

Использование автоматного
программирования для инкапсуляции
сложного сигнально-слотового
взаимодействия объектов в библиотеке Qt



Янушковский Владимир (9в)

Научный руководитель: И.Р.Дединский

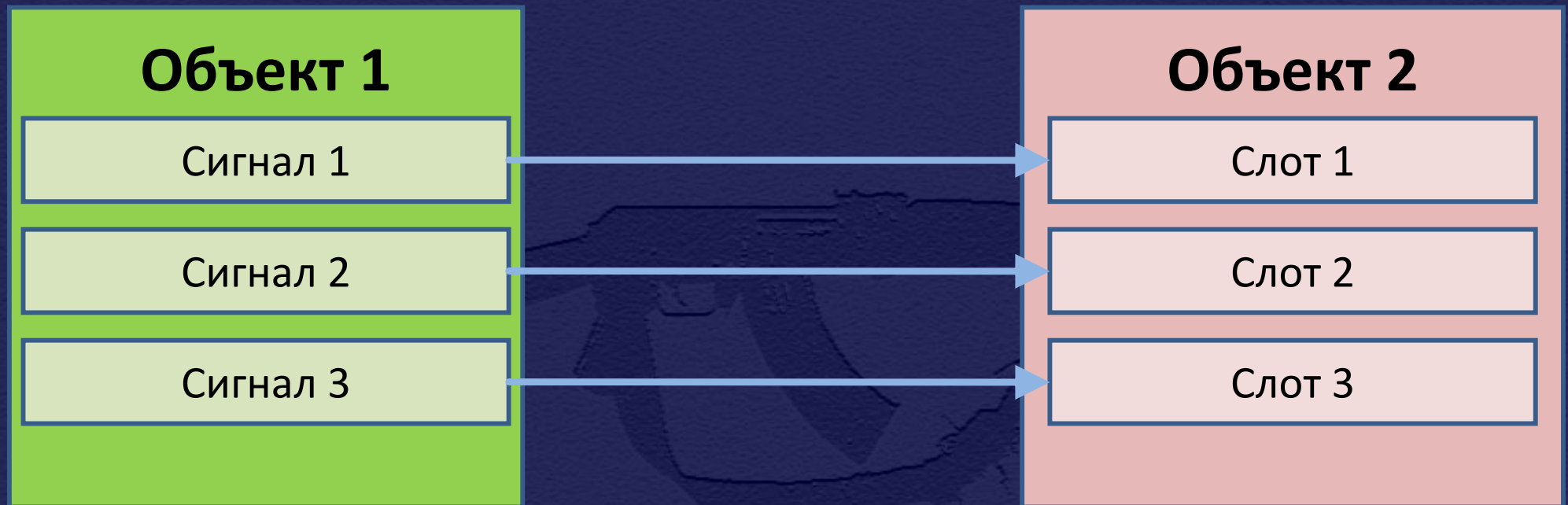
Цель работы:

- Применение автоматов для реализации сложных взаимодействий между объектами Qt, для которых соответствие между сигналами и слотами определяется текущим состоянием.

Задачи работы:

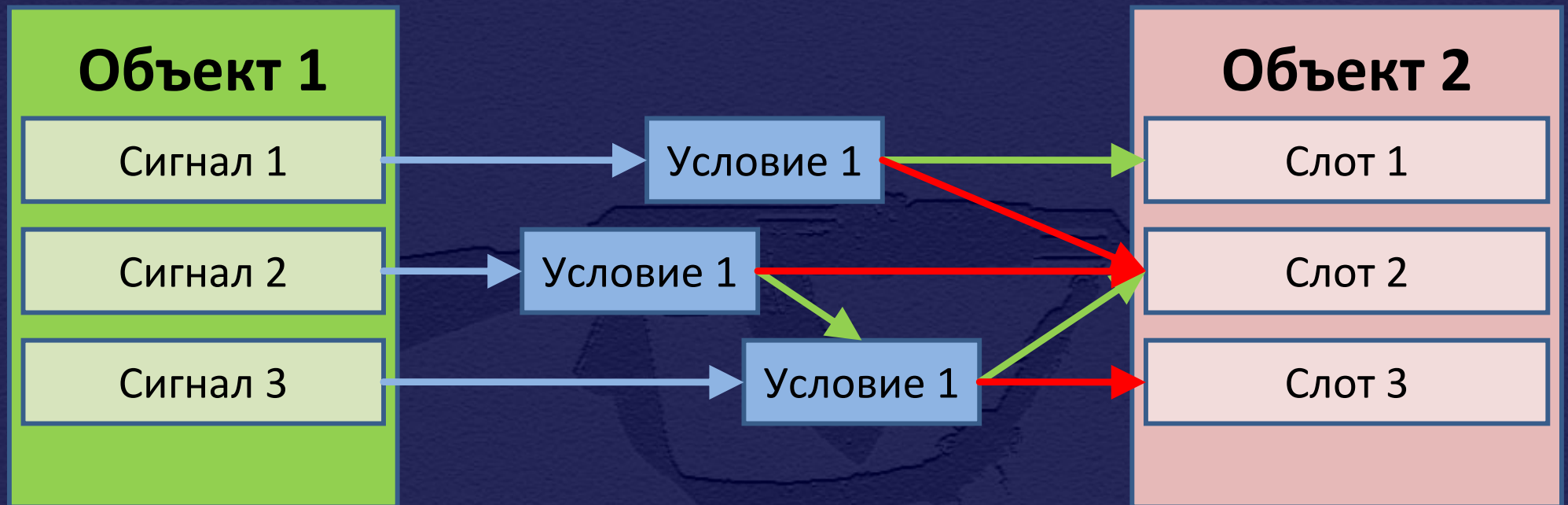
- Изучение автоматов, ознакомления со спецификой их работы.
- Изучение инструментов, облегчающих работу с ними.
- Применение автоматов для взаимодействий объектов в Qt.
- Написание простого Qt-приложения с применением автоматов.

