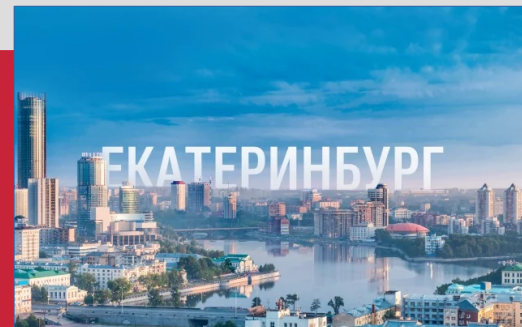


Сетевая технология программирования визуальных интерфейсов



Павел Васёв
Сектор визуализации
ИММ УрО РАН, г. Екатеринбург
<https://cv.imm.uran.ru>

Постановка задачи

- **Необходимо программировать интерфейс пользователя со сложной логикой. Интерфейс — нужен.**
- **При этом нет желания завязываться на какую-то очень конкретную технологию GUI.** Печален опыт, когда было много вложений и последующее устаревание. Delphi, Borland C++, MFC, Qt, QML.
- **Желательно также уметь программировать 3D графику.** Чувствуется что технология может быть общей для GUI и 3D.

Идея решения

- Интерфейс программируется в одной языковой среде.
- А выполняется — в другой среде в другом процессе.
- Взаимодействие организуется по сети.
- Вводятся модели по определению интерфейса.
- **Предметнее.** Программировать интерфейс (состав, логику) на языке Python, а изображать интерфейс - на Qt или Веб.

Похожие подходы:

- ЛОГОС-Платформа (Росатом)
- Yandex Divkit

Непохожие подходы:

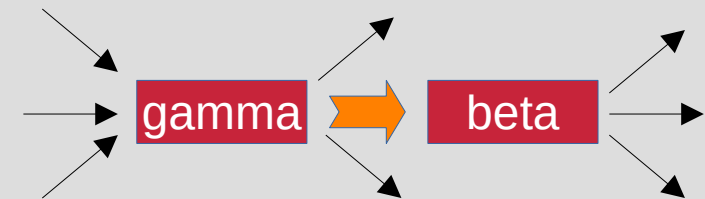
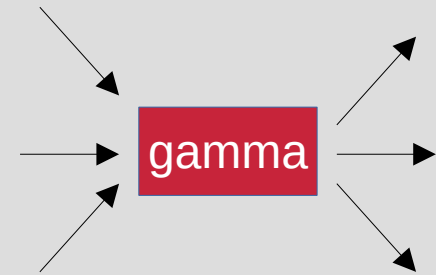
- X Window system
- GUI-библиотеки Python

Сущности

Исполнитель – процесс операционной системы (ОС), работающий на некотором оборудовании. Каждая компонента выполняется в некотором заданном исполнителе.

Компонента - это логический процесс в смысле Хоара. Над компонентами вводится отношение *родитель-дети**. Из компонент создаются интерфейсы, 3d сцены, и др.

- **Канал** — сущность для передачи данных. Каналы могут быть локальными (в текущем процессе) и сетевыми* (в некотором общем окружении, пространстве имён).
- **Реакция** – запуск некоторого кода при появлении сообщения в канале*.
- **Связь между двумя каналами** — специальный процесс, который производит передачу сообщений записываемых в первый канал во второй канал*.



Операции

Открытие канала:

`open_channel: channel_id → channel`

Запись сообщения в канал:

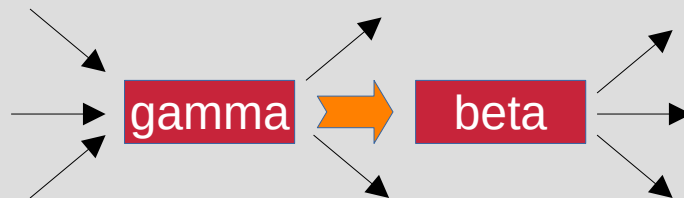
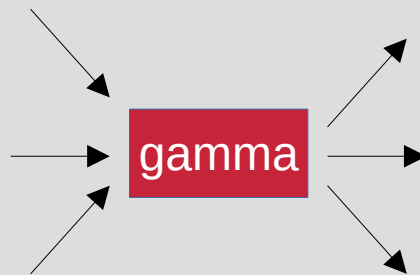
`put: channel, msg → void`

