Motivation

Software engineering can only succeed, if it is equipped with a rigorous experimental component that provides support to:

- study theories about the nature of software,
- investigate application domains and programming paradigms, and
- analyze developer behavior and decisions as manifested in the code they produce.

Stevens (1946): “Measurement is never better than the empirical operation by which it is carried out.”

Design

jCT is a stand-alone Java application. It operates on byte code rather than source code. Byte code is isomorphic to source code, yet due to the absence of any source-level syntactic sugar byte code offers a more direct access to the desired information. Except for the IL instructions, jCT is language-neutral and does not use a built-in meta model. Also, jCT does not provide any default measures. It relies on a viewpoint-agnostic light-weight extension mechanism to supply suitable metrics definitions. The resulting flexibility allows for tailoring measures to the specific needs of the intended analysis.

Features

We have developed jCT specifically to help developers and researchers understand software systems from empirical data. To be effective, we require jCT to:

- process non-trivial software systems,
- support the addition of new measures easily,
- yield a natural and adaptive approach for meaningful interpretation of metrics data, and
- enable repeatable studies.