

# サブミリ波単一鏡分光装置における解析ソフトウェア開発： 大学主導計画の視点から

○谷口暁星（名古屋大学）、DESHIMA collaboration、sdarray developers

## 概要（Abstract）

地上のサブミリ波分光観測において、装置の応答関数較正および大気雑音除去を担う解析ソフトウェアは、サイエンスの達成に必要な不可欠な開発である。本講演では、広帯域化、広視野化が進む開発において重要かつ複雑さが増すソフトウェアの技術と課題を、大学主導計画の視点から議論する。特に、人員が流動的な大学間連携の開発における、パッケージ共通化、バージョン管理、継続的インテグレーションの導入事例と現状を紹介する。

### 1. はじめに

地上のミリ波サブミリ波単一電波望遠鏡（単一鏡）は、大口径を生かした高感度分光観測や電波カメラによる広域マッピング観測など、ALMAをはじめとする干渉計観測と相補的な役割を果たす観測装置である。また、国内外の単一鏡は、新しい受信機や分光計のコミッションングを受け入れており、装置開発の面でも電波天文学コミュニティに重要な役割を果たしている。我々は、DESHIMA（Deep Spectroscopic High-redshift Mapper） [1, 2] と呼ばれる、超伝導検出器MKIDとフィルターバンク技術によりサブミリ波帯で数百GHzに及ぶ同時分光を実現する分光観測装置を開発している。プロトタイプDESHIMAはASTE望遠鏡に2017年に試験搭載され、系内天体や系外銀河の分子輝線の検出に成功した。DESHIMAは検出素子や冷却光学系などのハードウェア開発をオランダの研究者が、ソフトウェア開発や観測データの解析・評価を日本の研究者がリードする。それぞれの国内では複数の大学や研究機関が連携する、大学主導かつ大学間連携のプロジェクトであると言える（図1）。このようなプロジェクトでは、（1）大学間でバージョンと品質が管理されたソフトウェア開発の仕組みと、（2）大学間で共通の環境で行ったデータ解析とその結果の共有が重要な課題となる。以下では、DESHIMAのソフトウェア開発における各種ツール群の導入による課題解決の現状と、今後の課題について紹介する。

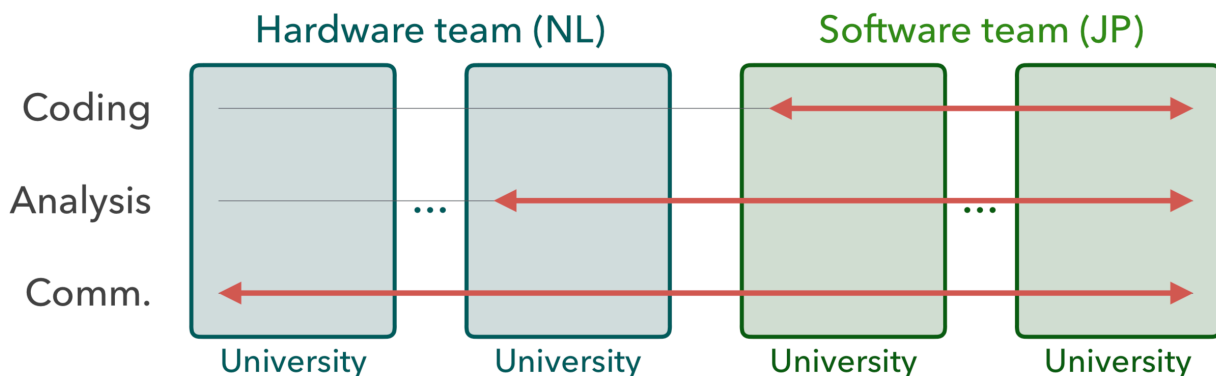


図1：DESHIMAにおける大学・研究機関連携のモデル。図中に示された矢印は、ソフトウェア開発（Coding）・データ解析（Analysis）・コミュニケーション（Comm.）の各項目でどの範囲の連携が必要なのかを表している。

