Data modelling - a view from the NMR world

Rasmus Fogh, Global Phasing

ISpyB meeting, Grenoble
January 2017

GΦL

Contents

- **Introduction**
- Data modelling in NMR
- Possible approaches

GΦL

Introduction

- Global Phasing has recently joined the ISPyB consortium
  - (I have recently joined GΦL)
- ISPyB is expanding its scope
  - From mainly experiment and sample tracking
  - Adding emphasis on calculations and results

Likely requirements

- Not all calculation and validation parameters are shared between programs
  - Each program has specific input and output
  - Proper representation requires program-specific data
  - Programs continuously change and user interfaces must change to keep up
  - E.g. Autoproc now uses (and ourputs) ellipsoidal (instead of spherical) completeness
- ISPyB must support multiple parallel and changing data structures and interfaces

GΦL

Similar problems dealt with

- ISPyB: Separation of central core from varied user interfaces
- MXCuBE: Different hardware, similar data collection protocols

GΦL

But this also affects the core

- User interfaces belong to specific sites
  - Each interface developed and maintained on the site that uses it
- Different applications – and their specific needs - apply across all sites at once
- Differences in data model as well as user interfaces.
  - Database or XML document?

GΦL

Contents

- Introduction
- **Data modelling in NMR**
- Possible approaches

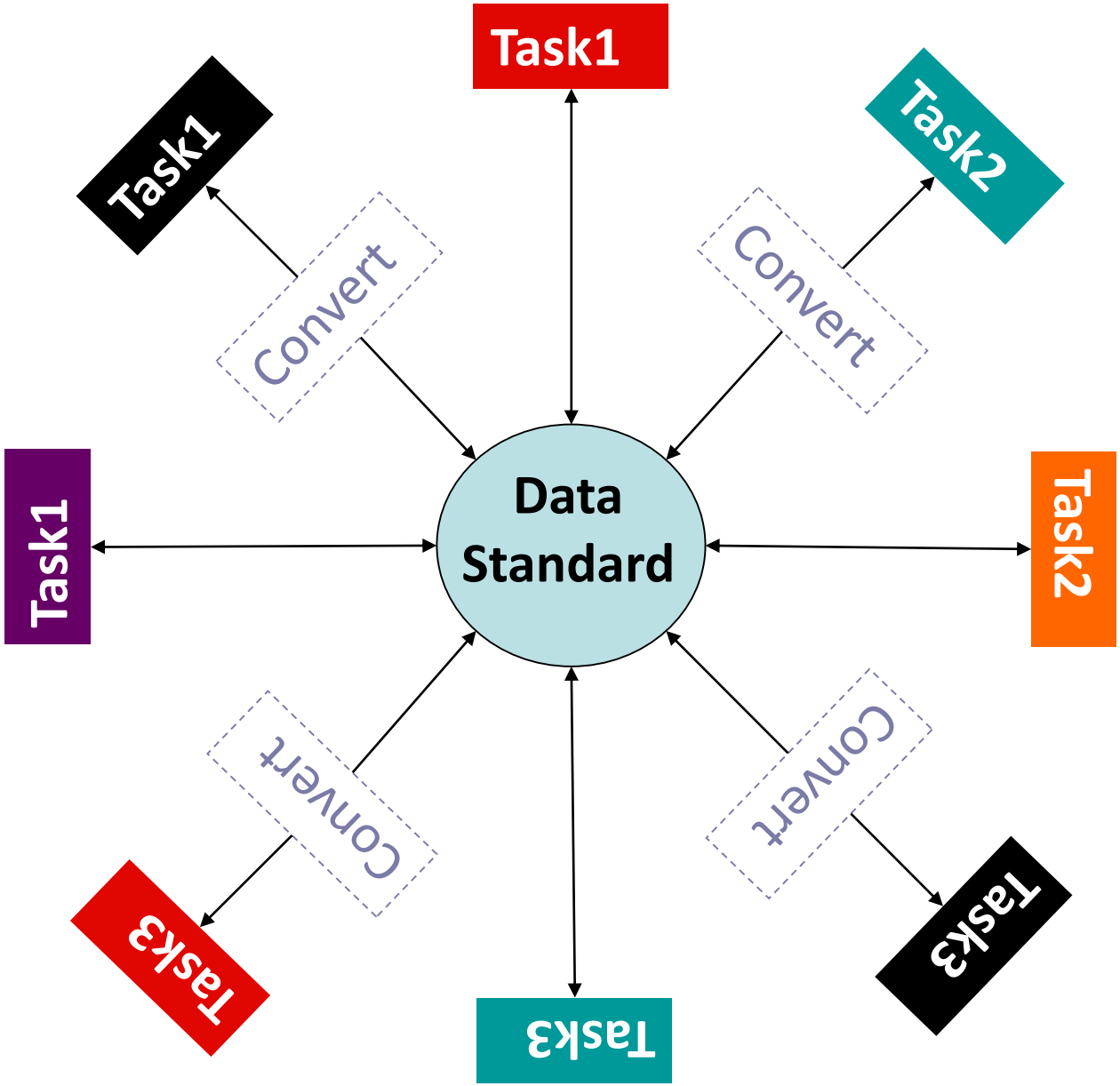
GΦL

Problem well known in NMR

- Omnicomprehensive data model
  - CCPN
  - NMRSTAR (BioMagResBank)
- ‘Sharing in one application’
  - CCPN and PIMS
- Shared core with program-specific extensions
  - Nmr Exchange Format

