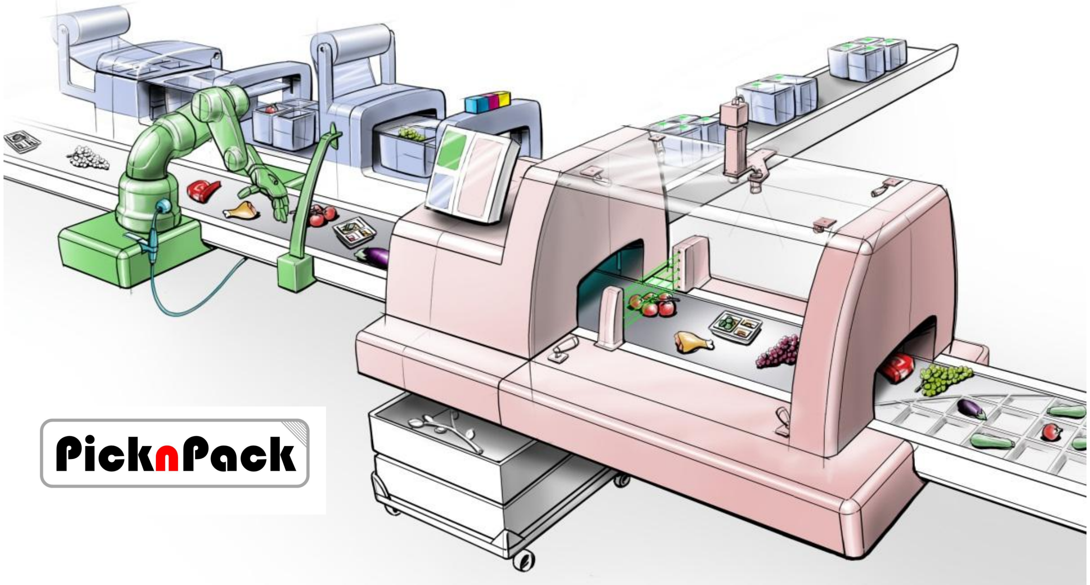
D2.6 — Framework for self-descriptive GUIs, demonstrated on at least two sub-systems

Herman Bruyninckx

25/10/2015



Flexible robotic systems for automated adaptive packaging of fresh and processed food products

The research leading to these results has received funding from the European Union Seventh Framework Programme under grant agreement n^o 311987.

Dissemination level		
PU	Public	**X**
PR	Restricted to other programme participants (including the EC Services)	
RE	Restricted to a group specified by the consortium (including the EC Services)	
CO	Confidential, only for members of the consortium (including the EC Services)	

This document is about the *software prototypes* for the *Graphical User Interfaces* (GUI), that have been developed to support the integration of various modules in a Pick-n-Pack food processing line.

The approach taken in the developments was to base them on the very recent and active developments in the broad domain of "the Web", as driven by major net-based companies like Facebook and Google, that are bringing a revolution to using a standard web browser as the commonly available, extremely flexible, and ever more efficient infrastructure for so-called *Single Page Applications*, which are flexible GUIs connected over a network to a system of functionalities.

The project hosts all its commonly shared and developed software on a *GIT* server, https://gitlab.mech.kuleuven.be/rob-picknpack/pnp-line/tree/gui, and access to it is via a username and password. Reviewers interested in exploring the concrete code of the prototype are invited to request a login to the GIT server by email to herman.bruyninckx@mech.kuleuven.be.

The prototype consists of the following two major components:

- *VirtualDOM* via *React*: this Facebook library is becoming very rapidly a key piece of infrastructure in GUI design, since it offers a very clearly specified, and efficiently programmed, interface to the browser on the one hand (reducing the number of redraws to a minimum), and to the application at the other hand (allowing composition of widget components of various unrelated or related origins).

- *mediator*, via our own Pick-n-pack specific development: a "mediator" is a software pattern that descibes how to develop the piece of software that has the knowledge about two or more sub-systems that it has to connect, while allowing those subsystems to be developed very independently. In this particular case, our mediator has the knowledge about which modules the GUI can connect to, and how to get the data from these different sources and present it in a way that the GUI widgets can display.

- *visualisations* of several types of data and information, via *d3* for time series plotting, *Alpaca* for text and JSON model forms, *GeoJSON/TopoJSON* for geospatial map data.