Java Community Process

Community Development of Java Technology Specifications

JSR 385 Early Draft Report

October 18, 2018

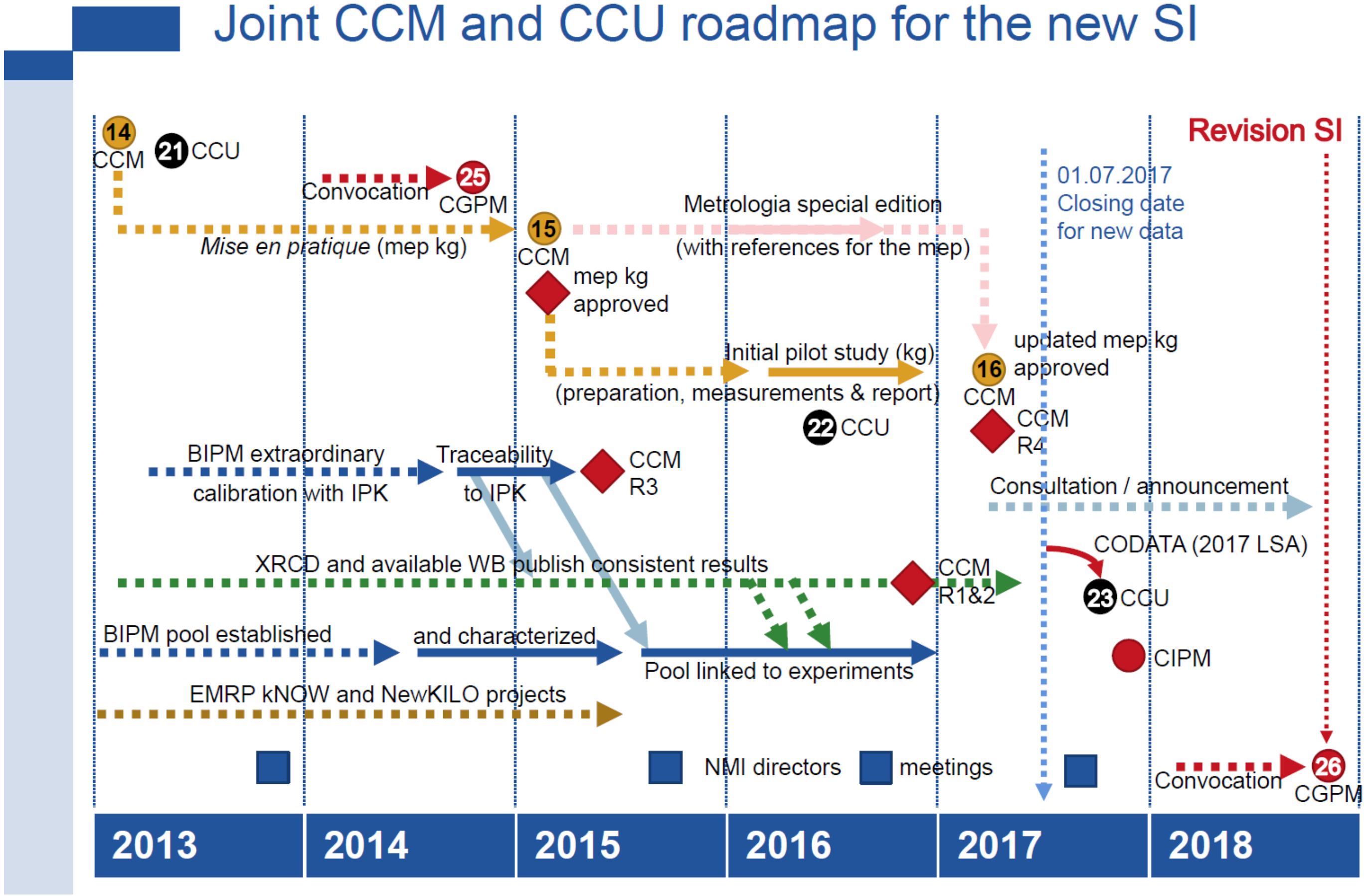
Werner Keil
Otavio Santana

History

- A framework supporting robust representation and correct handling of quantities.
  - For example, it may be unclear whether a person's mass is expressed in pounds, kilograms, or stones.
- JSR 363 established safe and useful methods for modelling physical quantities.
- Interfaces and abstract classes supporting unit operations including
  - Checking of unit compatibility
  - Expression of measurement in various units
  - Arithmetic operations on units
- Concrete classes implementing standard unit types (base, derived) and unit conversion.

