Coq — Recent History

SIGPLAN Programming Language Software Award
POPL 2014 — San Diego, USA
January 23rd 2014

Matthieu Sozeau, for the Coq team
My Scientific Genealogy

Gérard Huet
Christine Paulin
Jean-Christophe Filliâtre
Coq 7.*
The 8.* generation (2004 —)

Hugo Herbelin
Our success story

• “Coq” is pronounced at many POPL talks

• Google Scholar says ~3k papers mention Coq

• Our recent survey indicates that 300+ were published in the POPL/PLDI/ICFP and ITP/CPP communities during the last 2 years

• And that Coq has been taught to 1500+ students!
Your contribution

- Teaching material: Coq’Art, Software Foundations, CPDT, OPLSS lectures, Summer Schools, ...

- Major collaborative developments:
  - PL: Java, C, assembly languages, JavaScript, ...
  - Security and cryptography: CertiCrypt, JavaCard, ...
  - Mathematics: C-CoRN, 4 color, EU ForMath project, Feit-Thompson, HoTT, ...

- Systematic use in PL research
Coq as a Programming Language – Recent Evolutions

On top of CIC, a full spectrum DTPL (Cayenne, Agda):

- Extraction
- Fast computation, reflection: VM, native compilation
- Higher-level tools: Function, Program, Notations
- Inference: Type Classes, Canonical Structures
Coq as a Proof Development Environment

– Recent Evolutions

• Ssreflect

• CoqIDE, forthcoming document editing support

• Many user contributed decision procedures and domain-specific libraries (i.e. Separation Logic, AC rewriting, coinduction, …)
Future

- Foundations: Homotopy Type Theory, Observational Type Theory, CIC*, CIC^, CoqMT, ...
- Higher Level: user interfaces and API, search tools, package management
- Diffusion: Coq for education
- Certifying Coq itself
Public Announcements

- Coq 8.5 will be released this year with many new features.
- The 6th Coq Workshop will take place in Vienna, collocated with ITP.
- Envisioned regular users/developers meetings at PL conferences.
- Efforts to rationalize and open up the development process (100+ contributors). Developer bootcamp in 2014~2015.
Acknowledgments

“A most important, but also most elusive, aspect of any tool is its influence on the habits of those who train themselves in its use. If the tool is a programming language this influence is, whether we like it or not, an influence on our thinking habits.”

– Edsger Dijkstra