

The Future of Automation

Programmer as a discipline

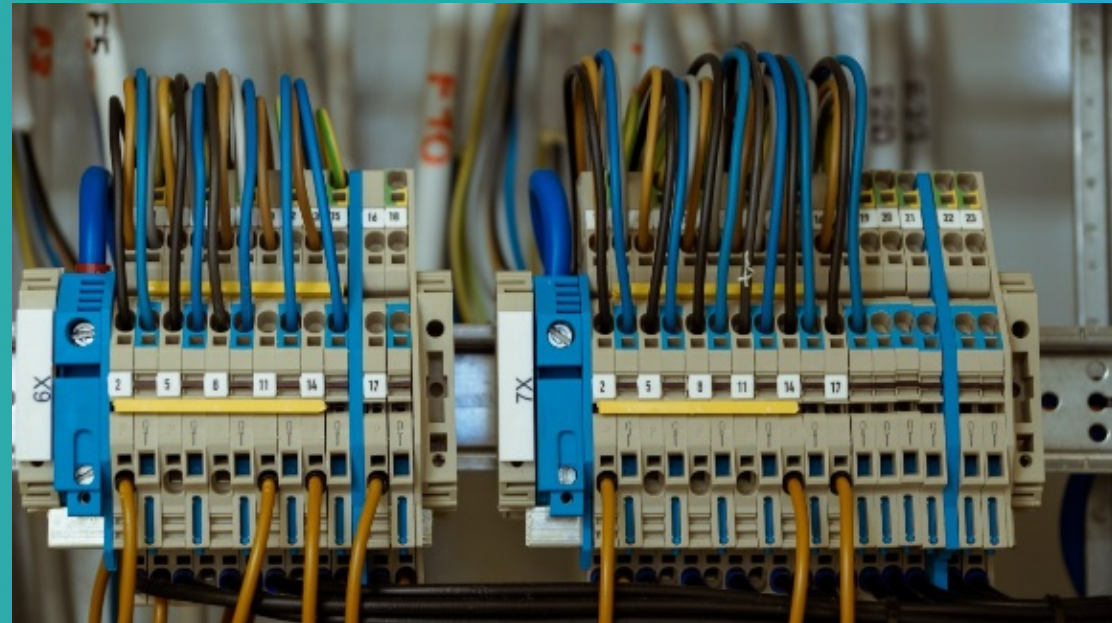
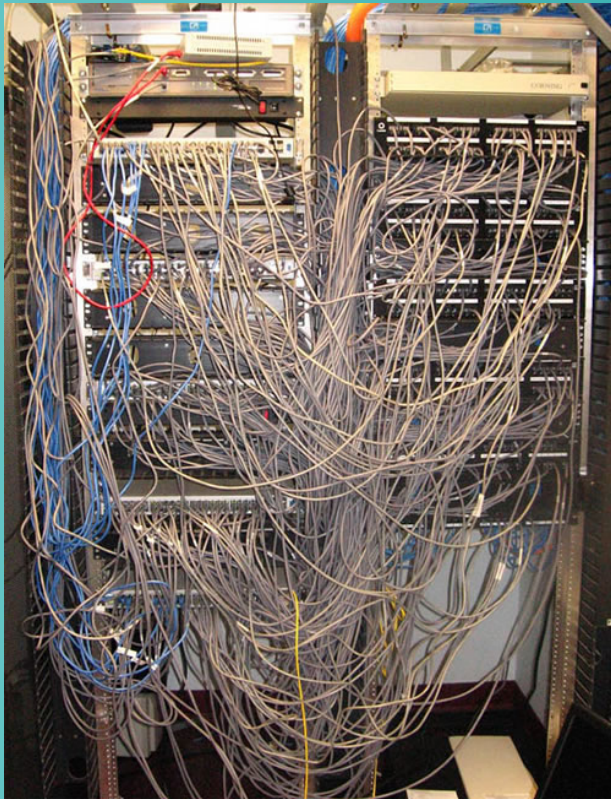
// TIA Portal Openness

```
controller = null;  
DeviceAggregation devices = tiaProject.Devices  
foreach (Device device in devices)  
{  
    controller.Add(device);  
    devices.Where(device => device.Subtype == "S7-1500").ForEach(device =>  
    {  
        Select(device => device.DeviceItems)  
        SelectMany(deviceItems => deviceItems)  
        AllType<ControllerTarget>();  
    }  
}
```

