Test Framework

Introduction &
Overview of the Test Framework

Author(s): *imbus AG*
MoReq2 test development team

Date: *18/04/2008*

Version: *1.0*

Status: *Approved*

Customer: *Serco Consulting*

Contents

T1. Introduction

This document is the Test Framework for MoReq2. It is based on the MoReq2 requirement specification [1]. Before reading this document it is recommended to read and understand at least the first two chapters of MoReq2.

T1.1 Background

MoReq2 updates and extends the functional requirements of the original MoReq specification. The scope of the update and extensions is defined in the DLM Forum Scoping report [2]. A major addition introduced by MoReq2 is a compliance testing regime allowing suppliers to demonstrate unequivocally their compliance with the specification. This Test Framework represents the specification of the tests. The Test Framework has been prepared as part of the MoReq2 development project by imbus AG (www.imbus.de).

T1.2 Purpose and Scope of the MoReq2 Test Framework

The purpose of the Test Framework is to provide a consistent means to verify compliance with MoReq2. The Test Framework is designed to support tests of Electronic Records management Systems (ERMSs), which are assumed to be Off-the-Shelf products. The term Off-the-Shelf product and its quality requirement are defined in the ISO standard ISO25051 [4]. The MoReq2 Test Framework expects that these quality requirements are met by the tested ERMS products.

The Test Framework is structured as a set of testing modules that correspond directly with the requirement modules contained in the MoReq2 specification. Within each module at least one test case is provided for every functional requirement.

Non-functional requirements are beyond the scope of the Test Framework.

T1.3 How to Use this Specification?

The MoReq2 Test Framework is intended to be used:

- **by official MoReq2 Compliance Test Centres:** to support the implementation of a consistent compliance testing regime;
- **by potential ERMS users:** as a basis to prepare an invitation to tender;
- **by ERMS users:** as a basis for auditing or checking an existing ERMS and for preparing acceptance testing for an existing ERMS;
- **by academic institutions:** as a teaching resource;
- **by ERMS suppliers and developers:** as a guide to product development and in preparation for undertaking MoReq2 compliance tests;

The specification is written with focus on usability. The intention throughout has been to develop a specification which is useful and usable in practice.

T1.4 Constraints and Limitations

The purpose of this Test Framework is to verify the compliance of an ERMS with the MoReq2 requirements. The Test Framework will NOT verify functional correctness or completeness and it will NOT provide a substitute for component or system tests. Nevertheless the Test Framework may help to cross check if all MoReq2 functional requirements are covered by system tests.

The Test Framework is designed in such a way that the test cases can be executed economically and within parameters of time and effort. By necessity the Test Framework has to limit the degree of test detail.

The Test Framework is designed to test a “generic” ERMS. Therefore the test cases are designed independently of any specific ERMS implementation, of any vendor, and independently of the different business sectors, different scales and different organisation types where an ERMS is used.

Consistent with the MoReq2 specification, test cases are also designed independently of country specific regulations, legislations or requirements. The MoReq2 requirement allows country specific enhancements to be addressed in a “Chapter Zero” which is developed by a member. This Test Framework does not cover any “Chapter Zero” requirements.

The test specification adheres to the vocabulary of the MoReq2 requirement.

T1.5 Using the Test Framework

This specification has been prepared so that it can be used in paper or electronic form. It is published electronically as Word 2003 format and PDF.

It supports the execution of the test cases to document the test results.

T1.6 Organisation of the Test Framework

This document is structured along the following chapters.

- Chapter 2 (below) provides an overview of the MoReq2 Test Framework, terminology and usage.
- Chapters 3 to 10 (published separately) define the test cases for the corresponding chapters of the MoReq2 requirement specification.

The chapters in the Test Framework are numbered identically to chapters in the MoReq2 specification. Within each corresponding chapter there is no similar correspondence between requirements and test paragraph numbers. To ensure a clear distinction between references to test cases and references to requirements each test case is prefixed with “T” (e.g. T3.2.1 indicates a test case, whereas 3.2.1 is a requirement that is not necessarily related directly to T3.2.1).

