Anatomy of a Test Assertion

A test assertion (or TA) must identify:

- The specification requirement(s) that it intends to verify.
- An object under test - what is supposed to conform to a specification requirement (which object is at risk of failing the test assertion)
- The conditions under which the test applies to this object, including some possible test action(s) or external event(s), and/or some predicate describing how the object under test qualifies for the test. The general term *antecedent* is used here to designate these conditions (action, event, predicate or any combination of these).
- The behavior (action, event) or quality that the object under test is expected to exhibit, under the previous conditions, as output of the test. The general term *consequence* is used here to designate this test output.

Also, the nature (not the details) of the test to perform in order to verify the requirements, must be clear enough, if not explicitly stated in the TA.

Consider the following specification requirement (let us identify it as "req 101"):

Specification Requirement (req 101): *"Purchase Order Receipt documents must be XML documents that are valid against the XML schema po-receipt.xsd, and that must use the approved format for product identifiers."*

The nature of the test associated with this requirement is clear: a schema validation operation, and a verification of the product ID content.

NOTE 1: although we could argue that there are two different tests involved (schema validation, content format), and two requirements actually merged into one, both define the single quality of being a P.O.Receipt: it is unlikely that either one of such requirements will be used separately in the specification. It is then appropriate to consider writing a single Test Assertion for both.

The related Test Assertion – named here TA-101 - can be described as:

Test Assertion: TA-101
Object: A document.
Antecedent: The document "claims" to be a P.O. Receipt. This will need to be specified further, as explained later (e.g. is it a response document related to a previously sent P.O.?, Or is it any XML document with a root element <PurchaseOrderReceipt>? or both?)
Consequence: The document validates against the po-receipt.xsd schema, and the <productRef> field has a value conform to the ProductNorm1234 specification, version 2005.

NOTE 2: The specification requirement req 101 is vague about the product ID format. This is because the specification editor is making two assumptions: (a) the reader knows which field of the document contains the product ID, and (b) the reader knows what the "approved format" is. Both (a) and (b) may have been more precisely specified somewhere else in the specification, or even externally to the specification. But the TA must be explicit and unambiguous on this, as it must be understood by a team of test case writers that may not have much expertise on the specification itself.

The wording of the *consequence* is factual. It states a *fact* that is expected to be observed in the test environment. It does not mimic a *requirement* which would make use of keywords such as MUST, SHOULD, etc. These are appropriate in a specification but not in a test assertion, which only *asserts* a result to be observed or not.

The object of a test assertion, may depend on the conformance target that is assumed for a specification requirement. For example, req 101 could have been read from the perspective of the entity (message handler, or business application) that produces Purchase Order Receipt documents. A slightly different wording might make this clearer - e.g.: (req 101b): *"*Purchase Order Receipt documents *in messages produced by a business endpoint* must be XML documents that are valid *…."* The related test assertion below (TA-101b) is defined for verifying conformance of the *business endpoint* instead of the conformance of any document claiming to be a P.O. Receipt:

Test Assertion: TA-101b
Object: a business endpoint.
Antecedent: the endpoint sends a message that contains a document "claiming" to be a P.O. Receipt. (e.g. it is a response document related to a previously received P.O. in a way that is detailed later in this section.)
Consequence: the document in the message validates against the po-receipt.xsd schema, and the <productRef> field has a value conform to the ProductNorm1234 specification, version 2005.

Consider now the following requirement (identified as *req 102*) in the same specification as previous req 101:

Specification Requirement (req 102): "*A message containing a PurchaseOrder document must always be responded within 24h by a PurchaseOrderReceipt message that refers to the original PurchaseOrder by having same Order reference value*."

In order to write a test assertion for req 102, one must identify the object under test. Here, it is no longer the document that is under test, but the business endpoint that is sending the *PurchaseOrderReceipt* document. This business endpoint may include the business application, or may be restricted to the message handler in case this handler is supposed to send automatically such receipts. Let us assume the latter. The message handler is the one that can fail this timing requirement. However, there are implicit assumptions behind a test assertion focused on the timing of this P.O. transaction. These assumptions are (a) the *PurchaseOrderReceipt* message contains a document that qualifies as PurchaseOrderReceipt, as verified by TA-101, and (b) a similar assumption for the *PurchaseOrder* message. We can see that this new test assertion will depend on other(s), and we will come back on this later.

For now, let us write a test assertion for req 102 by simply stating these assumptions as part of the antecedent and of the consequence:

Test Assertion: TA-102
Object: a message handler.
Antecedent: A message is received, that contains a document qualifying as PurchaseOrder.
Consequence: Within 24h, a message containing a document that qualifies as PurchaseOrderReceipt and that has an <orderRef> field with same value as the one in PurchaseOrder document, is sent by the handler.

