Ideas for data management and data handling concept for COMPACT

Daniel P. Mohr[1,2], Christina A. Knapek[1], André Melzer[1]

[1] Institute of Physics, University of Greifswald, Greifswald, Germany
[2] Institute of Atmospheric Physics, DLR German Aerospace Center, Oberpfaffenhofen, Germany

Abstract

The future complex plasma facility COMPACT [1] will allow the investigation of large three-dimensional complex plasmas under microgravity conditions aboard the International Space Station (ISS). COMPACT is a project with international scientific contributions, supported by space agencies (DLR, NASA, ESA) and NSF.
Data generated by experiments on the ISS have a considerable value considering the effort needed to repeat an experiment. To maximize the use of the unique data in the scientific community, data management and data handling must be designed sensibly. We have learned from previous projects that it makes sense to deal with this at an early stage. Ultimately, the data must be handled on many levels.
On the one hand, the data must be stored in a trustworthy manner with sufficient metadata during the experiment. On the other hand, the FAIR principles [2, 3] should be followed as early as possible. This creates confidence in the scientific results.
The data should already be processed as reliably as possible in an early stage: during the experiment. This includes protecting both the volatile memory and the non-volatile storage. For example, the usage of ECC main memory, and zfs with checksums are suitable for this. This is all the more important as silent data corruption certainly occurs more frequently under the radiation conditions on the ISS and the large data volumes of several TBs per experiment day.
The chain of trust should then be continued and made available to the scientific community in a suitable infrastructure. The basic concept of RIAF [4, 5] could be applied here.
Finally, the research data on which the papers are based should also be published in data repositories (e. g. [6]) – tools such as deploy2zenodo [7] could be used for this purpose.

Data Handling and Storage during Experiment

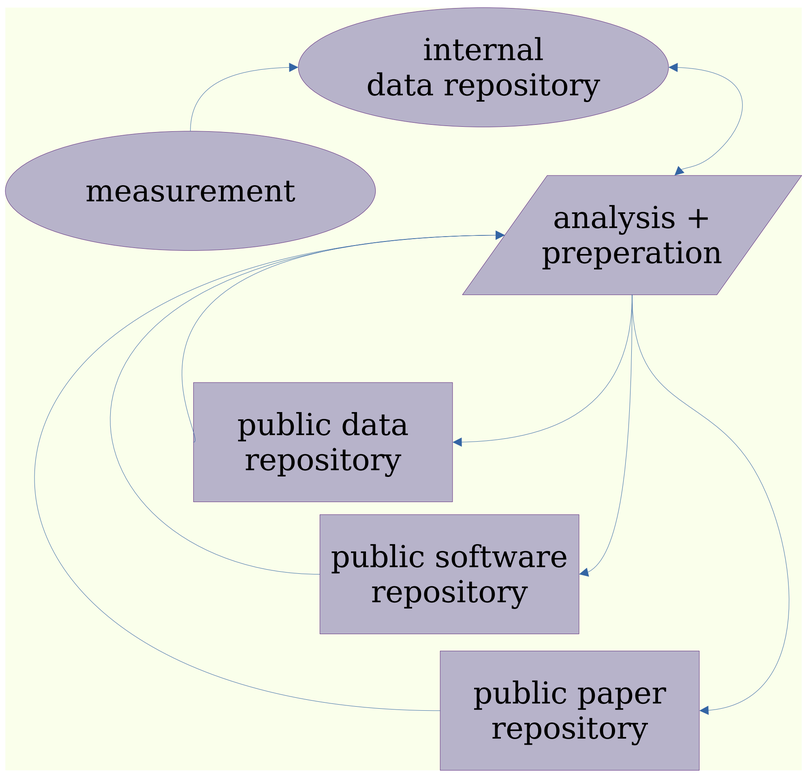
In order to obtain reliable measurement data from experiments, some requirements must be met:
- coordinated, synchronized data acquisition
- store measurement data with necessary/associated metadata
- uniform time for the time stamps of the measurement data (e. g. real time UTC)
- protect the data integrity

Solution: Use adequate hard- and software, e. g.:
- common trigger signals
- storing metadata during genesis of data

The biggest challenge: protection of data integrity.
- secure communications
- failure detection in RAM
- suitable file system for non-volatile memory

Possible solutions are:
- TCP/IP communication
- ECC memory
- zfs file system:
  - ACID transactions
  - Merkle tree: check-summing the complete storage tree (verifying data integrity)
  - redundant storage ← space station?!

⇒ redundancy has to be established as soon as possible on ground (error correction)



Repository Infrastructure for Data

As a first step, it is good to obtain the research data from the experiments carried out. To analyze and share this data in the scientific community that participates in COMPACT or the individual experiments, an infrastructure is necessary.

