Software Architecture Document (SAD) for the Interferometric Modules of the Next ESA SAR Toolbox (NEST)

Contract number:
20809/07/I-LG

Prepared by:
PPO.labs

Prepared for:
The European Space Agency

Revision history:

Version	Date	Prepared/Revised by	Description
1.01	12.04.2013	Petar Marinkovic	Initial version and revisions
1.1	23.04.2013	Petar Marinkovic	2nd revision and reorganization
1.2	24.04.2013	Petar Marinkovic	Update and revision of Figures

Executive Summary

This document gives an architecture overview of the Interferometric processing system as implemented in Next ESA SAR Toolbox (NEST). It contains high-level descriptions and high-level diagrams on the architecture and components of the developed interferometric processor. The document gives a broader view, and it is prepared for the users, and not developers. Formal design documents, including the detail software documentation, data-flow diagrams, class dependencies, and other development related documentation, are available via the project web-page.

Table of Contents

1 Scope

1.1 Identification

PPO.labs submits this Software Architecture Document (SAD) identified as PPO-NEST-InSAR-RS04-2013, to the European Space Agency (ESA).

This document provides the high-level description of the design, system, and individual components of the interferometric modules of NEST. The document identifies development, operational and support environment, as derived from the Technical Specifications [Ref 1].

1.2 Project Overview

In October 2007, ESA awarded Array a contract to develop the Next ESA SAR Toolbox (NEST), [Ref 1]. NEST is an open source, developed and distributed under GNU GPL license, toolbox for reading, post-processing, analysis and visualization of the large archive of SAR data (Level 1 and higher). The toolbox supports processing of ESA SAR missions, including ERS-1 and 2, and ENVISAT. In addition, NEST can handle products from third party missions, including ALOS PALSAR, TerraSAR-X, Radarsat-2 and Cosmo-Skymed. NEST has been built on top of BEAM Earth Observation Toolbox and Development Platform.

For development of interferometric modules, ESA contracted PPO.labs, a company with specific InSAR background. In the initial development cycle, The Delft University of Technology (TU Delft) had a role as the consultant. This dedicated development of interferometric modules in NEST has started in November 2009, [Ref 2]. The interferometric modules are integrated in NEST, with the support for all interferometric acquisition modes, of both ESA and third party missions.

The interferometry modules are released in NEST since version 3C.

In the initial development cycle, as algorithmic prototype for Interferometry modules, DORIS (Delft Object-oriented Radar Interferometric Software), [Ref 6], developed by TU Delft was used. Specifically DORIS software was used in development iterations for NEST versions 3C and 4A. In all following releases, 4B, 4C and 5A, library for interferometric processing in Java developed by PPO.labs, within the development contract for interferometric modules in NEST, is used.

1.3 Document Overview

This architecture document explains the implemented system for interferometric processing as a whole. It gives an overview on how the processing chain is implemented. It contains high-level diagrams that explain how different parts of NEST communicate with the developed library for interferometric processing. It also give an explanation of the main components and features of the interferometric processing library.

Other design documents, prepared having a specific interferometric feature in mind, together user manuals, guides, and tutorials are available via the project web-page.

1.4 Document Organization

The remainder of this document is organized as follows:

- Section 2 – gives the list of references;
- Section 3 – summarizes the design methodology and development environment;
- Section 4 – reviews the architecture and design of interferometry modules;
- Section 5 – gives details about the organization of the interferometric library, lists core classes, and packages.
- Section 6 – lists used acronyms and abbreviations.

2 Referenced Documents

2.1 Applicable Documents

[Ref 1]:
Title: **NEST Statement of Work**
Reference: **ENVI-DTEX-EOPG-SW-06-0005**
Date: **January 2007**

[Ref 2]: Title: **Proposal for Contract Change Notice 2 (CCN 2) to Import DORIS Software into the Next ESA SAR Toolbox (NEST) Software**
Reference: **ARR-NEST-RS07-042**
Date: **June 2009**

[Ref 3]: Title: **2012 Maintenance and Support for the Next ESA SAR Toolbox (NEST) Project**
Reference: **ARR-NEST-RS07-056**
Date: **October 2012**

