Gecode

an open constraint solving library

Christian Schulte
KTH – Royal Institute of Technology, Sweden

Gecode

- Generic Constraint Development Environment
  - open source
  - C++ library
  - constraint propagation + complete (parallel) search
  - finite domain and finite set constraints
  - complete documentation (reference, tutorial, papers)
  - thousands of users

Overview

- History and facts
- Modeling (interfacing) & programming
- Openness

History

- 2002
  - development started
- 1.0.0
  - Dec 6, 2005
- 2.0.0
  - Nov 14, 2007
- 3.0.0
  - Mar 13, 2009
- 3.5.0 (current)
  - Feb 1, 2011

24 releases

43 kloc, 21 klod

77 kloc, 41 klod

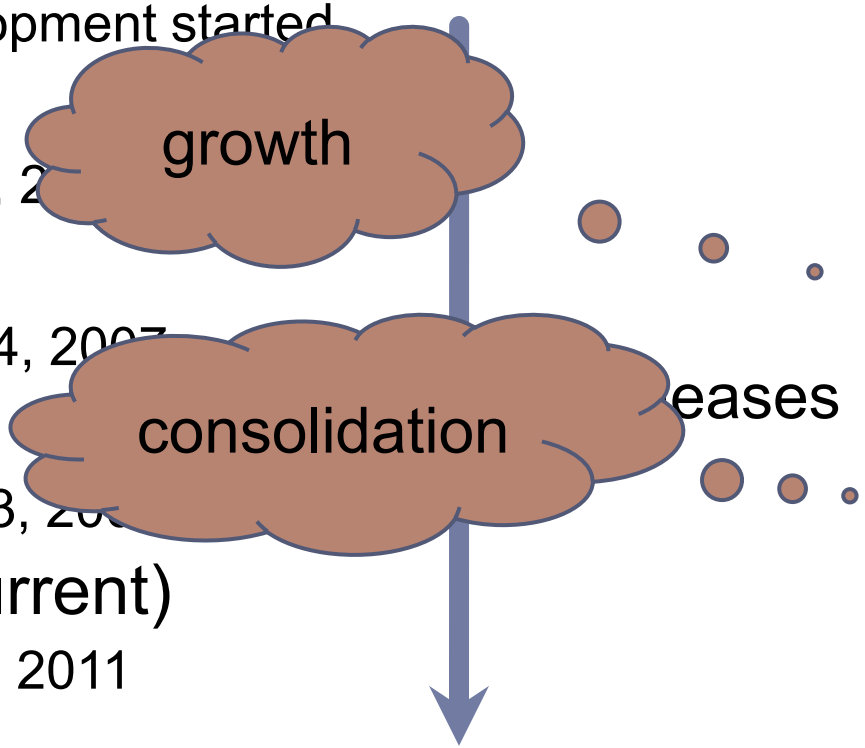
81 kloc, 41 klod

126 kloc, 52 klod

- … 4.0.0 in 2012

History

- 2002
  - development started
- 1.0.0
  - Dec 6, 2
- 2.0.0
  - Nov 14, 200
- 3.0.0
  - Mar 13, 20
- 3.5.0 (current)
  - Feb 1, 2011