Простая схема взаимодействия объектов



При такой схеме взаимодействия каждому сигналу однозначно ставится в соответствие некоторый слот, который будет вызван независимо от состояния объекта.

Сложная схема взаимодействия объектов



С этим случае слот, который вызывается данным сигналом определяется не только самим сигналом, но и состоянием объектов.

Стандартная реализация сложного взаимодействия и её проблемы

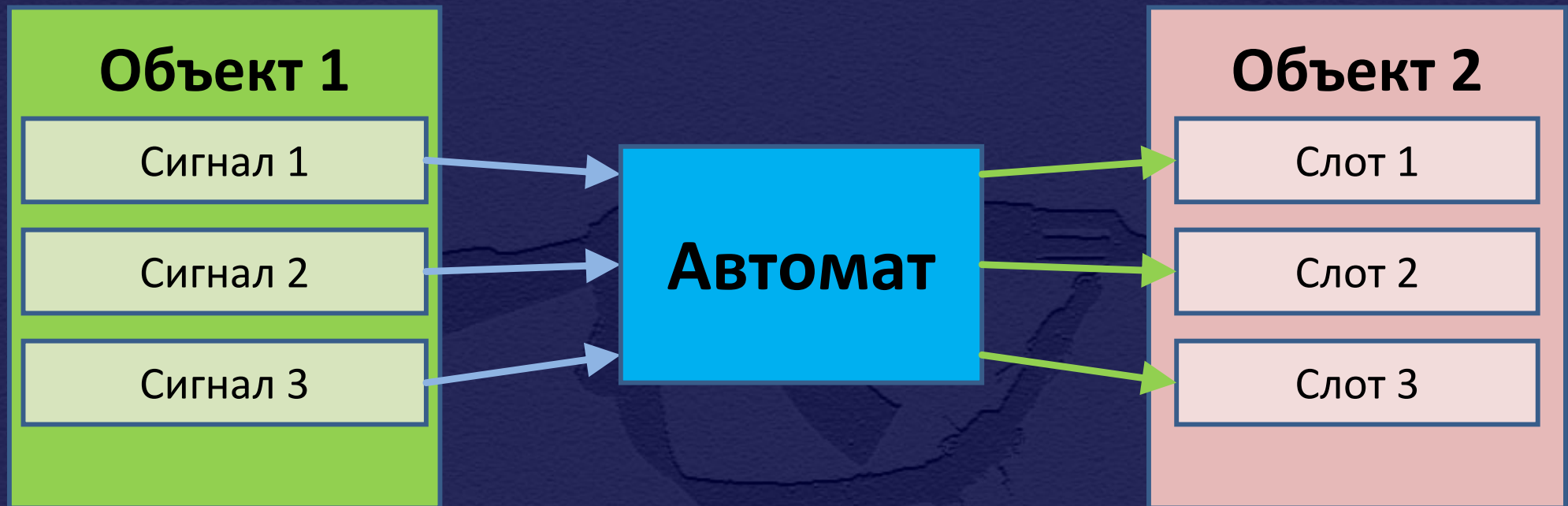
Стандартный способ реализации

- Для каждого сигнала пишется ряд операторов if и блоков switch, которые определяют вызываемый слот.

Проблемы данного способа

- При большом количестве условий, размер и сложность условных конструкций может достичь уровня, когда разобраться в них будет практически невозможно, а внести какие-то изменения - тем более.
- Условные конструкции засоряют определения самих сигналов и слотов, что приводит к ухудшению логики кода.

Автомат как возможное решение



Между объектами устанавливается автомат – объект или функция, которая инкапсулирует в себе выбор соответствия между сигналами и слотами.

