Nemiver:
A GNOME Debugger

Dodji Seketeli
dodji@redhat.com

Presentation map

- Overview
- A bit of history
- Features
- Architecture
- Future ?
- Questions

Overview

- Debugger for C/C++ programs
- Free time powered volunteered
- GNOME Community
- Simple to use
  - Integrated in GNOME
- Common and simple use cases first
- Reasonable resources usage

History

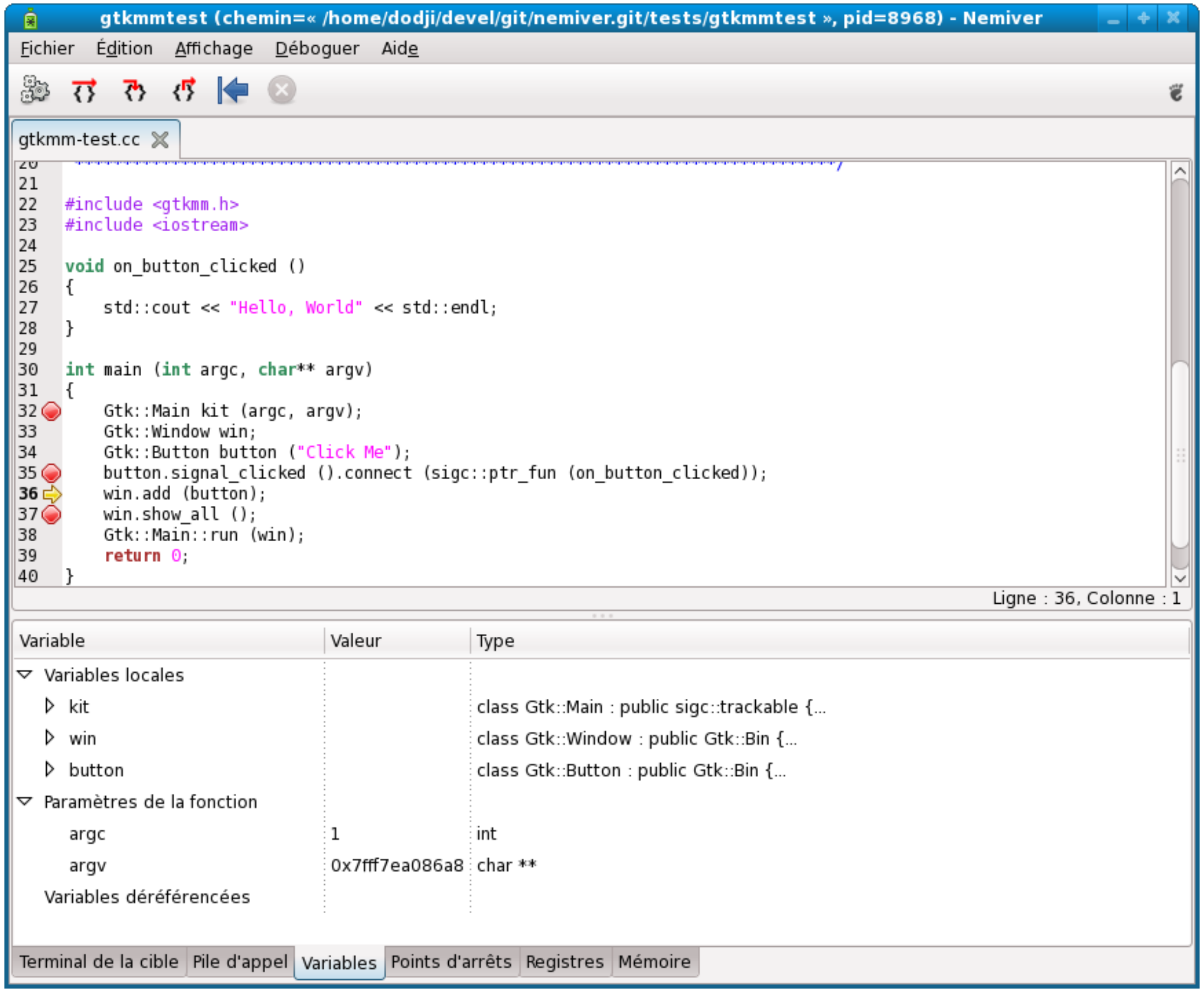
- Got aware of the need in 2006
  - Endless GDB questions on IRC
  - Ex Colleagues shouting against GDB
  - DDD being ugly and counter intuitive for simple tasks.
  - ==> abuse of interpreted languages
- Started coding during the summer 2006
- History still going ...

