CS446
SOFTWARE ENGINEERING

Richard Bartels

University of Waterloo
Faculty of Mathematics
Department of Computer Science

May 28, 1996

GP

CS446
GP OVERVIEW

Richard Bartels

University of Waterloo
Faculty of Mathematics
Department of Computer Science

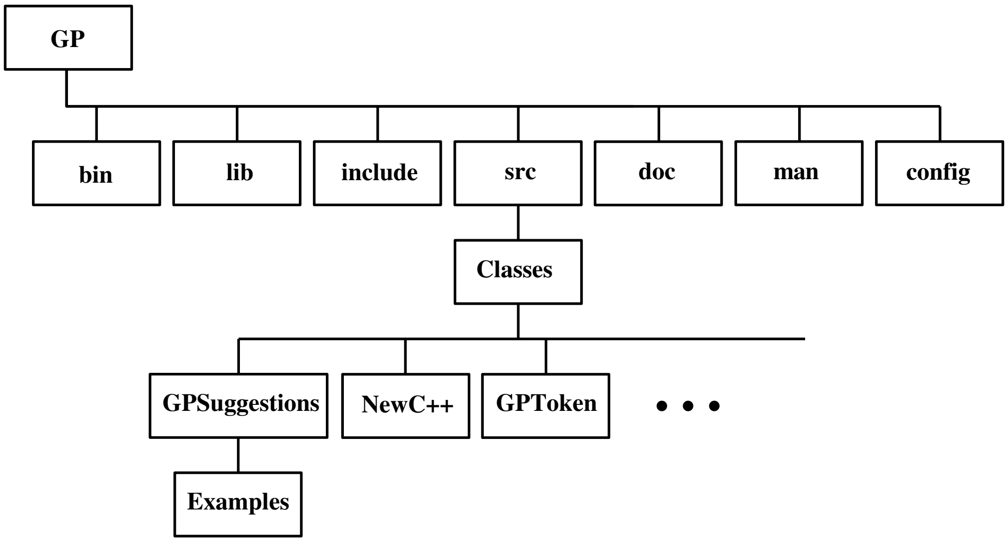
May 28, 1996

Outline

- Directory Structure
- Class Development
- Commands and make Targets
- Archiving
- Examples and Verification
- man Pages

GP Environment Overview

- Root directory is `...GP`
- Set `PATH` to include `...GP/bin`; e.g.:
  `` setenv PATH `showpath ~/CS446/GP/bin current` ``
- Extensive documentation:
  `~cs446/Formatted/Manual.ps`
- `man` pages can be read using: `gpman` *item*
- Directory structure:



Class Development

- Creating a class sector

  `gpnewsector` *sectorname*

- Results in a directory structure

  ./*sectorname*, ./*sectorname*/`Examples`, ...

- Creating a new class in ./*sectorname*

  `gpnewclass [` -*t* `]` *classname* [*baseclassname*]

- Results in files

  classname{`.h,.cc`}, `Imakefile`

- Creating a new example program in ./*sectorname*/`Examples`

  `gpnewexample [` -*t* `]` *examplename classname*

- Results in files

  examplename{`.cc,.input,.output`}, `Imakefile`

- Creating a new application is similar
- Creating a `Makefile` from an `Imakefile`

  `gmkmf`

- `make` targets

  `make`, `make verify`, `make clean`, ...

Remarks

- GP's `gpmkmf/Imakefile` extends `imake`
- `gpnew`*xxx* commands generate starter files/directories
- All starter files have common, designated style
- `.h` file format automatically creates `man` page
- File/directory structure enforced:

  sectorname/*classname*`.h`

  sectorname/*classname*`.cc`

  sectorname/`Examples/`*classname*`Example.cc`
- Example programs conform to general `#include` useage

  `#include "`*sectorname*/*classname*`.h"`
- Central library and include facility (`make install`)

  `...GP/lib`, `...GP/include`
- Individual GP environments are easy to establish

  Copy `GP` directory tree to new site

  Edit new site into `...GP/bin/LOCATIONS`

  Reset `PATH` to new `...GP/bin`

Archiving and Version Control

Unix facilities are minimal (could be much better):

- RCS
  - Versions of file stored in `diff` format

    `Foo.xx` → `RCS/Foo.xx,v`
  - Tree numbering or (optionally) labelling supported
    - ▹ `release.level.branch.sequence`
    - ▹ `name:release.level.branch.sequence`
  - Keywords logged in the file, for example
    - ▹ `$Log$` (all check-in comments)
    - ▹ `$Author$` (last person checking in)
    - ▹ `$Revision$` (latest r.l.b.s or name)
  - `rlog` reads log and status of file
  - File management
    - ▹ File checked out to one person (`co -l` command)

      `Foo.xx` is under lock-edit to that person

      `RCS/Foo.xx,v` has unchanged version
    - ▹ File checked in (`ci -u` command)

      `RCS/Foo.xx,v` has all versions including latest

      `Foo.xx` is a read-only shadow copy of latest
- SCCS (a similar Unix version-control system)

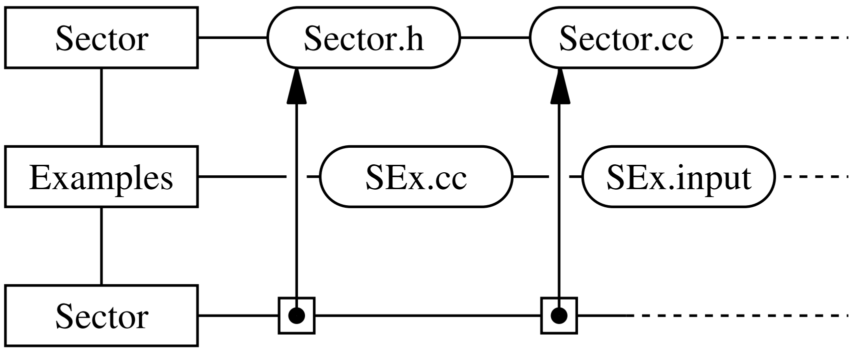
GP Provides RCS for File Collections

- Administered by `Imakefile`
  - Root directory and tree implicit in `SUBDIRS` lists
  - All files listed in `GPArchive` macros
- `make` targets
  - `make checkout`
    - ▹ Check out all files in tree to me
  - `make checkin`
    - ▹ Check in all files, leaving shadow copies (all files will have common r.l.b.s)
  - `make revision number=r.l.b.s`
    - ▹ Re-register all checked-in files with new revision number
  - `make symbolic name=n:r.l.b.s`
    - ▹ Assign symbolic name to revision number
- Management intention
  - Individual files assigned to individuals
  - Individuals manage versions through `co/cp/ci`
  - Low-level management through `make`
  - High-level management through `gp` commands
    - ▹ `gpextractsector`
    - ▹ `gpinsertsector`

Verification Examples

Each sector has `Examples` subdirectory

- Example programs with example input and output
- `make` targets to administer
  - `make`, `make verify`
- Examples use same include structure as applications
  - But some includes needed from `GP/include`
  - Sector includes needed from upper-level directory
- Resolve by *sectorname* subdirectory and symbolic links
  - `I` flag structure: *-I. -IGP/include*



man Pages

Each header file has a strict format

```
#ifndef Thing_H
#define Thing_H
// Sector/Thing \- one line
//
// Words
//
// .SH USAGE
// .EX
// Code
// .EE
//
// .SH LIBRARIES
// .SH SEE ALSO
// .SH AUTHOR(S)
// .SH COPYRIGHT
class{...};
#endif
```

Format is converted to man page

initial segment is *-man* layout

final section is parsed for formatting