T2. Overview of the Test Framework

T2.1 Key Terminology

The testing terms are used according to the ISTQB Certified Tester standard [5]. The ISTQB syllabus document [6] gives an overview of the terms structured by the specific contexts.

All MoReq2 specific terms are used according to the MoReq2 glossary (chapter 13 of the MoReq2 specification).

In addition the Framework uses a list of terms (see below), which are mainly used to express actions/operations within test cases. There may exist a different understanding of some terms. Consequently we hope that the following explanation will help to make the test case more readable and comprehensible.

Term	***Explanation***
Attempt	The verb “To attempt something” is used when a user shall attempt to execute an action though we expect the ERMS to deny execution. There may exist different ways in which the ERMS denies actions. For instance, an unauthorised user could be able to change a title within a graphical user interface, but he is not able to store the changes. Alternatively the unauthorised user may not be able to change a title because all necessary fields in the graphical user interface are read-only, or are hidden.
Assign	The verb “to assign” is used synonymously for “to allocate” and “to add” as well as other synonyms.
Bring in	The verb “to bring in something” is used to put in a record into the ERMS. There exist different options to do this. It may be the declaration or the import. In relation with the test environment other options may be possible, too. However, when this term is used the tester can decide which option he wants to use.
Create	The term “to create something” is used to mean “create and store” a configuration item.
Complete	The term “to complete” is used in case of proceedings e.g. workflows, declaration of records as defined by user systems. However, we do not state the explicit steps to follow to complete the respective proceedings. In each case follow the steps predefined by the system.
Configuration time	The term “configuration time” – usually used in the preconditions of a test case – means that the test case has to be executed when the ERMS under test is set up and configured.
Configure	The term “to configure something” is used for the configuration of system parameters or settings at configuration time (compare with MoReq2 glossary).
Change	The term “to change something” is used if properties of an item should be changed, yet the test case does not state how the property is changed.
Initiate	The term “to initiate something” is used in the sense of “to trigger a predefined action”. The initiation could be achieved manually by a user or automatically by the ERMS (e.g. recognition to specific record types etc.). Where initiate is used without further instruction the tester can choose how to trigger a proceeding or action.

Term	*Explanation*
Reset	The term “to reset something” is used to generate the precondition of a test cases. The testers brings the ERMS into a status that allows to create or import a classification scheme. If the ERMS does not support multiple classification schemes or where an additional classification scheme cannot be added into an existing one this would imply another set up of the ERMS.
Set up	The term “to set up something” is used to describe the activity of installing the ERMS. In some cases the tester needs to re-set up to have the possibility to change specific configuration parameters of the ERMS.

T2.2 Structure of the Test Framework

T2.2.1 Test Modules

The Test Framework is structured as a set of test modules (e.g. “T3 Classification Scheme and File Organisation”), each consisting of one or several sub-chapters (e.g. “T3.1 Configuring the Classification Scheme”). The structure of these test modules and sub-chapters corresponds directly to the structure of the MoReq2 specification.

Please read the general remarks of each sub-chapter as they give detailed information to test data and the test execution itself. If there is a “global” precondition for all test cases of the sub-chapter, you will find a short note there. In such a case the “global” precondition(s) will not be repeated in the specific precondition field of each test case.

T2.2.2 Test Case Structure

As far as possible the test cases are written in plain English and are presented in tabular form to make them both easily readable and easy to follow. Each test case is presented in an IEEE 829 [3] compliant standard format, as illustrated below.

I. Global test case information	
test case id:	<TEST CASE UID> e.g. T1.2.2
test case priority:	<Mandatory | Optional | Not Testable>
test case description:	Overview of goals and approaches of the test case. Remarks to be observed by the tester
req.-ID:	List of MoReq2 specification requirement ids that are covered by this test case. E.g.: 3.1.2, 5.2.3, 10.1.2

II. Test case		
a. precondition		
• <Description text of precondition 1> • <Description text of precondition 2> • …		
b. test steps		
step	action/operation	check/ expected result
1.	<Test step Action >	<Corresponding (succeeding) check to verify the test results of the step>
2.	<Next test step >	
3.	…	
c. postcondition		
• <Description text of post condition 1> • <Description text of post condition 2> • …		

III. Test result	
defects / deviations	*verdict*
List of defects and deviations detected by the test. If the defects are expected to be tracked in a defect management system, it is sufficient to list the IDs of the defects here. E.g. Bug01007, Bug Bug01220	☐ Passed ☐ Failed <Documentation of the test result>
Remarks	*tester*
Space to add remarks / observations during the test execution	<Name of the executing tester> <Signature of the executing tester> date, signature

The test cases include cross references to the IDs of the requirements they are designed to verify (field “req.-ID”).