43 kloc, 21 klod

77 kloc, 41 klod

eases

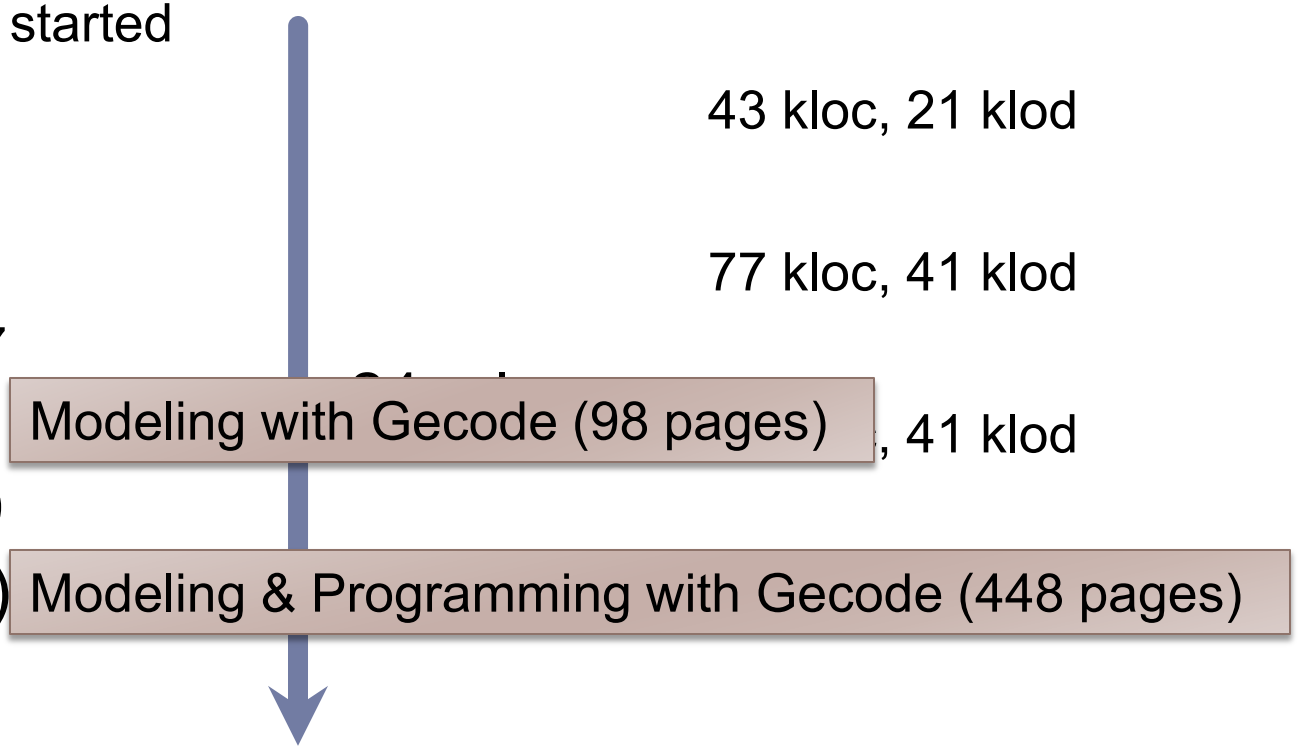
81 kloc, 41 klod

126 kloc, 52 klod

- … 4.0.0 in 2012

History: Tutorial Documentation

- 2002
  - development started
- 1.0.0 — 43 kloc, 21 klod
  - Dec 6, 2005
- 2.0.0 — 77 kloc, 41 klod
  - Nov 14, 2007
- 3.0.0 — , 41 klod
  - Mar 13, 2009
- 3.5.0 (current)
  - Feb 1, 2011
- … 4.0.0 in 2012



People

- Core team
  - Christian Schulte — KTH – Royal Institute of Technology, Sweden
  - Guido Tack — K.U. Leuven, Belgium
  - Mikael Z. Lagerkvist — KTH – Royal Institute of Technology, Sweden

- Code
  - contributions: David Rijsman, Denys Duchier, Filip Konvicka, Gabor Szokoli, Gregory Crosswhite, Håkan Kjellerstrand, Patrick Pekczynski, Raphael Reischuk, Tias Guns.
  - fixes: Alexander Samoilov, David Rijsman, Geoffrey Chu, Grégoire Dooms, Gustavo Gutierrez, Olof Sivertsson.
- Documentation
  - Seyed Hosein Attarzadeh Niaki, Vincent Barichard, Felix Brandt, Markus Böhm, Roberto Castañeda, Gregory Crosswhite, Pierre Flener, Gustavo Gutierrez, Gabriel Hjort Blindell, Sverker Janson, Andreas Karlsson, Håkan Kjellerstrand, Chris Mears, Flutra Osmani, Dan Scott, Kish Shen.