## 2. 大学間連携のソフトウェア開発

サブミリ波の新たな装置開発では、装置専用のデータ解析環境やソフトウェアを開発することが多い。例えば、DESHIMAプロジェクトでは、De:code [3] と呼ばれるPythonで書かれたデータ解析パッケージを開発・提供している。大学間でソフトウェア開発を円滑に行うために、我々は以下の2つの観点からツールを導入している。なお、以下に挙げるソフトウェア開発のデモンストレーションは著者のGitHubリポジトリ [4] で公開されているので参考にされたい。

### ・ソフトウェアのバージョン管理

データ解析パッケージは、データのI/O・応答関数較正・大気雑音除去などの複数の機能を担う。このため、Pythonのようなスクリプト言語の開発でも合計数千〜数万行のプログラムを管理することが必要である。我々は、分散バージョン管理システムのGit [5] を使用して、大学間に分散した開発に置いてコードの変更履歴を完全に把握できる仕組みを導入している。また、Gitのリポジトリ（データベースに相当）は、ホスティングサービスのGitHub [6] を利用してオープンソースかつクラウド上で管理している。これにより、ユーザは解析パッケージを自由にダウンロードし、機能追加やバグ修正を提案できる。

### ・ソフトウェアの品質管理

ソフトウェア開発では、プログラムに小さな更新（機能追加・バグ修正）を加える度に、テストプログラムを走らせて更新後のプログラムに問題ないかを確認し、必要であればビルドまで行う仕組みを導入することが多い。これを継続的インテグレーション（Continuous Integration; CI）と呼ぶ。GitHubを使った開発ではCIをクラウド上で自動的に行うためのCIサービス（例えばGitHub Actions [7]）が提供されている。これを使用することで、ソフトウェアのバージョン管理をしつつ、バグを含んだ変更はそもそもリポジトリに反映されないため、品質管理も達成することができる（図2）。

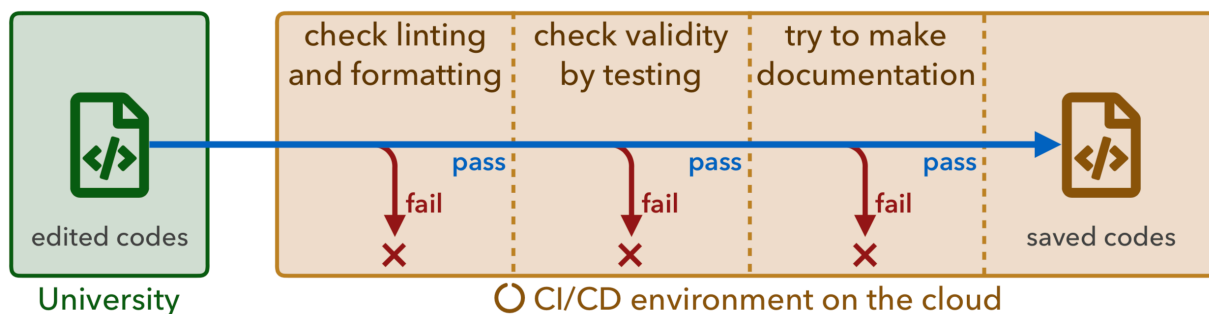


図2：GitHubとCIサービスを利用したソフトウェアのバージョン・品質管理の例。大学で機能追加やバグ修正などの更新が行われたプログラム（edited codes）を、リポジトリに反映させるための要求（プルリクエスト）を行ったとする。この際、プログラムの文法チェック・テストプログラムの実行・ドキュメンテーションの作成がCIサービスによって自動的に実行される。全てのステップでエラーが発生しなかった場合のみ、更新がリポジトリに反映される（saved codes）。

### 3. 大学間連携のデータ解析

2で開発したソフトウェアなどを使ったデータ解析では、大学間の複数のコンピュータ上で共有の解析環境を提供し、解析結果の再現性を保証する必要がある。また、解析結果は見やすい形でチーム内で共有できることが望ましい。このような観点から、我々は以下に挙げた共有の仕組みを導入している。

