Course "Automated Planning: Theory and Practice"
Chapter Lab 02: Hierarchical Task Networks

Teacher: Marco Roveri - marco.roveri@unitn.it

M.S. Course: Artificial Intelligence Systems (LM)
A.A.: 2023-2024
Where: DISI, University of Trento
URL: https://bit.ly/3zOkGk8

Last updated: Wednesday 25th October, 2023

Terms of Use and Copyright

The material (text, figures) in these slides is authored by Jonas Kvarnström and Marco Roveri.

The HDDL Language

- The *Hierarchical Domain Description Language* (HDDL) Höller et al. [10] is an extension of PDDL 2.1 Fox and Long [4].
- It is the standard language used for the International Planning Competition (IPC) 2020 on Hierarchical Planning.
  - `http://ipc2020.hierarchical-task.net`
    `http://gki.informatik.uni-freiburg.de/competition/`
- History
  - SHOP2 Nau et al. [14]
  - ANML Smith et al. [18]
  - HTN-PDDL Gonzalez-Ferrer et al. [8], González-Ferrer et al. [9]
  - HATP de Silva et al. [2], Lallement et al. [11]
  - HPDDL Alford et al. [1]
  - GTOHP Ramoul et al. [16]

HPDDL: Types and Predicates

- We use a simple transportation domain
- They are the same as in PDDL

```
(define (domain transport)
 (:types
    package - locatable
    capacity_number - object
    location - object
    target - object
    vehicle - locatable
    locatable - object
 )
 (:predicates
    (road ?arg0 - location ?arg1 - location)
    (at ?arg0 - locatable ?arg1 - location)
    (in ?arg0 - package ?arg1 - vehicle)
    (capacity ?arg0 - vehicle ?arg1 - capacity_number)
    (capacity_pred ?arg0 - capacity_number ?arg1 - capacity_number)
 )
 ...
)
```

HPDDL: Primitive tasks (aka actions)

- They are the same as in PDDL

```
(:action drive
 :parameters (?v - vehicle ?l1 - location ?l2 - location)
 :precondition (and (at ?v ?l1) (road ?l1 ?l2) )
 :effect (and (not (at ?v ?l1)) (at ?v ?l2) ) )

(:action noop
 :parameters (?v - vehicle ?l2 - location)
 :precondition (and (at ?v ?l2) ) :effect () )

(:action pick_up
 :parameters (?v - vehicle ?l - location ?p - package
              ?s1 - capacity_number ?s2 - capacity_number)
 :precondition (and (at ?v ?l) (at ?p ?l)
                    (capacity_pred ?s1 ?s2) (capacity ?v ?s2) )
 :effect (and (not (at ?p ?l)) (in ?p ?v) (capacity ?v ?s1)
              (not (capacity ?v ?s2)) ) )

(:action drop
 :parameters (?v - vehicle ?l - location ?p - package
              ?s1 - capacity_number ?s2 - capacity_number)
 :precondition (and (at ?v ?l) (in ?p ?v)
                    (capacity_pred ?s1 ?s2) (capacity ?v ?s1) )
 :effect (and (not (in ?p ?v)) (at ?p ?l) (capacity ?v ?s2)
              (not (capacity ?v ?s1)) ) )
```

HPDDL: (abstract) tasks

- In this domain: `deliver`, `get_to`, `load`, `unload`

```
(:task deliver
 :parameters (?p - package ?l - location)
 :precondition ()
 :effect ()
)

(:task get_to
 :parameters (?v - vehicle ?l - location)
 :precondition () ; No preconditions, but in principle possible
 :effect ()       ; No effects, but in principle possible to specify them
)

(:task load
 :parameters (?v - vehicle ?l - location ?p - package)
 :precondition ()
 :effect ()
)

(:task unload
 :parameters (?v - vehicle ?l - location ?p - package)
 :precondition ()
 :effect ()
)
```