Requirements for this infrastructure are:
- storage of large amounts of data
- verifiable verification of data integrity
- user management (scientists are spread all over the world)
- easy access for users
- practical access to the data for software during data analysis

⇒ basic concept of RIAF [4, 5], including decentralized version control (lesson learned from software development over decades)

RIAF is a repository infrastructure to accommodate files.
- FAIR principles (reproducibility of the data)
- applied in early part of the data life cycle
- enable checks on metadata, e. g. maDMP (machine actionable data management plan)
- cryptographic timestamping
- creation of public landing page from metadata

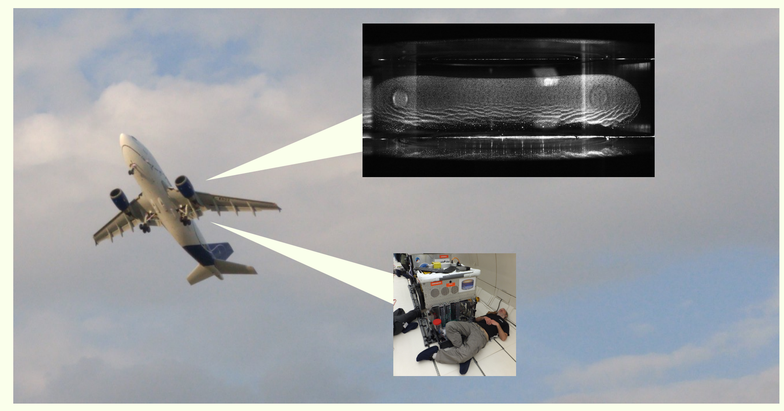
In this concept of RIAF most data is stored in a repository and can be easily distributed. This allows the data genesis in a private environment (e. g. aircraft, ISS, ...) without network access and later share the data using a central server instance. Already during data genesis (e. g. raw data, physical data, scientific data) the possibility to share data and track changes is given. In the end after preparing a publication the data can be transported to a public data repository.

Collaboration platforms such as Gitea [12], GitLab [11] and GitHub [13] have emerged, particularly in open source software development. These platforms enable other desired features:
- fine grade access control
- collaboration
- issue tracking
- automated processes
- project management
- structured and traceable processing
- labeling and/or release management

⇒ RIAF and other projects (e. g. DataPLANT [14]) already use GitLab together with Git LFS [15] to use the knowledge from software engineering to store and work with data.

Experience in Data Handling and Storage

We have already gained experience in data handling and storage of experiment data during the COMPACT project and previous projects.
- achieve consistency in time by PTP (redundant)
- experiment software stores data + metadata:
  - camera images
  - camera time, system time
  - camera serial number
  - exposure time
  - …
- using zfs file-system for almost 10 years
  - snapshots (delete protection, replicate data)
    - parabolic flight → backup → permanent storage
    - incl. checksum, permissions, access time, …



Data Publication

To satisfy the FAIR principles [2, 3], publications should be deployed to an open repository. In this way the publication gets a PID [8] and at least the metadata is publicly accessible, findable and citable. Furthermore, current discussions about KPIs [9] for software and data publications also lead to the need to generate PIDs for software and data. Scientific data comprises:
- measurements
- software
- results such as papers

For every data managed in a version control system an automatic publication to an open repository is useful [3].
⇒ citability of individual versions

deploy2zenodo [7, 10] is a shell script to deploy data to the open repository zenodo:
- flexible by environmental variables
- simple integration into various environments
- e. g. in GitLab [11] CI pipeline
  - ➔ curation by merge request
  - ➔ curation in zenodo web interface

References

[1] C. A. Knapek *et al.*, DOI: 10.1088/1361-6587/ac9ff0
[2] M. Wilkinson *et al.*, DOI: 10.1038/sdata.2016.18
[3] HMC et al., DOI: 10.3289/HMC_publ_01
[4] D. Mohr, DOI: 10.5281/zenodo.7189120
[5] riaf-data.org
[6] zenodo, DOI: 10.25495/7gxk-rd71
[7] D. Mohr (2024), DOI: 10.5281/zenodo.10112959
[8] en.wikipedia.org/wiki/Persistent_identifier
[9] en.wikipedia.org/wiki/Performance_indicator
[10] D. Mohr (2023), DOI: 10.5281/zenodo.10137956
[11] GITLAB is a trademark of GitLab Inc. in the United States and other countries and region
[12] about.gitea.com
[13] github.com
[14] www.nfdi4plants.org
[15] git-lfs.com

DOI: 10.5281/zenodo.11094321

Corresponding Authors:
daniel.mohr@uni-greifswald.de
christina.knapek@physik.uni-greifswald.de

