CCI Toolbox

Technical Design Document (TDD)

Issue Date: 18.06.2018
Version: Issue 7.0
Contract Nr.: 4000115250/15/I-NB
Technical Officer: Dr. Simon Pinnock, ESA
Simon.Pinnock@esa.int
Technical Manager: Ed Pechorro, ESA
Ed.Pechorro@esa.int
Signatures

Title : CCI Toolbox Technical Design Document (TDD)

Issue Date : 18.06.2018

Authored :

Norman Fomferra, BC

Approved :

Marco Zühlke, BC

Approved :

Jānis Gailis, [S&T]

Approved :

Address : 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 following PM2 and RIDS from ESA (CCI_Toolbox_D2.2_v1.0_RID.docx)</td>
<td>BC</td>
</tr>
<tr>
<td>1.2</td>
<td>26.09.2016</td>
<td>Further revision following ESA comments (in CCI_Toolbox_D2.2_v1.1_RID.docx)</td>
<td>BC</td>
</tr>
</tbody>
</table>
| 2.0              | 14.10.2016 | New document version relating to software release 0.5 and the CCI Toolbox Documentation 2.0 [RD-4] whose chapter Architecture has the following changes:  
  • Section *Common Data Model* rewritten  
  • Section *Object Input/Output* added  
  • All sections updated  
  Chapters *API Reference* and *Detailed Design* now reflect release 0.5 of the CCI Toolbox software. | BC         |
| 3.0              | 20.01.2017 | Changed software name from “ECT” to “Cate”.                                                                                              | BC         |
| 4.0              | 25.04.2017 | Update of user documentation  
  • Restructured user documentation  
  • Included an “About Cate” section that introduces the three main software interfaces and the Cate’s core concepts  
  • Various minor changes reflecting API evolution  
  • Included a terminology | BC         |
| 5.0              | 23.06.2017 | Update of user documentation  
  • Included development guide  
  • Included GUI quickstart  
  • Included GUI reference  
  • Updated operation specifications | BC and [S&T] |
| 6.0              | 07.09.2017 | Update of user documentation                                                                                                          | BC and [S&T] |
- Included new section **Support / Known Issues** to let users know workarounds for frequently seen problems
- Updated development guide
- Updated and extended GUI reference

<table>
<thead>
<tr>
<th>Issue</th>
<th>Revision</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0</td>
<td>0</td>
<td>18.06.2017</td>
<td>Fixed all links as we have renamed the CCI Toolbox ReadTheDocs project from “ect-core” to “cate”. Numerous user guide documentation updates to which this document refers to (<a href="#">#409</a>).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Replaced references to Cate 1.0 to Cate 2.0 and updated most of figures</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Rewrote section about setup including installation and configuration</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Added a new section about the new <strong>STYLES</strong> panel</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Updated section about <strong>PLACES</strong> panel to include information on how to generate a polyline, polygon, and box</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Updated section about <strong>LAYERES</strong> panel to describe its new elements</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 ........................................................................................................9
   1.1 Purpose and Scope of the Document .........................................................................................9
   1.2 Structure of the Document .........................................................................................................9
   1.3 Reference Documents .............................................................................................................9
   1.4 Applicable Documents .............................................................................................................9

2 Technical Documentation Approach ..............................................................................................11
   2.1 Problem ....................................................................................................................................11
   2.2 Solution ....................................................................................................................................11

3 Software Architecture ....................................................................................................................13
   3.1 Overview ....................................................................................................................................13
   3.2 Common Data Model (CDM) .....................................................................................................13
   3.3 Data Stores and Data Sources ..................................................................................................13
   3.4 Operation Management ..........................................................................................................13
   3.5 Workflow Management ..........................................................................................................13
   3.6 Task Monitoring .....................................................................................................................14
   3.7 Command-Line Interface .......................................................................................................14
   3.8 Plugin Concept .......................................................................................................................14

4 Software Detailed Design ..............................................................................................................15

5 Application Programming Interface Reference ............................................................................15
List of Figures

Figure 2-1: Technical documentation approach .............................................. Error! Bookmark not defined.
1 Introduction and Background

1.1 Purpose and Scope of the Document
This document constitutes the Technical Design of the CCI Toolbox software, whose public name is Cate.

As explained in more detail in the next chapter, this is a meta-document for the actual TDD, which has been made part of a superordinate CCI Toolbox software documentation that also targets interested external readers.

