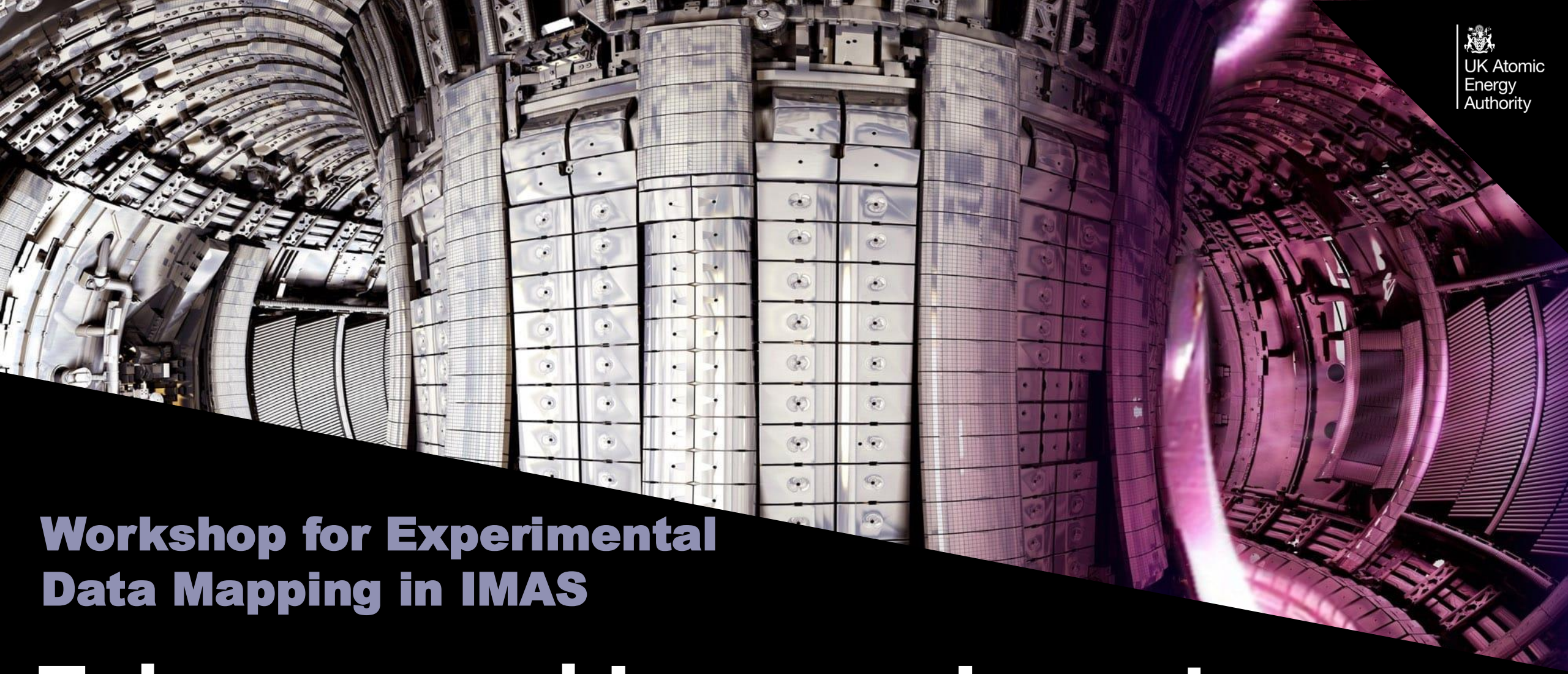


UK Atomic Energy Authority

Workshop for Experimental Data Mapping in IMAS

Tokamap and Lessons Learnt

Adam PARKER, Jonathan HOLLOCOMBE, Stephen DIXON
UK Atomic Energy Authority

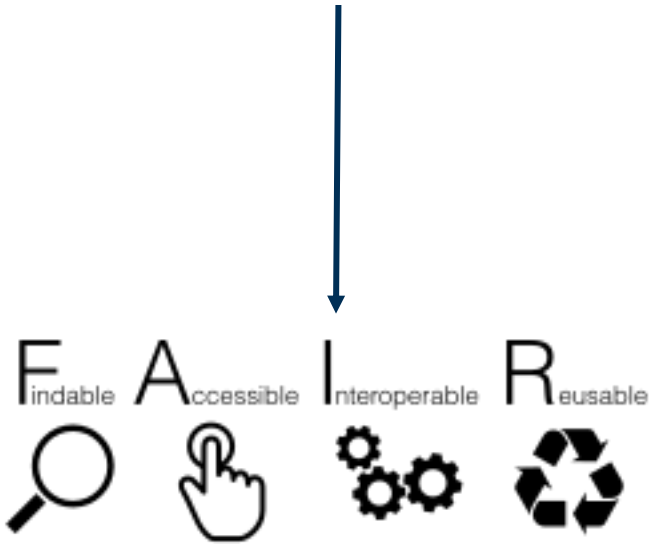
Outline

- TokaMap
- Lessons learnt and return on experience
- Presented later: LibTokaMap
  Underlying mapping library shown by J Hollocombe

Motivation

- No common representation of mappings
- Often written as bespoke mapping code (in Python, Matlab, etc.)
- These mappings cannot be easily shared or run in different environments
- Difficult to track provenance and version information

- Can we make the mappings themselves FAIR?



https://unsplash.com/photos/a-ball-filled-with-lots-of-different-types-of-beads-4KFAGIu-Abk

TokaMap

- TokaMap is a JSON schema-based framework for mapping experimental data
- Defines a common set of mapping types (90% of cases)
- Allows definition of data readers to allow it to read from custom data sources, i.e. UDA, MDS+, etc.
- Expressions: i.e. `mod(2*PI-atan2(Z,R), 2*PI)`
- Mapping files can be validated against the schemas

- Tools can be designed for using and sharing of mappings
  - We would like to build towards an online mapping portal with tools to design & share mappings

TokaMap – Structure

```
mappings/
└── experiment_name/
    ├── mappings.cfg.json     # Experiment configuration and metadata
    ├── globals.json          # Top-level globals
    └── group_name/
        └── partition_value
            ├── globals.json  # Partition globals
