**Functional Requirements for Controls Design Tool:   
Briefing Document and Breakout Group Scopes for Design Charette June 2018**

**Overview**

There will be a design charrette on June 11 to brainstorm the process and tools for designing control sequences for a project and then delivering them, in a neutral format, to the controls vendor/contractor/integrator. This process has the potential to provide verification of correct implementation. We are working closely with the ASHRAE Guideline 36 committee (though are not restricting our process and tools to G36 sequences) and have interest from a few controls vendors. The aim of the charrette is to start the development of a high-level function spec for the controls design tool.

The meeting will start with identification of what the tool will do (and not do) and, in particular, how it will fit into the broader design process and how it will support verification of correct installation. After this initial scoping, we'll split into two groups, one focusing on performance – modeling, comparison of strategies and evaluation of sequences - and one focusing on the additional functionality to be specified in order to generate a complete sequence: start-up, safeties *etc*.

**Existing Use Cases High Level Descriptions**

* Use the controls design tool to design a control sequence and export it as a CDL-compliant specification
  + load a sequence from a library (e.g. Guideline 36, but not necessarily)
  + customize and configure a sequence
  + compare the performance of two or more sequences
* Use the controls design tool to automatically create:
  + block diagrams representing the control sequences, for visualization and communication, and produce a points list with a standard naming convention and/or tagging convention, a plain language sequence of operation, and verification that the control diagram includes all instrumentation required to complete the control sequence,
* Use the OBC tool-chain to bid on a project and, when selected for the project, implement the control sequence in a building automation system,
* Use the commissioning and functional verification tool during commissioning

**New/Needed Use Cases**

1. Add sequence code that goes beyond normal, continuous operation
   * + identify needed content, e.g. start-up / shut-down, safeties / emergencies
     + … ?
   * identify sources of content, create libraries …

**Current Draft Functional Requirements for the Controls Design Tool**

**[** Items in **bold** regular font are in scope for the Performance breakout group. Items in ***bold italics*** are in scope for the Additional Functionality breakout group. Items that are in **bold and underlined** are in scope for both groups.]

1. **The controls design tool shall contain a library of predefined control sequences for HVAC primary systems, HVAC secondary systems and active facades in a way that allows users to customize these sequences.**
2. The controls design tool shall contain a library with functional and performance requirement tests that can be used during design and during commissioning.
3. **The controls design tool shall allow users to add libraries of custom control sequences.**
4. The controls design tool shall allow users to add libraries of custom functional and performance requirement tests.
5. **The controls design tool shall allow testing energy, peak demand, energy cost, and comfort (for each instant of the simulation) of control sequences when connected to a building/system model.**
6. The controls design tool shall allow users to test control sequences coupled to the equipment that constitutes their HVAC system.
7. **When the control sequences are coupled to system models, the controls design tool shall allow users to tag the thermofluid dependencies between different pieces of equipment in the object model. [For example, for any VAV box, the user can define which AHU provides the airflow, which boiler (or system) provides the hot water for heating, etc.]**
8. **The control design tool shall include templates for common objects.**
9. **A design engineer should be able to easily modify the library of predefined control sequences by adding or removing sub-blocks, limiting the need to modify the elemental blocks that make up the visual programming language.**
10. **The controls design tool shall prompt the user to provide necessary information when instantiating objects. For example, the object representing an air handler should include fan, filter, and optional coil and damper elements (each of which is itself an object). When setting up an AHU instance, the user should be prompted to define which of these objects exist.**
11. **To the extent feasible, the control design tool shall prevent mutually exclusive options in the description of the physical equipment. [For example, an air handler can have a dedicated minimum outside air intake, or it can have a combined economizer/minimum OA intake, but it cannot have both.]**
12. **The controls design tool shall hide the complexity of the object model from the end user.**
13. **The controls design tool shall integrate with OpenStudio.**
14. **The controls design tool shall work on Windows, Linux Ubuntu and Mac OS X.**
15. **The controls design tool shall either run as a webtool (i.e. in a browser) or via a standalone executable that can be installed including all its dependencies.**

