GAP Webpage: Infrastructure Overhaul

Max Horn

The status quo

The problem

- Maintaining GAP involves lots of boring tasks nobody (?) *loves* to do…
- … but which *have* to be done or else everything gets worse
  - maintain website,
  - administrate mailing lists,
  - maintain CI solutions,
  - make releases,
  - handle support requests,
  - fix bugs,
  - write documentation,
  - …

More on the problem

- Taking care of these tasks requires …
  1. volunteers who
  2. know what needs to be done and
  3. have access permissions
- GAP 4.11 was scheduled for September 2019
- GAP 4.11 was actually released in March 2020
- Oops?

Why the holdup?!?

- Steps for making a GAP release (heavily abbreviated!)
  1. prepare release in the `gap` repository,
  2. generate release notes,
  3. run scripts from `gap-distribution` repository …
  4. … on a computer set up just the right way,
  5. prepare (compile) Windows binaries,
  6. upload resulting archives to a server few can access,
  7. update `GapWWW` repository (on the computer from step 3)

… and it gets worse

- Making a release requires too many too complex steps
- Very few people even know about all these steps
- And even for those who do, it is easy to screw up
- Even fewer people (1?) have access to all involved parts
- Result: even trivial website updates essentially only happen during a GAP release
- Want to change how package pages look? Have to modify releases scripts! But how to test this w/o making a release?

What can we do?

- Reduce the complexity
- Untangle tasks
- Automate as much as we can
- Document the rest and make it as easy as possible
- *Do it incrementally!*
- In this talk, we focus on the webpage

What's wrong with our webpage?

- Content & design ⇒ topic of the next talk!
- This talk will focus on technical issues
- I will discuss the current status and suggest alternatives
- More details on `bit.ly/gap-infra`

Website data management

- Good: website *code* is in a public repository
  `github.com/gap-system/GapWWW`
- Bad: not all data is in that repository, e.g. file downloads, manuals
- Ugly: *deployment* is to a private server
- Solutions:
  - ensure all text content is in repo: e.g. manuals
  - separate file downloads ⇒ add `files.gap-system.org`
  - perhaps also use GitHub file release system?
  - move to Cloud hosting, with convenient access management?

Mixer vs. Jekyll

- Good: uses a static website generator:
  `github.com/gap-system/Mixer`
- Bad: it's hand-rolled, so nobody knows it
- Let's switch to Jekyll?! Well-known, extensive docs, used by many GAP packages
- See `github.com/gap-system/GapWWW/pull/142`

Migrating to Jekyll

- See `github.com/gap-system/GapWWW/pull/142`
- During transition, use Jekyll and Mixer together:
  - Mixer converts `.mixer` files to `.html`
  - Jekyll converts `.html` to `_site/.html`
- *Do it incrementally!*
- ⇒ let's look at `README.jekyll.md` inside that PR

Special cases

- packages
- bibliography
- GAP3 website
- … more ?

How you can help

- help convert the `GapWWW` repository to the new setup
- initial work done in PR #142 on the `GapWWW` repository
- help with special cases (packages, bibliography, …)
- help set up `files.gap-system.org` and migrate files
- look at `bit.ly/gap-infra` for details & more ideas
- if we join forces, much can achieved this week!