Tutored Project
Estate agency management

29 / 05 / 2012

Gilles Coulais <gilles.coulais@etu.u-bordeaux1.fr>

Sébastien Piquemal <sebastien.piquemal@etu.u-bordeaux1.fr>

Icham Sirat <icham.sirat@etu.u-bordeaux1.fr>

Plan

➢ Let's start the project

➢ Design: ACSI, UML

➢ Development: Database, classes, tools

I) Start of project

Two sets of technologies tested:

➢ PHP, Html / CSS
➢ Qt / C++

PHP

Designed to build dynamic websites

Works primarily with HTML (Hypertext Markup Language)

PHP - Advantages

➢ Html / Php code integration
➢ Easy hosting
➢ Easy to learn

PHP - Drawbacks

Main drawbacks from our point of view

> ➢ Mixed code impacts maintenance and readility

> ➢ Naming conventions

Qt/C++

Our choice

➢ A step forward for us
➢ A cross platform environment

Database

PostgreSQL

Management system of relational databases

Other tools

A free software

Distributed Version Control System

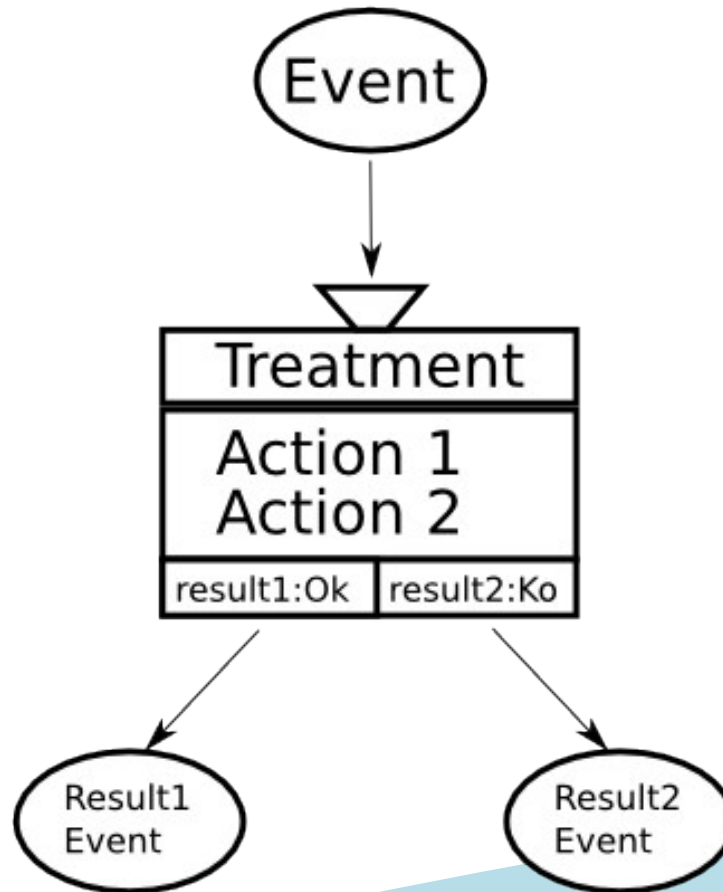# *II) Design*

- ➢ ACSI
- ➢ GUI Prototyping
- ➢ UML

ACSI

Design of a chart actor flow (GAF):

Objectives :

