0.0.1 Laborantin

Report by:	Lucas DiCioccio
Status:	Working, development for new features

Conducting scientific experiments is hard. Laborantin is a DSL to run and analyze scientific experiments. Laborantin is well-suited for experiments that you can run offline such as benchmarks with many parameters.

L aborantin encourages users to express experiments parameters, experiment results, as well as execution, startup, and teardown procedures in a methodical manner. For instance, the following snippet defines a network 'ping' experiment with a destination and packet-size parameters.

```
ping = scenario "ping" $ do
  describe "ping to a remote server"
  parameter "destination" $ do
    describe "a destination server (host or ip)"
    values [str "example.com", str "dicioccio.fr"]
  parameter "packet-size" $ do
    describe "packet size in bytes"
    values [num 50, num 1500]
  run $ do
    (StringParam srv) ← param "destination"
    (NumberParam ps) ← param "packet-size"
    liftIO (execPing srv ps) ≫= writeResult "ping.out"
execPing :: Text → Rational → IO (Text)
execPing host pktSz = let args = ["-c", "10"
  , "-s", show (round pktSz), T.unpack host]
  in fmap T.pack (readProcess "ping" args "")
```

Laborantin also lets users express dependencies between experiments. Laborantin is designed to allow multiple backend (where to run and store experiments) and multiple frontends (how a user interacts with Laborantin). The current backend stores experiment results on the filesystem and provides a command line frontend.

C ontributions are welcome. In the future, we plan to enrich Laborantin with helper modules for common tasks such as starting and collecting outputs of remote processes, reformatting results, and generating plots (e.g., with Diagrams). Laborantin would also benefit from new backends (e.g., to store results in an SQL database or HDFS) and new frontends (e.g., an integration in IHaskell).

Further reading

Hackage page: http://hackage.haskell.org/package/laborantin-hs Example of web-benchmarks: https://github.com/lucasdicioccio/laborantin-bench-web