Signatures

Title : CCI Toolbox Technical Requirements Document (TRD)

Issue Date : 18 June 2018

__________________________________________
Norman Fomferra, BC; Anna Corlyon, TVUK

__________________________________________
Marco Zühlke, BC

__________________________________________
Jānis Gailis, [S&T]

__________________________________________
Brockmann Consult GmbH
Max-Planck-Str. 2
21502 Geesthacht
Germany

Copyright : © Brockmann Consult 2017
Document Change Record

<table>
<thead>
<tr>
<th>Document, Version</th>
<th>Date</th>
<th>Changes</th>
<th>Originator</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>24.06.2016</td>
<td>Initial version</td>
<td>BC</td>
</tr>
<tr>
<td>1.1</td>
<td>04.07.2016</td>
<td>Revised issue following PM2 meeting and ESA RIDS (CCI_Toolbox_D2.1_v1.1_RID.docx)</td>
<td>BC</td>
</tr>
<tr>
<td>1.2</td>
<td>26.09.2016</td>
<td>Further revisions following comments on inclusion of Shapefiles (in CCI_Toolbox_D2.1_v1.1_RID.docx).</td>
<td>BC</td>
</tr>
<tr>
<td>2.0</td>
<td>14.10.2016</td>
<td>New document version relating to software release 0.5 and the CCI Toolbox Documentation 2.0 [RD-4] whose chapter Detailed Design now reflects release 0.5 of the CCI Toolbox software.</td>
<td>BC</td>
</tr>
<tr>
<td>3.0</td>
<td>20.01.2017</td>
<td>Changed software name from “ECT” to “Cate”.</td>
<td>BC</td>
</tr>
<tr>
<td>4.0</td>
<td>25.04.2017</td>
<td>Minor changes:</td>
<td>BC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• changed team</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• external user documentation update to which this document refers to</td>
<td></td>
</tr>
<tr>
<td>5.0</td>
<td>23.06.2017</td>
<td>External user documentation update to which this document refers to</td>
<td>BC</td>
</tr>
<tr>
<td>6.0</td>
<td>07.09.2017</td>
<td>External user documentation update to which this document refers to</td>
<td>BC</td>
</tr>
<tr>
<td>7.0</td>
<td>18.06.2018</td>
<td>Fixed all links as we have renamed the CCI Toolbox ReadTheDocs project from “ect-core” to “cate”.</td>
<td>BC</td>
</tr>
</tbody>
</table>
## Distribution List

<table>
<thead>
<tr>
<th>Name</th>
<th>Affiliation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ed Pechorro</td>
<td>ESA</td>
</tr>
<tr>
<td>Simon Pinnock</td>
<td>ESA</td>
</tr>
<tr>
<td>Carsten Brockmann</td>
<td>Brockmann Consult</td>
</tr>
<tr>
<td>Marco Zühlke</td>
<td>Brockmann Consult</td>
</tr>
<tr>
<td>Norman Fomferra</td>
<td>Brockmann Consult</td>
</tr>
<tr>
<td>Kevin Halsall</td>
<td>Telespazio VEGA UK</td>
</tr>
<tr>
<td>Anna Corlyon</td>
<td>Telespazio VEGA UK</td>
</tr>
<tr>
<td>Janis Gailis</td>
<td>S&amp;T</td>
</tr>
<tr>
<td>Kenneth Hauglund</td>
<td>S&amp;T</td>
</tr>
<tr>
<td>Rainer Hollmann</td>
<td>DWD</td>
</tr>
<tr>
<td>Petra Fuchs</td>
<td>DWD</td>
</tr>
<tr>
<td>Stephan Herzog</td>
<td>DWD</td>
</tr>
<tr>
<td>Chris Merchant</td>
<td>University of Reading</td>
</tr>
<tr>
<td>Kevin Pearson</td>
<td>University of Reading</td>
</tr>
<tr>
<td>Frank Paul</td>
<td>University of Zurich</td>
</tr>
</tbody>
</table>
Table of Contents

1 Introduction and Background........................................................................................................8
  1.1 Purpose and Scope of the Document.........................................................................................8
  1.2 Structure of the Document........................................................................................................8
  1.3 Reference Documents................................................................................................................8
  1.4 Applicable Documents...............................................................................................................8

2 Technical Documentation Approach...............................................................................................10
  2.1 Problem ..................................................................................................................................10
  2.2 Solution ..................................................................................................................................10

3 Technical Requirements ................................................................................................................12

4 Operation Specifications .................................................................................................................13

5 Features and User Stories...............................................................................................................14
List of Figures

Figure 1: Technical documentation approach .................................................................11
Figure 2: Technical requirements in the detailed design (screenshot) ..............................13
Figure 3: Feature and user story in the issue tracker (screenshot) .....................................15
Figure 4: Labels used in feature lists (screenshot) .......................................................15
1 Introduction and Background

1.1 Purpose and Scope of the Document

This document constitutes the Technical Requirements for the CCI Toolbox software, whose public name is Cate.

As explained in more detail in chapter 2, this is a meta-document for the actual TRD content, which has been made part of a superordinate CCI Toolbox software documentation that also targets interested external readers. This documentation provided as an annex to this document.