1.2 Structure of the Document
This document comprised 3 parts:

   Chapter 1: Technical Documentation Approach
   Chapter 2: Software Architecture
   Chapter 3: Detailed Design
   Chapter 4: Application Programming Interface Reference

1.3 Reference Documents

RD-1 CCI Toolbox User Requirements Document v4.0

RD-2 CCI Toolbox User Requirements Document Appendix I – State of the Art Analysis

RD-3 CCI Toolbox Technical Requirements Document v6.0

RD-4 CCI Toolbox Documentation 6.0 provided as annex to this document (separate document CC_Toolbox_Documentation_v6.0.pdf generated from cate.readthedocs.io on 07.09.2017)

1.4 Applicable Documents


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 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 of the evolving software quickly becomes outdated. 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.

Finally, 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 to 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 Python core of the CCI Toolbox in a dedicated GitHub organisation CCI-Tools.
documentation is published through the open ReadTheDocs service. The software build and 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.

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 (Cate.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.

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

3 Software Architecture

The software architecture documentation is described in the dedicated chapter 6 Architecture of the CCI Toolbox Documentation annex [RD-4] which is provided as annex to this document and online at cate.readthedocs.io.

Side note: The chapter Architecture is generated from reStructuredText (RST files) and includes UML diagrams generated with PlantUML tool whose sources are also plain text (PUML files) and under version control on GitHub.

3.1 Overview

Section cate.readthedocs.io/en/latest/architecture.html#overview

Provides the high-level design, including all main packages. Explains that package cate.core is the generic heart of the software which are documented here. Informs that cate.ops and cate.ds are implemented as plugins that provide the concrete (climate) operations, and various data stores, respectively.

3.2 Common Data Model (CDM)

Section cate.readthedocs.io/en/latest/architecture.html#common-data-model

Describes the generic concept of the CDM and how it is applied the software. Refers to the Data module mentioned in the SoW [AD-1].

3.3 Data Stores and Data Sources


Describes the generic concept of data stores, which provides data sources, and data sources, which are the factory for instances of the CDM. Data stores are the fundamental data providers for the software, e.g. the ESA CCI Data Portal’s FTP service. Refers together with the CDM to the Data module mentioned in the SoW [AD-1].

3.4 Operation Management

Content: cate.readthedocs.io/en/latest/architecture.html#operation-management

Describes a generic concept of how operations such as data processors and other functions that operate on CDM instances are represented and registered in the software. Refers together with the CDM to both the Logic and Presentation modules mentioned in the SoW [AD-1]. Presentation, because operations can be of any type.

3.5 Workflow Management

Content: cate.readthedocs.io/en/latest/architecture.html#workflow-management

Describes a generic concept of how operations (and other executable processing steps) can be combined as workflows. The workflows is basic data model for the Graph Builder, that may later become part of the GUI. Refers together with the CDM to both the Logic module mentioned in the SoW [AD-1].
3.6  Task Monitoring
Content: [cate.readthedocs.io/en/latest/architecture.html#task-monitoring](cate.readthedocs.io/en/latest/architecture.html#task-monitoring)
Describes a generic concept of how operations, workflows, and workflow sub-steps can be monitored and controlled by the process that invoked them. Refers together with the CDM to both the Logic and Control modules mentioned in the SoW [AD-1].

3.7  Command-Line Interface
Content: [cate.readthedocs.io/en/latest/architecture.html#command-line-interface](cate.readthedocs.io/en/latest/architecture.html#command-line-interface)
Describes both the concrete CLI implementation and a generic concept of how the CLI can be extended. Refers to the Control module mentioned in the SoW [AD-1].

3.8  Plugin Concept
Content: [cate.readthedocs.io/en/latest/architecture.html#plugin-concept](cate.readthedocs.io/en/latest/architecture.html#plugin-concept)
Describes a concrete concept of how the software can be extended by plugins and lists the currently available extension points. Refers mostly to the Cardinal Requirements stated in the SoW [AD-1].
4 Software Detailed Design

The software detailed design documentation is described in the dedicated chapter 8 **Detailed Design** of the CCI Toolbox documentation at cate.readthedocs.io.

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 and Data Processor 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 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.

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 the Glacier and Ice sheets CCI product use Shapefiles for their data files. However, the ECT 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.

5 Application Programming Interface Reference

The software detailed design documentation is described in the dedicated chapter 7 **API Reference** of the CCI Toolbox documentation at cate.readthedocs.io.

The official API of the software is basically a carefully selected subset of public classes, functions, and global variables that the team will try to keep as stable as possible from release to release.

A table of content of the **API Reference** 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.