Goals

- Research
  - architecture of constraint programming systems
  - propagation algorithms, search, modeling languages, ...
- Efficiency
  - competitive (winner MiniZinc challenges 2008-2010, all categories)
  - proving architecture right
- Education
  - state-of-the-art, free platform for teaching

Users

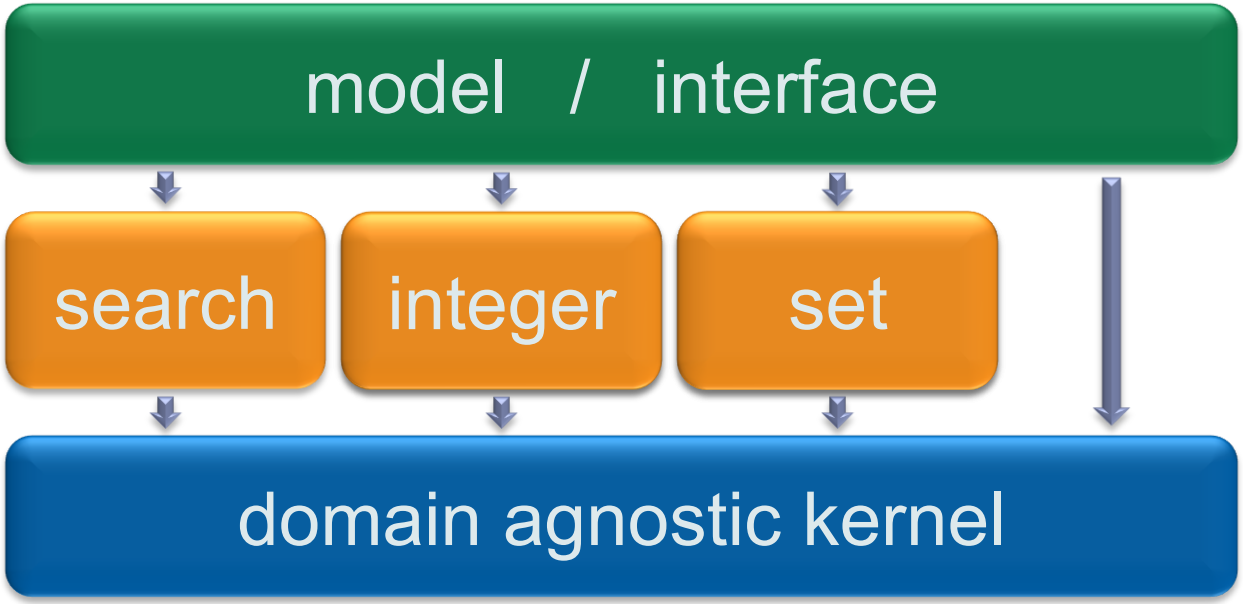
- Research
  - own papers
  - papers by others: experiments and comparison
  - Google scholar: some 500 references to Gecode
- Education: teaching
  - KTH, Uppsala U, U Freiburg, UC Louvain, Saarland U, American U Cairo, U Waterloo, U Javeriana-Cali, …
- Industry
  - several companies have integrated Gecode into products (part of hybrid solvers)