#### ・Python解析環境の共有

Pythonをはじめとするスクリプト言語では、言語のバージョン、およびソフトウェアとこれが依存するライブラリのバージョンを全て指定した環境を構築する（決定論的ビルド）ことが重要である。Pythonでは、このような環境構築を目的としたツール（例えばPipenv [8] やPoetry [9] ）が近年登場し、解析環境の共有を手軽に行えるようになった。例えば、De:codeの場合はPipenvを使用している。ユーザはリポジトリをダウンロード（クローン）した後、コマンド一つ（pipenv install）のみで即座にPython環境構築を完了させることが可能である。

#### ・データ解析結果の共有

スクリプト言語を用いたデータ解析では、Project Jupyter [10] によって解析プログラム・コメント・解析結果の画像をノートブックと呼ばれる単一のファイルに保存し、ウェブブラウザ上で解析・閲覧できるようになった。また、解析結果や議論の共有には、従来型のWikiに加えてオンラインサービス（例えばKibela [11] やesa [12] ）が登場しており、様々な場面で使われている簡便なMarkdown記法を使った情報共有が、サーバの管理を行うことなく可能になっている（図3）。例えば、DESHIMAではJupyter notebookを使ってデータ解析を行い、これをKibelaにアップロードすることで、チームメンバであれば誰でも解析結果を閲覧し、必要であれば自身の環境で解析結果を再現することも可能である。

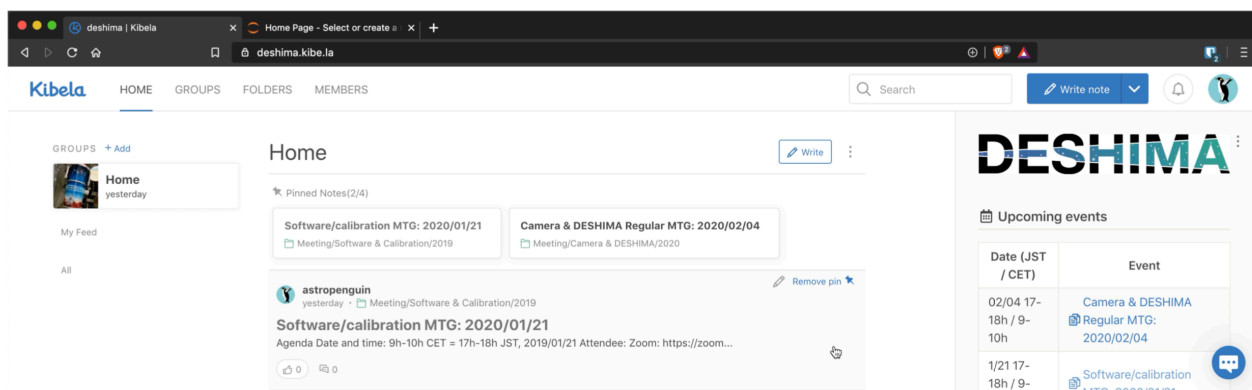


図3：DESHIMAプロジェクトで使われている情報共有ツールKibela。ユーザはアカウントでログインすることでページの閲覧・編集が可能である。なお、教育等の非営利目的では無料で使用できるサービスもあるため（Kibelaとesaはこれに該当）、導入の敷居は比較的低い。

### 4. 今後の課題と解決策

2と3では大学間連携プロジェクトにおけるソフトウェア開発と各種ツール群の導入について述べた。これらは、単一プロジェクトにおける円滑なソフトウェア開発とデータ解析に今後も大きな役割を果たすと考えられる。一方、日本国内における単一鏡装置開発では、複数のプロジェクトで同様なデータ解

