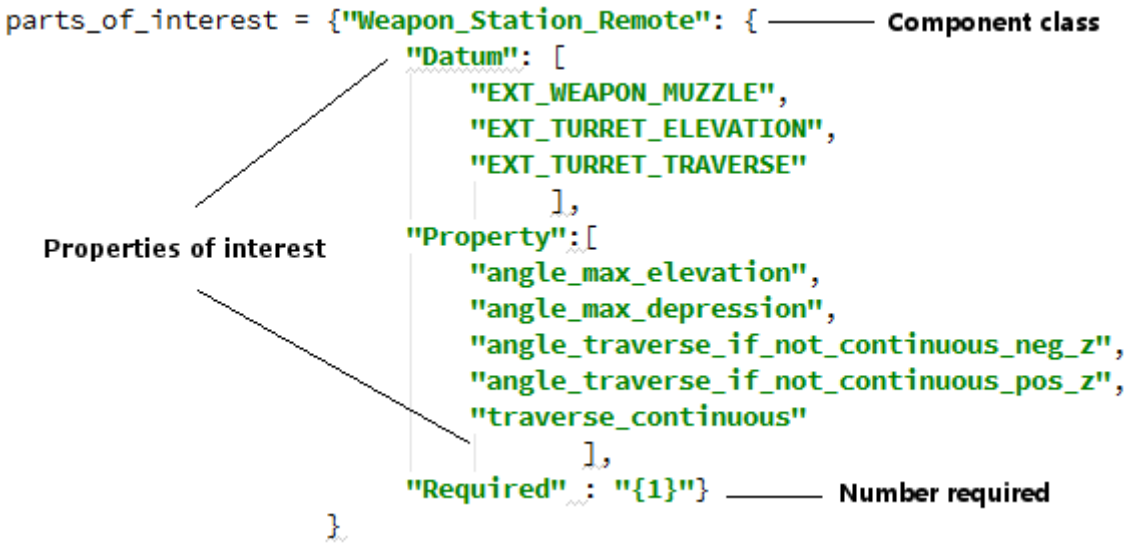
Human Factors/Dynamics Required Components

Overview of Human Factors/Dynamics Required Components for Test Benches

May 2, 2014

1.0 Overview

For each test bench, there are three sections: *explicit* requirements (test bench will forcibly stop if part is not present), *implicit* requirements (certain features, like plotting, may yield bad results or fail outright if the part is not present), and *manually specified information* (where applicable).
For explicit requirements, the dictionary of requirements is pasted directly. This includes the name of the component class, the name of the properties being queried for that class, and the number of parts of that type that must be in the assembly (see guide below).



The "required" field can take several kinds of value:

- {*} = any number of parts of this type can be present (including 0)
- {1+} = At least 1 component of this type; no upper limit
- {2,20} = The number of parts must fall in the range 2-20 (inclusive)

Some test benches further require that the user manually enter certain parameters (by editing *settings.js*). Due to how test benches are structured, this means that the user must run the test bench once (to generate the file structure) before they can input the parameters to get an accurate result!

1.2 Potential caveats/ sources of error

Components with a custom classification may not be recognized by the test bench. The class name must exactly match one of those listed; slight substitutions or derived classes are not recognized.
Custom components must have all of the properties, datums, etc specified in the parts_of_interest dictionary (explicit dependencies). ***The test bench will fail if these entities are not present exactly as named.*** Refer to the sample components for the proper placement and usage of the named properties- the burden is on the designer to sanity-check the placement of coordinate systems etc.

2.0 Egress

2.1 Explicit requirements

```
parts_of_interest = {
      "Manikin": {"Datum": ["SWRI_H_POINT", "BOOT_LEFT_CSYS"],
                                  "Property": ["vehicle_role"]},
                     "Hatch_Assembly_Rear_Ramp": {"Datum": ["EXT_HULL_APERTURE"],
                                                  "Required": "{*}"},
                     "Hatch_Assembly_Personnel_Door": {"Datum":
["EXT_HULL_APERTURE"],
                                                       "Required": "{*}"}}
```

2.2 Implicit requirements

The manikins must have specific roles assigned, and assumptions are made about how the vehicle is staged; see documentation.
In addition to the required properties/datums defined above, the overall vehicle geometry is processed to find interior and exterior regions. This creates dependencies on the following classes (note that the hull is either-or, but one of the two hull classes must be used).