+321.089 kWh
out of range

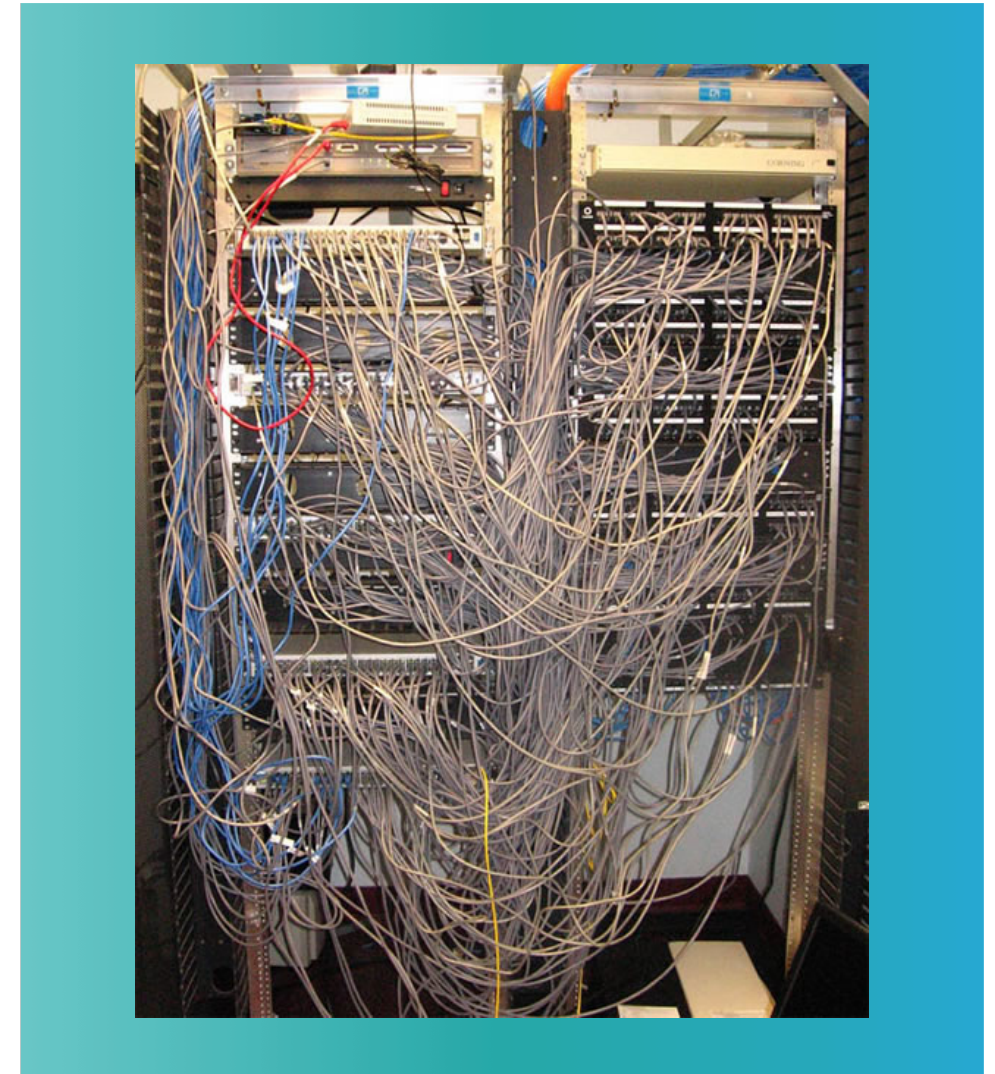
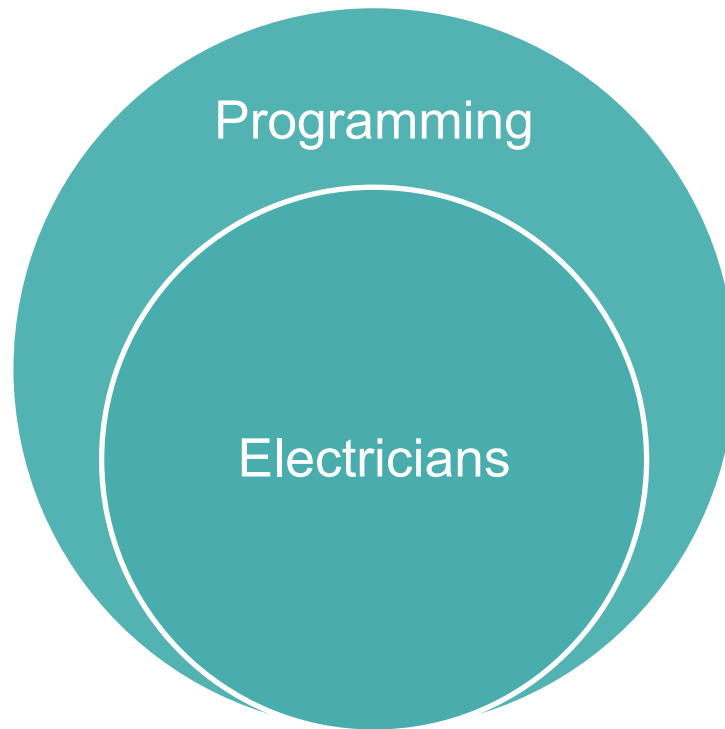
The Future of Automation – Why programmers?

