Good Practice Guides SmartCom Validation

(1) Test for communication interfaces used for the exchange of metrological data

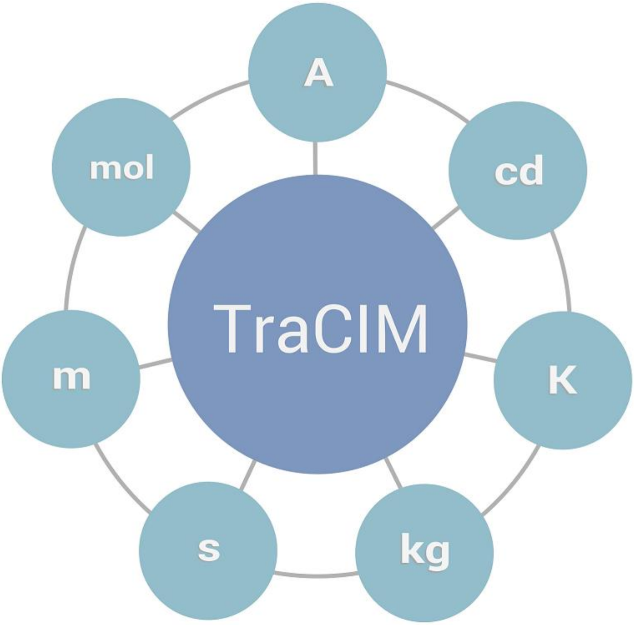
(2) Conformity test for unified DCCs

(3) TraCIM system

TraCIM D-SI

EN

DOI: 10.5281/zenodo.3816696



Good practice guide

TraCIM System

Integration of the SmartCom online validation system into TraCIM

Version 1.0

DOI: 10.5281/zenodo.3816696

Editors

Ostfalia University of Applied Sciences, Germany:
L. Heindorf, B. Müller

Physikalisch Technische Bundesanstalt, Germany:
D. Hutzschenreuter, J. H. Loewe

Comprising the results from our research and the fruitful and intensive discussions with all our other project partners and stakeholders worldwide.

Contact: smartcom@ptb.de

Braunschweig June 2020

DOI: 10.5281/zenodo.3816696

Table of Contents

DOI: 10.5281/zenodo.3816696

1 Introduction

The SmartCom online validation system (SmartCom Expert Extension – or in short SE) was developed by Ostfalia University [1] in the scope of the European joint research project EMPIR 17IND02 SmartCom [2]. It establishes a service that can be integrated into a TraCIM validation system to test and certify XML documents for correct usage of metrological data based on the D-SI format, such as Digital Calibration Certificates [3,4]. It can be used with the TraCIM Server 1.0 [5,6,7] and the new TraCIM Server 2.0 (provided by Ostfalia University [1]).

Normal TraCIM Expert Extensions deliver test data that will be used by a system under test to calculate a result (e.g. [8]). Then this result will be validated by the specific Expert Extension.

The validation process of the SE slightly differs from this normal process. The SE validates any well-formed XML document and therefore doesn't require test data and test data generation. Nevertheless, the corresponding REST API of the TraCIM server must be called in order to get an individual test process key for using the validation service. However, the SE doesn't generate test data.

DOI: 10.5281/zenodo.3816696

2 Integration of SmartCom online validation into TraCIM

The SmartCom online validation system (SE) is provided as a jar module for the TraCIM server application. This section explains how to use Maven [9] to build the module and run it with the TraCIM system.

This version was developed, tested and prepared to use with the new TraCIM server 2.0 and the corresponding Expert Extension API. Parts of the validation methods for the SE are provided through the SiChecker API. In the scope of the SmartCom project this API was implemented by PTB.

Because the TraCIM server as well as the SE come with their own installation instructions these instructions here serve the purpose to give an overview. Eventually, this document will not be updated on a regular basis. In doubt, please take a look into the TraCIM and SmartCom documentation with your used versions.

2.1 Set up a TraCIM service

In order to use the SE, you need a running TraCIM service. Refer to [1,7] to obtain and set up a TraCIM service.

The `jboss-deployment-structure.xml` of the TraCIM server must be extended with the following entry.

```
  <module name="deployment.smartcom-expert-
extension.jar" services="import" meta-
inf="import" optional="true" />
```

2.2 Build the SmartCom Expert Extension

The SE uses the PTB SiChecker for validation. Before building the SE, you have to install the corresponding dependency to your local repository.

```
  mvn install:install-file -Dfile=SiChecker-
1.0.jar -DgroupId=de.ptb.si.tools -
DartifactId=si-checker -Dversion=1.0 -
Dpackaging=jar
```

If you choose to use the integrated `SiChecker` you can remove the PTB SiChecker dependency from the Project Object Model (POM) and can continue without the PTB SiChecker.