HPDDL: METHODS

- In this domain: `m_unload`, `m_load`, `m_i_am_here`, `m_drive_to`, ...

```
(:method m_unload
 :parameters (?l - location ?p - package
              ?s1 - capacity_number ?s2 - capacity_number ?v - vehicle)
 :task (unload ?v ?l ?p)
 :subtasks (and (task0 (drop ?v ?l ?p ?s1 ?s2)) ) )

(:method m_load
 :parameters (?l - location ?p - package
              ?s1 - capacity_number ?s2 - capacity_number ?v - vehicle)
 :task (load ?v ?l ?p)
 :subtasks (and (task0 (pick_up ?v ?l ?p ?s1 ?s2)) ) )

(:method m_i_am_there
 :parameters (?l - location ?v - vehicle)
 :task (get_to ?v ?l)
 :subtasks (and (task0 (noop ?v ?l)) ) )

(:method m_drive_to
 :parameters (?l1 - location ?l2 - location ?v - vehicle)
 :task (get_to ?v ?l2)
 :constraints (not (= ?l1 ?l2)) ; additional to preconditions
 :subtasks (and (task0 (drive ?v ?l1 ?l2)) ) )
```

HPDDL: methods (cont.)

- In this domain: ... `m_drive_to_via`, `m_deliver`

```
(:method m_drive_to_via
 :parameters (?l2 - location ?l3 - location ?v - vehicle)
 :task (get_to ?v ?l3)
 :subtasks (and (task0 (get_to ?v ?l2))
                (task1 (drive ?v ?l2 ?l3)) )
 :ordering (and (task0 < task1) ) )

(:method m_deliver
 :parameters (?l1 - location ?l2 - location ?p - package ?v - vehicle)
 :task (deliver ?p ?l2)
 :subtasks (and (task0 (get_to ?v ?l1))
                (task1 (load ?v ?l1 ?p))
                (task2 (get_to ?v ?l2))
                (task3 (unload ?v ?l2 ?p)) )
 :ordering (and (task0 < task1)
                (task1 < task2)
                (task2 < task3) ) )
```

HPDDL: Problem

- Similar to PDDL

```
(define (problem pfile01)
 (:domain transport)
 (:objects
    package_0 package_1 - package
    capacity_0 capacity_1 - capacity_number
    city_loc_0 city_loc_1 city_loc_2 - location
    truck_0 - vehicle )
 (:htn
  :parameters ()
  :subtasks (and (task0 (deliver package_0 city_loc_0))
                 (task1 (deliver package_1 city_loc_2)) )
  :ordering (and (task0 < task1) ) )
 (:init
  (capacity_predecessor capacity_0 capacity_1)
  (road city_loc_0 city_loc_1) (road city_loc_1 city_loc_0)
  (road city_loc_1 city_loc_2) (road city_loc_2 city_loc_1)
  (at package_0 city_loc_1) (at package_1 city_loc_1)
  (at truck_0 city_loc_2) (capacity truck_0 capacity_1)
 )
)
```

The PANDA HTN Planner

- The PANDA planning system allows to solve different kinds of planning problems. In particular hierarchical task network (HTN) problems defined in HDDL.
  - Original web page:
    `https://www.uni-ulm.de/en/in/ki/research/software/panda/panda-planning-system/`
  - Download Java JAR file:
    `https://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.inst.090/panda/PANDA.jar`
    `https://didatticaonline.unitn.it/dol/mod/resource/view.php?id=1011619`
- Requirements: Java VM
  - `sudo apt-get install openjdk-14-jre openjdk-14-jre-headless`
- Usage: `java -jar -parser hddl domain-file problem-file`
  - Example:
    `java -jar ../PANDA.jar -parser hddl transport-htn.hddl transport-p01.hddl`
  - Full set of options:
    `java -jar ../PANDA.jar -help`
    `java -jar ../PANDA.jar -help [OPTION or KEY]` e.g.
    `java -jar ../PANDA.jar -help -parser` for help on the `-parser` option

The Transporter example

- The Domain:
  { `Domains/Transporter/transport-htn.hddl` }
- The first problem:
  { `Domains/Transporter/p01.hddl` }
- The second problem:
  { `Domains/Transporter/p02.hddl` }
- The third problem:
  { `Domains/Transporter/p03.hddl` }

The Rover example

- The Domain:
  { `Domains/Rover/rover-domain.hddl` }
- The first problem:
  { `Domains/Rover/pfile1.hddl` }
- The second problem:
  { `Domains/Rover/pfile5.hddl` }
- The third problem:
  { `Domains/Rover/pfile15.hddl` }

The Satellite example

- The Domain:
  { `Domains/Satellite/satellite.hddl` }
- The first problem:
  { `Domains/Satellite/1obs-2sat-1mod.hddl` }
- The second problem:
  { `Domains/Satellite/2obs-2sat-2mod.hddl` }
- The third problem:
  { `Domains/Satellite/3obs-3sat-2mod.hddl` }