A logically identical program is not necessarily just as good



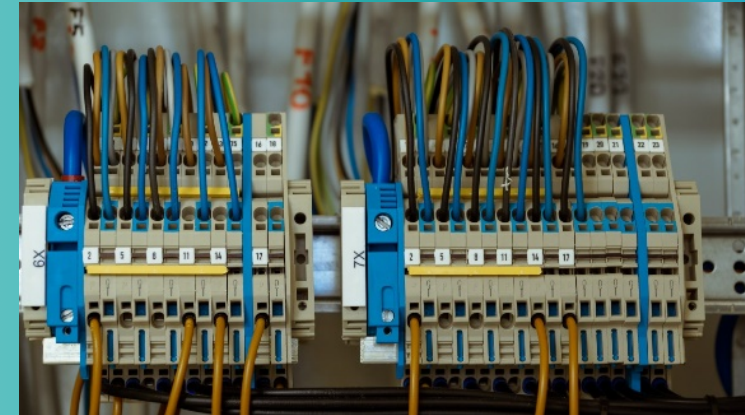
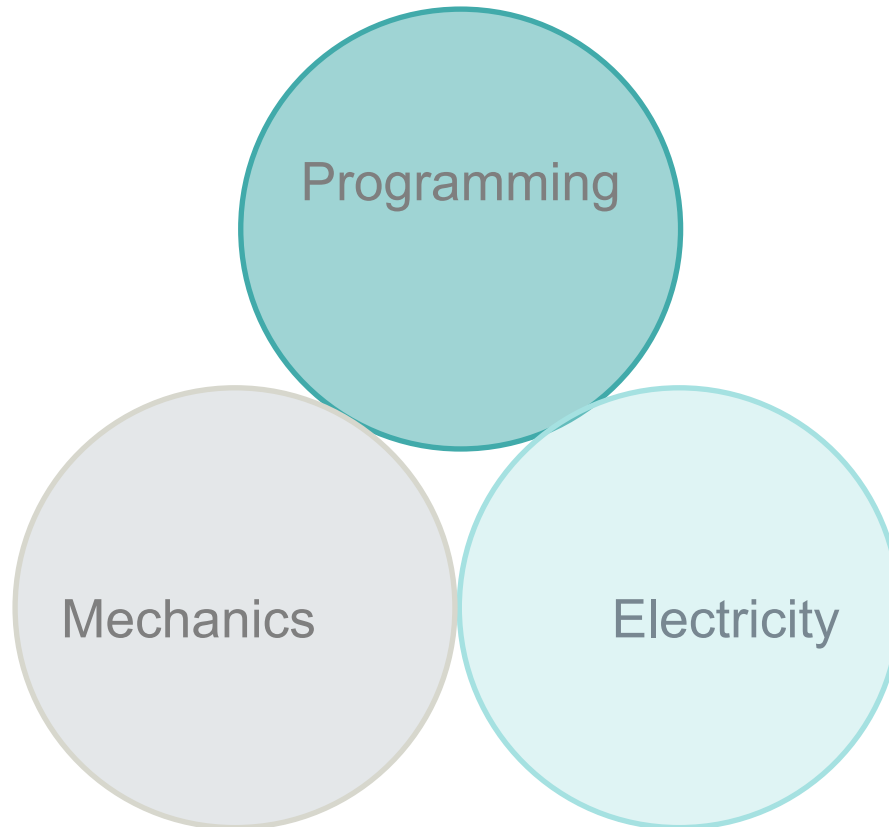
The Future of Automation – Why programmers?