Features

- Breakpoints setting
- Variables inspection
- Call stack inspection
- Memory inspection
- Context saving
- Integrated with GNOME

redhat

A pic of the beast


gtkmmtest (chemin=« /home/dodji/devel/git/nemiver.git/tests/gtkmmtest », pid=8968) - Nemiver
Fichier Édition Affichage Déboguer Aide
gtkmm-test.cc

```
#include <gtkmm.h>
#include <iostream>

void on_button_clicked ()
{
    std::cout << "Hello, World" << std::endl;
}

int main (int argc, char** argv)
{
    Gtk::Main kit (argc, argv);
    Gtk::Window win;
    Gtk::Button button ("Click Me");
    button.signal_clicked ().connect (sigc::ptr_fun (on_button_clicked));
    win.add (button);
    win.show_all ();
    Gtk::Main::run (win);
    return 0;
}
```

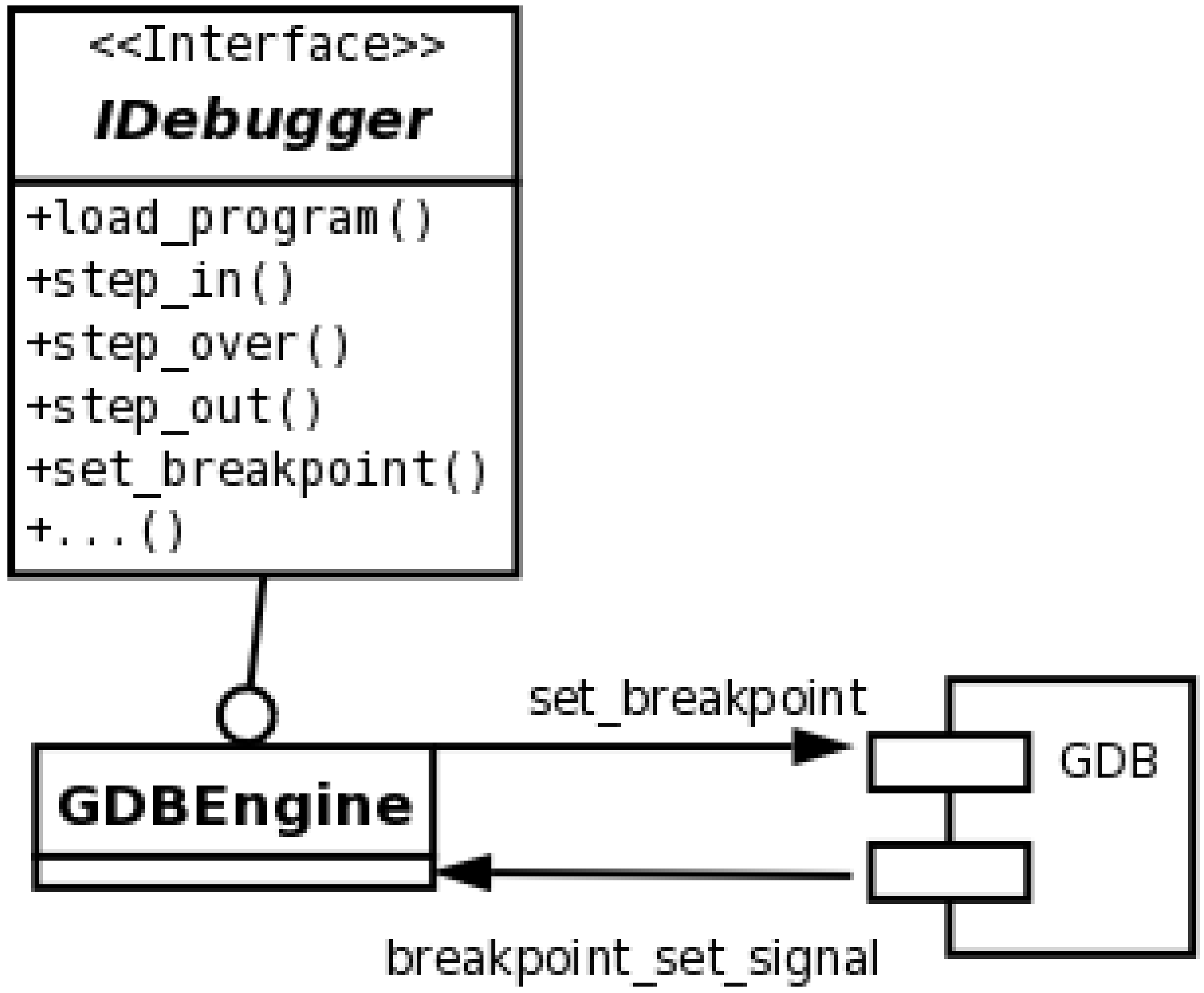
Ligne : 36, Colonne : 1

Variable	Valeur	Type
Variables locales		
kit		class Gtk::Main : public sigc::trackable {...
win		class Gtk::Window : public Gtk::Bin {...
button		class Gtk::Button : public Gtk::Bin {...
Paramètres de la fonction		
argc	1	int
argv	0x7fff7ea086a8	char **
Variables déréférencées		