Deployment & Distribution

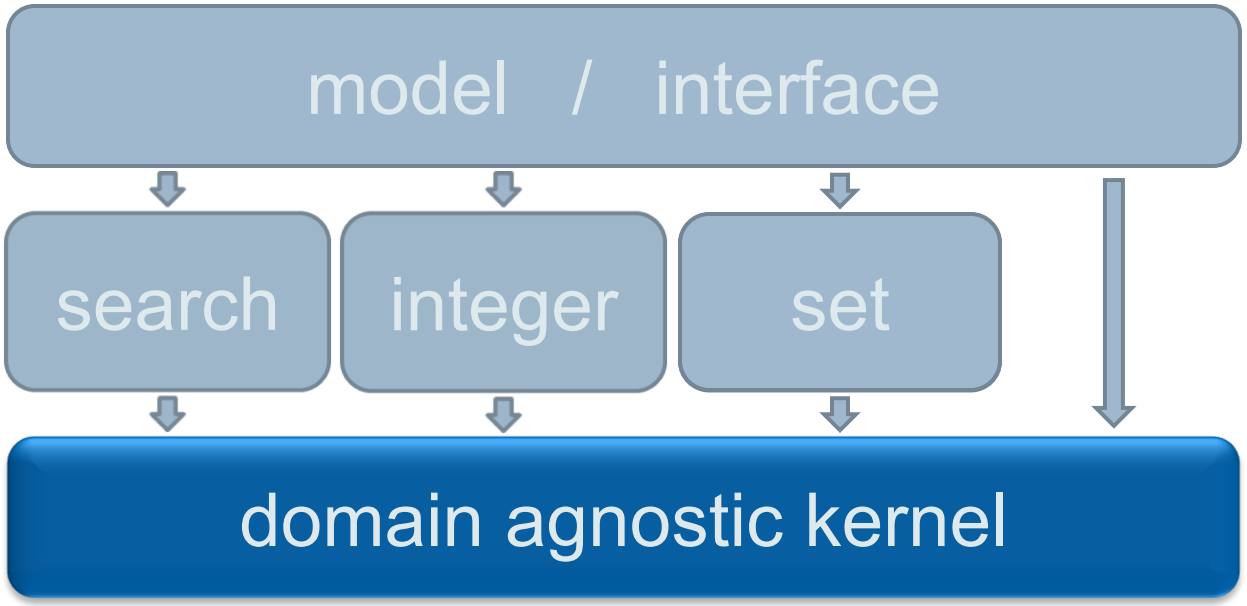
- Open source ≠ Linux only
  - Gecode is native citizen of: Linux, Mac, Windows
- High-quality
  - extensive test infrastructure (around 16% of code base)
- Downloads from Gecode webpage
  - software: between 25 to 125 per day
  - documentation: between 50 to 300 per day
- Included in
  - Debian, Ubuntu, FreeBSD