People with a background in electricity have to learn programming



The Future of Automation – Why programmers?

Someone with programming background is better suited to solve programming problems, especially complex ones



A good program

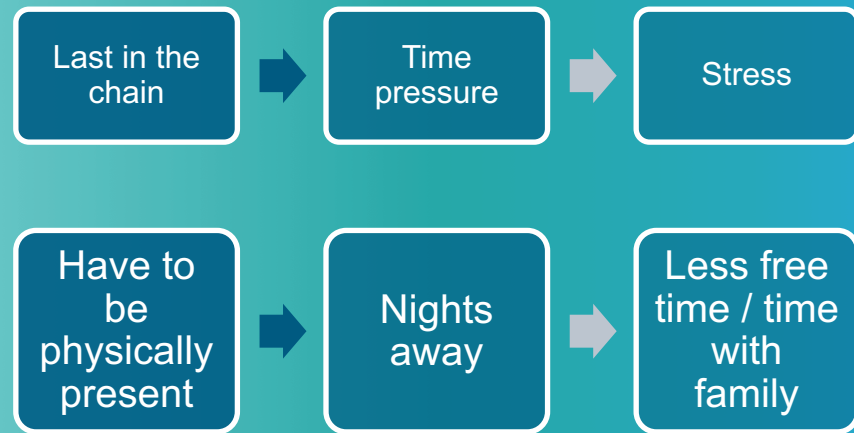
- Generalised
- Reusable
- Modular

Advantages

- Easier to debug
- Easier to expand upon
- Less bugs

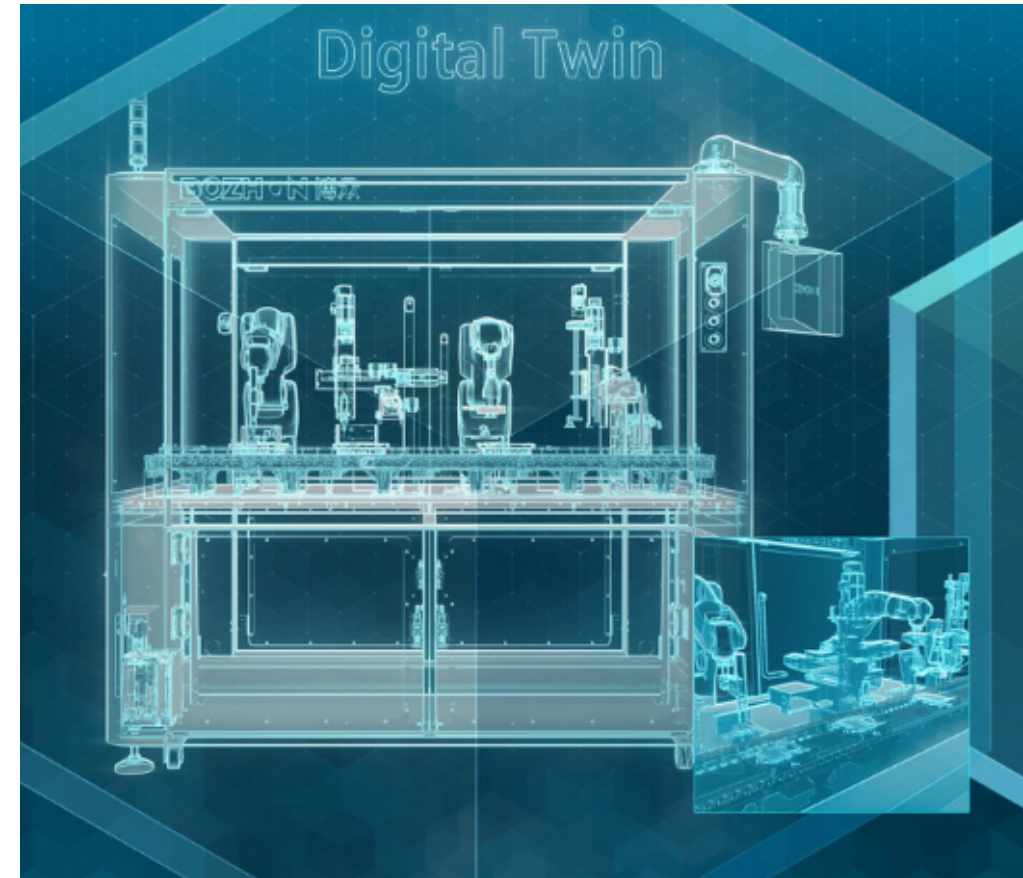
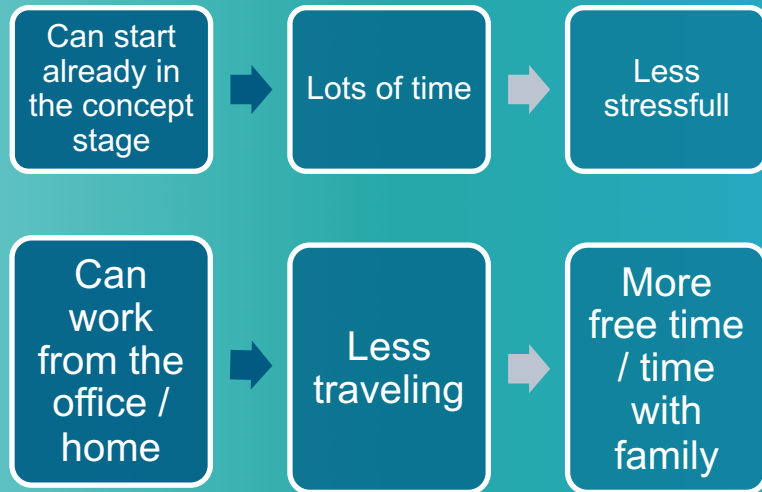
The Future of Automation – How to attract programmers?