            └── mappings.json # Actual mappings
mappings/
└── MAST/
    ├── mappings.cfg.json
    ├── globals.json
    └── magnetics/             # mappings for the different IMAS IDSs
        ├── 0/                 # mappings for shot 0 - 27999
        │   ├── globals.json
        │   └── mappings.json
        └── 28000/             # mappings for shot 28000+
            ├── globals.json
            └── mappings.json
```


https://unsplash.com/photos/brown-wooden-door-with-silver-door-lever-iIX-jNXydeY

TokaMap – Mapping File Structure

UK Atomic Energy Authority

mappings.cfg.json – contains details about how the mappings are structured and metadata

```
{
   "metadata": {
      "experiment": "mastu",
      "author": "Adam Parker",
      "version": "0.1.0"
   },
   "partitions": [
      { "attribute": "shot", "selector": ”MAX_BELOW" }
   ],
   "groups": ["magnetics", "pf_active"]
}
```

Required metadata about the mappings

The mappings can be selected at run-time based on ‘partition’, i.e. you can partition by ‘shot number’ and then select the correct mapping

Specifying what directories exist that contain mappings

TokaMap – Mapping File Structure

UK Atomic Energy Authority

globals.json – contains configuration for the data sources and any global data that is useful in the mappings:

```
"data_source_config": {
  "UDA": {
    "args": {
      "source": "{{ shot }}",
          "host": "uda.mastu.uk",
      "port": 56565
    }
  }
},
...
```

You can specific arguments that will be passed in each time the data source is used