➢ Identify internal and external actors of the system

➢ Identify trade flows between actors

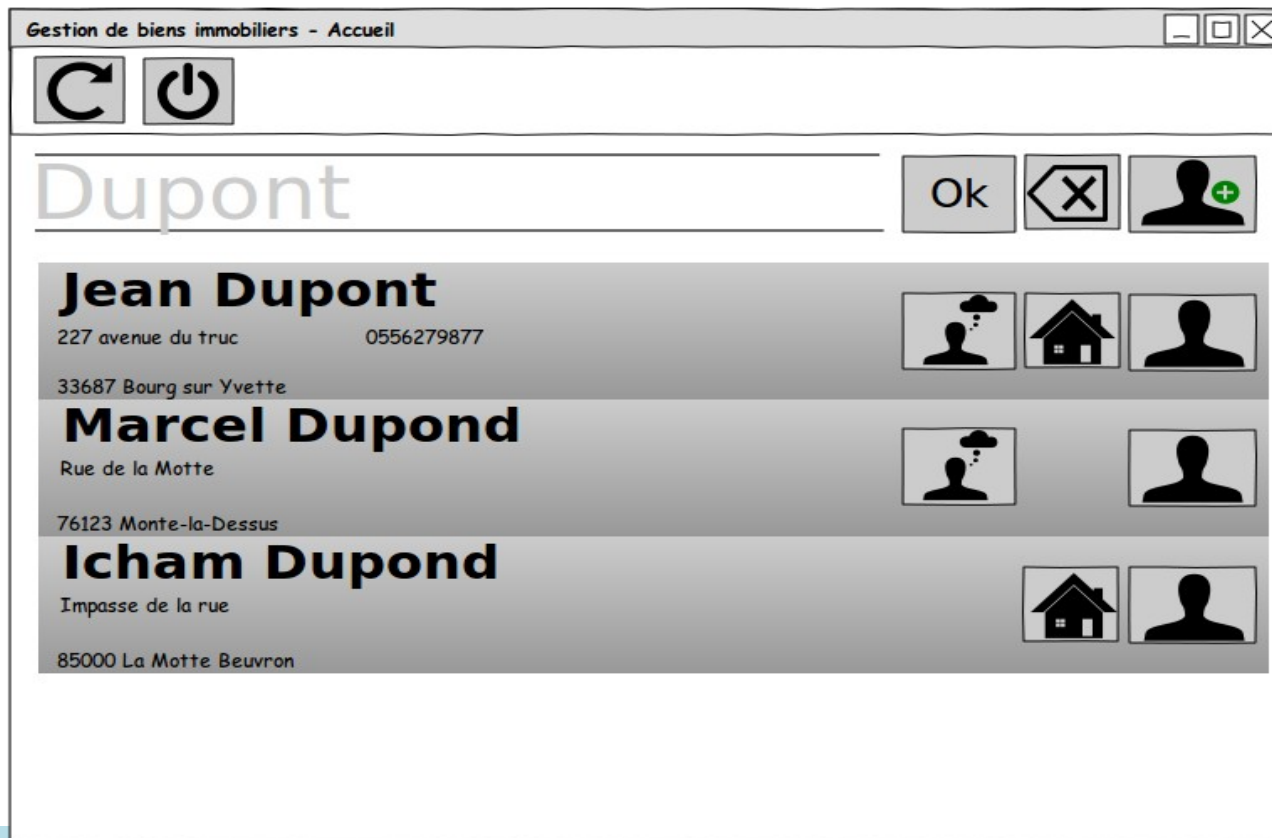➢ Flows between external actors outside the study area
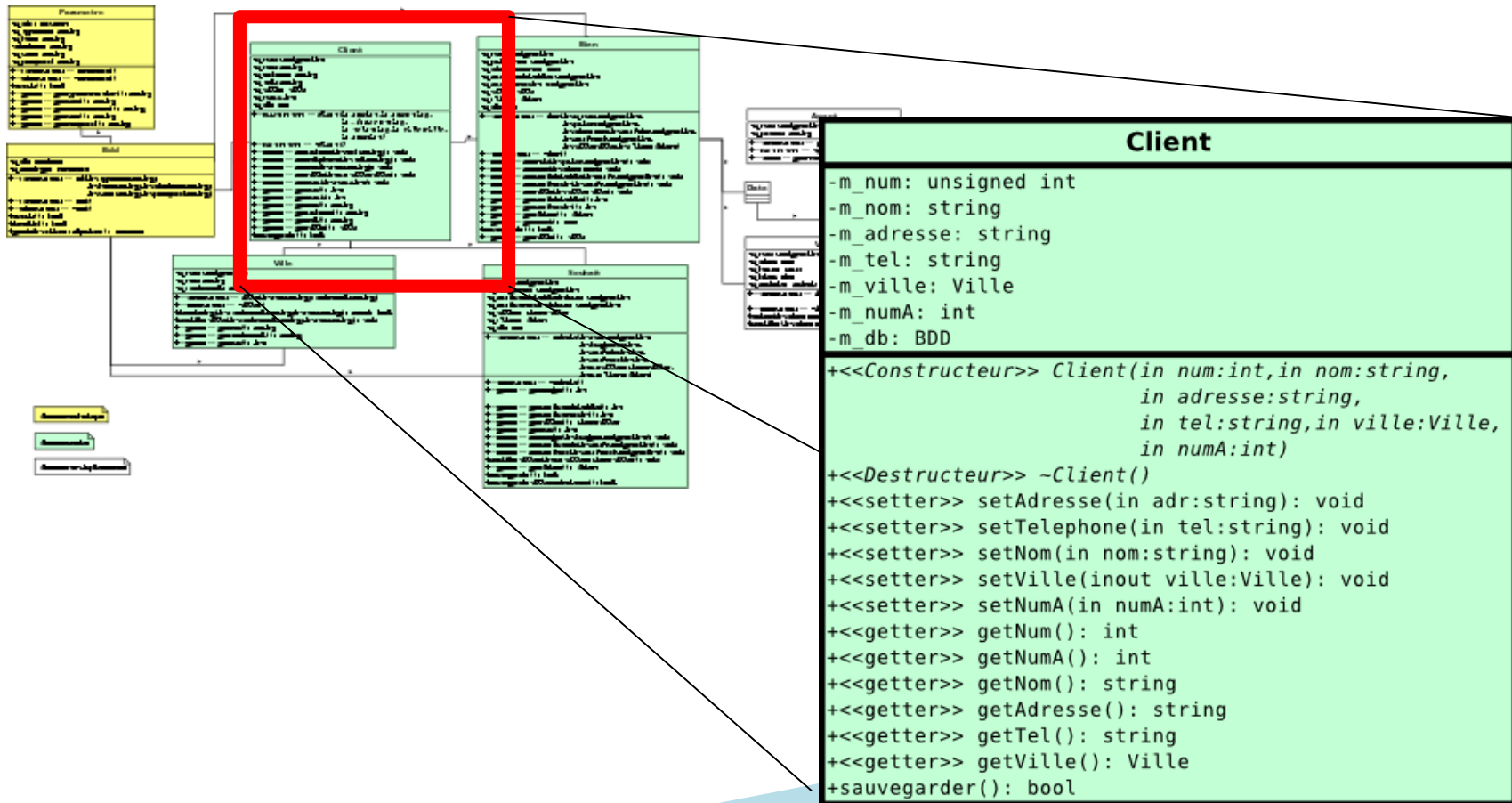
ACSI