[Ref 4]: Title: **BEAM Architecture Document**
Reference: **Norman Fomferra, Brockmann Consult**
Date: **May 2006**

[Ref 5]: Title: **Software Architecture Document (SAD) for the Next ESA SAR Toolbox (NEST)**
Reference: **NEST project website**
Date: **December 2012**

2.2 Other documents

[Ref 6]: Title: **Delft Object-oriented Radar Software (DORIS) Manual**
Reference: **DORIS project website**
Date: **March 2009**

[Ref 7]: Title: **Radar Inter, Data Interpretation and Error Analysis**
Reference: **Kluwer Academic Publishers**
Date: **December 2000**

3 Design Method and Development Environment

Design methodology and development environment for development and deployment of interferometric modules of NEST is the same as for development of non-interferometric modules.

The main principles are listed here, while more details can be found in the Software Architecture Document (SAD) for the Next ESA SAR Toolbox (NEST), [Ref 4 and 5].

The main components of **Design Methodology** are:

- Object-Oriented Design,
- Test-Driven Development.

The main points regarding **Development Environment** are:

- Java,
- Maven Build Tool,
- Integrated Development Environment,
- Performance Profiling,
- Version Control,
- Continuous Integration,
- Coding Standards,
- Design Documentation,
- Bug Tracking.

4 Architecture of Interferometric modules of NEST

4.1 Architecture Overview

The architecture of interferometric modules of NEST is visualized in Figure 1. A dedicated library and application interface (API) for handling and processing of SAR data for interferometric application is designed and developed. The functionality core library is independent from NEST/BEAM core. The NEST/BEAM libraries are only used in the construction of the operators and in forming of interferometric graph.

Specifically, the interferometric functionality of NEST is structured in two layers. A low-layer API/library that implements all interferometric functions, and another higher level layer that exposes this interferometric functionality to a user through a NEST/BEAM Graph Processing Framework (GPF) and operators, Figure 1 and 2.

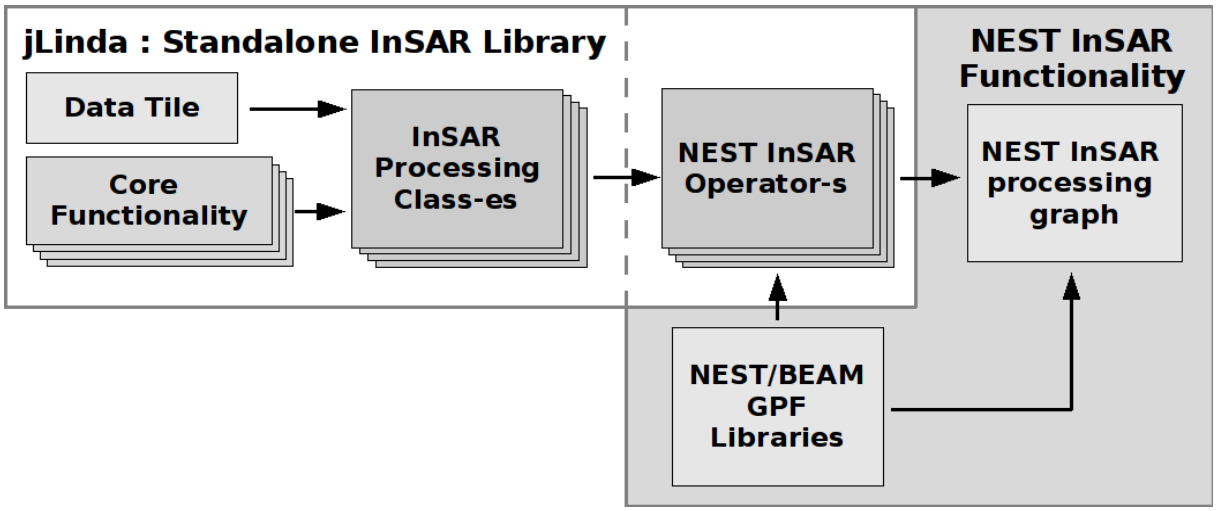

Figure 1: Flowchart of implementation strategy of interferometric modules of NEST.