We can see how verifying the test assertion TA-102 on a message handler, implies verifying TA-101 over PurchaseOrderReceipt documents produced by this message handler. This dependency is of a *pre-requisite*: TA-101 is a pre-requisite of TA-102 for the verification of the Consequence. For a message handler to pass TA-102, some document produced by this message handler must pass TA-101.
This dependency must be made explicit in TA-102, by adding a pre-requisite item:

Pre-requisite:
TA-101 (object: document, antecedent: sent by the handler and related to P.O. by <orderRef>).

Definition: *A test assertion TA1 is pre-requisite of a test assertion TA2, if verifying TA2 over its object requires also verifying TA1 over a related object.*

In the above definition, TA2 is also called a *post test assertion* of TA1.

Now, we could have another pre-requisite about the PurchaseOrder document received by the message handler, assuming there is a specification requirement (called req 100) that defines the qualification for PurchaseOrder documents. The related test assertion could be:

Test Assertion: TA-100
Object: a document.
Antecedent: the document "claims" to be a PurchaseOrder. It is an XML document with a root element <PurchaseOrder>.
Consequence: the document validates against the po.xsd schema, and the <productRef> field has a value conform to the ProductNorm1234 specification, version 2005.

NOTE 3: The antecedent - which defines under which conditions this TA must be exercised on a document – may depend on the test environment. In our example, when *the test environment has no control on the production of PurchaseOrder messages, and can only capture messages as they flow between messaging endpoints, some hints are necessary to detect when TA-100 must apply. This hint is here – quite arbitrarily - the presence of a root element <PurchaseOrder> in the document carried by the message. If the test environment allows for controlling the sending of PurchaseOrder documents – e.g. using a test driver that generates these – then it is possible to simply state as antecedent: "the document is claimed by sender to be a PurchaseOrder."*

TA-100 is another pre-requisite for TA-102: For a message handler to pass TA-102, some document used in the test must pass TA-100 (otherwise, it would not make sense to even attempt to verify TA-102 if the document supposed to be understood by the handler, is not conforming to the definition of a PurchaseOrder).

NOTE 4: Should then TA-100 and TA-101 always be exercised when a test case verifies TA-102 over a message handler? This remains the choice of test case writers. A test suite may decide to verify separately in a first phase that all PurchaseOrderReceipt documents produced by the message handler are conform (TA-101). This being assumed, the test case implementing TA-102 would not repeat this verification. Similarly, once the test harness has been proved to generate valid PurchaseOrder documents (TA-100), a test case implementing TA-102 may assume this is always the case, and not repeat TA-100. But the TA writer should not be influenced by these practicalities: from a logical viewpoint, passing TA-102 always requires some related objects to pass TA-100 and TA-101. This must be reflected in the definition of TA-102.

The full definition of TA-102 with its pre-requisite TAs becomes:

Test Assertion: TA-102
Object: a message handler.
Antecedent: A message is received, that contains a PurchaseOrder document.
Consequence: Within 24h, a message containing a document that qualifies as PurchaseOrderReceipt and that has an <orderRef> field with same value as the one in PurchaseOrder document, is sent by the handler.
Pre-requisites:
TA-100 (*target: antecedent*, object: document, antecedent: received by the handler, has a root element < PurchaseOrder>).
TA-101 (*target: consequence*, object: document, antecedent: sent by the handler, related to P.O. by <orderRef>).

NOTE 5: the two pre-requisites do not intervene in the same part of TA-102: TA-100 applies to an object used in the antecedent of TA-102, while TA-101 applies to an object used in the consequence of TA-102. In order to avoid any ambiguity, it is recommended to specify the target of these pre-requisites as either the antecedent or the consequence of the post TA (here TA-102).

NOTE 6: the antecedent of TA pre-requisites may be altered in order to comply with the *testing environment assumed by the post TA. Here, TA-102 assumes an exchange { PurchaseOrder, PurchaseOrderReceipt} which creates for TA-101 a very specific antecedent: the document to be tested is one that correlates with a previously selected PurchaseOrder.*

Could TA-102, TA-100 and TA-101 all be consolidated in a single TA? Yes if verifying conformance of PurchaseOrder and PurchaseOrderReceipt documents does not need be done outside the context of message transactions described by req 102. But if there is a need to verify conformance of such documents in other circumstances, then it is wise to keep TA-100 and TA-101 as distinct, to be reused as pre-requisites of possibly several other test assertions. Keeping the definition of all these TAs separate may also give greater latitude to test case writers in deciding when these pre-requisites may be verified in a test suite, e.g. for avoiding useless repeat of tests.