Build the project.

```
  mvn clean package
```

2.3 Deploy the SmartCom Expert Extension

The build generates a jar with and without dependencies. You are free to use either of those. If you use the jar without dependencies, you must deploy all dependencies separately and edit the `jboss-deployment-structure.xml` accordingly.

The simple and recommended way is to use the jar with dependencies.

Use your preferred method for deployment, for example the deployment scanner:

```
  cp target/smartcom-expert-extension.jar
<WILDFLY_HOME>/standalone/deployments
```

or

```
  cp target/smartcom-expert-extension-jar-
with-dependencies.jar
<WILDFLY_HOME>/standalone/deployments/smartc
om-expert-extension.jar
```

(Renaming the jar with dependencies is not necessarily required as long as the file name is the same as the file name in the `jboss-deployment-structure.xml` of the TraCIM server.)

If you have deployed the SE while the TraCIM server was running, restart Wildfly or deploy the TraCIM server again.

2.4 Run the Integration Tests

We are using the REST Interface of the TraCIM server for integration testing. Therefore, a TraCIM server with the SE under test must be running on localhost and port 8080. The Maven command to run the integration test is

```
mvn clean test failsafe:integration-test
```

2.5 Run the OWASP Dependency Check

The OWASP Dependency Check is used by our POM. Please make sure to use the most recent version. To run the OWASP dependency check, one of the following two commands can be used:

```
mvn verify
```

or

```
mvn dependency-check:check
```

2.6 Customization

2.6.1 PDF reports

Each SmartCom test provides a report of the test results in PDF format. It may be required to change the output PDF document for various reasons such as to use company specific logos, comply with different reporting policies, or incorporate updates to the validation scope.

The package de.ostfalia.smartcom.pdf in the SE contains the Java source files that generate a PTB-specific test report. These files can be changed to modify the report. The open source Java tool Apache PDFBox [10] is used to create the PDF documents.

It is also possible to integrate an alternative PDF creation library. In this case, the PDF generation must properly be linked and run in the generateValidationResult method in the main class SmartComExtension.java in package de.ostfalia.smartcom of the SE.

2.6.2 D-SI XML Schema changes

The D-SI XML schema (the path of the schema) that should be used for validating the SI elements is defined in the SmartComValidator class of the SE. In order to validate against a new schema, the schema must be added as a resource to the project and the path to the schema in the SmartComValidator class must be modified accordingly.

The package de.ostfalia.smartcom.si contains the Java representation for the elements defined by the schema. If the schema changes, make sure to update the Java objects accordingly. When adding new types, make sure to explicitly declare the name in @XmlRootElement annotation. Additionally, they must be added to the jaxb.index file in folder resources/de/ostfalia/smartcom/si. If the namespace changes (prefix or URI), update the namespace in package-info.java.

No further changes are required. Namespace and tag names are automatically determined by reflection from those annotations. The version information of the schema (i.e. as printed in pdf) is automatically determined from the version tag inside the schema.

After modifying the elements, it might be necessary to build and distribute the new API for the adaption of the external SiChecker (see the following subsection).

2.6.3 Build the SiChecker API

For the development of external implementations of the SiChecker (i.e. PTB SiChecker), the API is needed. Currently the API is included in this project. In future the API might be extracted into a separate project.

In order to extract and build the API, one can use the script `build_api.sh` from the SmartCom Expert Extension Maven project.

As described above, the validation of a single SI element is performed by an implementation of the interface `SiChecker`. The used implementation can be changed in the `SmartComValidator` class. As default, the `PTBSiChecker` implementation is used. Additionally, the project contains a second implementation called `OstfaliaSiChecker` which can be used as an alternative.

Caution: The `OstfaliaSiChecker` checks only the units of real elements based on a regular expression as defined in the D-SI brochure [11]. Additional checks of the latest regulations [3] are not implemented.

2.6.4 Other properties

When the SE is used with the TraCIM system 2.0, additional information can be customized that will be displayed by the TraCIM server to users of the SmartCom validation. The relevant information is in the project folder `src/main/resources/META-INF`. The file `webshop.properties` contains general web shop properties and `resources/service.xhtml` allows the creation of a web page that includes a detailed description of the service.

DOI: 10.5281/zenodo.3816696

2.7 Other TraCIM Systems

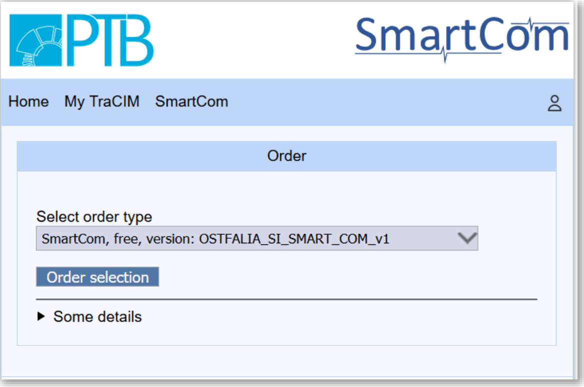
In order to use the SE with the old TraCIM server 1.0 (e.g. [6,7]) you have to change the version of the Expert Extension dependency in the POM.