MCT : Conceptual Model of Treatments
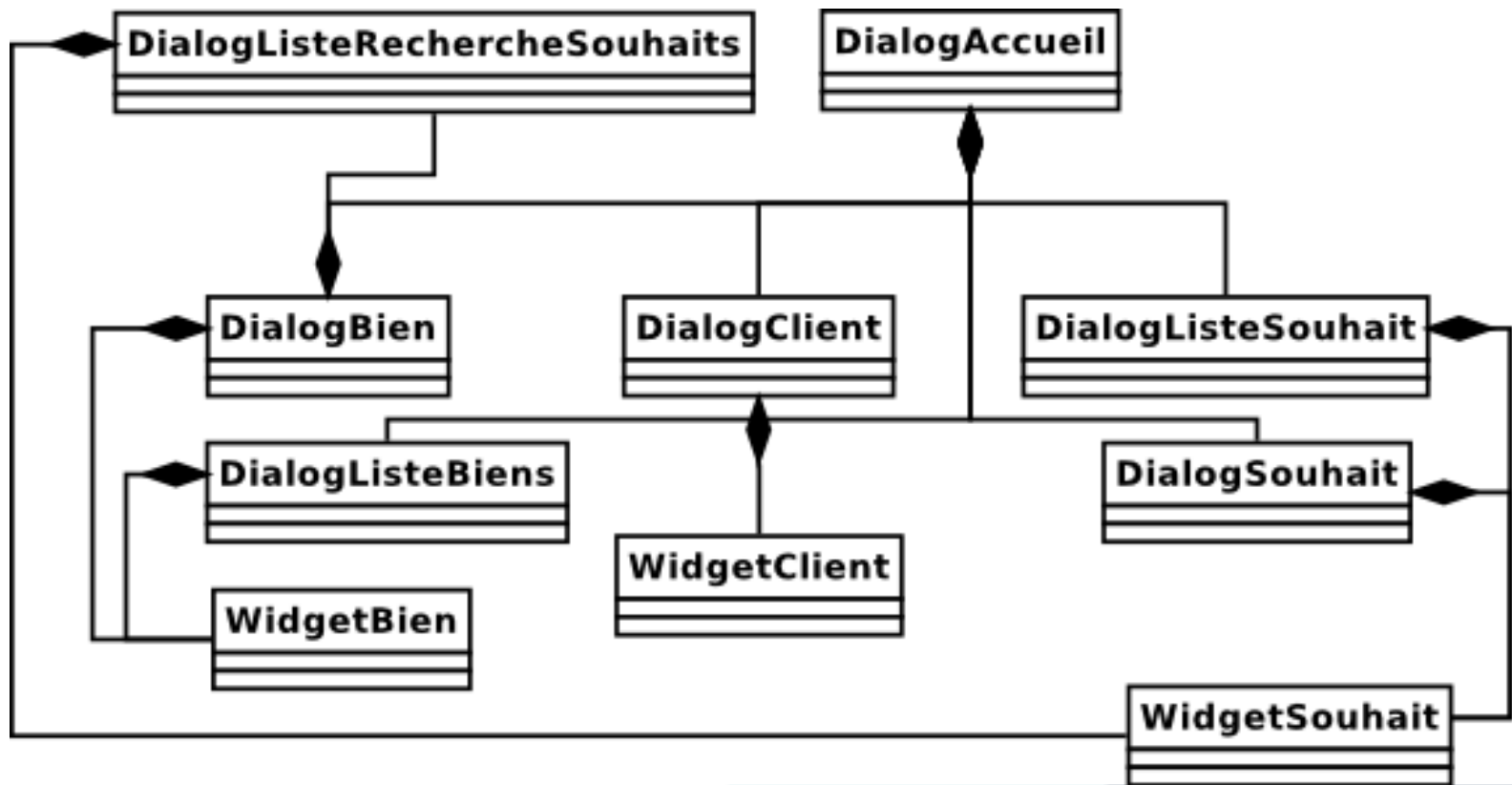
Pencil

