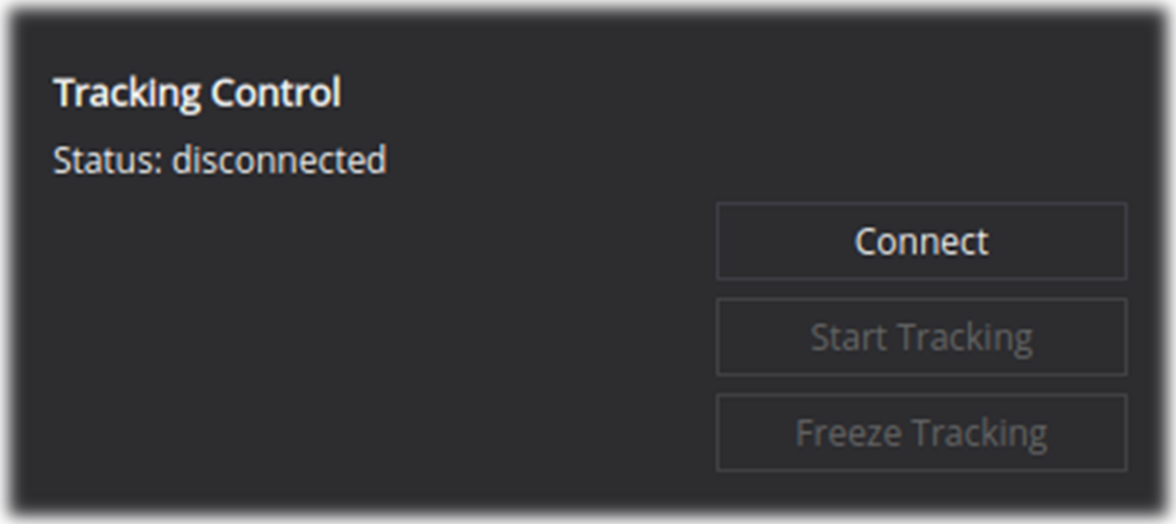
Controlling UI elements without spaghetti code

... by using the Qt State Machine Framework

Community Day Talk - Matthias Eisenmann

The spaghetti code incident

On button clicked:
if X, do this
else, do that



The spaghetti code incident

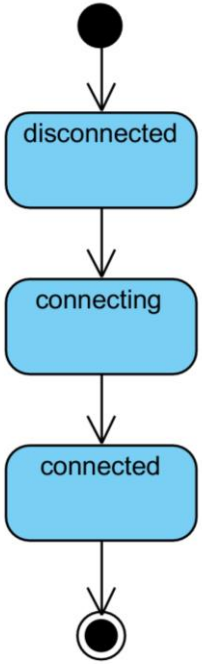
On button clicked:
if X, do this
else, if Y and Z
        if A and not B, do that
        else, do that other thing
else, do something great

Issues

- if-statements → state is implicit
- Control flow and business logic jumbled together
- Hard to understand, maintain and extend

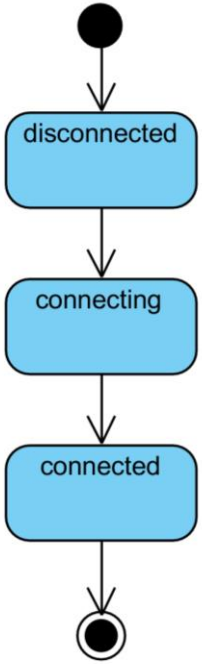
Image source: https://s3.amazonaws.com/lodcf/wp-content/uploads/2011/08/09141906/Spaghetti-Mess-Baby.jpg

State machines: What‘s in for you?



- Have your design and implementation speak the same language
- Cope with incremental complexity
- Write more robust code

State machine set-up recipe



- Create state machine

```
QStateMachine *machine = new QStateMachine();
```

- Create states

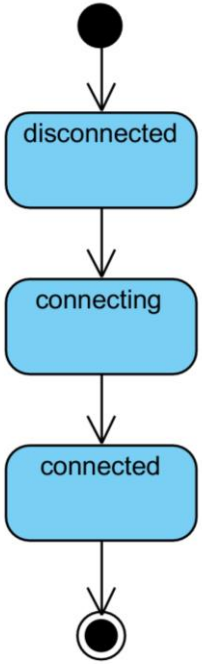
```
QState *disconnectedState = new QState();
QState *connectingState = new QState();
QState *connectedState = new QState();
```

- Create transitions between states and hook up to signals

```
disconnectedState->addTransition(connectionButton, &QPushButton::clicked, connectingState);
connectingState->addTransition(connectingState, &QState::finished, connectedState);

connect(connectingState, &QState::entered, device, &mitk::TrackingDevice::OpenConnection);
```

State machine set-up recipe



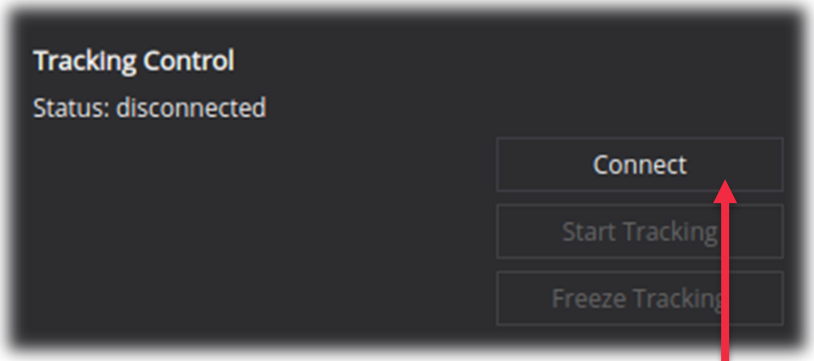
- Set initial state

```
machine->addState(disconnectedState);
machine->addState(connectingState);
machine->addState(connectedState);
machine->setInitialState(disconnectedState);
```

- Start the machine

```
machine->start();
```

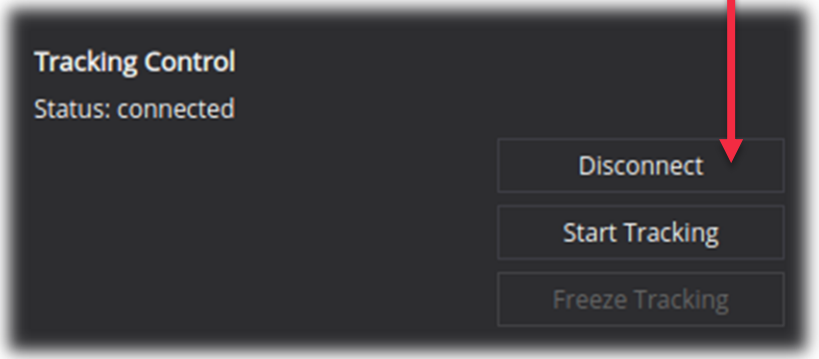
Configure properties of UI elements




```
disconnectedState->assignProperty(connectionButton, "text", "Connect");
disconnectedState->assignProperty(connectionButton, "enabled", true);

connectingState->assignProperty(connectionButton, "enabled", false);

connectedState->assignProperty(connectionButton, "text", "Disconnect");
connectedState->assignProperty(connectionButton, "enabled", true);
```



Some more features

- Grouped states
- History states
- Parallel states

References

- The Qt State Machine Framework:
  - http://doc.qt.io/qt-5/statemachine-api.html
  - https://de.slideshare.net/qtbynokia/qt-state-machine-framework