



Ensuring Efficiency in Factory and Production Planning

# Building complex QML-GUIs with QtQuick2

- Introduction to IPO.Plan's software
- Data model concept
- Worker Gantt - View
- Material Zone Editor - View
- Rack Planogram - View
- Live implementing

- Located in Leonberg (services) and Ulm (research and development department)
- 60 employees approximately
- Our process experts provide manifold services, supported by our homemade expert tools
- Main customers are Daimler AG, Siemens AG, Eisenmann GmbH & Co. KG, Liebherr GmbH

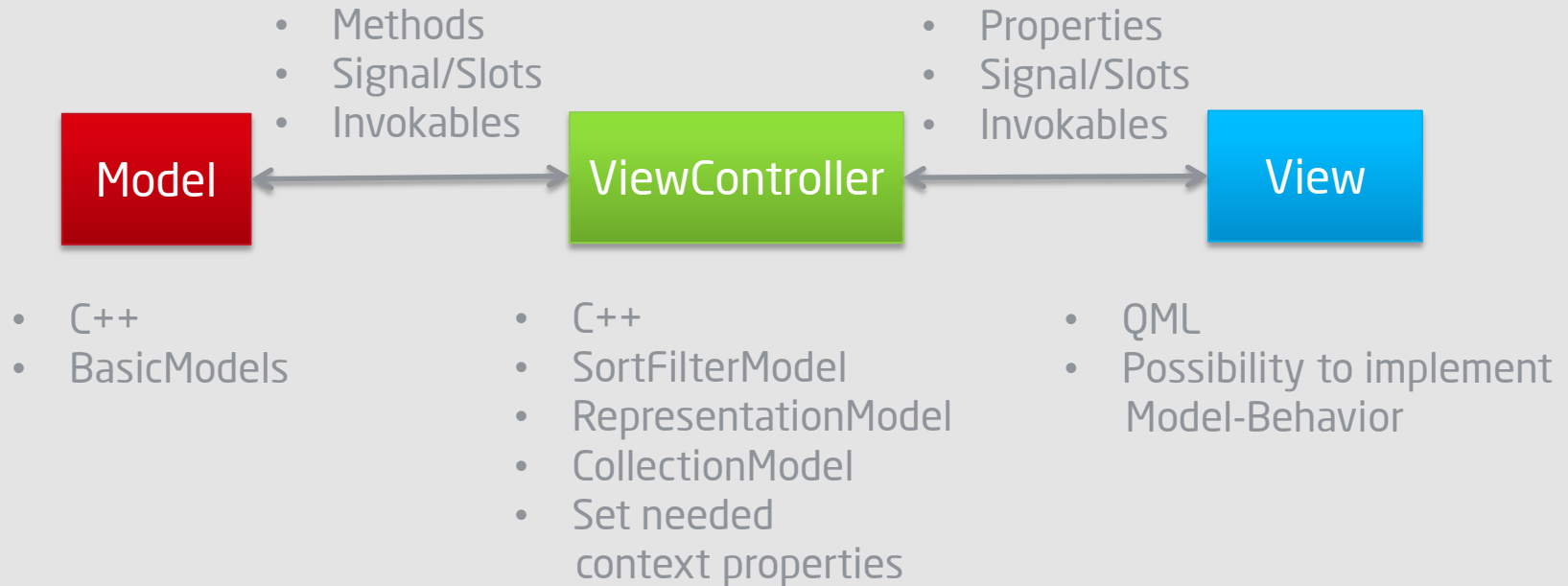
## ■ Software products made by IPO.Plan:

Purely software



Software + Hardware



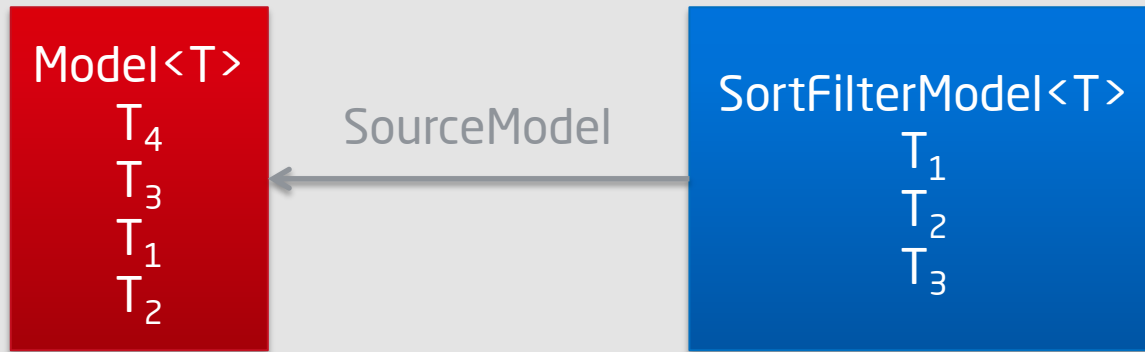


- Using only one custom-defined data role („objectRole“)
- Providing instance references directly to QML-Code by this role
- You have to inherit the „QAbstractListModel“

→ Detailed description in attachments

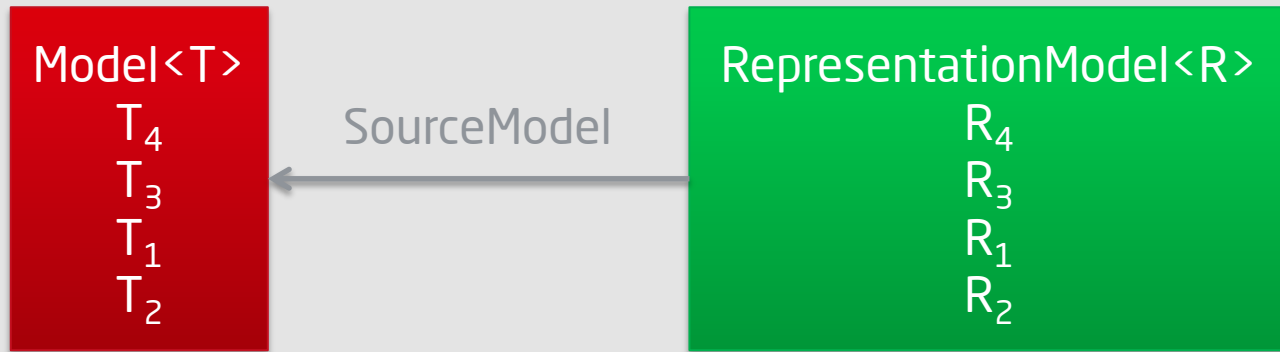
### Example

- Ascending index order
- Filter indices < 4



### Example

- Use a type R to represent a single source instance

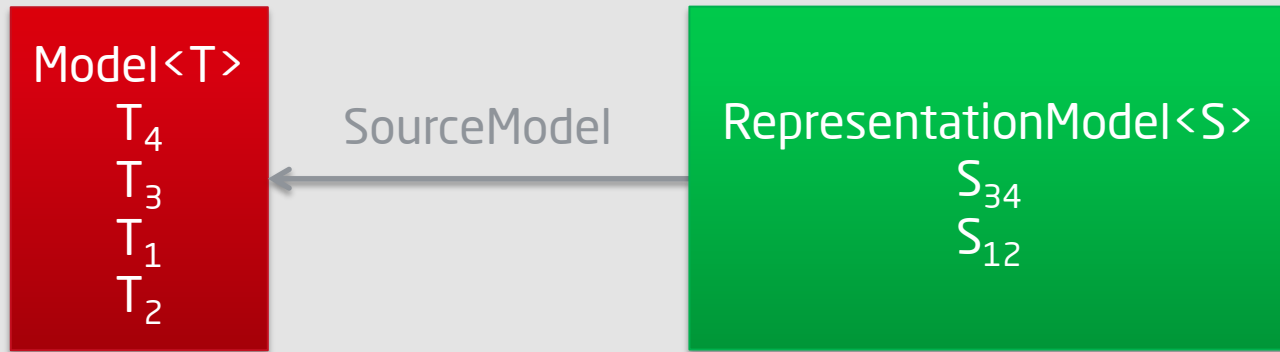


Keep in mind that instances of R may already exist or be allocated by the RepresentationModel