Diagrams and GUI prototyping

UML

Engine Classes : software architecture

Client

-m_num: unsigned int
-m_nom: string
-m_adresse: string
-m_tel: string
-m_ville: Ville
-m_numA: int
-m_db: BDD

+<<Constructeur>> Client(in num:int,in nom:string,
                         in adresse:string,
                         in tel:string,in ville:Ville,
                         in numA:int)
+<<Destructeur>> ~Client()
+<<setter>> setAdresse(in adr:string): void
+<<setter>> setTelephone(in tel:string): void
+<<setter>> setNom(in nom:string): void
+<<setter>> setVille(inout ville:Ville): void
+<<setter>> setNumA(in numA:int): void
+<<getter>> getNum(): int
+<<getter>> getNumA(): int
+<<getter>> getNom(): string
+<<getter>> getAdresse(): string
+<<getter>> getTel(): string
+<<getter>> getVille(): Ville
+sauvegarder(): bool

UML
Graphics Classes

III) Development

- ➢ Database
- ➢ Engine and User Interface
- ➢ Some examples

Database

➢ Installation of the PostGreSQL Engine
➢ Creation of the database
    ➢ Database structure
    ➢ Test Data
➢ PgAdmin III
  ➢ graphical administration
  ➢ querying tool

Qt

➢ A cross platform integrated development environment
➢ A set of tools and modules built around the C++ language

Qt Creator

➢ C++ code editor

➢ Integrated UI designer

➢ Project and build management tools

➢ Seamless integration with gdb and CDB debuggers

➢ Support for version control systems

QT Designer

➢ GUI layout and forms builder
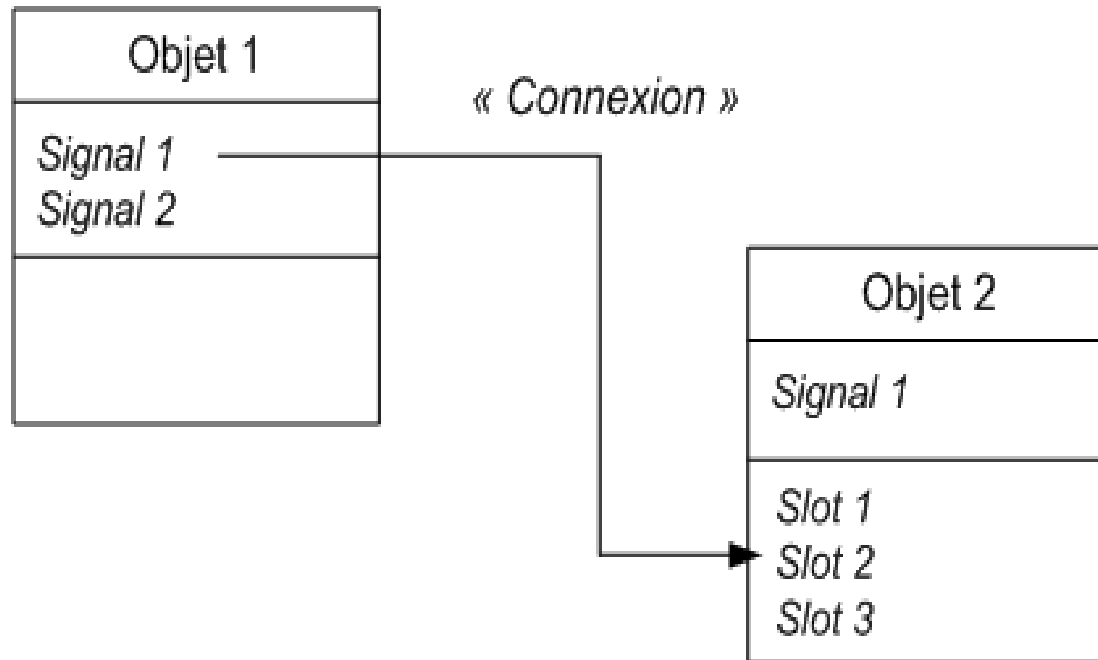➢ Generates C++ code from the interface prototypes