SI Roadmap

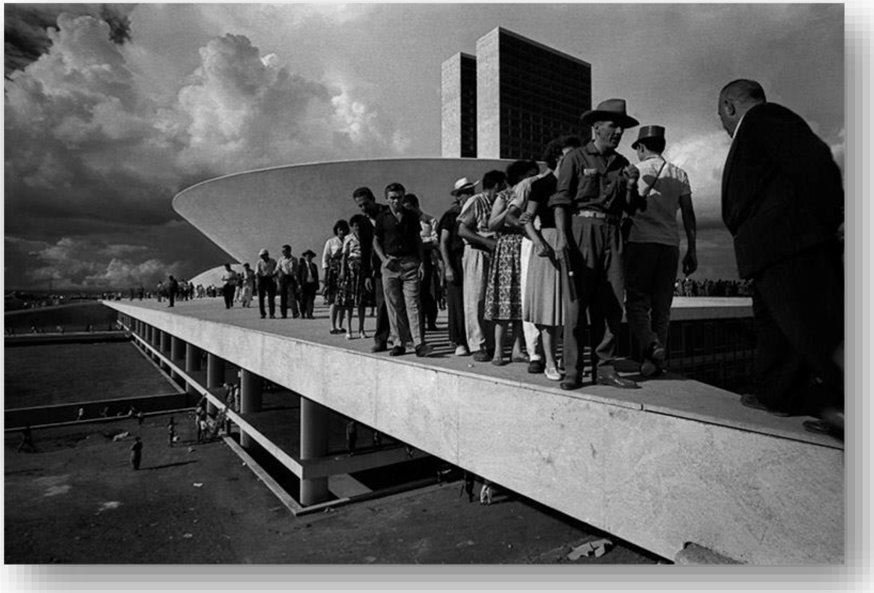
- The SI Standard will be revised in May 2019
  - The General Conference on Weights and Measures (CGPM) is expected to approve SI reforms in November 2018



Joint CCM and CCU roadmap for the new SI

1960

- JFK was elected US President
- Real Madrid won European Cup 7:3 against Eintracht Frankfurt
- Brasilia becomes Capital of Brazil



Motivation

- Because the new SI Revision is the most significant change to the Metric System since it was introduced in its current form 1960, this was a key reason to increase the version number to release 2.0.
- Another argument is adopting to a rapidly changing Java Platform and Runtime which itself tends to increase the major version of the JDK twice a year now.
- Time and the community will tell new requirements beyond that, but it is likely Units of Measurement 2.x should be the standard for at least the 2020s or beyond.

New Features in EDR

- Embracing Java 8 features like Lambdas or the new Date/Time API via a bridge in the RI (similar to what uom-se did for JSR 363)
- Align with Java 9 features where appropriate. The API should at least run under Java 8. Where applicable we use the Multi-release JAR feature of Java 9 to support multiple JDKs with a single API or implementation.
- QuantityFormat in the API
- Prefix in the API
- Including the 2 most common prefixes
  - Metric Prefix
  - Binary Prefix

Planned Features

- Adopting proposed changes in the **new SI Revision**, mostly Spec or Javadoc, but some of it may also influence code.
- Support “Compound Units” and “Compound Quantities”, like “5ft 2in”´by making `QuantityFormat` implementations aware of the compound nature
- Support different **levels of measurement**, e.g. `Interval` for quantity types like `Temperature`.
- Improved support for JVM languages like Kotlin

See https://waffle.io/unitsofmeasurement/unit-api

and https://waffle.io/unitsofmeasurement/indriya

Schedule

- Schedule for JSR 385.
  - Submitted: December 15, 2017
  - Creation approved: January 8, 2018
  - EDR finished: October 13, 2018
  - Public Review planned: Q4 2018
  - Proposed Final Draft planned: Q1 2019
  - Final Release planned: Q2 2019

Allowing extensions like SI Units to roll out just in time for the new SI Revision Spring 2019