### Example 2

- Use a type S to represent multiple source instances

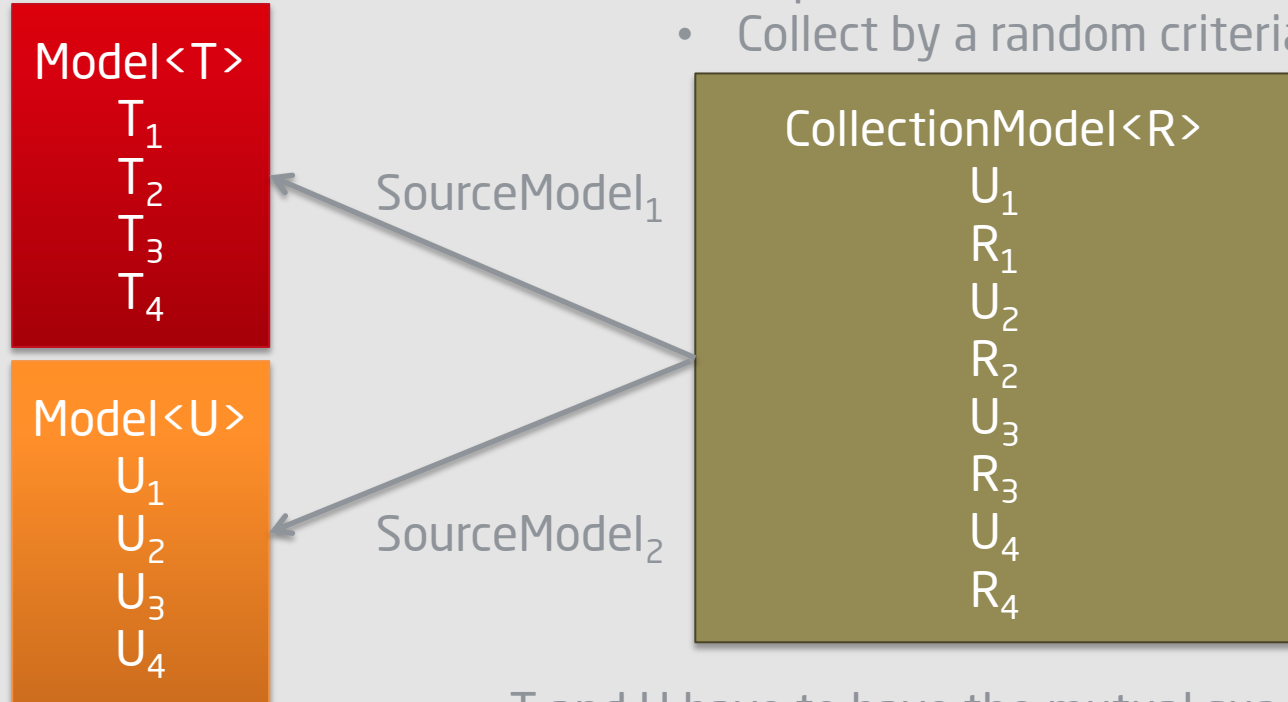


# Data model concept

## CollectionModel

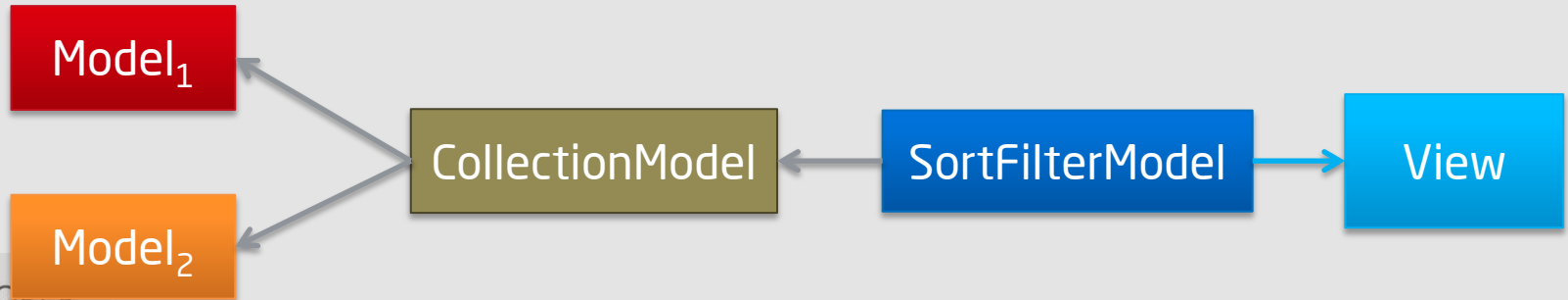
### Example

- Collect by a random criteria



T and U have to have the mutual super class R

- You can combine the mentioned models to create manifold behavior
- Just queue them in a model chain
- Later: RackPlanogram in IPO.Log