Further on, this functionality is implemented in such a way that first the core classes with “processing per-tile” support are developed, and later this functionality is exposed to the user in the interferometric operators and graphs.

4.2 Library for Interferometric Data Processing

The core processing element of this interferometric library is a SAR data tile. All the algorithms are implemented to efficiently handle and do InSAR processing on the tile level. Then this per-tile processing that is structured in library, is “glued” together and exposed to the user within processing operator and GPF processing setup.

This InSAR library, because of its modular and hierarchical architecture, can be easily re-used in other software packages. Only the processing engine, that tiles the input data and “feed” the patches of data to the InSAR processing classes should be replaced. This feeding of data to the InSAR classes, is at the moment realized by NEST via the processing operators.

Finally, because of its architecture, the InSAR library can be straightforwardly extended without any interference nor strict dependency on the NEST / BEAM libraries.

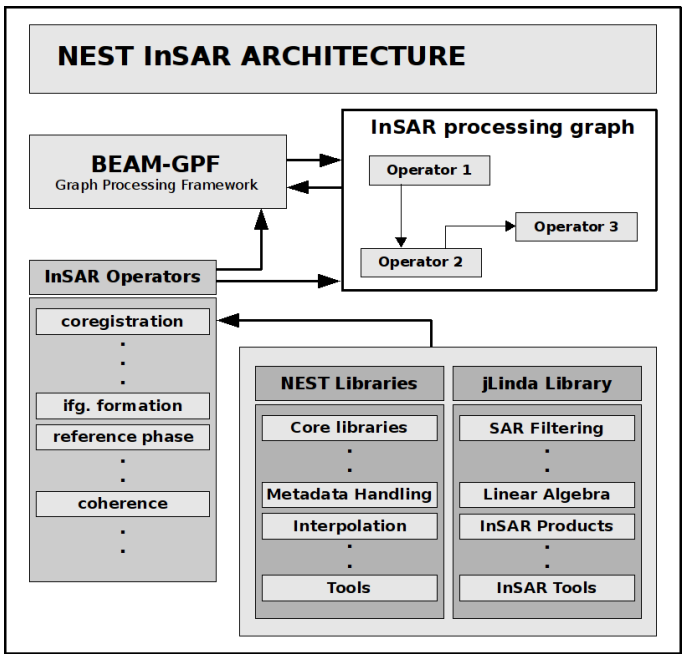

Figure 2: Diagram visualizing implementation strategy and dependency between standalone InSAR library, NEST/BEAM modules for the development of interferometric modules of NEST

- Additional library and modules are developed to support interferometric development,
- Both InSAR and BEAM/NEST libraries are used in “constructing” operators that will be linked within a Graph Processing Framework (GPF) forming an InSAR processing chain.

4.2.1 Key Features of Interferometric Processing Library

The key components and features of this newly developed InSAR library are:

- **Object-oriented and modular**
  Through hierarchical decomposition of the interferometric processing chain, and well structured object-oriented design the complexity of the processing chain is dramatically reduced, and decomposed into workable and testable smaller parts of the processing chain.
- **Encapsulated Data and Meta-data model**
  The library has its own data and meta-data model that is optimized for processing of SAR and InSAR data. Considering that JAVA does not provide a native support for complex number, becomes significant, because it provides a systematic way process data of complex types.
- **Algorithm robustness and algorithmic independence**
  All implemented functionality follow best algorithmic practices, and build on the previously documented/published/public algorithms. Moreover, algorithms are implemented as to minimize on the number of external dependencies.
- **Flexible, extensible, maintainable**
  Any of the core classes of this library can be easily extended, or even fully replaced, with a minimal impact on the functionality and how it is exposed to the users. In other words, all the development is happening at the library/API level, operators are just gluing this processing per tile and exposing the functionality to the user.
- **Tested and documented**
  For any of the core processing steps of the interferometric chain (see Figure 3), there is an extensive coverage of tests and documentation.
- **Portable**
  Because of its architecture and fully encapsulated data model, and algorithmic in-dependability on other libraries the library can be easily used within other processing frameworks.
- **Multi-core and thread-safe**
  All core functionality is multi-core and thread-safe.

A complete processing chain for Interferometry, including integrated phase unwrapping modules, visualized in Figure 3, is developed and deployed in NEST version 5A.