Today



The Future of Automation – How to attract programmers?

The future?



The Future of Automation – How can programmers learn automation?

Frameworks

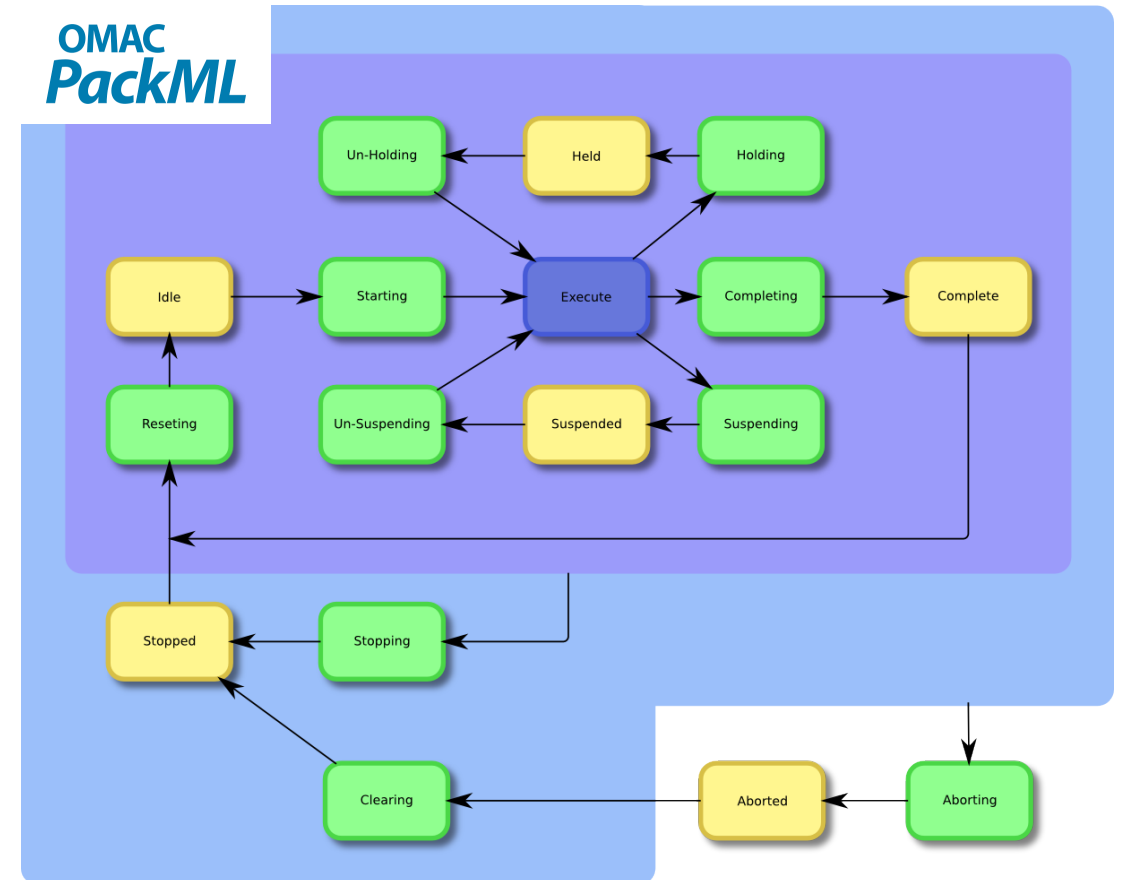
Usability

Safety

Consistency

KPI + OEE

OMAC