Qt signals and slots

Object oriented development

Object
Properties
Methods
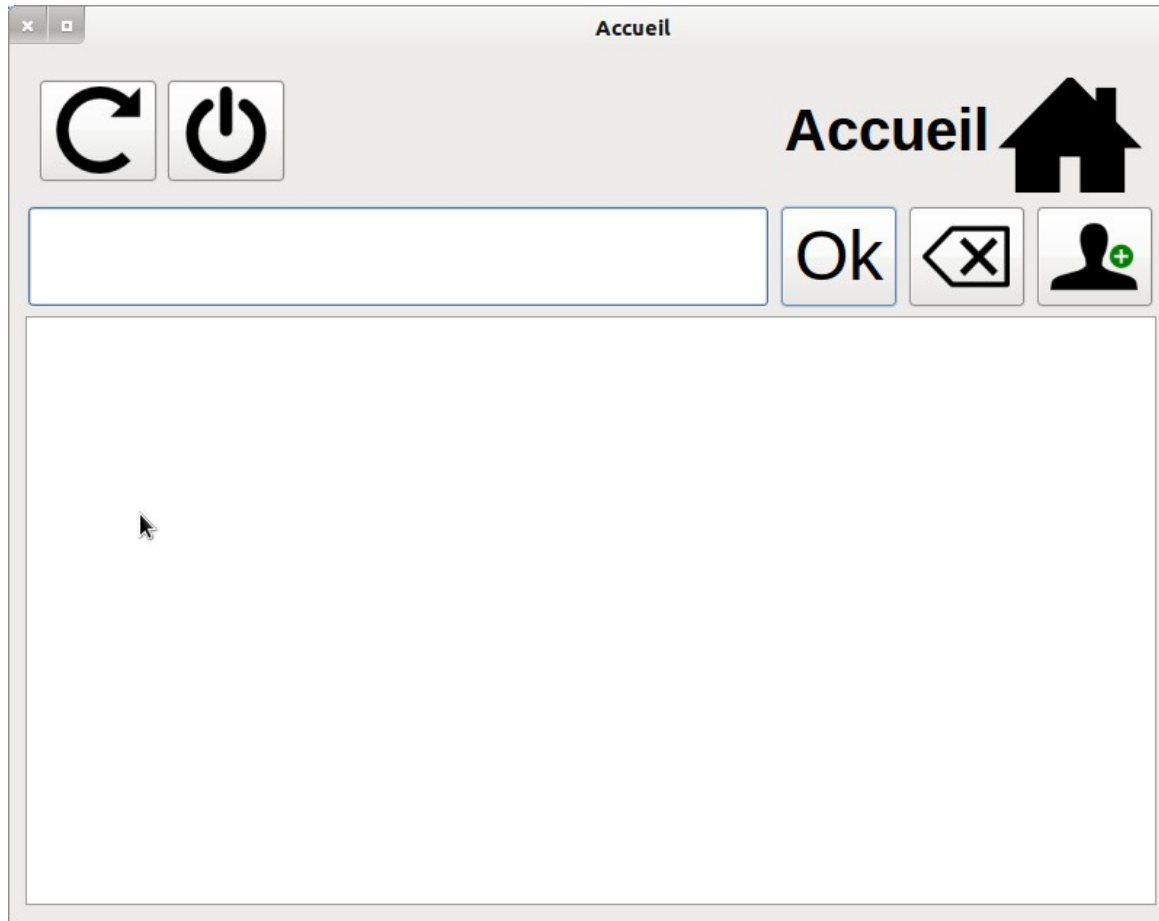
Object
Properties
Methods
Signals
Slots
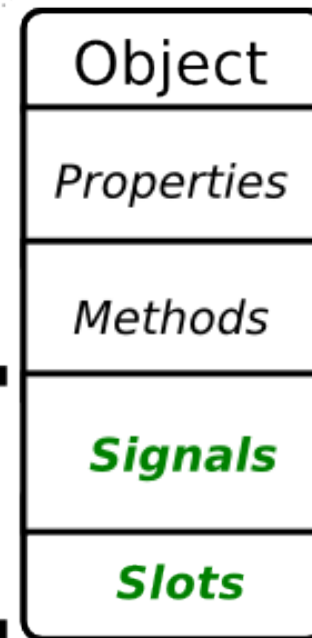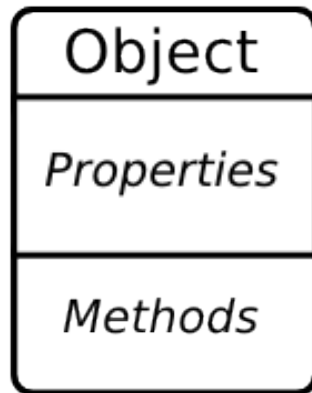
Object oriented development
with Qt