Furthermore, the TraCIM server is used with PostgreSQL [12] and WildFly [13]. However, it should be possible to port to any RDBMS and Java EE application server. From WildFly, the `jboss-deployment-structure.xml` is used to tweak class loading.

3 Examples for the integration in TraCIM

This section gives a short outline on the integration of the SE into a TraCIM system 2.0 that was run as example by PTB within the SmartCom project [2].

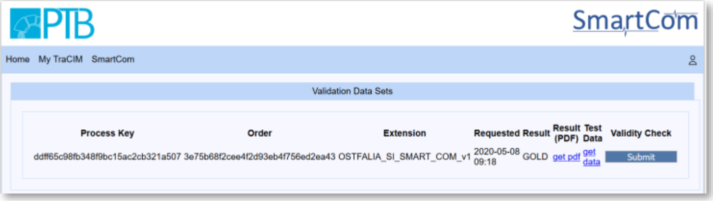
After registration and login, users can order the SmartCom validation in the order menu within "My TraCIM" as shown in Figure 1. There is the option to choose between a free test and a paid test in this example.



Select order type
SmartCom, free, version: OSTFALIA_SI_SMART_COM_v1

Figure 1: *Example of a SmartCom validation service selection in the TraCIM system 2.0.*

The user can also utilize the TraCIM web page to send D-SI XML data for validation and obtain the result by downloading the PDF result certificate (see Figure 2).



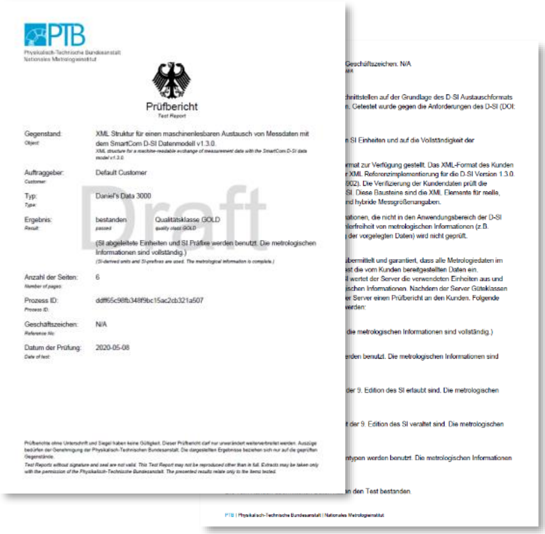
Validation Data Sets

Process Key	Order	Extension	Requested Result	Result (PDF)	Test Data	Validity Check
ddff65c98fb348f9bc15ac2cb321a507	3e75b68f2cee4f2d93eb4f756ed2ea43	OSTFALIA_SI_SMART_COM_v1	2020-05-08 09:18	GOLD	get pdf	get data

Figure 2: *Example of information on SmartCom tests that is made available by the TraCIM system 2.0.*

It is not necessary to build a TraCIM client as it was the case with the TraCIM version 1.0.

Figure 3 shows an example of the first page of the PDF report that is formatted in compliance with the PTB policy for test report.



Physikalisch-Technische Bundesanstalt
Nationales Metrologieinstitut

Prüfbericht
Test Report

Gegenstand: *Object*	XML Struktur für einen maschinenlesbaren Austausch von Messdaten mit dem SmartCom D-SI Datenmodell v1.3.0. *XML structure for a machine-readable exchange of measurement data with the SmartCom D-SI data model v1.3.0*	
Auftraggeber: *Customer*	Default Customer	
Typ: *Type*	Daniel's Data 3000	
Ergebnis: *Result*	bestanden *passed*	Qualitätsklasse GOLD *quality class GOLD*
	(SI abgeleitete Einheiten und SI Präfixe werden benutzt. Die metrologischen Informationen sind vollständig.) *(SI-derived units and SI-prefixes are used. The metrological information is complete.)*	
Anzahl der Seiten: *Number of pages*	6	
Prozess ID: *Process ID*	ddff65c98fb348f9bc15ac2cb321a507	
Geschäftszeichen: *Reference No*	N/A	
Datum der Prüfung: *Date of test*	2020-05-08	

Prüfberichte ohne Unterschrift und Siegel haben keine Gültigkeit. Dieser Prüfbericht darf nur unverändert weiterverbreitet werden. Auszüge bedürfen der Genehmigung der Physikalisch-Technischen Bundesanstalt. Die dargestellten Ergebnisse beziehen sich nur auf die geprüften Gegenstände.