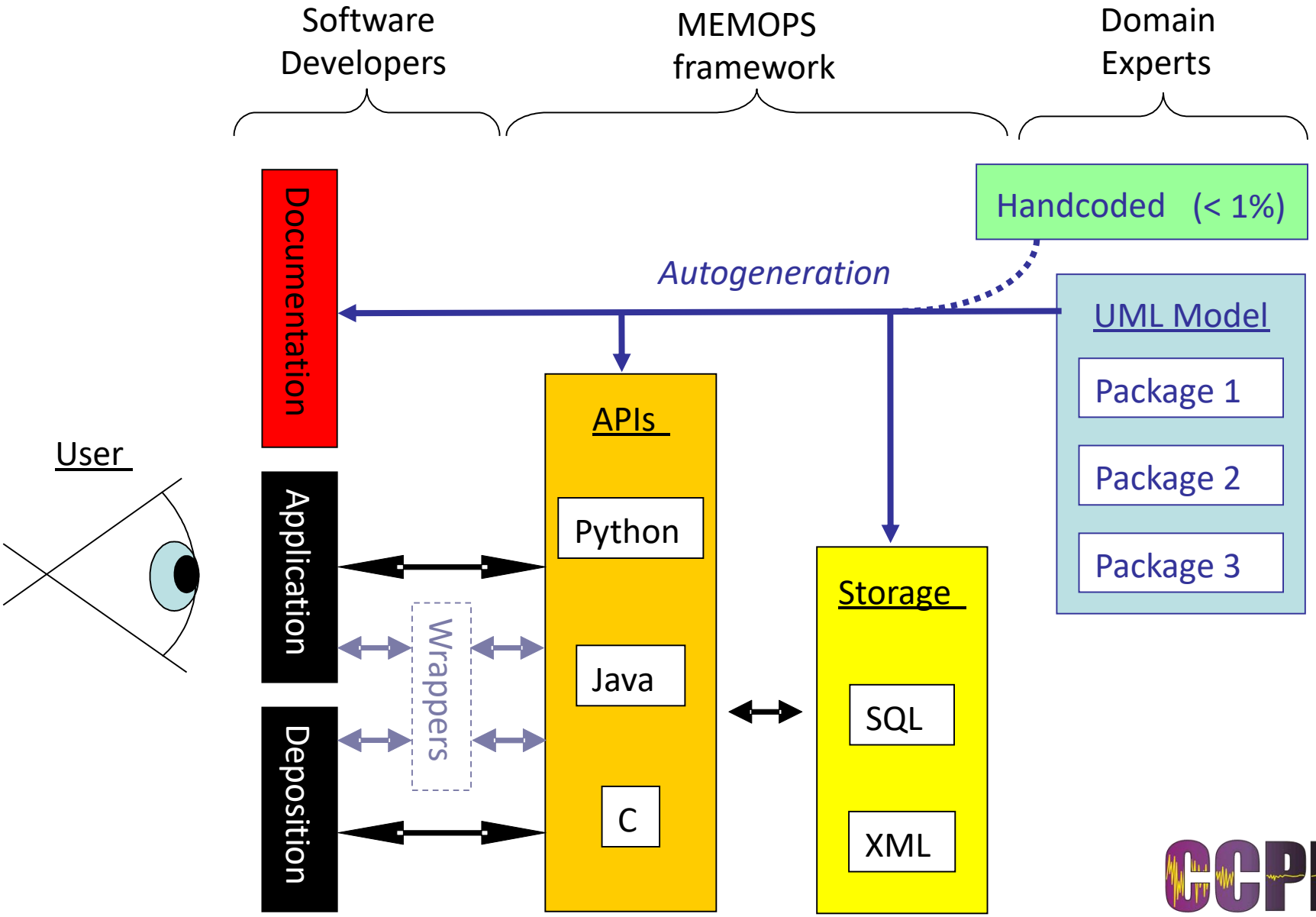
GΦL

Common Data Standard



CCPN

Code Generation Framework



CCPN APIs

- **A**pplication **P**rogramming **I**nterface and implementation
  - Is the standard
  - Data access subroutine library
  - Classes and objects, with functions (get, set, …)
- Precise, comprehensive Data Model
  - 50 packages
  - 477 classes
  - 3400 fields
  - Python-XML implementation has ca. 850 000 (autogenerated) lines of code
- Lots of supporting code:
  - Integrated, transparent I/O (file or database)
  - Complete validity checking against model constraints
  - Protection against casual change (data encapsulation)
  - Versioning and backwards compatibility
  - Consistency with underlying model guaranteed
  - APIs in Python, C, and Java

CCPN

Minimal uptake outside CCPN

- Great success within CCPN, but …

- Seen as ‘owned’ by CCPN
- Extension and model modification too hard
- Introduces major dependency

- ‘comprehensive’ + ‘detailed’ + ‘precise’ = ‘huge and complex’
- ‘Data modellers disease’

GΦL

NMRSTAR (BioMagResBank)

- STAR data format
- Used directly for deposition
- 1:1 match to BioMagResBank internal data structures
- Heavily promoted

GΦL

Little uptake outside BMRB

- Seen as ‘owned’ by BioMagResBank
- Extension and model modification hard or impossible
- STAR format not well supported
- STAR is
  - Human readable but imprecise (if denormalised)
  - Precise but unintelligible (if normalised)

- ‘comprehensive’ + ‘detailed’ + ‘precise’ = ‘huge and complex’
- ‘Data modellers disease’

GΦL

Single application (PIMS)

- Collaboration on new LIMS system

- CCPN working with PIMS
  - To extend open data model to cover LIMS area
  - Using same architecture and design throughout
  - Without being paid
  - ‘Open source’ collaboration model

- PIMS working with CCPN
  - To acquire bespoke data storage layer for PIMS
  - With full control over all implementation decisions
  - With detailed optimisation for PIMS way of working
  - ‘Project management’ collaboration model

GΦL

Collaboration dissolved

- The more stakeholders, the more complications
- Shared interests and goals insufficient to justify necessary sacrifices
- Much effort – little result