PackML



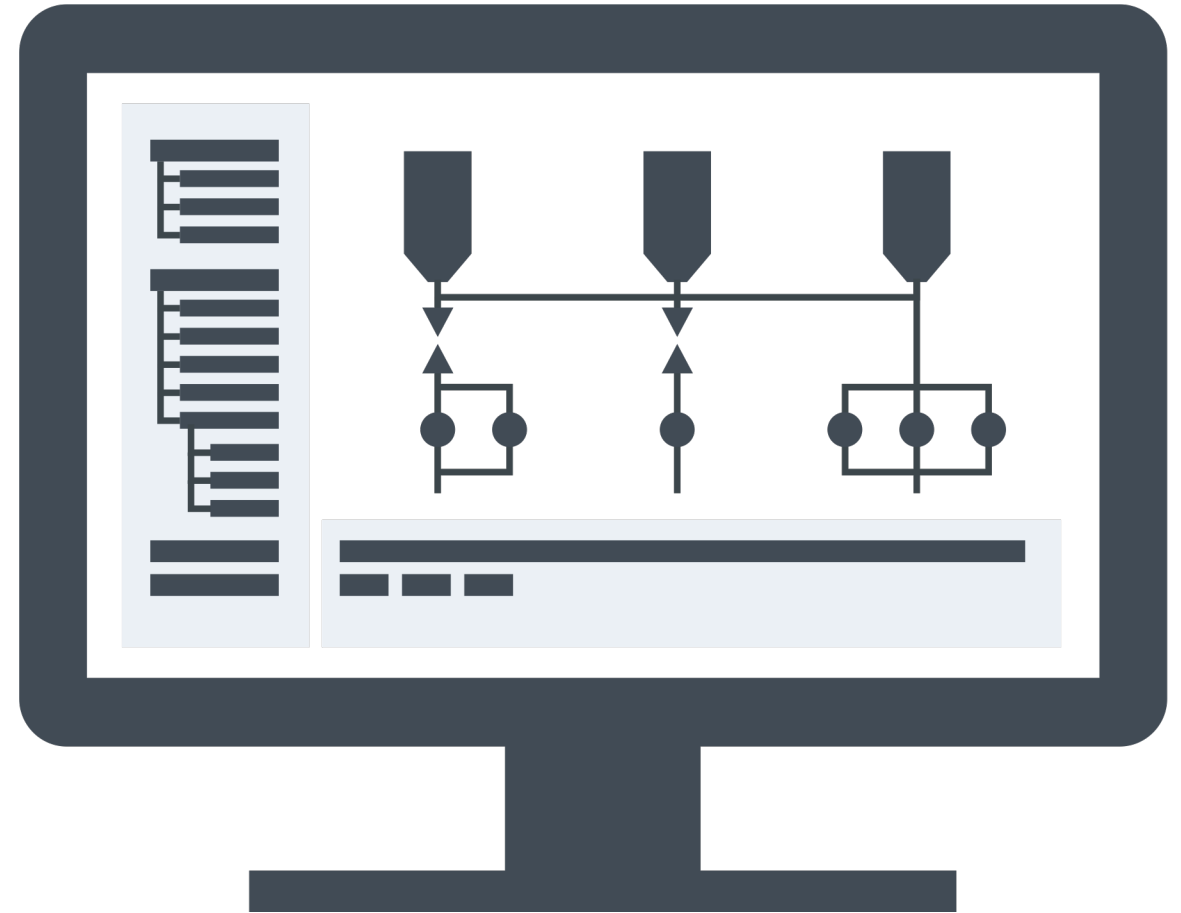
The Future of Automation – How can programmers learn automation?

Simulation

Experimenting

Error handling

Behavior
libraries



The Future of Automation – How can programmers learn automation?

