Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

A temporal agent based approach for testing open distributed systems

Salma Azzouzi *, Mohammed Benattou, My El Hassan Charaf

LARIT: Laboratory of Research in Computer Science and Telecommunication Faculty of Science Ibn Tofail University, Kenitra, Morocco

ARTICLE INFO

Article history:
Received 13 February 2012
Received in revised form 13 November 2014
Accepted 27 January 2015
Available online 4 February 2015

Keywords:
Distributed testing
Timing constraints
Multi-agent systems

ABSTRACT

The development of distributed testing frameworks is more complex, where the implementation process must consider the mechanisms and functions required to support interaction as long as the communication and the coordination between distributed testing components. The typical reactions of such systems are the generation of errors'set: time outs, locks, observability, controllability and synchronization problems. In other side, the distributed testing process must not only check if the output events have been observed, but also the dates when these events have been occurred. In this paper, we show how to cope with these problems by using a distributed testing method including timing constraints. Afterwards, a multi-agent architecture is proposed in the design process to describe the behavior of testing a distributed chat group application on high level of abstraction.

1. Introduction

Several reference models, architectures and frameworks such as ODP (Open Distributed Processing), CORBA (Common Object Request Broker Architecture), and TINA (Telecommunications Information Networking Architecture) have already been designed and proposed for developing distributed systems [1]. However, even though models provide a good basis for developing working open distributed applications, conformance testing approaches are required for gaining confidence in final products and guaranteeing their integration and interoperability within open distributed environment [2].

The conformance testing may be seen as mean to execute an IUT (Implementation under Test) by carrying out test cases, in order to observe whether the behavior of the implementation is conforming to its specification. Hence, the IUT (Implementation under Test) may be viewed in the context of open distributed systems as a system providing standardized interfaces for interacting with other systems. Based on testing of OSI communicating systems, conformance of an open distributed system can be assessed by attaching a related tester at each provided interface [4]. However, many problems influencing faults detection arise during the conformance testing process if there is no coordination between distributed testers.

In fact, the use of multiple testers introduces the possibility of coordination problems amongst remote testers. These potential problems are known as controllability and observability fault detections which are fundamental features of conformance testing [2–7]. To avoid these problems, many works [1–4] propose to introduce coordination messages which lead each tester to determine when to apply a particular input to the implementation under test (IUT) and whether a correct output from the IUT is generated in response to a specific input, respectively. Nevertheless, even if the introduction of such messages can resolve such mentioned issues, it leads to another kind of problems commonly known as synchronization problems. In this perspective, the distributed testing process must not only check if the output events have been observed, but also the dates when these events have been occurred especially if the system has to respect some timing constraints. Thus, two different time constraints are defined: transfer time i.e. the time required for a coordination message to travel from a tester to another, and reaction time, i.e. the time elapsed between the reception of an input by the IUT and the sending of the corresponding output by the IUT. We emphasize that even if some authors [4–6] have made a simplifying assumption that the time required for a transfer time, is greater than the reaction time of the IUT. Many academic researches [8,9] showed that controllability and observability problems are indeed resolved if and only if the test system observes those timing constraints.

In this paper, we focus on the temporal properties that specify the time required to exchange messages between the various components of the distributed test application. Thereby taking into consideration the temporal properties in the specification of the behavior of conformance testing it will provide a higher degree of expressiveness and flexibility. The proposed approach consists firstly on introducing a new architecture taking into account the delay of messages exchanged between testers and the IUT, and between testers. The main based idea of the proposed work is to develop an algorithm for generating Timing Local Test Sequences for each tester guarantying to avoid problems of coordination, observation and synchronization. Secondly, on high level of abstraction, we show how multi-agent based system used in distributed testing prototype realization contribute to capture the complex monitoring tasks of distributed tester behaviors.

* Corresponding author.
E-mail address: salma.azzouzi@gmail.com (S. Azzouzi).

http://dx.doi.org/10.1016/j.csi.2015.01.003
0920-5489/© 2015 Elsevier B.V. All rights reserved.

The paper is structured as follows: Section 2 describes the architecture and some modeling concepts of distributed testing application and presents the synchronization problems arisen in distributed testing execution. Section 3 is dedicated to introduce the architecture and modeling concepts of testing distributed applications with timing constraints. Section 4, presents the algorithm allowing the generation of timing local test sequences. Sections 5 and 6 describe the behavior of testing a distributed chat group application and design the test process of such application on high level of abstraction using a multi-agent architecture. Finally Section 7 gives some conclusions and identifies future works.

2. Distributed testing

Alongside the academic research activities, efforts have been made in the industry and standards bodies (ISO and ITU) to develop approaches to conformance testing of OSI protocols. A standardized methodology, tools, experiences and theories exist today with a certain degree of maturity [10]. This section is devoted to the extension of the results of testing protocols for dealing with the test of open distributed systems.

The principle is to apply input events to the IUT and compare the observed outputs with expected results. A set of input events and planned outputs is commonly called a test case and it is generated from the specification of the IUT.

2.1. Architecture

The basic idea of distributed testing architecture is to coordinate parallel testers called PTCs (Parallel Test Components) using a communication service in conjunction with the IUT. Each tester interacts with the IUT through a port PCO (Point of Control and Observation), and communicates with other testers through a multicast channel.

