the ICT research community's internal agreement, or Wittgenstein's *language games*) for excluding passive testing from the scope of the concept of testing.

The notion of testing being active is, apparently, *paradigmatic* of the "testing science". This state of affairs has various unfortunate consequences, some of which will be identified and, hopefully, remedied in the present paper. We concentrate on modelling issues. We argue that current system models, adequate for systems that are not subjected to testing at all, and for systems with an active tester, become deficient when a passive tester is introduced. We show the nature and consequences of these deficiencies. To remedy this situation, we develop a very simple model of a distributed system, in which a passive tester is treated on a par with other system entities (parts), and consequently can be *embedded* within a system.

The rest of the present paper is structured as follows. In Section 2 the distinction between *active* and *passive* testing is introduced and discussed, and the need for *embedded* passive testing is highlighted. Section 3 presents the current approaches to architectural modelling and shows the deficiencies of the currently used models. In Section 4, the proposed alternative model is developed. How this model "works" in practice, is shown in Section 5. Finally, the last section discusses the rôle of the proposed model within the comprehensive passive testing framework currently under development.

2. EXTERNAL AND EMBEDDED TESTERS

Active testing is understood as a joint activity of two *separate* systems: a *Test system* (**T**; also simply – a tester), and a *System under test* (**Sut**), one of the parts of which is a **Tut**, also referred to as *Implementation under test* (**Iut**). In this process, a tester: (a) *generates* and applies stimuli that provoke or stimulate phenomena within a **Sut**; (b) observes reaction to these stimuli – receives responses from

⋆ This research was conducted in the framework of a research project contracted by the Ministry for Science and Higher Education of the Republic of Poland, and financed with the 2007-2010 research funds.

a **Sut**; (c) analyses the relation (normally – some measure of similarity) between the actually obtained responses and those expected according to a pre-stated **Ref**; and (d) decides, basing on this analysis, on the assignment of a *verdict* $v \in \{\mathtt{P}, \mathtt{F}, \mathtt{I}\}$ (for Pass, Fail, and Inconclusive, respectively) that reflects the "behavioural correctness" of an **Iut** w.r.t. a **Ref** [1].

Practical active testing is a finite process (the execution of a *Test Suite* - a finite set of *Test Cases*), organized as a *Test Campaign*, and located at particular points in the life-cycle of a tested system (Brzeziński, 2009a). It is popularly associated with *the* "testing phase" that is placed at the end of the development (design and production) stage, before a system under test is commissioned for operational use [2]. Depending on the choice of an **Iut** within a **Sut** and the aims of testing, such tests may be divided into *conformance* and *interoperability* tests (in the telecommunications tradition), or into *unit*, *integration*, *system*, and *acceptance* tests (in the software engineering tradition). Active testing generally excludes the operational use of a **Sut**; if an operational system needs to be re-tested, it must be taken OOS (out-of-service) and suitably re-configured for active tests. When a test campaign is over, a tester is removed, and a **Sut** is further used *alone*. In this context, there is no clear reason for seeking the conceptual and implementation-related harmonization of the two systems – they are kept as separate as possible. In particular, there is no reason why a tester should, or could, be *embedded* in a **Sut**. On the contrary, the predominant view in telecommunications is that "design for testability" is explicitly prohibited – a tester may only access a **Sut** at its external interfaces, which have been provided for its normal operation (i.e., *not* strictly for testing), and a tester itself must be an *external* entity. Consequently, the modelling and implementation-oriented mechanisms of each of the two systems remain their "private" concern.