Each executed test case has a clear outcome – PASSED or FAILED. A test case is only passed if all test steps are passed. If one test step fails the complete test case gets execution status "Failed".

If the tester uses test data deviating from the description in the test data repository, the deviation as well as the cause of the deviation has to be written down in the field "remarks".

Prioritisation

The test cases are marked with priorities according to the following scheme:

- If at least one of the covered requirement is a mandatory requirement the test case will be marked as "mandatory";
- If only desired requirements are covered the test case will be marked as "optional";
- If requirements are marked as not testable we have introduced corresponding test cases as placeholders. These test cases are marked as "not testable".

In some cases, a requirement is mandatory only if a desirable requirement is met as a precondition. The priority of the corresponding test cases is set according to the rules above as "mandatory". If the optional precondition is not supported by the ERMS the test case can be regarded as "not testable" and will be therefore not relevant for further observations.

Section 1.12 of the MoReq2 Requirements specification ([1]) gives an example for this case:

- Requirement 3.1.17: The ERMS should support the export of all or part of a classification scheme;
- Requirement 3.1.18: Where the ERMS supports the export of all or part of a classification scheme (as in req. 3.1.17) this must include associated metadata […]

  This means that the functionality required by 3.1.18 is mandatory if, and only if, the desirable functionality required by 3.1.17 is provided.

MoReq2 includes several "partially testable" requirements. An example is "the ERMS should not limit the number of levels in the hierarchy." There is no way, formally, to test for the absence of a limit. However, during the testing it is possible that a limitation on the number of levels might be noticed; if this happens, then the ERMS under test has failed. More generally, if at any time during the testing, the tester notices that the ERMS does anything that is contrary to a partially testable requirement, then this is equivalent to failing a test, even though the observation might not result from a scripted test. In this example, it is possible that the tester notices that the ERMS limits the number of levels, possibly while testing some other function such as export; in this event the ERMS does not comply with MoReq2, as it has failed this partially testable requirement. In this event the tester should record the nature of the non-compliance and how it was detected and observed.

T2.2.3 Test Data Repository

The Test Data Repository (TDR) is used as a generic description of test data details. The TDR describes only those test data which are necessary for testing. Within the test modules the test cases refer to test data items by using specific identifiers. As the MoReq2 metadata model defines elements for such identifiers (e.g. M163 Identity.system_identifier) we use these elements for the linkage between test cases and the Test Data Repository. In existing ERMS

these values will often be system-generated and therefore do not have to match with the values in the TDR as long as it is possible to uniquely identify an item.

General Remarks

Please note that the Test Data Repository is just a guideline on how to use test data for effective testing. However, it might be possible that suggested test data do not match a specific ERMS configuration. In these cases the suggested test data cannot be a reason for failing the test and therefore stating none-compliance to MoReq2.

Because of this fact the tester may vary the test data, under the condition that the reason for this is entered within the test cases (field “remarks”; cf. the test case structure in chapter T2.2.2).

The following variation is an example of how this might occur:

- Regarding to retention and disposition schedules, MoReq2 does not state the value of a retention period. Some systems may express the retention period of retention and disposition schedules by using months, e.g. 24 months; others may express it by using years, e.g. 2 years.

- However, if the test data suggest 24 months as the retention period and the ERMS just supports years, it is of course acceptable to vary the test data. Both options are compliant to MoReq2.

General Structure

The Test Data Repository is organized within several sub-chapters – one corresponding (Sub) Test Data Repository for each test module. This reduces complexity and guarantees higher readability and comprehensibility for the user.

