Task Planner

1. **Definitionen**

Aufgabe: Vom Roboter zu erledigende Schritte, unterteilt in Grasp (G), Move (M), Deliver (D), Wait (W).

GUI: Guided User Interface – Benutzeroberfläche

Heuristik: Mit begrenztem Wissen möglichst präzise Aussagen treffen. Hier: Schätzt Kosten für verschiedene Wege.

Grundlagen der Recherche:

In der folgenden Präsentation von Michal Štolba werden verschiedene Task Planner beschrieben und bewertet.

https://cw.fel.cvut.cz/wiki/\_media/courses/a4m36pah/t2-planners.pdf

1. **Ziele**

Für den neuen Task Planner definieren wir die folgenden Anforderungen, die wir in Soft- und Hard- Requirements einteilen. Letztere sind für einen funktionierenden Roboter essentiell, wohingegen Soft- Requirements lediglich die Wettbewerbsfähigkeit erhöhen.

Hard- Requirements

* Interne und externe Bewertung von Aufgaben
  + Einführung eines „Preises“ für die verschiedenen Aufgaben
  + Orientierung an der zu erhaltenden Punktzahl im Wettbewerb.
  + Zeitoptimierte Ausführung der Aufgaben.
* Schnelle Planung
* GUI für aktuelle Prozesse
  + Interaktionsmöglichkeiten
  + Error Handling

Soft- Requirements

* Künstliche Intelligenz (Lernfähigkeit)
  + …wenn Aufgabe zu lange dauert, andere präferieren
* Nicht der optimale Weg ist nicht schlimm, aber vorzuziehen

1. **Auswahl zum Vergleich: Task Planner**

Task Planner: FD-ms (2011)

Heuristik: Merge & Shrink

http://ai.cs.unibas.ch/papers/helmert-et-al-jacm2014.pdf

Task Planner: FD-lmcut (2011)

Heuristik: LM-Cut landmark heuristic (for STRIPS Style Planning Tasks)

Infos: http://agents.fel.cvut.cz/~stolba/publications/stolba2015malmcut.pdf

http://gki.informatik.uni-freiburg.de/papers/bonet-helmert-oberseminarss2010-slides.pdf

Vorteile: **LM-cut heuristic** ist State-of-the-Art

1. **Theorie**

(Basics: http://ai.cs.unibas.ch/misc/tutorial\_aaai2015/planning02.pdf)

Die Planung einer Aufgabe erfolgt nach gegebenen Formalismen, die im Folgenden aufgezählt werden. Dabei geht es um das Definieren „der Welt“. Dies geschieht durch Anfangs- (Initial State) und Endwerte (Goals), sowie Aktionen des Roboters (Effects) etc.

Weit verbreitete Formalismen sind die folgenden:

* STRIPS
* SAS+ (Bäckström und Nebel)

Weiterhin ist eine Input-Language notwendig, wie z.B. für Plan- Systeme PDDL.