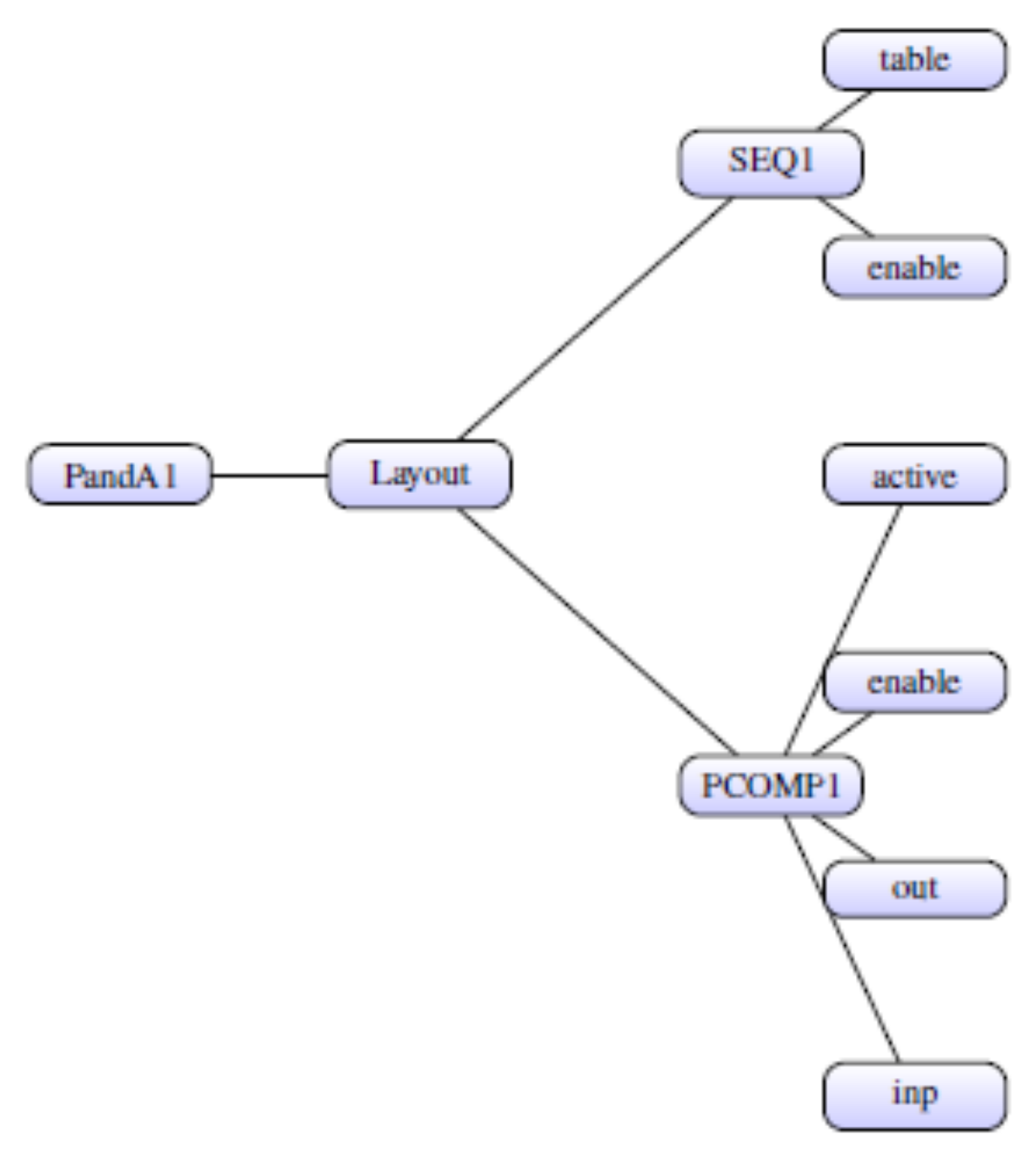
MALCOLMJS: A BROWSER-BASED GRAPHICAL USER INTERFACE FOR THE MALCOLM MIDDLELAYER FRAMEWORK

I. J. Gillingham, T. Cobb, Diamond Light Source Ltd, Oxfordshire, UK,

Malcolm is a middlelayer framework that implements high level configure/run behaviour of control system components like those used in continuous scans. It was created as part of the Mapping project at Diamond Light Source to improve the performance of continuous scanning and make it easier to share code between beamlines. To support the use of Malcolm middlelayer, a browser based graphical user interface is being developed, known as MalcolmJS.

