Configuration Library Semantics

C++Now 2015

Library in a Week

David Sankel

Some Definitions

a -> b: A pure function from type a to type b

‘a’ Editor: A pure function from type a to type b

Meaning of command line arguments

Command line arguments, when taken together, are a configuration editor where configuration is an arbitrary data type.

For example, the meaning of “--user jorfy --mode status” is a function which takes in a configuration and returns a new configuration where the user is set to “jorfy” and the mode is set to “status”.

Command line data

Command line data is a list of strings.

So...

- User develops their own custom configuration data type.
- User develops their own custom configuration editor data type.
- User writes a parser (via. spirit?) to parse the command line into the configuration editor.
- User then applies the configuration editor to a default configuration.

Note: The command line parser is separate from an .ini parser or whatever. Usually there are less options there.

Git-like configuration editor

```
struct ConfigurationEditor
{
  std::vector< Flag > flags;
  variant< SubmoduleCommand, StatusCommand > command;

  void operator()( Configuration & ) const;
};
```

Command line parser w/ imaginary parsing lib

```
auto flagParser = ...

auto submoduleCommandParser = ...

auto statusCommandParser = string(“status”) [ StatusCommand() ];

auto commandLineParser =
  (flagParser >> (submoduleCommandParser || statusCommandParser ))
  [ ConfigurationEditor]
```

Note: ‘>>’ doesn’t imply spaces, but extra strings.

User Code

```
ConfigurationEditor e = parseCommandLine( argc, argv, commandLineParser );
Configuration c = e( defaultConfiguration );
// ....
```

```
git –root=../foo/bar status
git –root=../foo/bar submodule update --init
```

Other thoughts

- Collapse the ConfigurationEditor structure to functions that are in the grammar?
- Use the grammar to generate documentation. Do we even want to do something like that in general? Wouldn’t work with something complex like git unfortunately.