The picture presented above does not fit non-paradigmatic *passive* testing. A tester has now to *recognize* the behaviours of an **Iut** that are meaningful w.r.t. a **Ref** (and thus allow a verdict to be produced), without issuing any stimuli. Technically speaking, a passive tester is not equipped with a "sending channel" (Brzeziński, 2005, 2009b). This kind of testing is *campaign-less* – it may never terminate, so there may be no point in time when a tester is removed from a **Sut**. Both systems may thus jointly participate in the *operational* mission of a **Sut**. The rôle of a passive tester in this mission may be to provide a stream of verdicts for on-line detection of failures (where these verdicts are further *externally* processed by a human operator or another, higher-layer tester), or to provide the source of feedback for an *internal* control loop, in which verdicts directly influence the consecutive behaviour of a system. The former is akin to *DES Diagnosis* (Sampath et al., 1996) and the latter – to *Supervisory Control* (Charbonnier et al., 1999) or *Execution Monitoring* (Bauer et al., 2002) of DES systems; note also the potential relevance and applicability of the *self-testing* concepts. These research subjects are studied using a peculiar language of their own, in which the term "passive tester" appears rarely, if ever. We suggest that the body of research on passive testing could be directly applied to these works (which, surprisingly, seems to have not been explicitly proposed before).

It is now conceivable and natural to regard a passive tester not only as an external entity (possibly as a part of an external *active* tester [3]), but also as an *internal (embedded) part* of a system under test. The latter possibility, however, suffers from the lack of clear concepts and models that would treat a passive tester "module" on a par with any other part (subsystem) of a **Sut**. There has been no *need* for models with such properties in the research on paradigmatic active testing. Models that have been adapted to passive testing suffer from using non-generic or "magical" concepts and terms, such as "spying on a channel". In the sequel we will address such conceptual and linguistic (terminological) inconsistencies in descriptions of a system and its behaviour, by proposing a very simple, or indeed – minimal architectural model of a system in which a passive tester could be *embedded*. For this, the notion of "embedding" must be made explicit.

A *system* is endowed with a boundary, which may be "real", or *de dicto* – a construct devised for discourse purposes (Varzi, 2008). The latter may be, for the said purposes, freely moved. To say that interactions (messages, signals) pass across a well-defined boundary of a system usually means that this boundary is equipped with suitable devices, variously referred to as *gates*, *ports*, or *interfaces*. Another crucial property of systems is that they are composed of distinguishable entities (*parts*); in this aspect systems are in the scope of *mereology* – a theory of part-whole relations (Varzi, 2009). Let us propose a makeshift definition of embedding: an *embedded entity* is simply a *proper part* of a system, in the mereological sense. An embedded *system A* is a pragmatic notion. It denotes a proper part of another system *B* – a part that has a distinguished, well-defined and relatively self-contained functionality of its own, and is able to operate *as a system* also when "extracted" from *B*.

3. CURRENT MODELLING APPROACHES

Fig. 1 illustrates the common problems with modelling a system, in which a tester may be present. In fig. 1(a) a generic, familiar model of a distributed system (a future **Sut**) is shown. The vertices correspond to active system entities (nodes, processes) that are considered as local (i.e., non-distributed), and the edges represent the possibility of direct communication between the entities (i.e., not through other entities – this is *not* the transitive closure of the "direct connectivity" relation). These edges thus

[1] To be precise, test verdicts pertain to an *Object of assessment* (**Ooa**). Within ICT it is normally assumed that **Ooa** is an **Iut**, but this is *not* the essence of testing in general. Test verdicts may also apply to a **Ref**, e.g., when testing is used for reverse engineering (Brzeziński et al., 2008), or for *testing a hypothesis* in natural sciences (Brzeziński, 2009b).

[2] This is a simplified view, inherent in the "waterfall" system life-cycle model. Multiple, repeated testing phases are also conceivable, but they remain *phases*, i.e., each one of them is started and finished.

[3] Such use of a passive tester to complement the operation of an active tester is based on the idea of separating the problem of choosing and applying *test stimuli* (the "pure" active part of testing) from the problem of assessing the behaviour of an **Iut**. It has been considered by von Bochmann and Bellal (1989); von Bochmann et al. (1989).

Download English Version:

https://daneshyari.com/en/article/719521

Download Persian Version:

https://daneshyari.com/article/719521

Daneshyari.com