An Integrated Software Framework for Magnetic Measurements
- From Raw Data to Assets

Matthias Bonora
Outline

- Magnetic measurement requirements
- FFMM – A Flexible Framework for Magnetic Measurements
  - Concept and idea
  - Components of a measurement script
  - Connection to webservice
  - Integration examples
- Development goals and future plans
Rotating coil systems

Helmholtz coils

3D mapper

Stretched wire systems
Magnetic Measurements

Ring-Sample Permeameter

Translating Fluxmeter

Fluxmeter

Physical object (magnet)

Magnetic flux density in the magnet bore

Capture noise (random, systematic)

Field Transducer

Raw signal (voltages)

Read-out noise (electronics, mainly random)

AD-conversion
Digital Integration, Drift correction

Processed data (fluxes)

Modelling, Calibration, Approximation errors

\[ B_x(r_0) = 2 \sum_{k=1}^{n} B_0(r_0, \phi_k) \sin \phi_k \]
A Magnetic Measurement Bench (1)
A Magnetic Measurement Bench (2)

- Setup
  - Measurement script
  - Parameters

- Acquisition
  - Raw data
  - Pre-processed data

- Analysis
  - Harmonics, multipoles, field, center offset, roll angles

- Results
  - Measurement report
  - Summaries
Magnetic Measurement Results
The TE-MSC-MM Value Shop

Client

Discuss measurement needs

MM Section

Infrastructure & Support

- Measurement benches
- Equipment
- Software
- Research
- Expertise

Choose measurement approach

Develop tools if required

Metrological Characterisation

Discuss, interpret, analyze results

Create results and reports

Validate, postprocess, analyze

Execute measurements
- Operation of many different measurement benches and types
- Similar, yet different acquisition systems
- Use of different sensors and actuators
- **Resource optimization**
  - Symbiotic benefit from measurement bench developments
  - Keep and reuse development expertise
  - Benefit from short term contracts and student contributions
- Need for an efficient software platform as base of operations
A Flexible Framework for Magnetic Measurements (FFMM)

- Idea for a software framework for magnetic measurements

- Reusable
  - Easy to implement small, independent blocks
  - Blocks are reusable when needed

- Flexible
  - Easy to write measurement applications
  - Still full control and extendability

- Framework
  - A software framework, in computer programming, is an abstraction in which common code, providing generic functionality, can be selectively overridden or specialized by user code for providing specific functionalities.
A Flexible Framework for Magnetic Measurements (FFMM)

- Works well with available resources
  - R&D with short term contracts
  - Implemented components stay integrated in framework
- Platform Independent, vendor independent
  - Build on open source software
  - Linux an option
  - Driver implementation for commercial devices optional
- Separation of Users, Test Engineers and Framework Developers
FFMM – Roles

- Developer context
  - Measurement software framework
  - Application Operator context
  - Measurement software application
  - ... 
  - Device 1
  - Device n

Test engineer context
First idea in 2007

First implementation for SM18 rotating coil benches

Extension to all rotating coil benches

Extension to wire benches (2011)

Coverage all of MM sections platforms
  - Around 40 systems for measurements plus R&D
  - Increased requirements on functionality and features
  - Coverage of functions beyond simple data acquisition

Many contributions from initial idea to current state by staff, students, short-term contracts, collaborations
  - Present developers: Matthias Bonora, Lucio Fiscarelli, Carlo Petrone
Implementation of many devices
- Data acquisition
- Multimeters / tesla-meters
- Motor controls
- Power converter controls
- Implementation of a GUI generator
- Description of a user interface in a few lines of text
- Generation of a user interface with parameter settings, plots, and user controls
Inputs for parametrizing a measurement
- Input fields
- Grouping of settings
- Tables
- Cycles
- Device settings (ports, locations, types)
- Measurement parameters

Simple plots
- Time series plots for fluxes/currents/DAQ signals
- Bar plots for histograms and multipoles
- Status signals

Inputs for measurement control
- Dialogues, status messages, dynamic controls
#Script = Techdemo 2

grouping = Group Groups
grouping.description = Text "Used to group a list of Parameters in a logical Unit."
grouping.check = Group "Checkable Groups" checkable
grouping.check.description = Text "Checkable groups have a checkbox. Their state is queried via groupname.enabled from the script"
grouping.flat = FlatGroup "A flat group"
grouping.flat.name = Text Flatgroup
grouping.flat.description = Text "A FlatGroup has no visual borders, but still groups elements within"
grouping.select = ComboGroup "Combo group" checkable
grouping.select.sub1 = FlatGroup "Subgroup 1"
grouping.select.sub1.description = Text "Combo groups store multiple items in the same space. The chosen subgroup is selected with a combobox. The name shown is the label of the subgroup"
grouping.select.sub2 = Group "Subgroup 2" checkable
grouping.select.sub2.description = Text "Any group can be a subitem of combogroup. To read any group can be a subitem of combogroup."