What is Required of the interface?

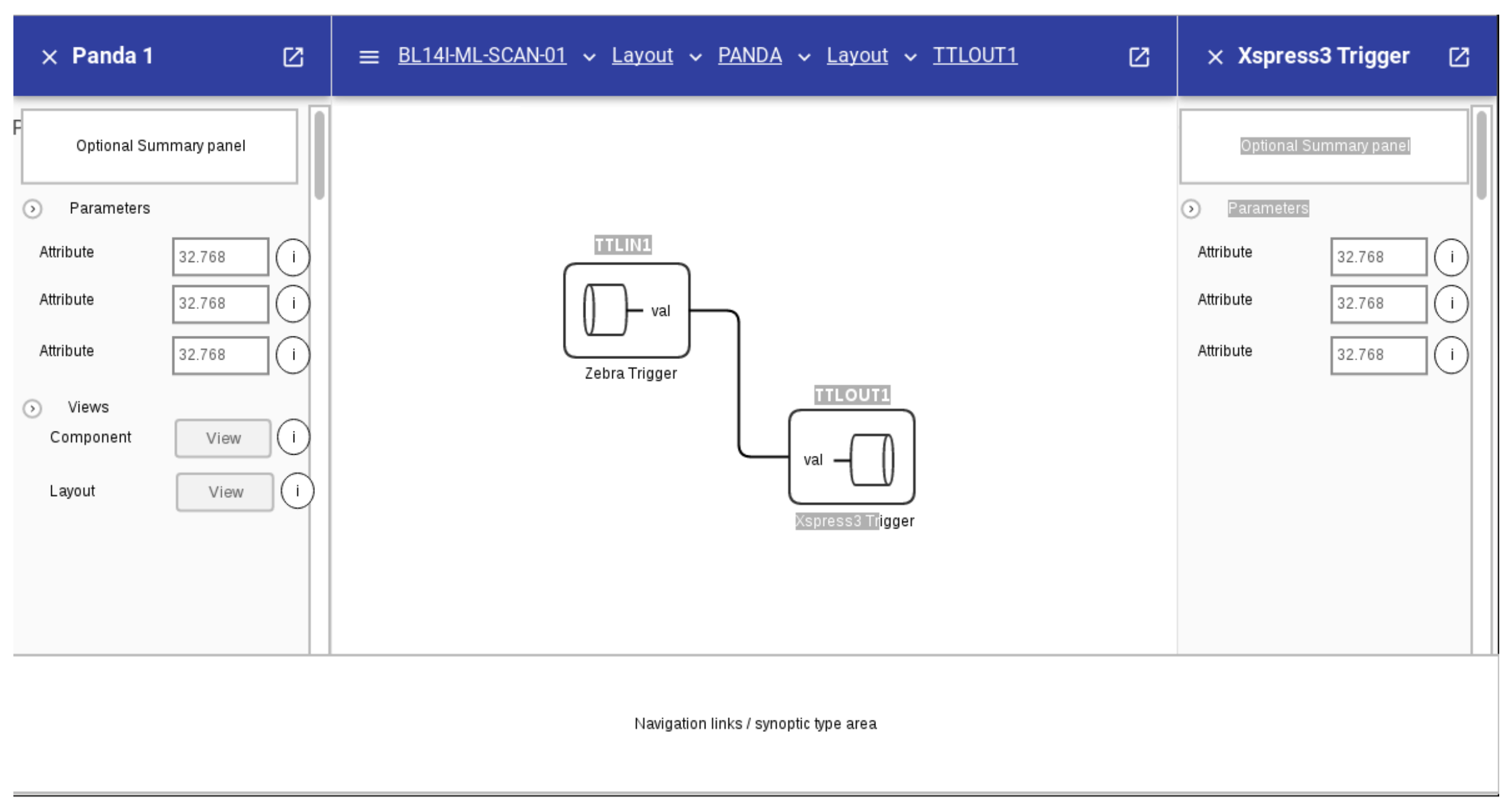
Malcolm objects form a network of parent child relationships with duplicated nodes. For example, the same PMAC motor controller is likely to be used in many scans. A typical tree of objects might look like this:



MalcolmJS must, therefore interrogate the attached device(s) for information on all the available function blocks, attributes and associated meta-data. The collection of internal resources then being listed in a "palette" in the right hand side-pane.

How should the GUI behave?

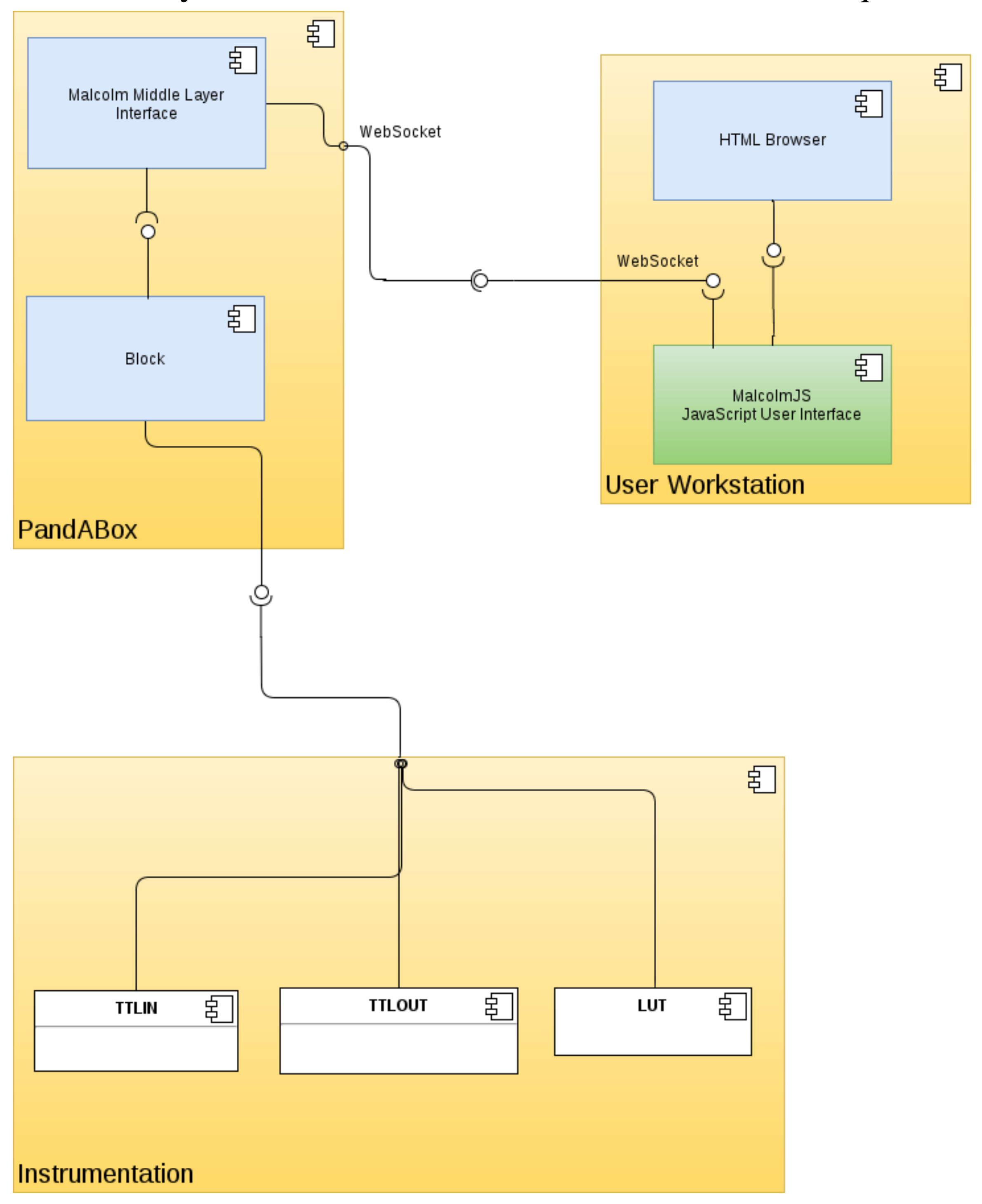
The user is to be able to drag function blocks from a side-pane onto the central main instrument configuration area (main-pane). Connections between blocks are made by selecting a defined port on one block and dragging a "wire" to a port on another block.