Modeling & Programming

Architecture

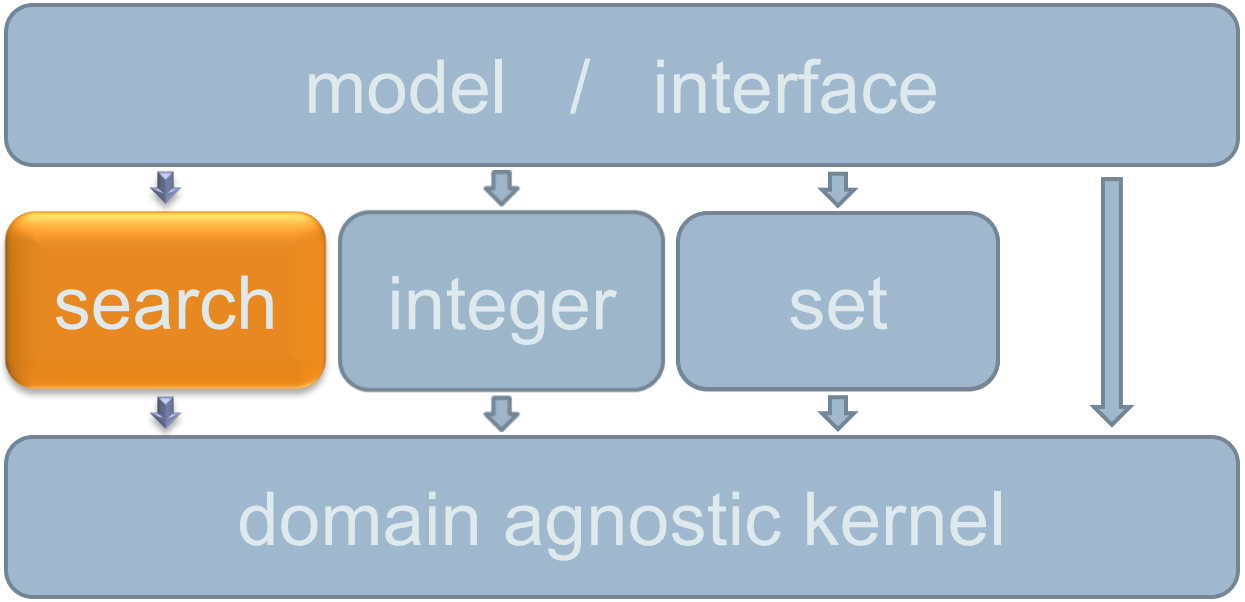


Architecture



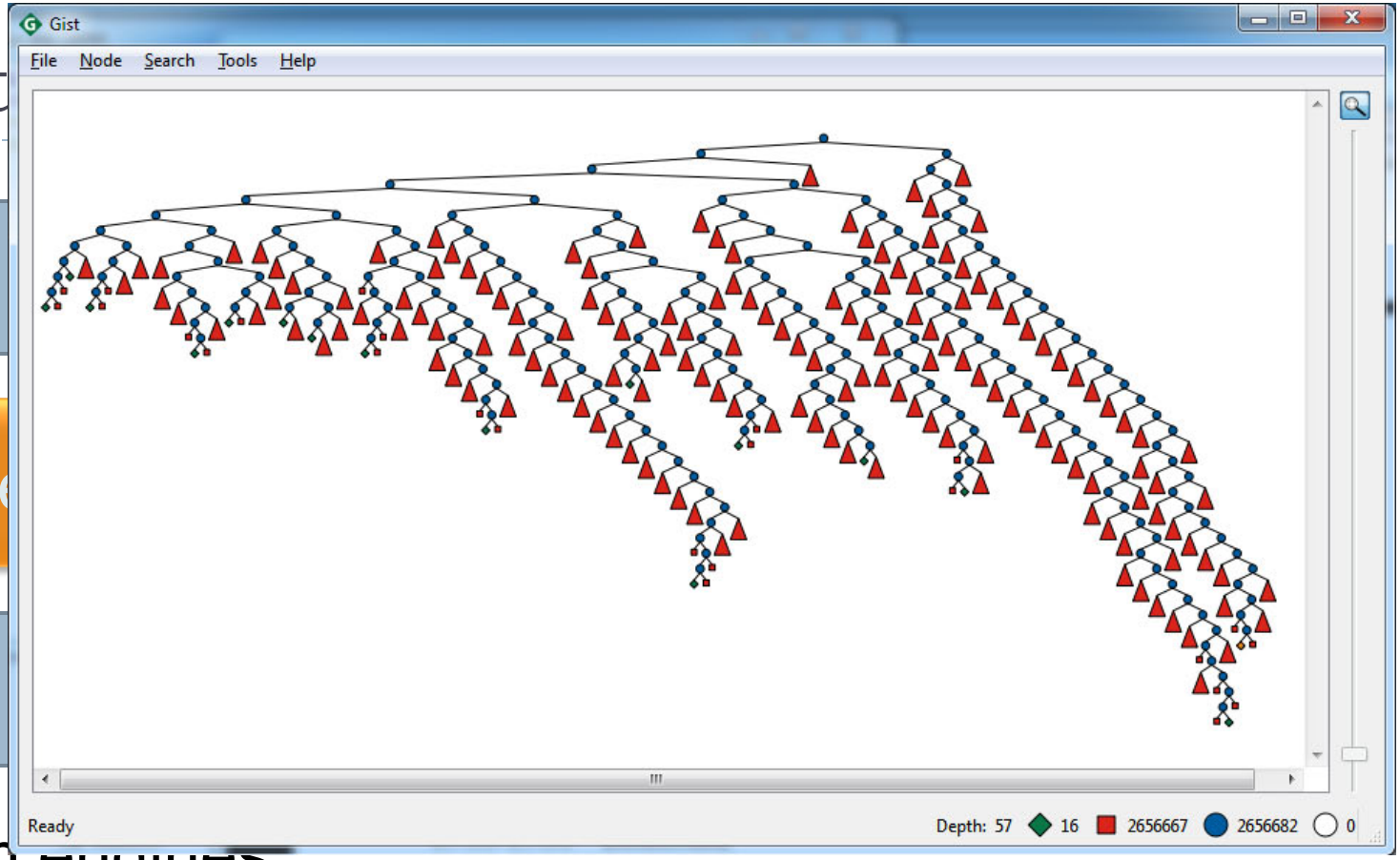
- propagation loop
- backtracking for search
- memory management