InputFields = Group "Text input elements"
InputFields.description = Text "For default value input, the Input element is used. Use can be defined with 3 different input types:"
InputFields.reals = Input "<Real> (decimal) values:" Real
InputFields.ints = Input "<Integer> values:" Integer
InputFields.txt = Input "<Text> values (anything):"
InputFields.desUnit = Text "An optional Unit info can be added at the end:" InputFields.unit = Input Length: Real mm
InputFields.cps = Input "Counts per second:" Integer cps
FFMM: Complex Plotting

- No limit in plotting functionality
- Simple plots in GUI description
- Complex plots as code within script
Provide features to run analysis online during data taking
- Immediate feedback
- Early results

Multiple approaches
- a) Use Matlab generated code for analysis (compiled C++)
  - Code sharing between offline post-processing scripts and online analysis
  - Quick implementation
- b) Use native C++ implementation
  - Use library for common needs (linear algebra, solvers, FFT calculation)
  - No dependencies on Matlab
FFMM: Webservices

Oracle

InforEAM
- Measurement Requests
  - Calibration
  - Assets

EDMS
- Raw Data
  - Measurement Information

MM Database
- Processed Data
  - Released Data

FFMM
Workorders derive from a client measurement request
- Track status of a measurement
- Central point for accessing information and data about a measurement
  - Results, data, execution information, used devices

Assets represent MM devices and sensors
- 2000 devices registered
- Rotating coil shafts, integrators, coils, probes, measurement benches,...
- Calibration data (including history)

Both integrated into FFMM
- Link a measurement to a workorder
- Track used assets for a measurement
- Add devices and equipment by barcode scan
- Access all measurement data through a workorder
FFMM: Data Storage

- Combination of InforEAM, EDMS, and custom Oracle database
- Trade-off between ease of access and ease of implementation
  - Simple results into InforEAM
  - Complex outputs into Oracle database
  - Raw data files and measurement parameters into EDMS
- Storage of all measurement parameters
  - Devices in use
  - Settings
  - Calibration data
- No raw data are discarded → reconstruction possible
  - In case of incorrect analysis parameters or errors
  - Future investigation of measurement results
FFMM: Measurement Dataflow

Client
- Workorder Creation
- Release Data

Measurement
- Select Assets
- Select Measurement Parameters
- Run Measurement
- Acquire Raw Data
- Online Processing
- Postprocessing
- Finish Measurement

InforEAM

EDMS Oracle
FFMM: Measurement Postprocessing

Online
- Run Measurement
- Acquire Raw Data
- Online Processing
- Finish Measurement

Offline
- Retrieve Measurement Data
- Combine, Merge, Filter, Clean up
- Check and Verify Results
- Reanalyze

EDMS Oracle

Additional Analysis
- Numerical / Simulation Models
  - FFMM
  - InforEAM
  - Matlab
  - SWAN / Python
- Fits
- Compare multiple measurements
- Compare Magnets

Measurement Report
Upcoming measurement campaign for HL-LHC
- High volume in measurements
- Reduction of feedback loop
Quality assurance and traceability of measurements
- Webservice implementation
  - Tracking of used devices and rotating coil shaft
  - Direct loading of calibration data
  - Storage of raw fluxes and processed multipole data
- All data linked to a measurement request
  - Traceability of measurement
- Online analysis for full system
  - Multiple segments, multiple apertures
- Faster access to results for postprocessing
- Immediate feedback of measurement
  - Connection errors show in analysis
  - Early possibility to restart measurement
Postprocessing with Python/Jupyter scripts
- Retrieve data by measurement request
- Perform data cleanup and checks on measurements
- Generate plots, tables, **release data into database**

Reusable
- Generic template with parameters
- Customized template for specific magnets
- Common core classes for processing, plotting and database access

Traceable
- Common code in version control
- Applied transformations stored in script on EDMS
- released data linked to EDMS document
FFMM: Summary of current state

- FFMM now established baseline for magnetic measurement scripts
- Continuous development of new features and integrations
- Integration of features beyond simple data acquisition
- An effective way of retaining implementations and R&D
  - Growing library of devices, components and features
  - R&D effort on a single script automatically integrated in framework
    - Mostly from students and short-term personnel

- What’s next?
Extend focus of FFMM beyond magnetic measurements
For any workflow using basic data acquisition or object manipulation
Framework provides
  - GUI
  - Visualisation
  - Device implementations
  - Access to InforEAM
    • Workorder access
    • Asset tracking
  - Access to EDMS
    • Store data
    • Load documents
  - Access to Oracle database
In-house calibration of coils and arrays

- Coils, arrays and PCBs added as assets to system
- Operator loads request for calibration in FFMM
- Operator performs calibration in FFMM
- FFMM uploads calibration values to InforEAM
- FFMM uses uploaded calibration values

- Example of script using simple daq + webservice features
- Mutual benefits from FFMM framework implementation
- Same concept easily adaptable to other domains wherever needed
  - FGC control (Current Cycles, pulsing, degaussing)
Increasing Requests for building measurement systems for external entities (INFN, CEA)
- Building and selling measurement systems not within CERN mandate

Concept: Measurement platforms based on commercial systems
- Rotating coil platform (PCB coils, commercial motors, commercial DAQ)
- Stretched wire systems (commercial stages, commercial DAQ)
- Measurement bench based on commercial products
- Provide a material list, technical drawings and instructions for setting up measurement benches

FFMM as measurement software
- Open source libraries
- Base for collaboration and extensions
- No dependencies on platform beyond device choice

Provide our expertise as service (collaborations)
⇒ Combine Models and Measurements
  - Avatar: Measurement, Simulation, Inference
    Based on Kirchhoff’s theorem
  - Twin: Fusion of measurements, models, empirical rules, machine learning

⇒ Track Avatars, Twins and applied algorithms in database

⇒ Improve models with measurement data

⇒ Better understanding of the magnet

⇒ Feedback for new transducers or measurements
Universal Magnetic Measurement Platform
- Covers majority of MM needs
- Separation of roles keeps development ongoing and operation simple
- Fully integrates with CERN webservices
- Results stored and ready for further development

Potential Extension of Scope
- Use outside CERN
- Use beyond magnetic measurements