Arrays of objects can be used in the mappings using the templating (see later)

```
"PF_COILS": [
  {
    "COIL_NAME": "D1U",
    "TYPE": 1,
    "GEOM_NAME": "d1_upper"
  },
  ...
],
...
```

TokaMap – Mapping File Structure

UK Atomic Energy Authority

mappings.json contain the actual mappings as key-value pairs

```
"ids_properties/homogeneous_time": {
   "map_type": "VALUE",
   "value": 0
},
...
```

Every mapping has a ‘map_type’ specify what type of mapping it is:

[DIMENSION, VALUE, DATA_SOURCE, EXPR, CUSTOM]

All the other arguments are specific to the given ‘map_type’

TOKAMAP

TokaMap – Map Types

UK Atomic Energy Authority

Value Mapping

```
"key/path": 42
...
```

```
"key/path": {
    "map_type": ”VALUE",
    "value": 42
}
```

Static value defined in the mapping:

- String, i.e. ”F_BL_02"
- Number, i.e. 42, 3.14
- Boolean, i.e. true
- Array, i.e. [1, 2, 3, 4, 5]

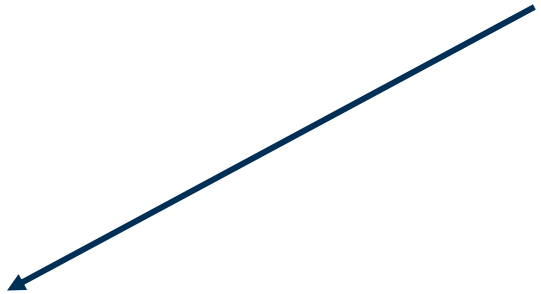
TokaMap – Map Types

Dimension Mapping

```
"key/path": {
   "map_type": "DIMENSION",
   "dim_probe": "myarray/path"
},
...
"myarray/path": [0.0, 1.0, 2.0]
```

Extract dimensionality of this mapping

i.e. dim([0.0, 1.0, 2.0]) = 3



TODO: Add "DIM_IDX" for selecting which dimensional indices to return (currently always returns first index)

TokaMap – Map Types

Data Source Mapping

```
"key/path": {
   "map_type": "DATA_SOURCE",
   "data_source": "source_name",
   "args": {
      "arg1": "value1",
      "arg2": true
   },
   "offset": 0,
   "scale": 1.0,
   "slice": "[0:10]"
}
```

Name of the data source, i.e. NetCDF, MDS+, UDA, etc.

Up to the implementing tool/library to make these available at run time

Arguments passed to the data source

Post processing options for the returned data

TokaMap – Map Types

UK Atomic Energy Authority

Expression Mapping

```
"key/path": {
   "map_type": "EXPR",
   "expr": "2*param1 + 5",
   "parameters": {
      "param1": "myarray/path"
   }
}
```

==> **[5.0, 7.0, 9.0]**

Mathematical expression to evaluate

Data from other mappings (i.e. VALUE or DATA_SOURCE) that can be used in the expression

TokaMap – Templating

UK Atomic Energy Authority

The mappings support Jinja style templating using Inja library:

```
"circuit": {
    "map_type": "VALUE",
    "value": "{{ length(PF_SUPPLIES) }}"
},
"circuit[#]/name": {
    "map_type": "VALUE",
    "value": "CIRCUIT_{{ PF_PS_NAME }}"
},
```

Various templating functions available.
See https://pantor.github.io/inja/ for details

Templates are rendered multiple times

This template is defined in the globals.json as:
"PF_PS_NAME": "{{ at(PF_SUPPLIES, indices.0).PS_NAME }}"

TokaMap – Parameterised Keys

UK Atomic Energy Authority

Mapping keys can be parameterised to avoid repetition:

"coil[#]/element[#]/geometry/rectangle/z"

Will work for any ‘coil’ and ‘element’ indices, i.e.:

"coil[0]/element[3]/geometry/rectangle/z"
"coil[2]/element[1]/geometry/rectangle/z"

Indices are available in the mappings via ‘indices’ variable.