Architecture (I)

- Nemiver uses GDB (surprise!)
  - But but, how to control GDB ?
    - No libgdb available
    - ==> GDB/MI
  - New interface: IDebugger
  - Event driven model
  - Implementation: Fork/Exec/Pipes
  - Glib event loop

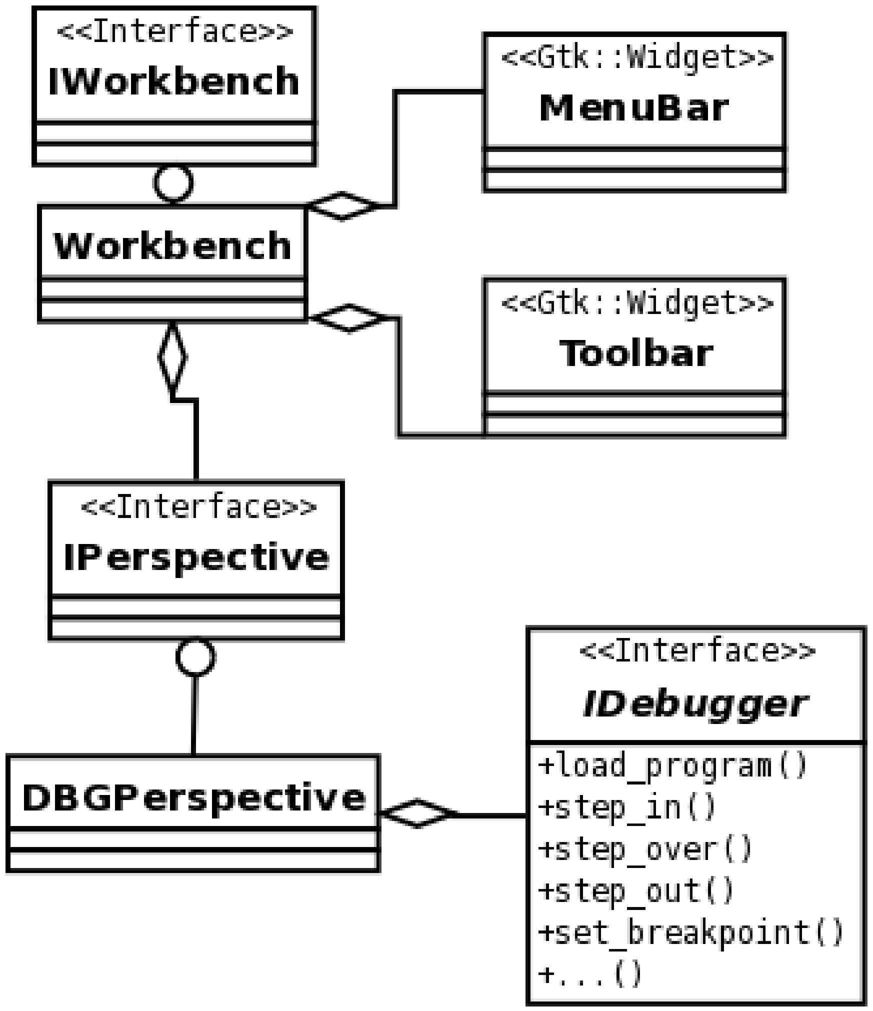
Architecture (I): Quick view



Architecture (II): Visible stuff

- Interfaces/implementations separation
- IWorkbench
  - Main window container
  - Tool and menu bar
  - perspective
- IPerspective
  - Widges collaborating for a task
    - IDBGPerspective
      - Uses IDebugger

Architecture (II): Big picture



Perspectives

- Cover more use cases (obviously)
- Scripting (Javascript ? Ruby ? Anyone ?)
  - More complex use cases
  - Cmd line interface on top of it ?
- Direct access to ELF binaries ?
  - Portability issue ?
  - More power
- External debugging library ?

Questions?

dodji@redhat.com