Work funded by DLR/BMWi (FKZ 50WM2161)

Ideas for data management and data handling concept for COMPACT

Deutsches Zentrum für Luft- und Raumfahrt
German Aerospace Center

Daniel P. Mohr[1,2], Christina A. Knapek[1], André Melzer[1]

UNIVERSITÄT GREIFSWALD
Wissen lockt. Seit 1456

[1] Institute of Physics, University of Greifswald, Greifswald, Germany
[2] Institute of Atmospheric Physics, DLR German Aerospace Center, Oberpfaffenhofen, Germany

Abstract

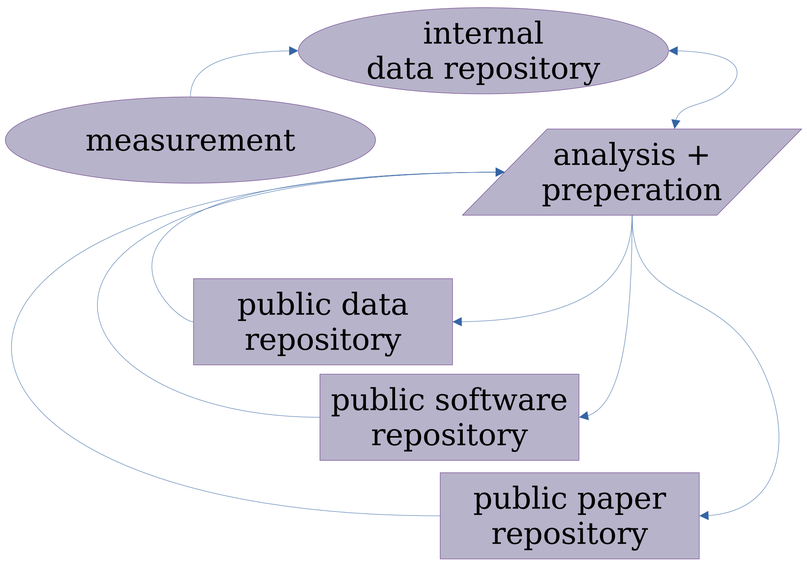
The future complex plasma facility COMPACT [1] will allow the investigation of large three-dimensional complex plasmas under microgravity conditions aboard the International Space Station (ISS). COMPACT is a project with international scientific contributions, supported by space agencies (DLR, NASA, ESA) and NSF.
Data generated by experiments on the ISS have a considerable value considering the effort needed to repeat an experiment. To maximize the use of the unique data in the scientific community, data management and data handling must be designed sensibly. We have learned from previous projects that it makes sense to deal with this at an early stage. Ultimately, the data must be handled on many levels.
On the one hand, the data must be stored in a trustworthy manner with sufficient metadata during the experiment. On the other hand, the FAIR principles [2, 3] should be followed as early as possible. This creates confidence in the scientific results.
The data should already be processed as reliably as possible in an early stage: during the experiment. This includes protecting both the volatile memory and the non-volatile storage. For example, the usage of ECC main memory, and zfs with checksums are suitable for this. This is all the more important as silent data corruption certainly occurs more frequently under the radiation conditions on the ISS and the large data volumes of several TBs per experiment day.
The chain of trust should then be continued and made available to the scientific community in a suitable infrastructure. The basic concept of RIAF [4, 5] could be applied here.
Finally, the research data on which the papers are based should also be published in data repositories (e. g. [6]) – tools such as deploy2zenodo [7] could be used for this purpose.

Data Handling and Storage during Experiment

In order to obtain reliable measurement data from the experiments carried out, a number of requirements must be met:
- coordinated and synchronized data acquisition
- store measurement data with necessary/associated metadata
- uniform time for the time stamps of the measurement data (e. g. real time UTC)
- protect the data integrity

All these points are not only an organizational requirement for the recording systems but also for the hardware used. Coordinated data acquisition could be achieved by common trigger signals. The metadata is already available when the data is created using suitable software.
The biggest challenge is the protection of data integrity. Every communication should be secured. Also the computer memory should at least be able to detect failures. A suitable file system must be selected for non-volatile memory.
Possible solutions are:
- TCP/IP communication
- ECC memory
- zfs file system:
  - ACID transactions
  - check-summing the complete storage tree (Merkle tree)
  - redundant storage

For example redundant storage maybe hard to achieve on the ISS. But then the data should be replicated to a redundant storage as soon as possible on ground. Checking the data integrity by verifying the check-sums of the file-system allows still to detect corrupted data. As soon as the data is stored redundantly, errors can also be corrected.



