Build system

- declarative specification of dependencies between targets
- build recipe is imperative list of commands and macros
- interpreter: only (re)build target if it's older than any dependency

1977 → ...: configurability
- configuration script: high-level, platform-independent description
- build script: generated per platform

MAKAO

Build systems contain valuable data for various stakeholders about all facets of software.

Makefile Architecture Kernel for Aspect Orientation (MAKAO):
- extract and offer this knowledge to all stakeholders
- enhance data gained by source-code techniques
- aid in re(verse)-engineering of build systems

How?
- represent dynamic traces of concrete builds
- Directed Acyclic Graph (DAG)
- retain links back to static build data

Approach

EXPLORER:
- explore and navigate through DAG
- discover available build concerns
- find spread of targets over build scripts
  example: add source code preprocessor

WEAVER:
Apply modifications:
- logically (in-memory)
- physically (scripts)

```make
weave_before( [T for (c,t,T) in base],
            [c for (c,t,T) in base],
            advice)
```

Conclusion & Future Work

MAKAO helps to re(verse)-engineer build systems ↔ TODO: more surveyable visualization, other case studies, DSL on top of Gython, ...