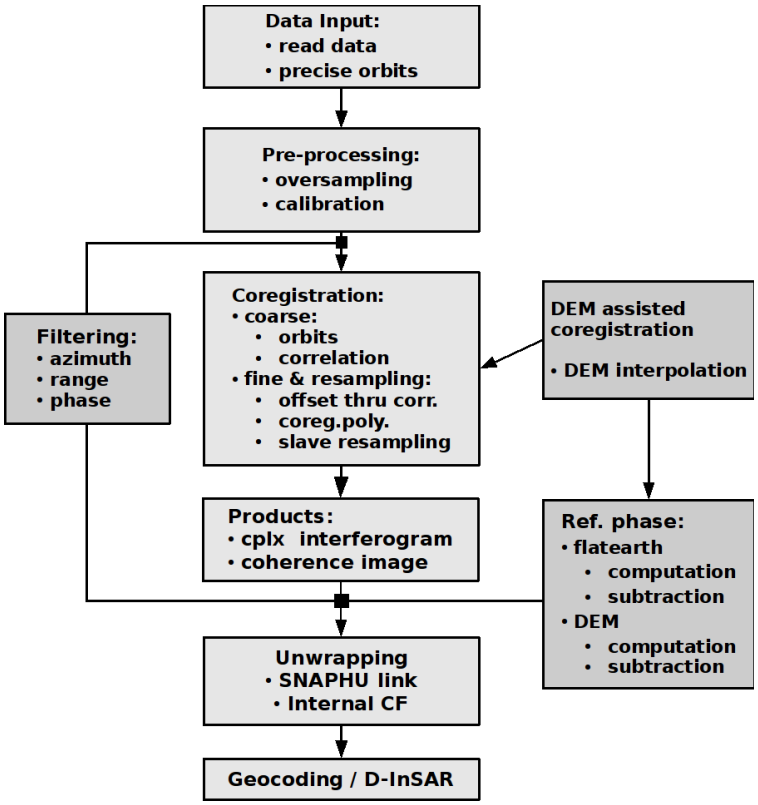

Figure 3: Full interferometric processing chain, [Ref 7]. The flowchart of the processing chain as implemented in NEST version 5A, including fully integrated Java implementation of the phase unwrapping.

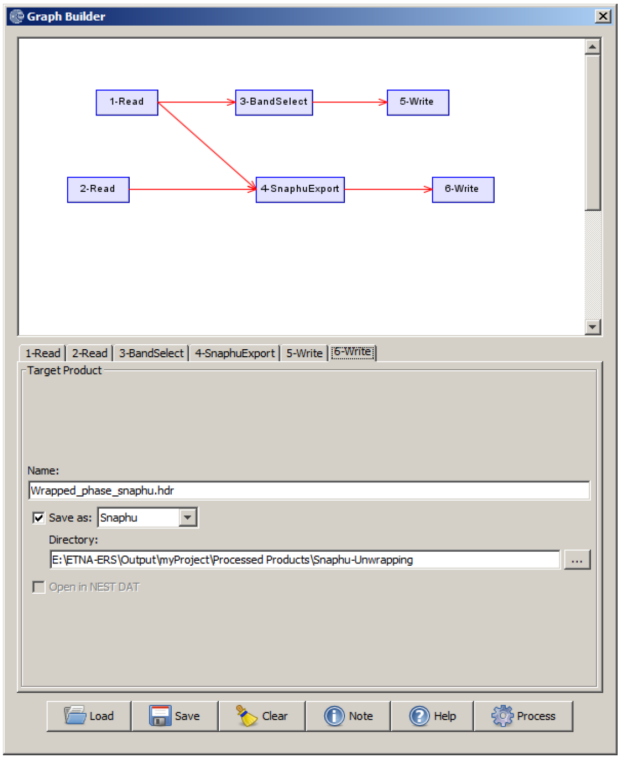

Figure 4: Example of inteferometric processing flowchart as implemented in NEST, in reference to Figure 1, 2, 3.

5 Organization of Interferometric Data Processing Library

Note: This technical document covers the library and API developed for NEST version 5A-beta. This overview exposes the minimum of the necessary information for understanding of the design and implementation principles. The expanded and detail version of library organization with API details is available via the project web-page.