Repository Infrastructure for Data

As a first step, it is good to obtain the research data from the experiments carried out. But to analyze and share this data in the scientific group that participated in COMPACT or the individual experiments, an infrastructure is necessary.
Requirements for this infrastructure are, for example:
- storage of large amounts of data
- verifiable verification of data integrity
- user management (scientists are spread all over the world)
- easy access for users
- practical access to the data for software during data analysis

The basic concept of RIAF [4, 5] provides exactly these requirements. In addition, lessons were learned from software development. Decentralized version control (e. g. git) has developed over decades in software development.
RIAF is a repository infrastructure to accommodate files. It enables to hold the data with the FAIR principles.
RIAF is designed to enable provenance and reproducibility of the research data in the early part of the data life cycle, i. e. prior to publication. It further is designed to enable checks on metadata relevant to research data management as defined e. g. in a machine actionable data management plan (maDMP). This concept of using CI pipelines for research data allows interesting features. The server could create cryptographic timestamps to inhibit silent changes of the history. Research data management can define relevant checks on metadata. From given metadata a public accessible landing page can be created.
In this concept of RIAF most data is stored in a repository and can be easily distributed. This allows the data genesis in a private environment (e. g. aircraft, ISS, ...) without network access and later share the data using a central server instance. Already during data genesis (e. g. raw data, physical data, scientific data) the possibility to share data and track changes is given. In the end after preparing a publication the data can be transported to a public data repository.
If we look at software development again. Collaboration platforms such as Gitea [12], GitLab [11] and GitHub [13] have emerged, particularly in open source software development. These platforms enable other desired features:
- fine grade access control
- collaboration
- issue tracking
- automated processes
- project management
- structured and traceable processing
- labeling and/or release management

RIAF and other projects (e. g. DataPLANT [14]) already use GitLab together with Git LFS [15] to use the knowledge from software engineering to store and work with data.

Experience in Data Handling and Storage

We have already gained experience in data handling and storage of experiment data during the COMPACT project and previous projects.
To achieve consistency in time we use PTP to synchronize all systems to one clock. PTP allows a redundant setup to overcome a not working system.
In our experiment software we already store metadata together with the measurement data, e. g. for camera images:
- camera time, system time
- camera serial number
- exposure time
- ...

We have been using zfs file-system for almost 10 years. Using snapshots, it not only provides protection against accidental deletion but also enables transport to other systems together with the metadata and checksums from the zfs dataset. In this way, data from parabolic flight campaigns is temporarily stored on backup hard disks and ultimately kept permanently in a storage system. The data transport is completely lossless and contains all data together with the metadata held by the file system (e. g. checksum, permissions, access time, ...)

Data Publication

To satisfy the FAIR principles [2, 3], publications should be deployed to an open repository. In this way the publication gets a PID [8] and at least the metadata is publicly accessible, findable and citable. Furthermore, current discussions about KPIs [9] for software and data publications also lead to the need to generate PIDs for software and data.
In principal the same is true for all kind of scientific data (e. g. measurements, software and results such as papers). For every data managed in a version control system an automatic publication to an open repository is useful [3].
Software in particular is subject to frequent changes, resulting in many versions. This leads to the urge to automate the publishing process. This is not only about making the software usable through software repositories, but also about the citability of individual versions.
deploy2zenodo [7, 10] is a shell script to deploy data to the open repository zenodo. It can be integrated in a GitLab [11] CI pipeline as an automatic workflow or various other environments for automatic publication.
Environmental variables allow very flexible use. Depending on the selected flags, the data can be curated before deployment in a merge request, in the zenodo web interface or not curated at all.

References

[1] C. A. Knapek *et al.*, DOI: 10.1088/1361-6587/ac9ff0
[2] M. Wilkinson *et al.*, DOI: 10.1038/sdata.2016.18
[3] HMC et al., DOI: 10.3289/HMC_publ_01
[4] D. Mohr, DOI: 10.5281/zenodo.7189120
[5] riaf-data.org
[6] zenodo, DOI: 10.25495/7gxk-rd71
[7] D. Mohr (2024), DOI: 10.5281/zenodo.10112959
[8] en.wikipedia.org/wiki/Persistent_identifier
[9] en.wikipedia.org/wiki/Performance_indicator
[10] D. Mohr (2023), DOI: 10.5281/zenodo.10137956
[11] GITLAB is a trademark of GitLab Inc. in the United States and other countries and region
[12] about.gitea.com
[13] github.com
[14] www.nfdi4plants.org
[15] git-lfs.com

DOI: 10.5281/zenodo.11094321

Corresponding Authors:
daniel.mohr@uni-greifswald.de
christina.knapek@physik.uni-greifswald.de

Work funded by DLR/BMWi (FKZ 50WM2161)