Bringing your notebook is recommended but not required. Attending Part 1 is helpful but not required to follow Part 2.
Users vs. Developers

MMT blurs the distinction between users and developers

- MMT is not an application itself
- It is a
  - API for the MMT language close relative of OMDoc
  - suite of reusable algorithms/services e.g., MKM services
  - set of few example applications e.g., the IDE used in Part 1
- Primary users: developers of math applications
- Secondary users: users of those applications
Basic Design of MMT Implementation

- Data structures for MMT language
  - documents, modules, declarations, objects
  - get, add, update, delete

- Core algorithms
  - parse, check, simplify, present, . . .
  - index, query, diff, build, . . .

- Backend
  - catalog for mapping MMT URIs to physical locations
    - in git repositories, databases, etc.
  - transparent loading/unloading into memory

- Frontend
  - API calls from Scala, Java
    - programmatic or interactive
  - shell, scripting language
  - HTTP server
    - core algorithms exposed
Extension Interfaces

MMT systematically exposes extension interfaces

**essentially everything can be extended or replaced**

- Customize core algorithms
  - add rules declared in MMT theories
  - replace with custom implementations
  - combine algorithms for structural and object level
- Adding language features
  - literals with external data/computation
  - pragmatic features with elaboration semantics
- Import/export interfaces for integrating other formats and build targets
- Outside interface
  - adding new command line syntax
  - web framework for adding new HTTP interfaces
- Change listening infrastructure for content events
Example: jEdit IDE

Application: IDE for MMT theories
- based on jEdit text editor
- induces IDE for any
  - language defined in MMT
  - language exported to OMDoc

Design: MMT as p MMT and jEdit
- Plugin for jEdit that wraps around MMT
- MMT and jEdit large projects
- But only little glue code needed
  mostly forwarding to MMT API functions
Example: HOL Light library browser

Based on

▶ definition of HOL Light in MMT/LF
▶ export of HOL Light library in OMDoc format

MMT provides out of the box

▶ HTML+MathML rendering of library including 2-dimensional notations
▶ as interactive documents definition lookup, type inference
▶ dependency graph of theorems
Example: Generation of GAP Inheritance Graphs

Based on GAP export to OMDoc
Only a few lines of code to
  - generate additional predicates for MMT’s relational index
  - register a new graph built from this predicate
yield
  - generation of inheritance graph between GAP filters
  - interactive, integrated with web browser
Example: Codec-Based Knowledge Exchange

Problem:
- mathematical data distributed over multiple databases
  LMFDb, OEIS, findStat.org, . . .
- in incompatible, unspecified low level encodings

Solution: Codec infrastructure of MMT
- MMT theory for codecs and codec operators
- Describe database schemas
  - using mathematics-near types
  - annotate codec expressions to define concrete encoding
- Treat databases as MMT backends

Effect: MMT provides uniform high-level interface to low-level databases
Example: MathHub

Application:
- Project hosting platform for mathematical library based on gitlab
- Web interface for browsing, interacting with libraries based on Drupal
- MMT used to process, interpret the content

Implementation: suite of MathHub-specific MMT extensions
- new sTeX importer for interpreting sources uses MMT’s build syste
- HTML presenter for producing web pages uses MMT’s MathML presenter for objects
- HTTP server plugin for high-level MathHub specific queries uses MMT web framework, API functions
- HTTP server plugin for browser-based editing uses same functions as jEdit IDE
Overview

1. Brief introduction to MMT-based Applications
2. 3 mini-demos of prototypical MMT-based applications
   easy for attendants to understand, reproduce, modify
   2.1 Changing equality by adding arbitrary rewrite or computation rules
   2.2 Using the MMT query interfaces to build a browser-based editor
   2.3 Using MMT’s export infrastructure to build an OpenMath Content Dictionary editor
Let’s Start

- I will show mini-demos on the screen
- The tutorial points to the self-documenting source files
- Some demos can be run/changed by you as well
- Main link: http://uniformal.github.io/doc/tutorials/applications/
  no need to type this — these slides are linked from the CICM program