## Worker

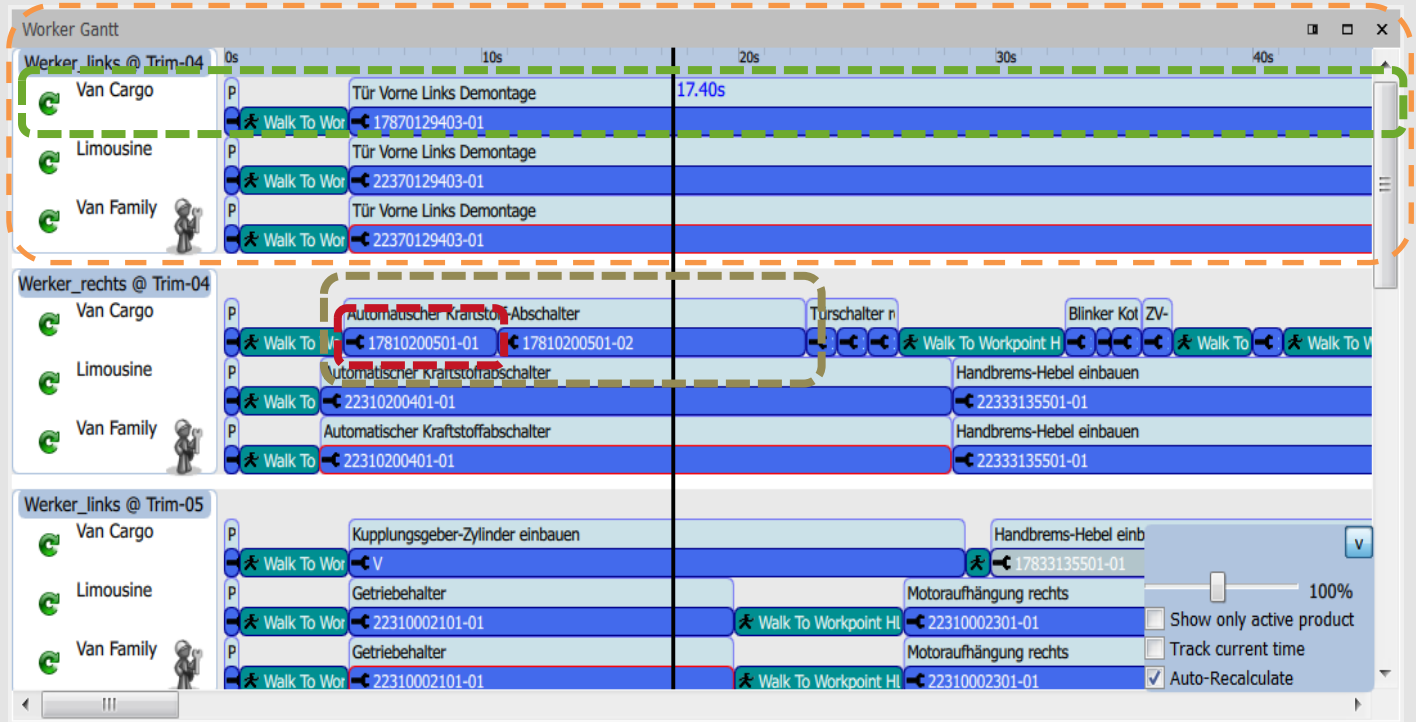
Use a SortFilterModel for Products

## ProductLine

Use a ListModel of Activities and ActivityGroups

## Activity

## ActivityGroup



## ■ The hierarchy of the Worker Gantt - View

WorkerGantt

↳ Worker

↳ ProductLine

↳ Activities and ActivityGroups

## Area

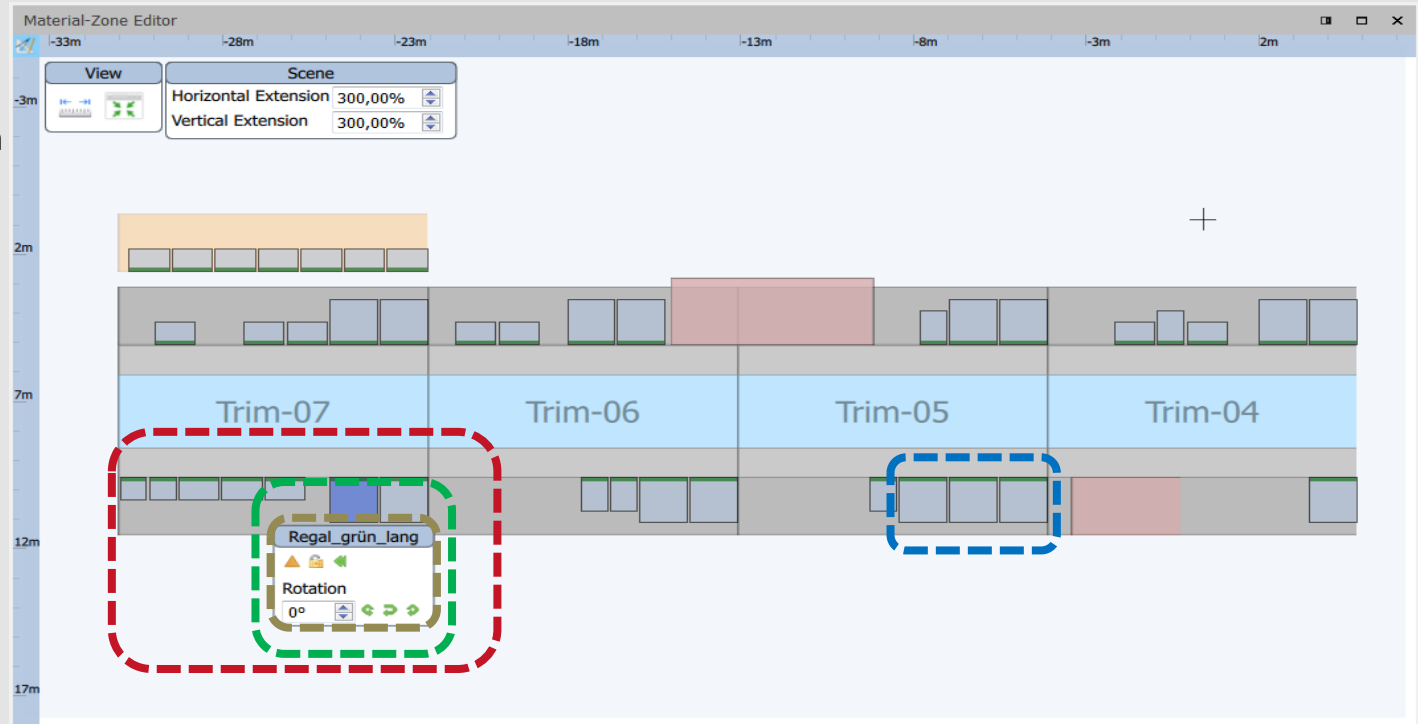
Contains all components in a simple ListModel

## ComponentOnArea

## ComponentOnAreaEdit

## ComponentOnAreaGroup

Uses a SortFilterModel of selected Components



Planogram

PlanogramArea

Uses a SortFilterModel

to get all areas with selected components

PlanogramRackComponent

PlanogramRackInfoGrid

PlanogramInfoGrid

The screenshot shows a software window titled "Rack Planogram" with a sub-header "Trim-05\_ML". The main area displays a 3D perspective view of a rack structure with four shelves labeled "Shelf 1" through "Shelf 4". The shelves are arranged in two columns, each labeled "Regal\_grün\_lang". A central panel lists components for each shelf:

- Shelf 4:**
  - 46807544 (Kupplungs-Geber-Zylinder) @ Van Cargo
  - 46828291 (Seilzug Motorhaube) @ Limousine, Van Family
- Shelf 3:**
  - 46740714 (Schlauch HydroKupplung) @ Limousine, Van Family
  - 46773925 (Getriebehalter) @ Limousine, Van Family
- Shelf 2:**
  - 51781077 (Schloss Motorhaube) @ Limousine, Van Family
  - 55182460 (Kupplungs-Schlauch) @ Van Cargo
  - 735276409 (Öffner Motorhaube) @ Limousine, Van Family
  - 46410854 (Sicherheitsgurt-Höheneinstellvorrichtung Komp.) @ Van Cargo
- Shelf 1:**
  - 46828845 (Schlauch-Halter Komp.) @ Van Cargo
  - 51750208 (Mittelkonsole-Halter hinten) @ Van Cargo
  - 51762065 (Motoraufhängung M11) @ Limousine, Van Family
  - 51772914 (Blinkleuchte) @ Van Cargo

On the right side, there is a "Boxes on Floor" section with a blue square icon. The interface includes various UI elements like zoom controls, a search bar, and a status bar at the bottom.

# Rack Planogram - View

## ■ The hierarchy of the Rack Planogram - View

Planogram

↳ PlanogramArea

↳ PlanogramRackComponent

↳ PlanogramRackInfoGrid

↳ PlanogramInfoGrid

PlanogramBoxGroupComponent

↳ PlanogramBoxGroupInfoGrid

↳ PlanogramInfoGrid



# We are hiring!

If you...

- ...are interested in our stuff
- ...are of course firm in **Qt, QML**
- ...are firm in **C++, C#, Java** or another language
- ...want to work in a **young, creative** and **flexible** team
- ...search for **challenges**
- ...looking for **virtual reality, simulation** and **3D graphic**

Maybe you can find the right job for you at **IPO.Plan**

→ You are also welcome to our **stand** here at **QtDeveloperDays**

THANK YOU 😊

## IPO.Plan GmbH



- Martin Lang  
E-Mail: [martin.lang@ipoplan.de](mailto:martin.lang@ipoplan.de)
- Office Ulm.  
Grünhofgasse 3 | D-71229 Ulm  
Fon: +49.7152.70010.82 | Fax: +49.7152.70010.99

## IPO.Plan GmbH



- Dipl.-Ing. Dennis Effmert  
E-Mail: [dennis.effmert@ipoplan.de](mailto:dennis.effmert@ipoplan.de)
- Office Ulm.  
Grünhofgasse 3 | D-71229 Ulm  
Fon: +49.7152.70010.82 | Fax: +49.7152.70010.99

## DataModel.h

```
class DataListModel : public QAbstractListModel
{
    Q_OBJECT
public:
    ...
    enum Roles {ObjectRole = Qt::UserRole + 1};
    void append(QObject *newObject);
    QHash<int, QByteArray> roleNames() const;
    QVariant data(const QModelIndex &index, int role) const;
    ...
protected:
    QList<QObject*> m_objects;
}
```

## DataModel.cxx

```
...
void DataListModel::append(QObject *object)
{
    beginInsertRows(QModelIndex(), m_objects.count(), m_objects.count());

    m_objects.append(object);
    // To observe the QObject::destroyed() signal
    if (m_tracking)
        trackObject(object);

    endInsertRows();
}
...
```

## DataModel.cxx

```
...
QHash<int, QByteArray> DataListModel::roleNames() const
{
    QHash<int, QByteArray> roles;
    roles[DataListModel::ObjectRole] = "object";
    return roles;
}
...
```

## DataModel.cxx

```
...
QVariant DataListModel::data(const QModelIndex &index, int role) const
{
    if (index.row() < 0 || index.row() >= m_objects.size())
        return QVariant();

    switch (role)
    {
        case ObjectRole:
            return QVariant::fromValue(m_objects.at(index.row()));
    }

    return QVariant();
}
...
```