Qt Development Tools

GammaRay

Volker Krause
volker.krause@kdab.com
@VolkerKrause

gdb is not enough

- Single-stepping through a scene graph?
- Debugging graphical glitches?
- Following asynchronous API?
- Keep a state chart in mind while looking at its generated code?

printf isn't much better

- Generates huge amount of data
- No way of finding higher-level correlations
- Needs code changes

We want more!

- Debugger should “understand” the frameworks we work with
- Visualize state on the same conceptual level

→ GammaRay

QObject

- QObject has runtime introspection support
  - list and edit properties
  - list and call slots
  - list and monitor signals
  - list existing signal/slot connections

QObject

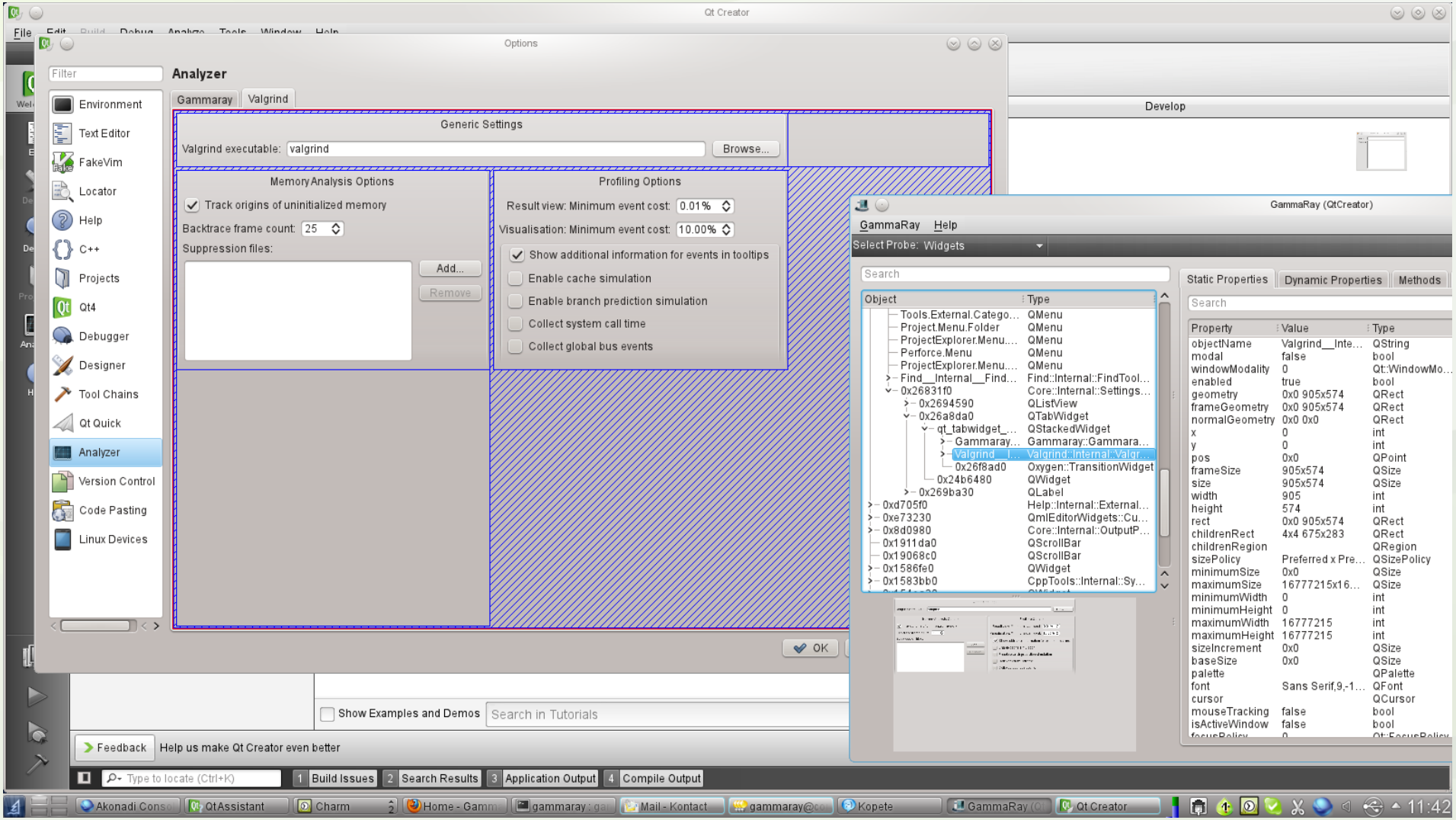
Demo

QLayout

- Powerful layouting framework for QWidget
  - box, form and grid layouts
  - margins and spacings
  - size policies
  - size hints
  - layout nesting
  - stretch factors