- HULLS = {"Hull_Assembly_Parametric", 'Hull_Assembly_Example_With_Connector'}
- DOORS = {'Hatch_Assembly_Rear_Ramp', 'Hatch_Assembly_Personnel_Door'}
- HATCHES = {'Hatch_Assembly_Driver_Commander', 'Hatch_Assembly_Cargo'}
- MANIKINS = {'Manikin'}
- LITTERS = {'Litter_Open'}

Beyond the above special cases, all geometry is considered (with the exception of components- such as headlights- that would never be inside the vehicle, and thus are not germane to finding an exit path).

2.3 Manually specified information

None of the parameters in *settings.js* will need to be edited by the average user. Some values (movement rate, allowed collisions, image rendering, etc) may be useful for select debugging purposes.

3.0 Ergonomics

3.1 Explicit requirements

```
parts_of_interest = {
"Manikin": {"Datum": ["EXT_GRASP_ZONE", "EXT_TOGGLE-BUTTON_ZONE",
                                        "EXT_PUSH-BUTTON_ZONE", "BOOT_LEFT_CSYS",
                                        "BOOT_RIGHT_CSYS", 'EXT_EYE_CSYS_LEFT',
                                        'EXT_EYE_CSYS_RIGHT'],
                              "Property": ["vehicle_role"],
                              "Required": "{2,20}"},
                  "Steering_Wheel": {"Datum": ["EXT_REACH_OBJ_GRASP"], "Required":
"{+}"},
                  "Gear_Select_Panel": {"Datum": ["EXT_REACH_OBJ_PUSH"], "Required":
"{+}"},
                  "Pedal_Accelerator_Electric": {"Datum": ["EXT_FOOT_POINT_PEDAL",
                                                           "INT_BOLT_15"],
"Required": "{*}"},
                  "Seat_Crew": {"Datum": ["CUSHION_FRONT_CSYS", "SRP_1472",
                                          "EXT_BOLT_4"], "Required": "{*}"},
                  "Pedal_Brake_Hydraulic": {"Datum": ["EXT_FOOT_POINT_PEDAL"],
                                            "Required": "{*}"},
                  "Pedal_Brake_Electric": {"Datum": ["EXT_FOOT_POINT_PEDAL"],
                                           "Required": "{*}"}}
```

3.2 Implicit requirements

When considering the placement of the manikin inside the vehicle, certain elements of vehicle geometry and component placement must be considered. Though it is possible for these components to be absent, excluding some of them may yield odd results.

```
class_set = {"Hull_Assembly_Parametric", "Display_Control_Module_Dve", "Periscope",
                    "Hatch_Assembly_Driver_Commander",
"Engine_Compression_Ignition_Diesel"}
```

3.3 Manually specified information

None of the parameters in *settings.js* will need to be edited by the average user. Some values (image rendering, tolerances for MIL stds, etc) may be useful for select debugging purposes.

4.0 Field of Fire

4.1 Explicit requirements

```
parts_of_interest = {
"Weapon_Station_Remote": {
                    "Datum": [
                        "EXT_WEAPON_MUZZLE",
                        "EXT_TURRET_ELEVATION",
                        "EXT_TURRET_TRAVERSE"
                            ],
                    "Property":[
                        "angle_max_elevation",
                        "angle_max_depression",
                        "angle_traverse_if_not_continuous_neg_z",
                        "angle_traverse_if_not_continuous_pos_z",
                        "traverse_continuous"
                               ],
                    "Required" : "{1}"}
                }
```

4.2 Implicit requirements

The field of fire test bench looks for geometry that could potentially obstruct the weapon fire. This requires a reasonably complete vehicle model- for example, the designer should include a hull to get a meaningful result. Some artifacts may occur if components are physically detached (for example, if the weapon station hovers over the vehicle).
All geometry is considered (with the exception of components- such as crew seats or engine- that would never be outside the vehicle, and thus would not interfere with line of fire.