  - Some useful modelling remains

GΦL

NMR Exchange Format (NEF)

- Exchange format
  - NOT universal data standard
  - Limited size core covering commonly used data.
- Fully extendable with program-specific tags
  - Each group can add its own tables and fields
- STAR-based, denormalised text format
  - Hackable, human readable
  - No code dependencies
- mmCIF used for structures
- **ALL major NMR software groups have committed to using NEF**

GΦL

Why did NEF work?

- Developed de novo as collaboration with software groups
  - Implementation choices by consensus
- Owned by consortium – changes in core require consensus of users
  - Safe from hijacking by maintainers
- Each group free to extend as needed

GΦL

Contents

- Introduction
- Data modelling in NMR
- **Possible approaches**

GΦL

Conclusions

- The NEF model was clearly superior for sharing and exchange
- But ISPyB is a mission-critical application, not an exchange format, so the answer may not transfer directly
- Political buy-in and mutual support for mutual needs are key to success

GΦL

Support for program-specific data

- One comprehensive model
  - Ideal, but rigid, cumbersome and one-sided?
- Shared core with unlimited extension (NEF)
  - How does that work technically in a database environment?
  - In database? Two databases? Or in XML data file?
- Modeled ‘Tag-value’ semantics (like XML)
  - Imprecise and harder to search?
- Separate data for each application
  - Possibly in the form of an XML file or text blob?
  - Requires program-specific searches for everything

GΦL

Program-specific user interfaces

- Bespoke interface for each program's data?
  - Much redundancy and much to maintain
- Data-driven automatic UI?
  - Not highest quality
  - Gives UI without custom coding
- One core UI with mechanisms for additional fields?
  - Desirable but demanding?

GΦL

Final slide

- Global Phasing hopes to participate and help sort out these issues.

GΦL

Some Protagonists

- John Ionides (CCPN architecture)
- Eldon Ulrich (NMRSTAR architect)
- Chris Morris (PIMS collaboration)
- Geerten Vuister (NEF collaboration organiser)

- CCPN project, present and past members

- BBSRC, MRC, EU for funding

GΦL