QLayout

Qt Developer Days 2013



QPainter

- Comprehensive 2D painting API
  - various geometric drawing methods
  - countless fill and outline options
  - coordinate system transformations
  - often performance sensitive

QPainter

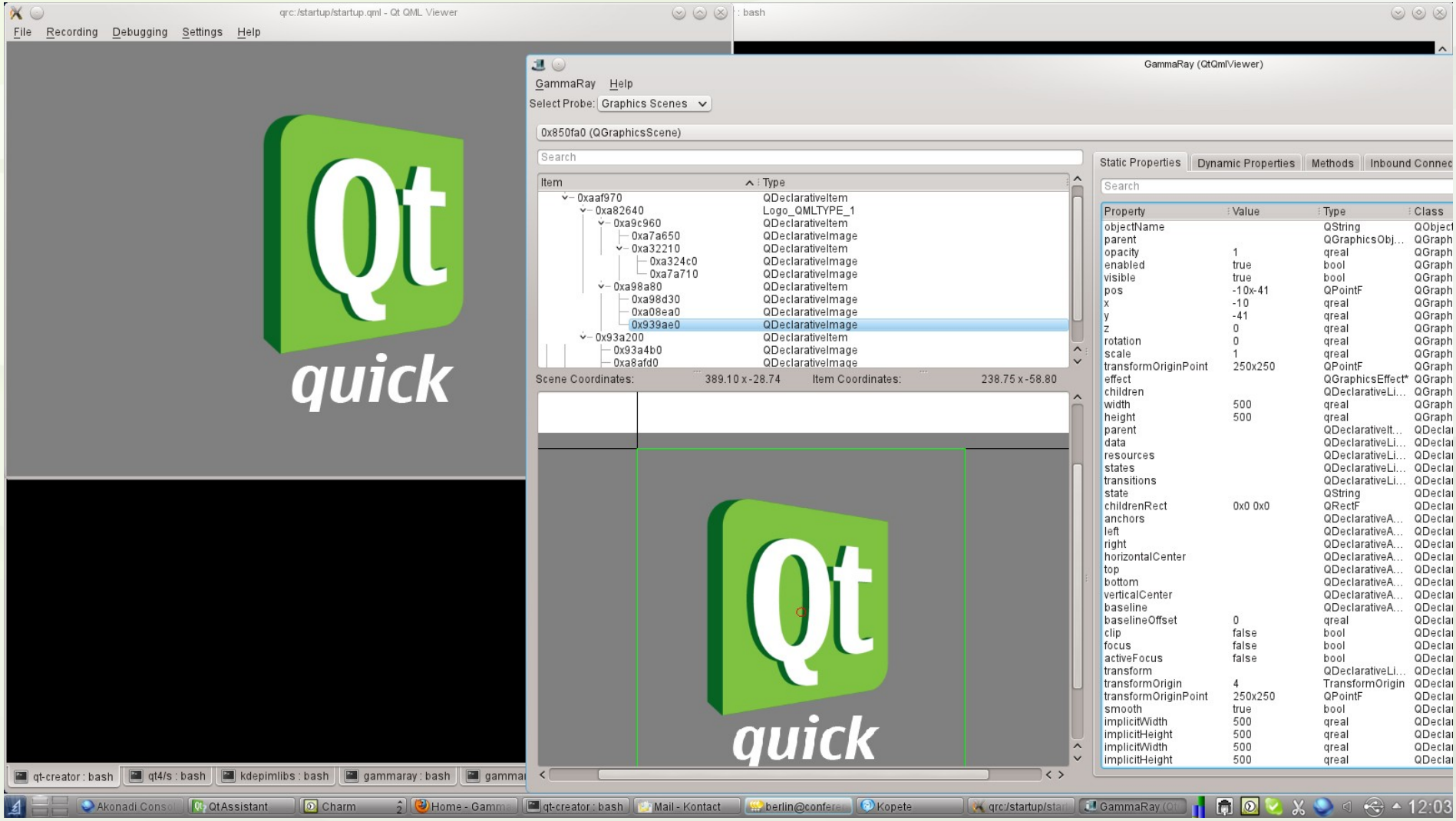
Demo

QGraphicsView

- 2D visualization framework
  - scene graph with BSP index tree
  - tree of graphics items with nested transformations
  - graphics items can have complex shapes
  - level of detail support
  - ...