Изменения в коде

Без автомата

```
void Object1::Method1 ()
{
    switch (state_)
    {
        case STATE_1:
            connect (...);
        case STATE_2:
            connect (...);
        case STATE_3:
            connect (...);
        case STATE_4:
            connect (...);
        case STATE_5:
            connect (...);
        case STATE_6:
            connect (...);
    }
    emit signal1 ();
}
```

С автоматом

```
void Object1::Method1 ()
{
    A0 (1);
    emit signal1 ();
}

//и все.
//Теперь условия внутри
//автомата.
```

Пользы данного подхода

- Улучшается логика кода, теперь объекты просто вызывают сигналы и выполняют слоты, все условия выделены в автомат.
- Облегчение внесения изменений в логику: теперь достаточно просто изменить автомат и не нужно трогать сигналы и слоты.
- Структура автомата проста и системна, что позволяет не писать код взаимодействия вручную, а сгенерировать его на основе схемы или другого ресурса, используя соответствующие инструменты.

Использование схемы переходов для задания автомата

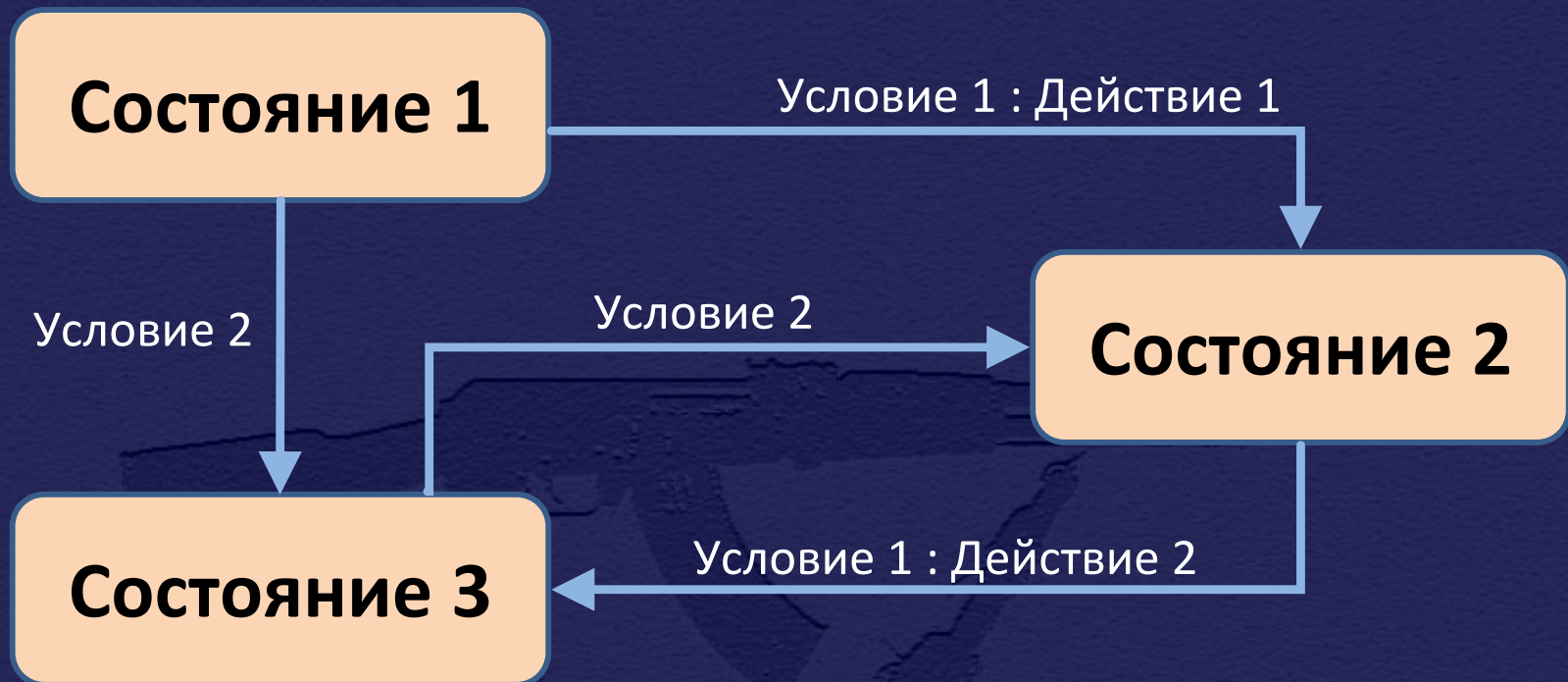


Схема переходов изображает состояния, в которых может находиться объект, переходы между ними, которые могут быть вызваны сигналами и слоты, которые могут вызываться при этих переходах. Существует ряд инструментов, которые способны переводить такие схемы в исходный код, например Visio2Switch, MetaAuto и др..

Выводы:

- Внедрение автомата позволяет улучшить логику сигнально-слотового взаимодействия объектов в библиотеке Qt, выделив в отдельную составляющую зависимость соответствия сигналов и слотов от состояния объектов. Также при таком подходе облегчается внесение изменений в код. И наконец, возможно задание автомата не ручным написанием кода, а, например, созданием схемы переходов состояний объектов.

Практические результаты работы:

- Реализовано приложение “Clocks”, которое демонстрирует применение автоматов для определения логики взаимодействия объектов в Qt.



Спасибо за внимание!