Please note: The system identifiers used in the Test Data Repository are unique only within one (Sub) Test Data Repository, they cannot be used to identify an item across more than one (Sub) Test Data Repository.

Each Test Data Repository has the following structure:

- **Agents & Access Controls**: This sub-chapter gives details about user profiles, user roles and user groups. Furthermore you will find explanation which user roles have access permission to functions, aggregations and records. In addition we explain which user groups have access permission to aggregations and records.

- **Classification Scheme(s):** This sub-chapter includes all classification scheme(s) being used within the corresponding sub-chapter of the test module. It lists information of the metadata of the classification scheme itself and all aggregations and records within it.

- **Record types:** This sub-chapter describes the record types that are required for the test cases.

- **Retention and Disposition Schedules:** This sub-chapter gives detailed information on the retention and disposition schedules used.

- **Used content of captured records:** This sub-chapter gives detailed information about the content of the items that are captured as records. As these items are described before they are captured into the ERMS, no metadata elements can be used for the description.

Used Metadata Elements

Each sub-chapter contains tables such as the following:

M163 Identity.system_ identifier	**M167** Description. title	**M189** Description. email.address	**M171** Relation.entity_ agent	**M166** Relation. has_role	**M165** Relation. s_member_of	**M169** Use. administrator	**M170** Use. inactive
<U01>	User1						
<U02>	User2			<R01>		No	
<U03>	User3			<R02>		Yes	

This table describes all metadata elements and their values that are relevant for testing. The heading shows several metadata elements (underlined in grey). These elements are taken from the MoReq2 metadata model. The metadata elements themselves are not used for testing, only the metadata values are used.

Example: The test case requires the creation of a user with the name USER1. For passing the test case a check is made to confirm that a user with the name USER1 is created. This implies one of the following situations:

- The value USER1 is stored explicitly in a specific metadata element of the ERMS. In this case the test case is NOT intended to confirm that the ERMS stores the value USER1 in a metadata element “Description.title” (M167) – it is only intended to ensure that the value USER1 is stored in any metadata element.

or

- The value USER1 is stored implicitly. Two examples for this circumstance are described in chapter A9.3 of MoReq2. In this case the vendor must explain how the value can be retrieved from this implicit storage.

If a concrete value of a metadata element is not relevant for an item, no value is given and the corresponding cell table is greyed out.

Entity/Agent

The MoReq2 standard introduces a special metadata element called ‘Entity/Agent’ in A9.7. This element is used to describe the access permissions of agents (that is users, user roles or user groups) to different entities which is especially useful if there is a many-to-many relation between users and entities. Therefore these metadata elements are fictitious metadata elements that will not exist in most ERMS systems. Nevertheless it is an adequate method to describe these access permissions. MoReq2 assumes that a user has access permissions to an entity if there is no entity/agent defined for this agent and this entity and if there is no entity/agent that can be inherited.

The following table shows how the access permission of a user to a special entity is described using the entity/agent metadata element.

M175	M177	M176	M180	M181	M179
Identity.system_identifier	Relation.applies_to_agent	Relation.applies_to_entity	Use.rights.permission	Use.rights.end_date	Use.rights.start_date
<EA01>	<U01>	<CS01/01>	NO		

Column 1 of this table contains an identifier that uniquely identifies the entity/agent within this module of the Test Data Repository. This value is used in the classification scheme.

Column 2 defines the agent to which this entity/agent applies. It can contain the system identifier of a user, user groups or user roles. Therefore these tables have to be searched to find out the intended agent.

The value in column 3 defines to which entity the defined entity/agent applies. As the identifiers that are referenced in this column uniquely identify an item across several classification schemes, it is not necessary to separately state to which classification scheme an entity/agent applies.

The value in column 4 shows which access permissions the defined agent has on the defined agent. The only valid value for this column is to state that the defined agent has no access permissions to the defined agent.