Fig. 1 gives an abstract view of what have could be an ODP distributed testing architecture. An IUT is the implementation of the distributed application to test. It can be considered as a “black-box”; its behavior is known only by interactions through its interfaces with the environment or other systems. Each tester sends some stimulus to the IUT via their attached interfaces called PCOs (Points of Control and Observations) and from which it observes the output IUT reactions. The external behavior of the IUT is observable via another interface type called IAP (Implementation Access Points). The difference between the PCO and the IAP is that PCOs are the logical points where communications are made, but the IAPs are the physical access points of the IUT. In order to control the test execution, PTCs exchange messages that encapsulate the information avoiding controllability and observability problems.

2.2. Test procedure

To approach the testing process in a formal way, the specification and the IUT must be modeled using the same concepts. The specification of the behavior of a distributed system is described by an automaton with n-port [11] (FSM: Finite State Machine) defining inputs and the results expected for each PCO. We denote Σ_k the input alphabet of the port “k” (PCO number k) and Γ_k the output alphabet of the port k. Fig. 2 gives an example of a 3p-FSM with: $Q = \{q_0, q_1, q_2\}$, q_0 the initial state, $\Sigma_1 = \{a_1, a_2\}$, $\Sigma_2 = \{b_1\}$, $\Sigma_3 = \{c_1\}$, and $\Gamma 1 = \{x_1, x_2\}$, $\Gamma_2 = \{y_1, y_2\}$, $\Gamma_3 = \{z_1\}$.

A test sequence of an np-FSM is a sequence in the form: $!X_1?Y_1!X_2?Y_2 \ldots !X_t?Y_t$ where for $i = 1,..,t$, X_i belongs to $\Sigma = \Sigma_1 U \ldots U \Sigma_n$ with $\Sigma_i \cap \Sigma_j = \emptyset$ for $i \neq j$ and Y_i is a subset of $\cup^n{}_{k=1}\Gamma_k$ such that, for each port k, $|Y_i \cap \Gamma_k| \leq 1$, i.e. Y_i contains at most one symbol from the output alphabet of each port of ***A***.

- $!X_i$: Denotes sending the message X_i to IUT.
- $?Y_i$: Denotes the reception of messages belonging to the Y_i from the IUT.

An example of a global test sequence (GTS) deduced from the 3p-FSM given in Fig. 2 is:

$$!a_1?\{x_1.y_1\}!b_1?\{x_2.y_2\}!c_1?\{z_1\} \tag{1}$$

Generally, test sequences are generated from the specification of the IUT and characterized by fault coverage. Several methods exist for generating test sequence from FSM specification. They are mainly used for detecting two basic types of faults output faults and transfer faults [12].

The work [4] allows generating local test sequences for each tester, and thus the behavior of the test application in each PCO is well defined. In fact, each tester executes its local test sequence (LTS), built from the global test sequence of the IUT. The generated LTS encapsulate the information that allows controlling the test execution. Indeed, many problems influencing fault detection during the conformance testing process arises if there is no coordination between distributed testers. These potential problems are known as controllability and observability fault detections which are fundamental features of conformance distributed testing. The controllability may be defined as the capability of the test system to realize input events at corresponding PCO in a given order, and observability may be defined as the capability of the test system to determine the output events and the order in which they take place at corresponding PCO [4]. To solve such problems, authors in [4] propose an algorithm to generate Local Test Sequences (LTS) from Global Test Sequence (GTS). The following LTS are the results given by applying the proposed algorithm to test sequence (1):

$$\begin{cases} W_1 = !a_1?x_1?x_2?O^3 \\ W_2 = ?y_1!b_1?y_2!C^3 \\ W_3 = ?C^2!O^1!c_1?z_1 \end{cases} \tag{2}$$

where:

- $!x$ denote sending of message x to IUT
- $?y$ denote receiving of message y from the IUT
- $!C^k$ denote sending coordination message to tester k and $?C^k$ receiving coordination message from tester k.
- $!O^k$ denote sending observation message to tester k and $?O^k$ receiving observation message from tester k.

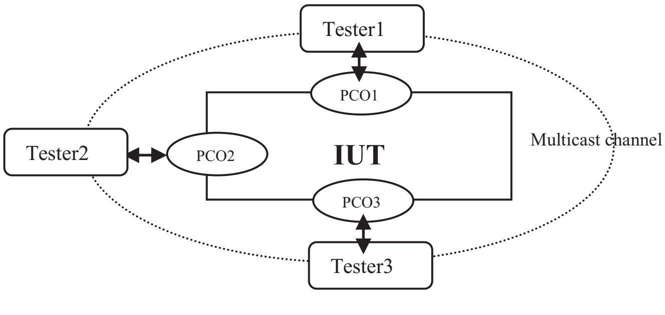

Fig. 1. Test architecture.

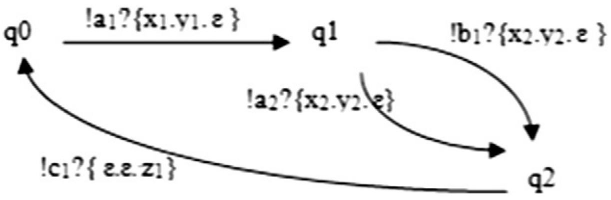

Fig. 2. An example of 3p-FSM.