1.2 Structure of the Document

This document comprises 3 parts:

- **Chapter 1:** Technical Documentation Approach explains the documentation approach taken and its application to the TRD.
- **Chapter 2:** Technical Requirements introduces the use of technical requirements and how they are represented in the TRD.
- **Chapter 3:** Operation Specifications explains the use of detailed technical specifications and how they are represented in the TRD.
- **Chapter 4:** Features and User Stories extends the concept of technical requirements, explains how use cases are broken down into features and how they are tracked and traced in the CCI Toolbox.

1.3 Reference Documents

<table>
<thead>
<tr>
<th>RD-1</th>
<th>CCI Toolbox User Requirements Document V1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RD-2</td>
<td>CCI Toolbox User Requirements Document Appendix I – State of the Art Analysis</td>
</tr>
<tr>
<td>RD-3</td>
<td>CCI Toolbox Software Verification and Validation Plan (SVVP) V5.0</td>
</tr>
<tr>
<td>RD-4</td>
<td>CCI Toolbox Documentation 5.0 provided as annex to this document (separate document CCI_Toolbox_Documentation_v5.0.pdf generated from cate.readthedocs.io on 23.06.2017)</td>
</tr>
</tbody>
</table>

1.4 Applicable Documents

|--------|-----------------------------------------------------------------------------------|
| AD-2   | Data Standards Requirements for CCI Data Producers v1.1, CCI-PRGM-EOPSTN-13-0009, 24 May 2013. Online copy: https://earth.esa.int/documents/10174/1754357/AD1-
CCI_Data_Requirements_Iss1.1_May2013.pdf


2 Technical Documentation Approach

2.1 Problem

The CCI Toolbox is a software development project that applies an agile, iterative software development process. The agile approach assumes that software may change at any time to make it compliant with the latest user and technical requirements.

In such as process, keeping technical documentation up-to-date can be a challenge, especially if the documentation is distributed over multiple (MS Word) documents that need to be maintained. Technical documentation quickly becomes outdated w.r.t. to evolving software. Also, considering that code is well documented by docstrings, there is a lot of textual duplication when the same text needs to be repeated in the technical documentation. This can imply expensive synchronisation effort. Last but not least, technical software documentation is often read by developers and scientists that would prefer online access but maintaining the technical documentation in MS Word documents is not the ideal format for online (HTML) publishing.

A solution for this dilemma has been developed that (1) fulfils the ESA SoW [AD-1] needs, must be therefore ECCS [AD-2] compliant, and (2) frees developers from the burden to write MS Word documents and code documentation and keep both in sync at the same time.

The project (document) deliverables that are concerned by the problem are

- Technical Requirement Document (TRD);
- Technical Design Document (TDD);
- Software Verification Plan (SVerP).

Partly also the

- Software Validation Plan (SVaIP);
- Iteration Plan (IPD).

Note: it has been decided to let the SVerP and SVaIP form a combined document, the Software Verification and Validation Plan (SVVP).

2.2 Solution

The technical documentation approach for the CCI Toolbox described here assumes that a considerable amount of technical documentation is written as docstrings in the software source code. However, documentation generated from source code often lacks readability and is more likely a detailed design documentation and reference.

Therefore, we have decided to use the Sphinx documentation system the generate the entire CCI Toolbox software documentation from a single source of reStructuredText (RST) pages and source code docstrings. RST pages have plain text format and can therefore be put under version control (git) together with the source code. Sphinx generates high-quality HTML outputs and has plugins that produce ePub, LaTeX, PDFs, and other formats. The single source is our GitHub repository for the CCI Toolbox Python package in a dedicated GitHub organisation CCI-Tools. The documentation is...
published through the open ReadTheDocs service. The software builds, unit-testing, and the documentation generation is done through the open Travis CI and CodeCov services.

The following diagram explains the approach in more detail.

Figure 1: Technical documentation approach

The team develops software, updates source code and RST pages, pushes them to GitHub. Changes on GitHub are detected by Travis CI, a build server, which pulls the current software version, builds it, performs the unit level tests, performs code coverage analysis, generates the documentation, and finally publishes it to ReadTheDocs as a web site, where ESA and the CCI Toolbox users can read it online or download it as PDF or as an eBook.

Instead of providing the documentation to ESA as MS Word documents (or its PDF version), the team provides to ESA the “meta documents” TDD, TRD, and SVVP in MS Word format, plus a PDF “content document” ESA CCI Toolbox Documentation (ECT.pdf in the diagram above) downloadable from ReadTheDocs. The meta-documents provide the table of content which links into the relevant chapters in the online and PDF content documents. They also tell the reader how the various chapter are related to each other. The ESA CCI Toolbox Documentation is provided as an annex to the TRD, TDD and SVVP.

All CCI Toolbox Documentation sources are under version control on GitHub at


The latest resulting CCI Toolbox Documentation [RD-4] is published on ReadTheDocs at

- cate.readthedocs.io/en/latest/index.html
3 Technical Requirements

The initial set of technical software requirements are mostly derived from the SoW [AD-1] and the URD [RD-1]. They have driven the preliminary and detailed design of the CCI Toolbox software given in the software Architecture and Detailed Design chapters of the CCI Toolbox documentation [RD-4] which is provided as annex to this document and online at cate.readthedocs.io.