5.1 Major Packages and Classes

As of NEST 5A version, the complete interferometric chain, visualized in Figure 3, is supported.

All developed interferometric functionality is organized in classes. These classes are then, depending on their application, organized and grouped in packages and sub-packages.

5.1.1 List and brief description of packages

The main packages are listed in Table 1.

Package Name	Description
jlinda-core	Encapsulates all functionalities and algorithms for interfereometric processing. Functionality implemented in this package operates on tiles of SAR data.
jlinda-nest	Consists of operators that are built on top of core classes from jlinda-package, together with NEST/BEAM libraries. Operators in this package expose the inteferometric functionality to users.

Table 1. List and brief description of InSAR library packages

The core package, jlinda-core, is further divided into sub-packages, each encapsulating specific processing and algorithmic functionality. Tables 2a and 2b summarize this organization.

(Sub) Package Name	Description
core	Package that contains all core classes.
coregistration	Classes for coregistration based on correlation optimization, and DEM assisted coregistration.
coregistration /estimation	Sub-sub-package with procedures for least-squares estimation, adjustment, and hypothesis testing. The initial implementation is optimized for the estimation of the coregistration polynomial, but is generic enough for any other type of the estimation problem.
delaunay	Delaunay triangulation interpolation.
filtering	SAR (azimuth, range), and InSAR phase filtering (Goldstein).
geocode	Phase to Height conversion, differential interferometry, and other InSAR geometry operations.
geom	Computation of reference and topographic phase, together with tools for processing of DEM tiles.

Table 2b. List and brief description of sub-packages of InSAR library core package.

(Sub) Package Name	Description
utils	Utility methods for mathematical, spectral, and SAR operations.
io	Input / Output tools
simulation	Class for interfereometric data simulation, used only for testing.
stacks	Functions for stack processing – functionality for an optimal master selection, theoretical coherence modeling.
unwrapping	Unwrapping methods, with link to SNAPHU unwrapping tool, and newly developed unwrapping functionality that is based on MCF approach.

Table 2b. List and brief description of sub-packages of InSAR library core package.

5.1.2 List and brief description of core-classes

Core classes of InSAR developed library are listed in Table 3.

Class Name	Description
SLCImage	MetaData class - Container for InSAR processing relevant metadata, and all operations on the metadata (e.g., doppler computation, etc).
Orbit	MetaData class - Encapsulates state vectors and all the numerical operations on them, eg, interpolation and propagation of state vectors.
Baseline	Functionality for baseline estimation for interferometric pairs.
Window	Defines ranges of data.
Point	Class for pixel, 2D, or 3D types of coordinates with set operations on coordinates.
Ellipsoid	Container for definitions of ellipsoids.
Constants	Aggregates all interferometry relevant constants.

Table 3. List and brief description of core classes of InSAR library

6 List of Acronyms and Abbreviations

Acronyms and Abbreviations	Explanation
2-D	Two-dimensional
3-D	Three-dimensional
ALOS PALSAR	Advanced Land Observing Satellite Phased-Array-type L-band Synthetic Aperture Radar
API	Application Interface
Array	Array Systems Computing Inc.
ASAR	Advanced Synthetic Aperture Radar
EO	Earth Observation
BEAM	Basic ERS & Envisat (A)ATSR and MERIS
CEOS	Committee on Earth Observation Sat
DAT	Display and Analysis Tool
DEM	Digital Elevation Model
ESA	European Space Agency
ERS	European Remote Sensing Satellite
ENVISAT	ENVIronmental Satellite
ESRIN	European Space Research Institute
GNU	GNU's Not Unix (recursive acronym)
GPF	Graph Processing Framework
GPT	Graph Processing Tool
GUI	Graphical User Interface
InSAR	Synthetic Aperture Radar Interferometry
IO	Input/Output
IDE	Integrated Development Environment
JDK	Java Development Kit
JERS	Japanese Earth Resources Satellite
NEST	Next ESA SAR Toolbox
OO	Object
SAR	Synthetic Aperture Radar
MCF	Minimum Cost Flow