Architecture



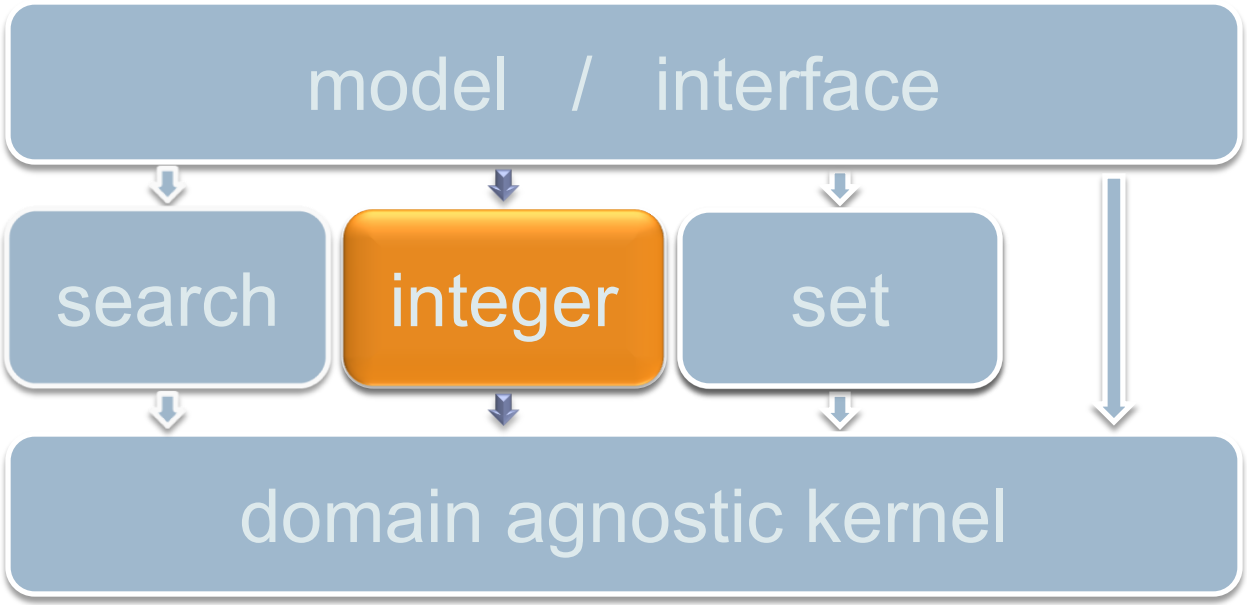
- search engines
  - depth-first (DFS) and branch-and-bound (BAB)
  - parallel search