SIGNALS and SLOTS operation

An example

Homepage of our software

An example

When clicked, the application is shutdown.


```
m_boutonQuitter
+clicked()
+pressed()
```

QPushButton

```
Application
+quit()
+closeAllWindows()
```

QApplication
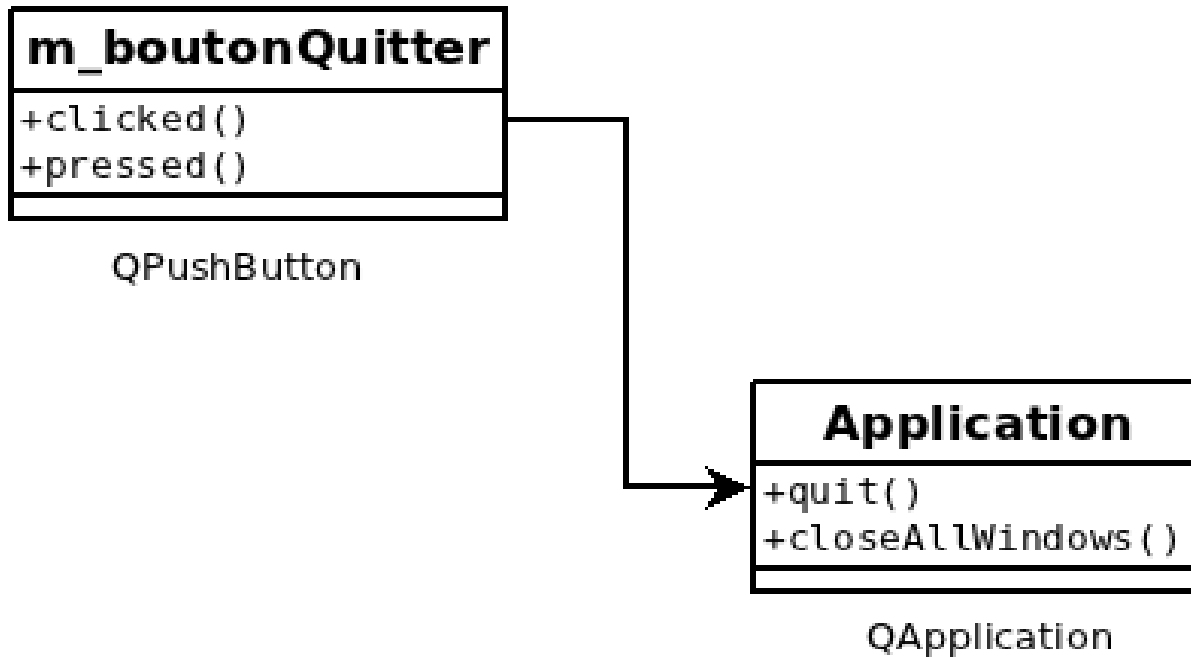
An example

The corresponding code

A single line of code is enough

```
QObject::connect(m_menu.pushButton_2, SIGNAL(clicked()), qApp, SLOT(quit()));
```

Conclusion

➢ A rich project
➢ A clean and functional software
➢ A professional approach

Questions ?