5.0 Field of View

5.1 Explicit requirements

```
parts_of_interest = {
 'Manikin': {'Datum': ['EXT_EYE_CSYS_LEFT',
                                        'EXT_EYE_CSYS_RIGHT'],
                              "Property":["vehicle_role"],
                              "Required" : "{+}"},
                "Periscope":{"Datum":["EXT_PSCOPE_INNER",
                                      "EXT_PSCOPE_OUTER"],
                              "Property":["focal_distance",
                                          "field_of_view_angle_left",
                                          "field_of_view_angle_right",
                                          "field_of_view_angle_down",
                                          "field_of_view_angle_up"],
                              "Required" : "{+}"}}
```

5.2 Implicit requirements

All geometry is considered (with the exception of components- such as crew seats or engine- that would never be outside the vehicle, and thus would not interfere with field of view.

6.0 Transportability

6.1 Explicit requirements

```
parts_of_interest = {
   "Eye_Welded": {"Datum": ["INT_LOAD_LOCATION"],
                               "PrimitiveProperty": ["load_rating.x",
                                                     "load_rating.y",
                                                     "load_rating.z"],
                               "Required": "{4}"},
                "D_Ring_Lashing": {"Datum": ["INT_LOAD_LOCATION"],
                                   "PrimitiveProperty": ["load_rating.x",
                                                         "load_rating.y",
                                                         "load_rating.z"],
                                   "Required": "{4}"},
                "Pintle_Tow": {"Datum": ["INT_LOAD_LOCATION"],
                               "PrimitiveProperty": ["load_rating.x",
                                                     "load_rating.y",
                                                     "load_rating.z"],
                               "Required": "{+}"},
                "Cleat_Mooring": {"Datum": ["INT_LOAD_LOCATION"],
                                  "PrimitiveProperty": ["load_rating.x",
                                                        "load_rating.y",
                                                        "load_rating.z"],
                                  "Required": "{+}"},
                "Eye_Bolted": {"Datum": ["INT_LOAD_LOCATION"],
                               "PrimitiveProperty": ["load_rating.x",
                                                     "load_rating.y",
                                                     "load_rating.z"],
                               "Required": "{+}"},
                "Drawbar_Bolted": {"Datum": ["INT_LOAD_LOCATION"],
                                   "PrimitiveProperty": ["load_rating.x",
                                                         "load_rating.y",
                                                         "load_rating.z"],
                                   "Required": "{+}"},
                "Drawbar_Eyebolt": {"Datum": ["INT_LOAD_LOCATION"],
                                    "PrimitiveProperty": ["load_rating.x",
                                                          "load_rating.y",
                                                          "load_rating.z"],
                                    "Required": "{+}"},
                "Drawbar_Welded": {"Datum": ["INT_LOAD_LOCATION"],
                                   "PrimitiveProperty": ["load_rating.x",
                                                         "load_rating.y",
```

```
                                                    "load_rating.z"],
                                "Required": "{+}"}}
```

6.2 Implicit requirements

In addition to the properties (such as lift eye location) that are explicitly requested, the transportability test bench requires a reasonably complete vehicle model. (the calculation of vehicle dimensions is not very useful if there are no wheels!)
Certain calculations- such as approach angles or lifting- require the geometry of specific parts:

- wheel_set = {"Roadwheel", "Sprocket_And_Carrier_Drive", "Wheel_Idler"}
- track_set = {"Track"}
- lift_set = {"Eye_Welded"}
- d_ring_set = {"D_Ring_Lashing"}

6.3 Manually specified information

In the *settings.js* file, the user must enter two values. These numbers are initially hard-coded, and relying on them will yield erroneous results in actual use:

- vehicle_curb_mass (in units of kg)
- cent_grav (cg position, in units of m)
- Transport vehicle dimensions are also provided, and should not need to be changed.

7.0 Dynamics

7.1 Manually specified information

- Vehicle mass for all test benches
- Fuel tank volume for Level_Road_Land_Range test bench