T2.3 Test Execution

T2.3.1 Getting Started for Test Execution

For all test cases, the following approach should be taken:

- Read the remark concerning the recent sub-chapter
- Read the global Test case information
- Read and execute the Test case (paying attention to the preconditions and following the references to the test data repository)
- Write down the Test result

T2.3.2 Logical Sequence of Test Cases

The Test Framework is designed in such a way that the test modules and their sub-chapters correspond to the modules and sub-chapters of the MoReq2 specification. Within each sub-chapter the test cases are ordered in a way that allows the sequential execution of the test cases. Nevertheless some additional actions between test cases might have to be performed to reach the state in which the next test case can be performed. Therefore it is very important to carefully consider the preconditions of each test case.

T2.3.3 Import of Classification Schemes

It will be necessary to prepare several classification schemes within a test module. There might be different ways to do so:

- Create a classification scheme first, export it, reset the ERMS and import the classification scheme again.
- Generate the classification scheme by another application and import the generated data.
- When a MoReq2 XML scheme is published, the classification scheme can be prepared and imported regarding to this scheme.

We used the option mentioned at the first bullet within the test modules. Where the ERMS does not support the export the Test Center may choose one of the other options.

T2.4 Existing Test Modules and Test Data Repositories

The following table gives an overview of the test modules and Test Data Repositories that are included in the MoReq2 Testing Framework.

MoReq2 specification	**Test Cases Available?**	**TDR Included?**	**Remarks**
3 Classification Scheme and File Organisation	yes	yes	
4 Controls and Security	yes	yes	
5 Retention and Disposition	yes	yes	
6 Capturing and Declaring Records	yes	yes	
7 Referencing	yes	yes	
8 Searching, Retrieval and Presentation	yes	yes	
9 Administrative Functions	yes	yes	
10.1 Management of Physical (Non-electronic) Files and Records	yes	yes	
10.2 Disposition of Physical Records	yes	yes	
10.3 Document Management and Collaborative Working	yes	yes	
10.4 Workflow	yes	yes	
10.5 Casework	yes	yes	
10.6 Integration with Content Management Systems	yes	yes	
10.7 Electronic Signatures	yes	yes	
10.8 Encryption	yes	yes	
10.9 Digital Rights Management	no	no	All requirements of this chapter are not testable.
10.10 Distributed Systems	yes	no	
10.11 Offline and Remote Working	yes	yes	
10.12 Fax Integration	yes	yes	
10.13 Security Categories	yes	yes	

Appendix 1 – References

[1] **MoReq2 Requirement Specification**
Available from the following urls:
http://www.moreq2.eu
http://www.DLM-Network.org
http://ec.europa.eu/transparency/archival_policy
Due to be published in paper form by the European Commission Office for Official Publications of the European Communities

[2] **Scoping report**
for the development of the Model Requirements for the management of electronic records (MoReq2),
DLM Forum Working Group for the development of MoReq version 3 - February 2006;

[3] **IEEE 829 -** Standard for Software Test Documentation
Institute of Electrical and Electronics Engineers, 1998;

[4] **ISO/IEC 25051:2006** - Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Requirements for quality of Commercial Off-The-Shelf (COTS) software product and instructions for testing;

[5] **ISTQB Glossary of terms used in Software Testing**, Version 1.2,
http://www.istqb.org/downloads/glossary-current.pdf;

[6] **ISTQB Syllabus Foundation Level V2005**, 1st July 2005,
http://www.istqb.org/downloads/syllabi/SyllabusFoundation.pdf;

Appendix 2 – Acknowledgements

The MoReq2 test development team want to thank Serco Consulting namely Mr Peter Campbell-Burns, Mr Tim Burrows and Mr Marc Fresko for their ongoing support during the development of the Test Framework.

Further we want to thank all reviewers who have shared their views on the MoReq2 Test Framework with us. Special thanks go to Dr. Ulrich Kampffmeyer and Mr. Christof Jeggle of PROJECT CONSULT Unternehmensberatung GmbH (Germany) who have provided a survey on the Test Framework.

Last but not least the MoReq2 Test Framework would not exist without the rising work of all members of the imbus test development team itself. Therefore a thank goes to Ms Claudia Schieber, Ms. Esther Kausz, Mr Michael Haimerl and Mr Michael Sill.

Thomas Rumi

imbus AG