

Beispiel JSON

```
*11c2d147-6b68-41b7-ba64-a40aa1727650*: (
  "uuid": "11c2d147-6b68-41b7-ba64-a40aa1727650"
 "start_time": -1,
 "after_time_state_uaid": "
  "treatments": {
   "2": "11c2d147-6b68-41b7-ba64-a40aa1727650"
   "3": "11c2d147-6b68-41b7-ba64-a40aa1727650
  "conditions": ( ... )
"664b7d1e-65cd-4f67-80b0-56e82adefcb2": (
  "uuid": "664b7d1e-65cd-4f67-80b0-56e82adefcb2
  "start_time": -1
```

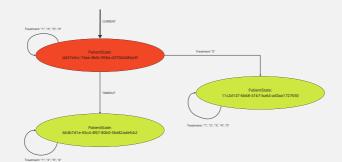
11": "664b7d1e-65cd-4f67-80b0-56e82adefcb2 "3": "664b7d1e-65cd-4f67-80b0-56e82adefcb2

"uuid": "dd37e2cc-74ae-6b0c-958a-d375b3486a3f "start_time": 1721376206, "timelimit": 10,

"after_time_state_uuid": "664b7d1e-65cd-8/67-80b0-56e82adefcb2", "1": "dd17e2cc-74ae-4b0c-958a-d175b3486a3f". "2": "dd17e2cr-74ae-4b0c-958a-d175b3486a2f". "2": "dd17e2cc-74ae-6b0c-958a-d175b3486a3f "5": "11c2d147-6b68-41b7-ba64-a40aa1727650

"conditions": (_)

ActivityDiagram and PatientStates



Idee:

- Ein Zustand h
 ält alle messbaren allgemeine Patienten Informationen (conditions).
- Ein Zustand kann die Conditions des Vorgängers erben.
- · 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 äguivalent 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
 - · vllt 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 können 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