Technical requirements have been placed into the docstrings of related (Python) modules, where they provide a useful reference for developers wishing to understand by which needs a given design was driven. Therefore, the chapter Detailed Design has been made the primary location of the technical requirements.

The chapter Detailed Design contains detailed sections for all packages and modules in the software. Every section provides a sub-section Description that provides the module’s purpose, contents, and possibly its usage. Module descriptions may link into Operation Specifications for further explanation and traceability of the detailed design. An optional sub-section Technical Requirements provides a mapping from URD [RD-1] requirements to technical requirements and software features that drove the design of a module. If available, links to verifying unit-tests are given in sub-sections called Verification. The sub-section Components lists all documented, non-private components of a module, including variables, functions, and classes.

Side note: The chapter Detailed Design is generated entirely from the docstrings attached to (Python) modules, classes, functions, and any global variables. The docstrings are written in reStructuredText (RST) format.

A table of content of the Detailed Design is not given here, because it is generated from source code and its structure and contents might change while the software is actively developed and a stable version has not yet been released.

To navigate to the technical requirements of a module, open chapter Detailed Design, and select a module section. Find the technical requirements in sub-section Technical Requirements:
The data model as described in RD-1, refers to the using, ingesting and supporting of shapefiles format. This is not the only vector representation available but is selected here because Glacier and Ice Sheets CCI product use Shapefiles for their data files. However, the Cate data model is not limited to the contents of Shapefiles. It rather exploits the *Shapefile schema* which became an early OGC data standard well known as the *Simple Features Model*. Other geo file formats such as GeoJSON, KML, GML have very similar data models and can easily be represented in the Simple Feature Model and vice versa.

4 Operation Specifications

Many user/technical requirements originating from the SoW [AD-1] and the URD [RD-1] cannot be converted straight forward into an implementation without a further, detailed technical specification of the operations to be implemented and used as part of a use case workflow. The CCI Toolbox’ operations that require a detailed specification are described in a dedicated chapter **Operation Specifications** in the **CCI Toolbox Documentation** annex.

These operations are derived from the Use Case Workflows which are on the other hand based on the Use Cases and their Problem Definition (see chapter on **Use Cases** at cate.readthedocs.io). As the
Use Cases, together with Source Documents (e.g. SoW [AD-1]) and user consultations provide the framework for the User Requirements, listed in the URD [RD-1], the latter are also reflected in the operations. Thus, each operation directly refers to a User Requirement, for instance the operation "Temporal Filtering" covers the user requirement CCIT-UR-LM0030 “The CCI Toolbox shall include functionality for temporal filtering/sub-setting by entering onset and end date.” For more details on this, the reader is referred to the Software Verification and Validation Plan (SVVP) [RD-3].

The operations (e.g. Product-Moment correlation (Pearson)) are grouped to Operation Categories (e.g. Data Inter-Comparison), which sometimes subdivide into Operation Subcategories (e.g. Correlation Analysis). They are specified by listing and describing:

- operation name, algorithm name, algorithm or method reference, a description of the task the operation is to fulfil, list of Use Case workflows comprising the operation
- applicable options
- input data
- output data
- parameters (also input data but have default values and are often used to control certain aspects of the algorithm behaviour)
- computational complexity
- convergence
- known error conditions
- code example

of the operation.

Every operation can form part of several Use Case workflows and every workflow, in turn, consists of several operations.

As the development of the ESA CCI Toolbox is supposed to be an agile and iterative process, operations are added and can also be re-specified during the toolbox evolution.

5 Features and User Stories

The CCI Toolbox development applies an agile development process. Each iteration of the toolbox will incorporate one (or more) of the specified Use Cases described in a dedicated chapter of the CCI Toolbox Documentation at cate.readthedocs.io. These were initially defined in the URD [RD-1]. The selected use cases will then be decomposed into specific software features to be implemented and any other software development tasks. A software feature is a named functional software component from end-user perspective (e.g. specific command, plot type, new processor or extension of a processor). Many features have a directly corresponding user/technical requirement in the URD, others may arise from the decomposition of Use Cases. To describe a feature in its context, we use user stories. A user story captures the “who”, “what” and “why” of a feature in a simple, brief, and concise way. It is important to note that both the Use Case decomposition and the derived user stories are an effort of the development team in close collaboration with the champion users.

All required features with their related user stories are maintained in the project’s issue tracking system provided by the GitHub repository:

Each feature is entered as an “issue” (also “ticket”). The GitHub issue tracker with the CCI Toolbox features are found at:
The issue’s name is the feature name, e.g. “Analysis report as HTML or PDF”; the issue’s description is the user story, see Figure 3.

Figure 3: Feature and user story in the issue tracker (screenshot)

Issues can and should be attached with labels, where labels such as feature together with labels such as uc09, uc11, cli can be effectively used to filter features and other relevant work units. Label examples are given in Figure 4.

Figure 4: Labels used in feature lists (screenshot)