Создание реакции:

`react: channel, fn → void`

Создание связи:

`create_link: source_id, target_id → link_object`



Компонента

- Выполняет ту или иную функцию (колонка, кнопка интерфейса, узел 3d сцены, и т. п.) с опорой на локальные API (dom, qt, etc).
- Содержит локальные входные и выходные каналы. Через них взаимодействует с внешним миром, путём их связи с сетевыми каналами.
- Может содержать под-компоненты.
- Деревья компонент — основной метод описания интерфейсов, сцен графики и т.п.

Создание дерева компонент на заданном исполнителе:

create_component: *description, runner_id, target_id, id* → *node*

- *runner_id* — идентификатор исполнителя (целевого процесса в сети)
- *target_id* – идентификатор родительского объекта
- *id* — идентификатор создаваемого объекта
- *description* — описание компоненты

Описание компоненты состоит из:

- *type* — тип компоненты (должен быть известен исполнителю)
- *params* — начальные значения параметров компоненты
- *links_in* – ссылки из глобальных каналов в каналы компоненты*
- *links_out* – ссылки из локальных каналов компоненты в глобальные
- *tags* — перечень множеств, в которые входит компонент (исследуется*)
- *items* — список описаний вложенных (детских) компонент

- type: "column"
- target_id: "mainwindow"
- items:
 - type: "textfield"
id: "txt1"
 - links_in:
value: "abc"
 - links_out:
changed: "text1"
 - type: "textfield"
links_out:
changed: "text2"
 - type: "button"
params:
value: "Поехали!"
links_out:
click: "perform"

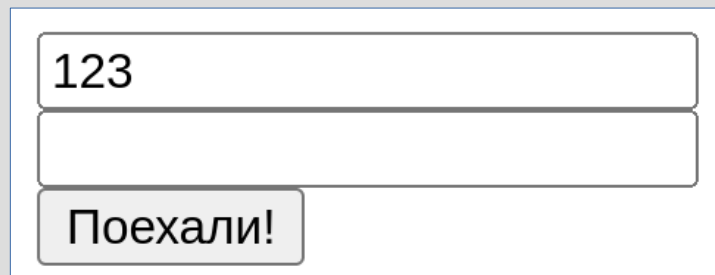
Пример

```
open_channel("abc").put("123")
```

```
open_channel("text1").react( lambda x: print x )
```

```
open_channel("text2").react( lambda x: print x )
```

```
open_channel("perform").react( .... )
```



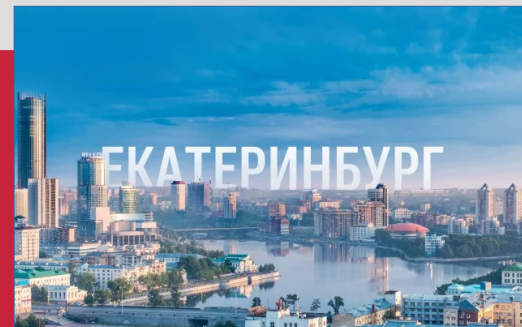
The image shows a window with three components: a text input field containing the number '123', a second empty text input field directly below it, and a button with the text 'Поехали!' (Let's go!) below the second input field.

Результаты

- Разработана программа для десктоп-среды
- Разработано несколько программ для веб-среды
- Реализованы компоненты: GUI (dom), 3D (threejs), ГИС (openlayers)
- Репозиторий*: <https://hub.mos.ru/pavel.vasev/grafix>

<https://hub.mos.ru/pavel.vasev/grafix>

Сетевая технология программирования визуальных интерфейсов



Павел Васёв
Сектор визуализации
ИММ УрО РАН, г. Екатеринбург
<https://cv.imm.uran.ru>