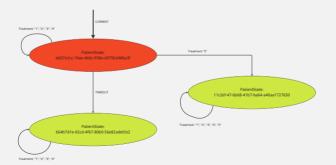


Beispiel ISON

```
"11c2d147-6b68-41b7-ba64-a40aa1727650":
  "uuid": "11c2d147-6b68-41b7-ba64-a40aa1727650
  "timelimit": -1
  "after time state usid": "",
 "treatments": {
    "11: "11:2d147-6b68-41b7-ba64-a40aa1727650"
   "4": "11c2d147-6b68-41b7-ba64-a40aa1727650"
  "conditions": (__)
 "after time state usid": "
   "2": "664b7d1e-65cd-4%7-80b0-56e82adefcb2"
   "4": "664b7d1e-65cd-4957-80b0-56e82adefcb2"
   "5": "664b7d1e-65cd-4f57-80b0-56e82adefcb2"
  "conditions": (__)
"uud": "dd27e2cc-74ae-4b0c-958a-d275b3486a2
"after_time_state_uuid": "664b7d1e-65cd-467-80b0-56e82adefcb2";
```

"1": "dd37e2cc-74ae-6b0c-958a-d37%-3486-47 "2": "dd37e2cc-74ae-4b0c-958a-d375b3486a38

ActivityDiagram and PatientStates



Idee:

- Ein Zustand hält alle messbaren allgemeine Patienten Informationen.
- Fin Zustand kann die Conditions des Vorgängers erhen.
- · Granularität der Zustandsunterschiede in den Conditions hängt von der Definitionfreude des Scenario Admin ab.
- Ein Zustand kann zeitabhängig invalidieren (wird im Hintergrund berechnet durch start time und timelimit) Die start time eines Zustandes definiert den Zeitpunkt, ab wann der Zustand aktiv ist.
- · Damit ein Zustand valide Zeitabhängig ist, muss ein folge Zustand angegeben werden (after time state uuid)
- · Treatments sind äquivalent zu Maßnahmen/Actions
- Treatments definieren die gerichteten Kanten im Graph
- Treatments können einen Zustand ändern
- · Treatments revealen key/values aus den Conditions.

Wichtige Design Aspekte:

- Derzeit (Stand 19.07) revealen die meisten Treatments nur Conditions, ohne den Zustand zu ändern
 - vilt sinnvoll im Backend einen Standard zu definieren
 - der Standard kann als Vorschau eingeblendet werden (so bekommt der Nutzer eine Idee vom Graphen)
 - · der Nutzer kann die Vorschau bearbeiten
- . Ein Zustand kann ein fertiges Template von Conditions laden
 - der Standard kann als Vorschau eingeblendet werden (so bekommt der Nutzer eine Idee vom Graphen)
 - · der Nutzer kann die Vorschau bearbeiten
- . Ein neuer Zustand kann Conditions von einem anderen Zustand erben.
- · dadurch wird die Verwaltungsarbeit des Nutzers drastisch reduziert, sollte der Nutzer Conditions feingranular definieren
- · Shit in Shit out im Groben verhindern:
 - · einfache Bedingungen überprüfen "Sind für einen zeitabhängigen Zustand Limit und neuer Zustand definiert?", etc.
 - um einen Zustand zu verlinken (Treatment oder Timeout) muss dieser vorher erstellt werden, da der Server die UUID bestimmt.

 - das Henne Ei Problem beachten: Sollten nur zeitabhängige Zustände definiert werden, können erst im nachhinein als zeitabhängig
 - konfiguriert werden, damit man den Zustand auswählen kann

Wichtige Backend Aspekte:

- Leere Zustände und Diagramme k\u00f6nnen im Patienten genutzt werden. Die Struktur ist getestet abgesichert.
- Sobald Zustände im Diagramm definiert sind, sollte ein EntryState (current) definiert werden!
 - · solange ein current leer ist kann einfach der erste Zustand der angelegt wird als current definiert werden. Der Nutzer kann das im nachhinein noch ändern.
- Zustände die zeitabhängig definiert sind, akzeptieren nur dann den Input, wenn after time state uuid und timelimit enthalten sind.
- Die UUIDs kommen immer aus dem Backend