Archit



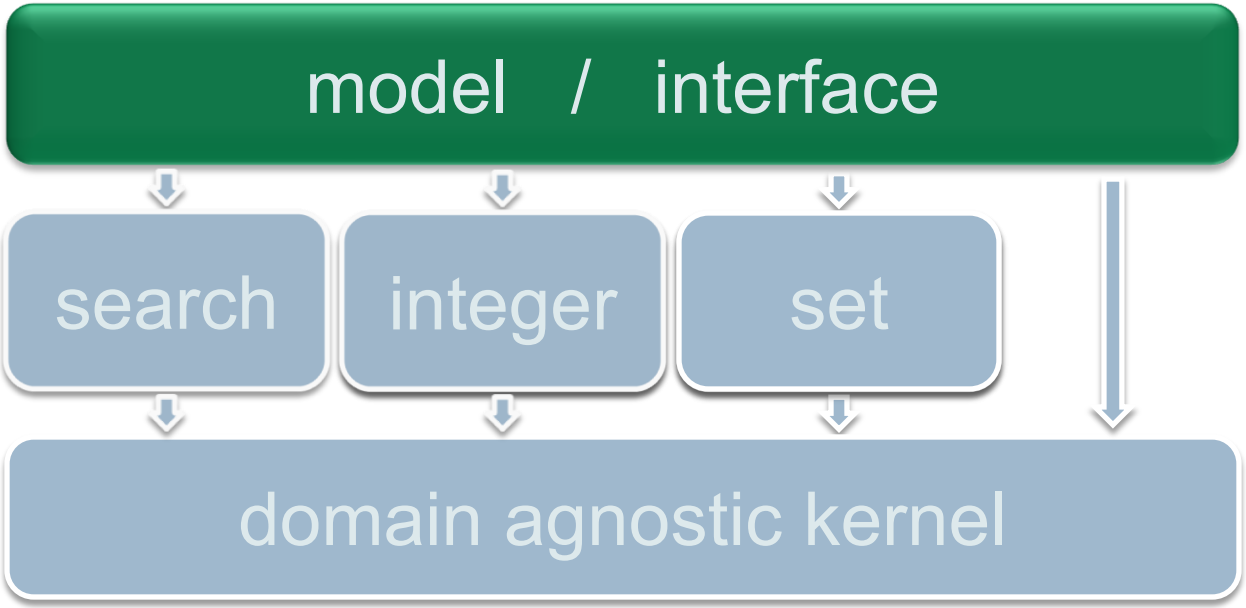
- search engines
  - depth-first (DFS) and branch-and-bound (BAB)
  - parallel search
- search tool: Gist (millions of nodes)

Architecture



- variables
- propagators (constraints)
- branchers (search heuristics)

Architecture



- direct C++ modeling or interfacing
- language interfaces: MiniZinc (see Guido's talk later), Java, JavaScript, Lisp, Ruby, Eclipse Prolog, …

Modeling (interfacing)

- Use modeling layer in C++
  - matrices, operators for arithmetical and logical expressions, …
- Use predefined
  - constraints
  - search heuristics and engines
- Documentation
  - getting started 30 pages
  - concepts and functionality 96 pages
  - case studies 76 pages

Modeling (interfacing)

- Constraint families
  - arithmetics, Boolean, ordering, ….
  - alldifferent, count (global cardinality, …), element, scheduling, table and regular, sorted, sequence, circuit, channel, bin-packing, lex, geometrical packing
- Families
  - different variants
  - different propagation strength

Programming

- Interfaces for programming
  - propagators (for constraints)
  - branchers (for search heuristics)
  - variables
  - search engines

Documentation	intro	advanced
propagators	40 pages	58 pages
branchers	22 pages	
variables		44 pages
search engines	12 pages	26 pages

Openness

Open Source

- MIT license
  - permits commercial, closed-source use
  - disclaims all liabilities (as far as possible)
- License motivation
  - public funding
  - focus on research
- Not a reason
  - attitude, politics, dogmatism

Open Architecture

- More than a license
  - **license** restricts what users **may do**
  - **code and documentation** restrict what users **can do**
- Modular, structured, documented, readable
  - complete tutorial and reference documentation
  - ideas based on scientific publications
- Equal rights: clients are first-class citizens
  - you can do what we can do: APIs
  - you can know what we know: documentation
  - on every level of abstraction

Open Development

- We encourage contributions
  - direct, small contributions
    → we take over maintenance and distribution
  - larger modules on top of Gecode
    → you maintain the code, we distribute it

- Prerequisites
  - MIT license
  - compiles and runs on platforms we support

Summary

- Open source libraries require open architecture
  - users need good code to build on
- Open architecture promotes equality
  - client code is first-class citizen
  - encourages code contributions
- Open development fosters research
  - collaboration
  - experiments are reproducible