TokaMap – Parameterised Keys

UK Atomic Energy Authority

```
"coil[#]/element[#]/geometry/rectangle/z": {
    "map_type": "PLUGIN",
    "plugin": "GEOMETRY",
    "args": {
        "signal": "/magnetics/pfcoil/{{ PF_GEOM_NAME }}",
        "key": "geom_elements.centreZ"
    },
    "slice": "[{{ indices.1 }}]"
},
```

This contains an array of the indices

TOKAMAP

TokaMap – Syntactic Sugar

UK Atomic Energy Authority

To remove some of the verbosity from common features some some ‘syntactic sugar’ has been added:

‘VALUE’ mapping expansion

"mapping_name": "value"

⇒"mapping_name": { "map_type": "VALUE", … }

"PF_COIL_{{ #0 + 1 }}”

⇒"PF_COIL_{{ indices.0 + 1 }}”

‘indices’ expansion

"{{ PF_SUPPLIES[#0].PS_NAME }}”

⇒{{ at(PF_SUPPLIES, indices.0).PS_NAME }}”

Inja array access expansion

TOKAMAP

TokaMap – Comments

UK Atomic Energy Authority

Trying to recover some of the readability and functions of YAML

```
"key/path": {
      "map_type": ”VALUE",
   "value": 42,
   "comment": "Optional description"
}

… "comment": "hello this will be done later"
```

TokaMap – MAST-U Wall Mappings


```
{
  "ids_properties/homogeneous_time": 2,
  "description_2d": 1,
  "description_2d[#]/type/name": "Equilibrium codes",
  "description_2d[#]/type/index": 0,
  "description_2d[#]/type/description": "A limiting surface for EFIT++",
  "description_2d[#]/limiter/unit": 1,
  "description_2d[#]/limiter/unit[#]/closed": 1,
  "description_2d[#]/limiter/unit[#]/outline/r": {
    "map_type": "DATA_SOURCE",
    "data_source": "GEOMETRY",
    "args": {
      "signal": "/limiter/efit",
      "key": "R"
    }
  },
  "description_2d[#]/limiter/unit[#]/outline/z": {
    "map_type": "DATA_SOURCE",
    "data_source": "GEOMETRY",
    "args": {
      "signal": "/limiter/efit",
      "key": "Z"
    }
  }
}
```


Dedicated MAST-U geometry data source

GEOM::get("/limiter/efit", {{shot}}).R

TokaMap – Tools

UK Atomic Energy Authority

Pip installing TokaMap provides the following tools:

tokamap --schemas-dir

tokamap --version

Simple CLI to provide information about installed version of the schemas

tokamap-validator /path/to/mapping/directory

Validator tool which checks a given mapping directory to confirm:

- Valid mappings.cfg.json
- Directory structure conforms to expected
- Globals.json and Mappings.json files conform to schemas
- Some static checks, i.e. “EXPR” expressions are valid

UK Atomic Energy Authority

Lessons Learnt (and return on experience)


2.0
1.5
0.0
5
0.0
-2.0

Time is Precious

UK Atomic Energy Authority

Physicists/diagnostians time hard to get

- Intuitive, to get people on board
- Richness of functionality – easy to read
- Often more important projects to prioritise or need funding for their help

Time is Precious 2

AP: “shouldnt be long then, 9-10mins is the average it was taking for me”

“passed it and still nothing...”