Test Reports without signature and seal are not valid. This Test Report may not be reproduced other than in full. Extracts may be taken only with the permission of the Physikalisch-Technische Bundesanstalt. The presented results relate only to the items tested.

Geschäftszeichen: N/A
N/A

chnittstellen auf der Grundlage des D-SI Austauschformats
n. Getestet wurde gegen die Anforderungen des D-SI (DOI:

n SI Einheiten und auf die Vollständigkeit der

rmat zur Verfügung gestellt. Das XML-Format des Kunden
r XML Referenzimplementierung für die D-SI Version 1.3.0.
902). Die Verifizierung der Kundendaten prüft die
SI. Diese Bausteine sind die XML Elemente für reelle,
nd hybride Messgrößenangaben.

ationen, die nicht in den Anwendungsbereich der D-SI
lerfreiheit von metrologischen Informationen (z.B.
der vorgelegten Daten) wird nicht geprüft.

übermittelt und garantiert, dass alle Metrologiedaten im
est die vom Kunden bereitgestellten Daten ein.
I wertet der Server die verwendeten Einheiten aus und
ischen Informationen. Nachdem der Server Güteklassen
er Server einen Prüfbericht an den Kunden. Folgende
werden:

die metrologischen Informationen sind vollständig.)

erden benutzt. Die metrologischen Informationen sind

der 9. Edition des SI erlaubt sind. Die metrologischen

t der 9. Edition des SI veraltet sind. Die metrologischen

ntypen werden benutzt. Die metrologischen Informationen

n den Test bestanden.

PTB | Physikalisch-Technische Bundesanstalt | Nationales Metrologieinstitut

Figure 3: *Example of the first page of a PTB Test Report for a free SmartCom test of XML data with D-SI elements.*

DOI: 10.5281/zenodo.3816696

4 References

[1] Ostfalia University of Applied Science, webpage: https://www.ostfalia.de/cms/en/ (accessed May 2020).

[2] EMPIR project 17IND02 SmartCom, webpage: https://www.ptb.de/empir2018/smartcom (accessed May 2020).

[3] Hutzschenreuter D., Härtig F., Heeren W., et al.: SmartCom Digital System of Units (D-SI) Guide for the use of the metadata-format used in metrology for the easy-to-use, safe, harmonised and unambiguous digital transfer of metrological data – Second Edition, doi: 10.5281/zenodo.3816686 (accessed May 2020).

[4] Wiedenhöfer T., Hutzschenreuter D., Smith I, Brown C.: Document describing a universal and flexible structure for digital calibration certificates (DCC), doi: 10.5281/ zenodo.3696567 (accessed May 2020).

[5] EMRP project NEW06 TraCIM, project webpage: https://www.ptb.de/emrp/tcim.html (accessed May 2020).

[6] NPL TraCIM system 1.0 service, webpage: https://tracim.npl.co.uk/tracim/index.jsf (accessed May 2020).

[7] PTB TraCIM system 1.0 service, webpage: https://tracim.ptb.de/tracim/index.jsf (accessed May 2020).

[8] Wendt K., Brand U., Lunze U., Hutzschenreuter D.: Traceability for computationally intensive metrology user manual for Chebyshev algorithm testing, TraCIM e. V.,

https://tracim.ptb.de/tracim/resources/downloads/ptbwhz _math_chebyshev/ptbwhz_math_chebyshev_ma nual.pdf (accessed May 2020).

[9] Apache Maven project, web page: http://maven.apache.org/ (accessed May 2020).

[10] Apache PDFBox project, web page: https://pdfbox.apache.org/ (accessed May 2020).

[11] Hutzschenreuter D., Härtig F., Heeren W. et al.: SmartCom Digital System of Units (D-SI) Guide for the use of the metadata-format used in metrology for the easy-to-use, safe, harmonised and unambiguous digital transfer of metrological data, doi: 10.5281/zenodo.3522631 (accessed May 2020).

[12] PostgreSQL object-relational database system, web page: https://www.postgresql.org/ (accessed May 2020).

[13] Wildfly web application server, web page: https://wildfly.org/ (accessed May 2020).

DOI: 10.5281/zenodo.3816696

DOI: 10.5281/zenodo.3816696

DOI: 10.5281/zenodo.3816696

The content presented was developed within the framework of the EU-funded project SmartCom *"Communication and validation of smart data in IoT-networks"* with the support of international partners from science and industry.

⌂

https://www.ptb.de/empir2018/smartcom
(accessed June 2020)

EMPIR EURAMET

The EMPIR initiative is co-funded by the European Union's Horizon 2020
research and innovation programme and the EMPIR Participating States

DOI: 10.5281/zenodo.3816696