This updated version of the digital twin includes the ability to bias agent actions based on preferences and requirement goals. It is noted that setting agent preferences and specifically requirements may result in infeasible team setups, so not all agent actions or results may not satisfy the desired user inputs.

Variables

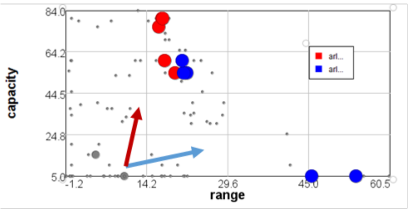
- Business : with general [min , max] values
    - profit : profit of a delivery plan based on weight delivered [0,12000]
    - cost : the cost of buying delivery drones [0,50000]
    - no_customers : the number of customers in a delivery plan [0,65]
    - total_weight : the total weight delivered in a plan [0,126]
    - parcel_weight : the total parcel weight delivered in a plan [0, 112]
    - food_weight : the total food weight delivered in a plan [0,14]
    - number_parcel : the number of parcel customers in a plan [0,65]
    - number_food : the number of food customers in a plan [0,10]
- Planner : with general [min , max] values
    - profit : profit of a delivery plan based on weight delivered [0,12000]
    - cost : the cost of buying delivery drones [0,50000]
    - no_customers : the number of customers in a delivery plan [0,65]
    - weight_delivered*: the total weight delivered in a plan [0,126]
    - parcel_delivered* : the total parcel weight delivered in a plan [0, 112]
    - food_delivered*: the total food weight delivered in a plan [0,14]
    - number_parcel : the number of parcel customers in a plan [0,65]
    - number_food : the number of food customers in a plan [0,10]

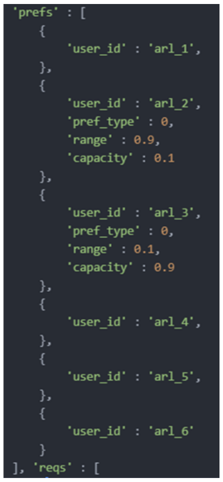
      *future updates will match business name for simplicity
- Designer : with general [min , max] values
    - range : the range or total distance the drone can travel [0,100]
    - capacity : the delivery weight it can carry [0, 100]
    - cost : the purchase cost of the drone [2000, 15000]
    - no_structures : the total number of structures with batteries included in the drone design [1,7]
    - no_ motors: the total number of motors included in the drone design [4,18]
    - no_ foils: the total number of foils included in the drone design [0,10]
    - no_components: the total number of components (batteries, motors, and foils) included in the drone design [5,19]

Preference for a user

- pref_type : 0 or 1
    - 0: weighted sums weighting values (-1.0 – 1.0), where all absolute preference values of cumulative variable values should sum to 1 (ex. 0.4 for range and -0.6 for cost). Positive weightings represent if a variable should be maximized and negative weightings represent if a variable should be minimized. Events of a user are then biased to choose events that performed better based on a weighted sums equation: $f = \Sigma(wi)x$, where wi is the variable weighting and x is the normalized event variable value.

arl_2 : weight = range:0.9, capacity:0.1
arl_3 : weight = tange:0.1, capacity:0.9

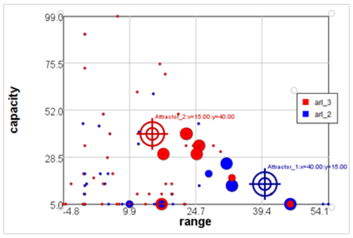


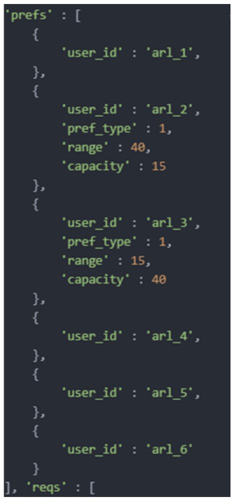


```
'prefs' : [
    {
        'user_id' : 'arl_1',
    },
    {
        'user_id' : 'arl_2',
        'pref_type' : 0,
        'range' : 0.9,
        'capacity' : 0.1
    },
    {
        'user_id' : 'arl_3',
        'pref_type' : 0,
        'range' : 0.1,
        'capacity' : 0.9
    },
    {
        'user_id' : 'arl_4',
    },
    {
        'user_id' : 'arl_5',
    },
    {
        'user_id' : 'arl_6'
    }
], 'reqs' : [
```

- 1: goal-based preference, where preference values represent a target or goal to try and reach using values (ex. 30 for range and 5000 for cost). Events of a user are then biased to perform better based on a distance metric from a goal. This approach is currently recommended, as it can overcome some downsides of a weight-sums approach, such as concave shaped datasets.

arl_2 : goal = range : 40, capacity : 15
arl_3 : goal = range: 15, capacity : 40



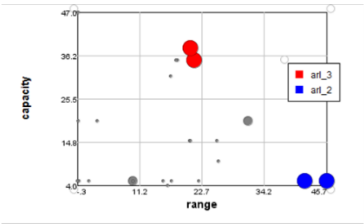


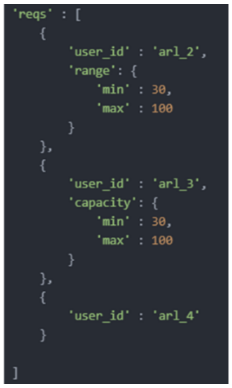
```
'prefs' : [
    {
        'user_id' : 'arl_1',
    },
    {
        'user_id' : 'arl_2',
        'pref_type' : 1,
        'range' : 40,
        'capacity' : 15
    },
    {
        'user_id' : 'arl_3',
        'pref_type' : 1,
        'range' : 15,
        'capacity' : 40
    },
    {
        'user_id' : 'arl_4',
    },
    {
        'user_id' : 'arl_5',
    },
    {
        'user_id' : 'arl_6'
    }
], 'reqs' : [
```

Requirements for a user: where user actions and submit events are biased to be located within a feasible range of values

- min : minimum value for a variable
- max : maximum value for a variable

arl_2 : range [30,100]
arl_3 : capacity [30,100]





```
'reqs' : [
    {
        'user_id' : 'arl_2',
        'range': {
            'min' : 30,
            'max' : 100
        }
    },
    {
        'user_id' : 'arl_3',
        'capacity': {
            'min' : 30,
            'max' : 100
        }
    },
    {
        'user_id' : 'arl_4'
    }

]
```

Future versions of the Digital Twin with preferences and requirements will be updated based on feedback and/or performance results