After checks for port type compatibility, the new association will be sent to the instrument, where the connection is made in internal logic. With the demand of high levels of user interactivity, it quickly became apparent that the user interface reactive, slick and intuitive.

Communicating with Malcolm

Historically, posting data from a web application has mostly been a straightforward exercise; however receiving data from external source used to be problematic and always a compromise, utilising long-polling and AJAX techniques. Recently, WebSockets have become a mature and reliable method of delivering a persistent and bi-directional interface, facilitating an external entity to actively push information to a browser-based application at any time. Also the application can push information to an external entity, without the need to establish a full HTTP request.



Websocket communications expose the structure of Malcolm Blocks via a JSON protocol over websockets. There is a client MalcolmJS library that is used to create a web GUI to allow configuration/load/save of Blocks from a web browser, like in a PandABox.

Design Architecture

Selecting Third Party Tools and Packages

·It important to consider the following options and questions before employing a package:

- When was the package last updated and is there likely to be future releases to maintain compatibility with the ever evolving React platform, which has short release intervals?
- How popular is the package and how well is it rated among peer developers?
- On how many other packages is this one dependent? Need to be aware of the risk of bloating the final application.

JSON Based Protocol

As the JavaScript Object Notation (JSON) [4] is widely supported in most programming languages and a number of open source libraries are also available, the communications protocol between the browser application and external systems has been developed with JSON as the information exchange language. JSON is easily readable by both computers and people, which brings a huge advantage to the development process.

NPM (Node Package Manager)

NPM [5] is the package manager for JavaScript and the worlds largest software registry. It has been chosen for the wealth of tools and libraries it brings. The NPM repository offers approximately 475,000 open source tools and code modules. A sample of some of the npm packages used within MalcolmJS is:

- interact.js – to manage user dragging of components on screen
- react-toolbox – a set of React components that implement Google's Material Design specification
- karma – a JavaScript tool to facilitate test driven development (TDD)

Webpack

Webpack is a module bundler for modern JavaScript applications, facilitating code to be designed and written and tested in a structured fashion, using the latest ECMAScript (ES) language edition, such as ES6. Most modern browsers fully support EC5, but not the full set of features of ES6 (and above) JavaScript editions. To realise the benefits of developing with ES6, it has been necessary to transpile the ES6 code down to ES5 (or lower), as a bundled monolithic javascript file. This is readily achievable using the Babel package along with Webpack

Summary

The Diamond Control Systems now implement a number of Malcolm compatible instruments across beamlines. Whilst these devices can be configured through other means, there is an increasing need to roll out an operational web interface - MalcolmJS. An alpha test release has been made internally and the feedback has been highly productive, with constructive changes to the original specification. It is likely that MalcolmJS will evolve with increasing versatility as user and instrument requirements change over time. Not only has the development of this application been beneficial to data acquisition and control of devices such as PandA, but is potentially paves the way for adoption with accelerator control systems generally, due to the responsive nature of websockets communication

For more information please contact
ian.gillingham@diamond.ac.uk

diamond