析が行われているにも関わらず、ソフトウェア開発が独立に行われている現状がある。また、一人の研究者が複数のプロジェクトに参加している場合も多い。ソフトウェア開発のマンパワーを考えると、開発を加速するためにソフトウェアの共通化に今後目を向ける必要がある。我々は、このような観点から単一鏡のデータのフォーマットを共通化するためのPythonパッケージsdarray [13] の開発を進めている。sdarrayでは単一鏡の観測データが時間×チャンネルの2次元で表されるとし、これにチャンネルの値やアンテナ座標などのメタデータを付与したオブジェクトをPythonパッケージで扱う(図4)。DESHIMAの次期コミッショニングに対応することを目標に、フォーマットの策定が今年度中に完了する予定である。

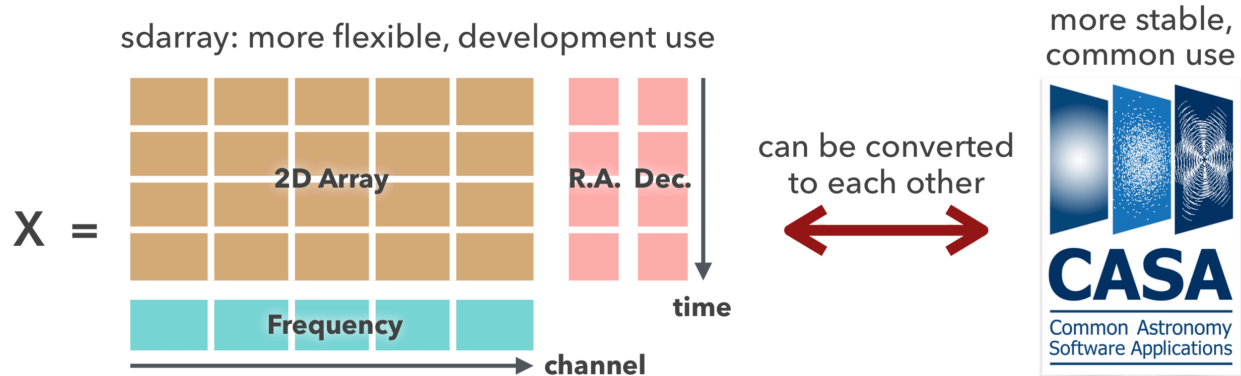


図4：単一鏡共通データフォーマットsdarrayの概念図。観測データは分光観測、連続波カメラ観測などに関わらず、時間×チャンネルの2次元配列として扱われる。また、電波天文学においてはALMAのデータ解析ソフトウェアCASA [14] がデファクトスタンダードとなりつつある。sdarrayは、実際の装置開発現場における柔軟かつ即応的な解析のための解決策として、CASAと相補的な立場を目指す。

## 5. 共著者リスト

DESHIMA - Y. Tamura, K. Suzuki (Nagoya-U), T. Takekoshi (UTokyo), A. Endo, (TU Delft), K. Karatsu (SRON), T. Tsukagoshi (NAOJ), and the DESHIMA collaboration

sdarray - Y. Tamura, K. Tachihara, K. Suzuki, K. Shiotani (Nagoya-U), T. Takekoshi, T. Ishida, Y. Yoshimura (UTokyo), T. Nitta, P. Mandal, Y. Murayama (Tsukuba-U), T. Oshima, M. Nagai, R. Kawabe (NAOJ)

## 6. 参考文献

- [1] [Endo et al. 2019a, JATIS, 5, 3, 035004](#)
- [2] [Endo et al. 2019b, NatAstron, 3, 989](#)
- [3] [De:code - DESHIMA code for data analysis](#)
- [4] [astropenguin/demo-python-package](#)
- [5] [Git - a free and open source distributed version control system](#)
- [6] [GitHub](#)
- [7] [GitHub Actions - automate your workflow from idea to production](#)
- [8] [Pipenv - Python Development Workflow for Humans](#)
- [9] [Poetry - Python packaging and dependency management made easy](#)
- [10] [Project Jupyter](#)
- [11] [Kibela - a new communication tool to fuel the team](#)
- [12] [esa - the document sharing service for motivated teams](#)
- [13] [sdarray - Python package for common single-dish data](#)
- [14] [CASA - the common astronomy software applications package](#)