QGraphicsView

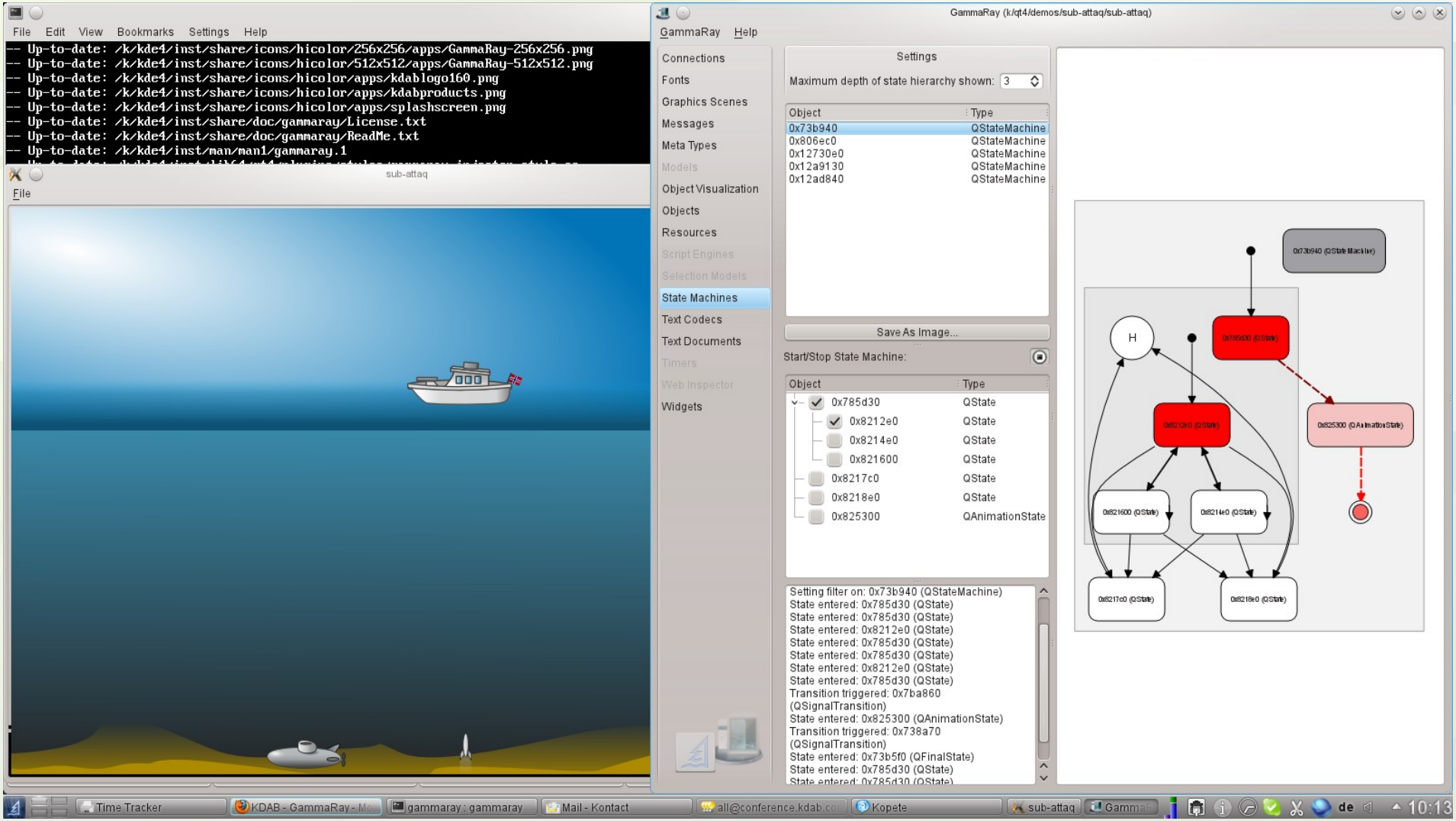
Qt Developer Days 2013



QStateMachine

- State machine framework
  - nested states
  - parallel states
  - history states
- Implementation is linear code

State Machines



And more!

- QTextDocument inspector
- (Proxy) model inspector
- Resource browser
- QStyle debugger
- Web inspector
- QScript debugger
- ...

Where do I get it?

- http://www.kdab.com/gammaray
- Free Software (GPL)
- Code is on Github
- Works on Linux, Mac, Windows, QNX/BB10