**New/Needed Functional Requirements for the Controls Design Tool**

1. ***The controls design tool shall contain a library of predefined, configurable sequence modules that implement functionality in addition to normal, continuous operation (“additional functionality”), e.g. start-up, safeties***
2. ***The controls design tool shall prompt the user to provide necessary information when instantiating additional functionality modules.***
3. **The controls design tool shall export complete or partial sequences in CDL and in natural, English, language. A complete export shall include the additional functionality modules required to operate the building.**

**Scope for Performance Breakout Group**

**Tool Functionality (Underlined Items Apply to both Breakout Groups)**

1. **The controls design tool shall contain a library of predefined control sequences for HVAC primary systems, HVAC secondary systems and active facades in a way that allows users to customize these sequences.**
2. **The controls design tool shall allow users to add libraries of custom control sequences.**
3. **The controls design tool shall allow testing energy, peak demand, energy cost, and comfort of control sequences when connected to a building/system model.**
4. **The controls design tool shall export complete or partial sequences in CDL and in natural, English, language. A complete export shall include the additional functionality modules required to operate the building.**
5. **When the control sequences are coupled to system models, the controls design tool shall allow users to tag the thermofluid dependencies between different pieces of equipment in the object model. [For example, for any VAV box, the user can define which AHU provides the airflow, which boiler (or system) provides the hot water for heating, etc.]**
6. **The control design tool shall include templates for common objects.**
7. **A design engineer should be able to easily modify the library of predefined control sequences by adding or removing sub-blocks, limiting the need to modify the elemental blocks that make up the visual programming language.**
8. **The controls design tool shall prompt the user to provide necessary information when instantiating objects. E.g., the object representing an air handler should include fan, filter, and optional coil and damper elements (each of which is itself an object). When setting up an AHU instance, the user should be prompted to define which of these objects exist.**
9. **To the extent feasible, the control design tool shall prevent mutually exclusive options in the description of the physical equipment. [For example, an air handler can have a dedicated minimum outside air intake, or it can have a combined economizer/minimum OA intake, but it cannot have both.]**

**Software Design**

1. **The controls design tool shall hide the complexity of the object model from the end user.**
2. **The controls design tool shall integrate with OpenStudio.**
3. **The controls design tool shall work on Windows, Linux Ubuntu and Mac OS X.**
4. **The controls design tool shall either run as a webtool (i.e. in a browser) or via a standalone executable that can be installed including all its dependencies.**

**Scope for Additional Functionality Breakout Group**

**Tool Functionality (Underlined Items Apply to both Breakout Groups)**

1. **The controls design tool shall contain a library of predefined, configurable sequence modules that implement functionality in addition to normal, continuous operation (“additional functionality”), e.g. start-up, safeties**
2. **The controls design tool shall prompt the user to provide necessary information when instantiating additional functionality modules.**
3. **The controls design tool shall export complete or partial sequences in CDL and in natural, English, language. A complete export shall include the additional functionality modules required to operate the building.**
4. **The control design tool shall include templates for common objects.**
5. **The controls design tool shall prompt the user to provide necessary information when instantiating objects. For example, the object representing an air handler should include fan, filter, and optional coil and damper elements (each of which is itself an object). When setting up an AHU instance, the user should be prompted to define which of these objects exist.**
6. **When the control sequences are coupled to system models, the controls design tool shall allow users to tag the thermofluid dependencies between different pieces of equipment in the object model. [For example, for any VAV box, the user can define which AHU provides the airflow, which boiler (or system) provides the hot water for heating, etc.]**

**Software Design**

1. **The controls design tool shall hide the complexity of the object model from the end user.**
2. **The controls design tool shall integrate with OpenStudio.**
3. **The controls design tool shall work on Windows, Linux Ubuntu and Mac OS X.**
4. **The controls design tool shall either run as a webtool (i.e. in a browser) or via a standalone executable that can be installed including all its dependencies.**