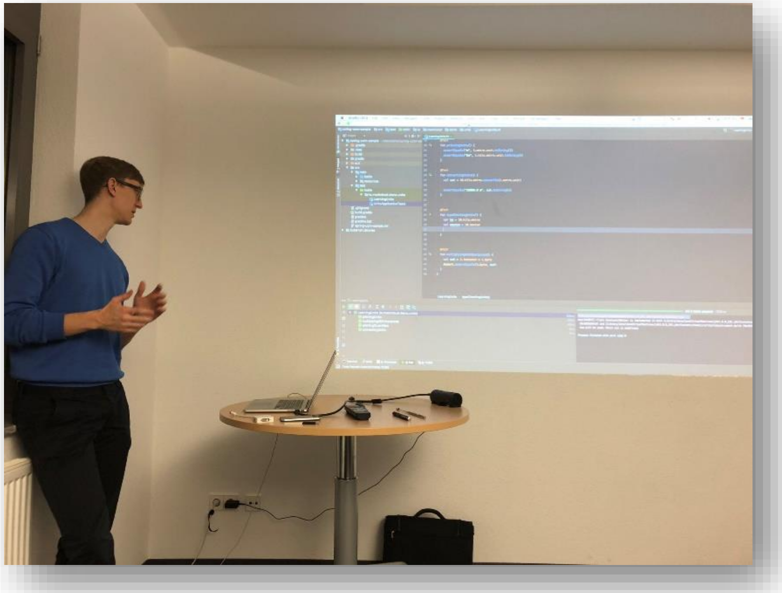
Publicity

- DevoXX UK 2015
- DevoXX BE 2015
- DevoXX US 2017

Publicity (2)

- JVM-Con 2017
- JUG Mainz January 2018
- Unconference at JavaLand 2018
- JavaZone 2018
- OOP 2019

org.jetbrains.kotlin:kotlin-stdlib-
org.jetbrains.kotlin:kotlin-stdlib-jre8:1.
Event Log
JVMCon
Die Konferenz für Java-Entwickler



Development

- We develop the JSR collaboratively through http://unitsofmeasurement.github.io
  - Committers: 8 EG Members (Daniel-Dos, dautelle, duckasteriod, jhg023, keilw, magesh678, otaviojava, Geomatys)
  - Contributors: 5 (Ellis Berry, Andi Huber, Rustam Mehmandarov, Anakar Parida, Filip Van Laenen)
  - 40+ GitHub users contribute to wider project
    - Demos
    - Integrating Bean Validation, JSON,…
    - Language Bindings (e.g. Kotlin, Groovy)

Development (2)

- The API continues to be under https://github.com/unitsofmeasurement/unit-api
- The RI is available under https://github.com/unitsofmeasurement/indriya and on public repositories like JCenter or MavenCentral
- The TCK is available under https://github.com/unitsofmeasurement/unit-tck

  All of them with an EDR tag: “2.0-EDR”

- All source-code repositories at https://github.com/unitsofmeasurement

Adopt-a-JSR

- We are participating in the Adopt-a-JSR program
- Some of the 4 JUGs continue to be involved:
  - Morocco JUG
  - SouJava
  - JUG Chennai
  - JUG Hyderabad
- Especially SouJava / with Otavio stepping up as Spec Lead after Leo/V2COM take care of other areas like Java ME/Embedded.
- The JUG Mainz also just did a presentation about JSR 363 and we'll reach out e.g. in JavaLand if they would like to contribute to or adopt JSR 385.

Communication

- Public mailing list(s) and/or forum(s)
  - Units-Dev on Google Groups: https://groups.google.com/forum/#!forum/units-dev
  - Units-Users on Google Groups: https://groups.google.com/forum/#!forum/units-users
- Collaboration happens via Google Groups, GitHub, Twitter or Stack Overflow in some cases.
- Our document archive is available on https://bintray.com/unitsofmeasurement/downloads

Users

- PCP Parfait
- GeoAPI and projects using it
  - Apache SIS
  - Eclipse LocationTech
- Eclipse SmartHome / OpenHAB
- Eclipse Science, UOMo
- Hibernate Validator 6 (BV 2 RI)
- Apache Tamaya
- Opower (Oracle)
- Jadira Bindings
- Several other commercial users

Questions, discussion, next steps

Thank you!
http://jcp.org/en/jsr/detail?id=385