Cooperation

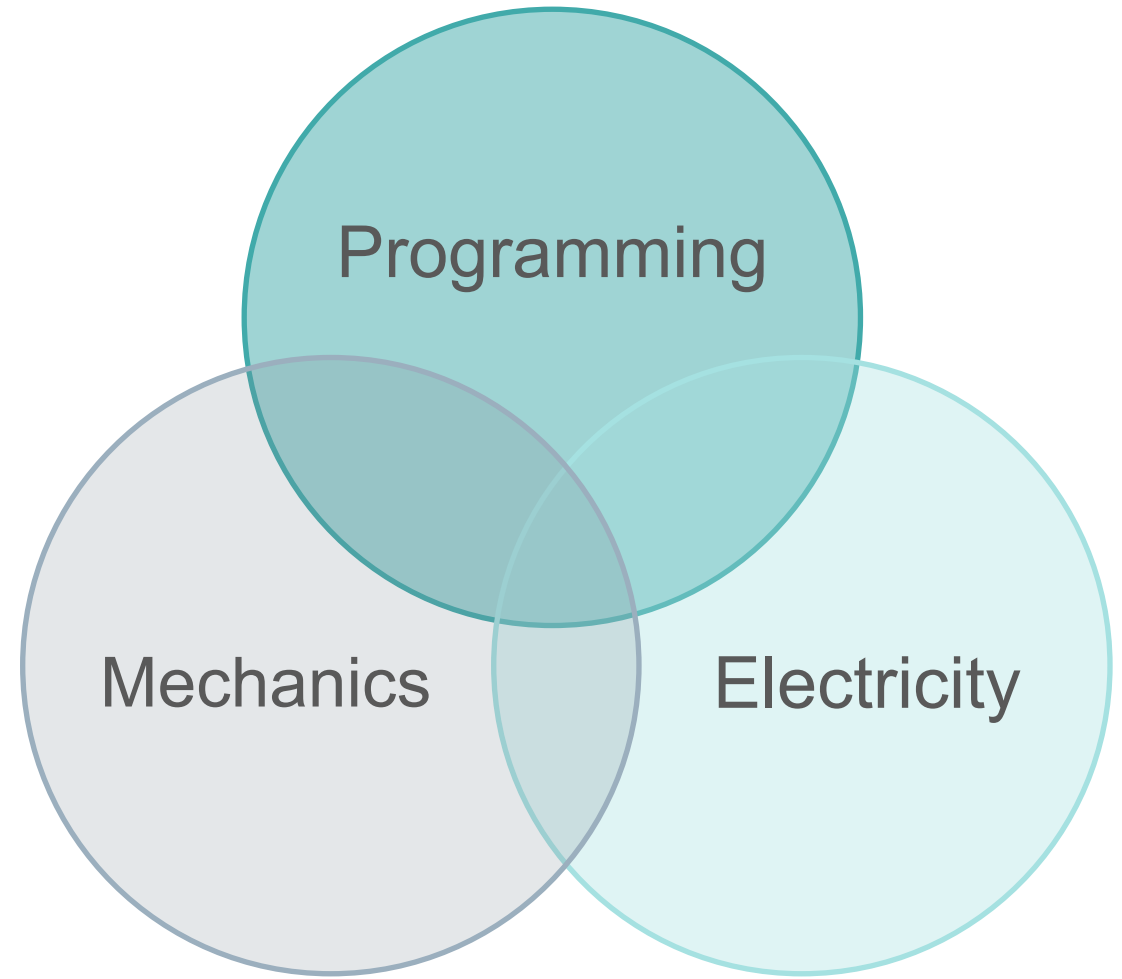
Help on the way

New perspectives

Useful insights

Exchange of knowledge

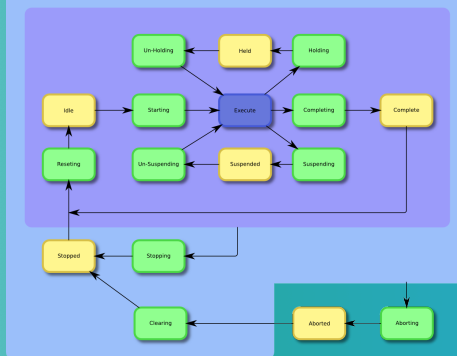
Optimal problem solving



The Future of Automation

Frameworks

OMAC
PackML



Simulation



Cooperation