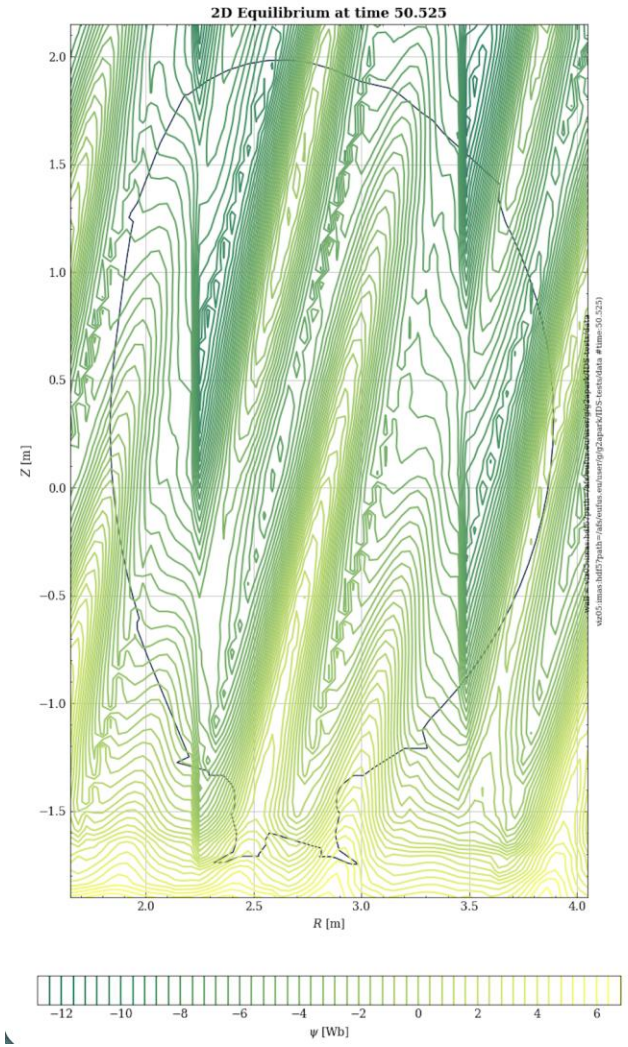
“i would hope even without caching we would make it faster since i am almost certain the CLISTE (equivalent to EFIT++) output at AUG is sampled much higher and would mean an hour.....”

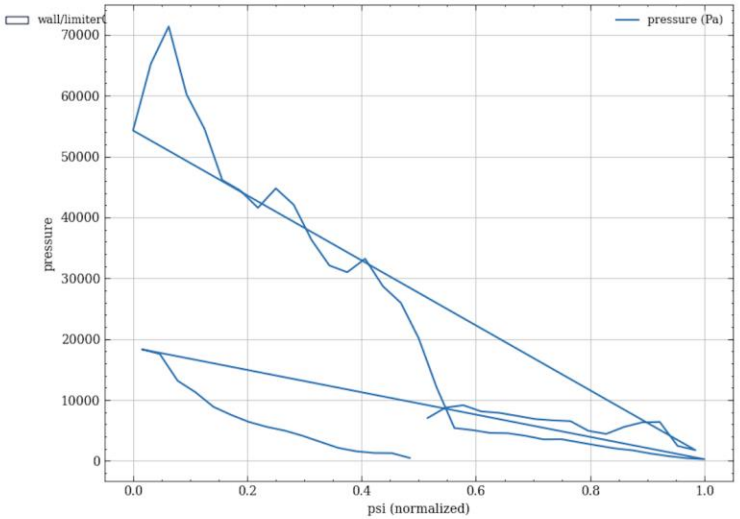
MAST-U IDS performance
magnetics ~20s
pf_active ~30s
pf_passive ~20s
wall ~1s
equilibrium ~30s

Aiming for sub-10s

- **Performance is very important**
  - Speed to map
  - Speed of mappings

Miscellaneous





- Data source dimensionality and ordering matters when using multiple data sources
- COCOS considerations take time
- Statically hardcoding mappings values is often quicker (but not ideal)

Benefits of Mapping Files

- Discovery, Validation, Versioning, Sharing, Reuse, Interoperability, Reproducibility
- Mappings are data (and can be controlled as such)
- Separation of definition and execution