The Wood Working example

- The Domain:
  { `Domains/Woodworking/woodworking.hddl` }
- The first problem:
  { `Domains/Woodworking/01--p01-complete.hddl` }
- The second problem:
  { `Domains/Woodworking/05--p02-part4.hddl` }
- The third problem:
  { `Domains/Woodworking/06--p02-complete.hddl` }

Other HTN Planners (from IPC 2020 on HTN)

- Tools
  - **Name** (hyperlink to singularity image)
    - LiloTane by Schreiber [17]
    - SIADEX by Fernandez-Olivares et al. [3]
    - HyperTensioN by Magnaguagno et al. [13]
    - PDDL4J-TO and PDDL4J-PO by Pellier and Fiorino [15]
    - pyHiPOP by Lesire and Albore [12]
- Usage
  - `singularity run singularityimage.sif domain.hddl problem.hddl`
  - Example (For LiloTane):
    - `singularity run IPC2020-competitor-1n.sif domain.hddl problem.hddl`

References I

[1] Ron Alford, Gregor Behnke, Daniel Höller, Pascal Bercher, Susanne Biundo, and David W. Aha. Bound to plan: Exploiting classical heuristics via automatic translations of tail-recursive HTN problems. In Amanda Jane Coles, Andrew Coles, Stefan Edelkamp, Daniele Magazzeni, and Scott Sanner, editors, *Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling, ICAPS 2016, London, UK, June 12-17, 2016*, pages 20–28. AAAI Press, 2016. URL http://www.aaai.org/ocs/index.php/ICAPS/ICAPS16/paper/view/13174. 3

[2] Lavindra de Silva, Raphaël Lallement, and Rachid Alami. The HATP hierarchical planner: Formalisation and an initial study of its usability and practicality. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2015, Hamburg, Germany, September 28 - October 2, 2015*, pages 6465–6472. IEEE, 2015. doi: 10.1109/IROS.2015.7354301. URL https://doi.org/10.1109/IROS.2015.7354301. 3

[3] Juan Fernandez-Olivares, Ignacio Vellido, and Luis Castillo. Addressing HTN planning with blind depth first search. In *Proceedings of 10th International Planning Competition: Planner and Domain Abstracts – Hierarchical Task Network (HTN) Planning Track (IPC 2020)*, pages 1–4, 2021. 15

[4] Maria Fox and Derek Long. PDDL2.1: an extension to PDDL for expressing temporal planning domains. *J. Artif. Intell. Res.*, 20: 61–124, 2003. doi: 10.1613/jair.1129. URL https://doi.org/10.1613/jair.1129. 3

[5] Hector Geffner and Blai Bonet. *A Concise Introduction to Models and Methods for Automated Planning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2013. ISBN 9781608459698. doi: 10.2200/S00513ED1V01Y201306AIM022. URL https://doi.org/10.2200/S00513ED1V01Y201306AIM022.

References II

[6] Malik Ghallab, Dana S. Nau, and Paolo Traverso. *Automated planning - theory and practice*. Elsevier, 2004. ISBN 978-1-55860-856-6.

[7] Malik Ghallab, Dana S. Nau, and Paolo Traverso. *Automated Planning and Acting*. Cambridge University Press, 2016. ISBN 978-1-107-03727-4. URL http://www.cambridge.org/de/academic/subjects/computer-science/artificial-intelligence-and-natural-language-processing/automated-planning-and-acting?format=HB.

[8] A. Gonzalez-Ferrer, J. Fernández-Olivares, and L. Castillo. JABBAH: A java application framework for the translation between business process models and HTN. In *International Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS)*, 2009. http://kti.mff.cuni.cz/~bartak/ICKEPS2009/download/ICKEPS2009_gonzalez-ferrer_p28-37.pdf. 3

[9] Arturo González-Ferrer, Juan Fernández-Olivares, and Luis A. Castillo. From business process models to hierarchical task network planning domains. *Knowl. Eng. Rev.*, 28(2):175–193, 2013. doi: 10.1017/S0269888912000410. URL https://doi.org/10.1017/S0269888912000410. 3

[10] Daniel Höller, Gregor Behnke, Pascal Bercher, Susanne Biundo, Humbert Fiorino, Damien Pellier, and Ron Alford. HDDL: an extension to PDDL for expressing hierarchical planning problems. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9883–9891. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/6542. 3

References III

[11] Raphaël Lallement, Lavindra de Silva, and Rachid Alami. HATP: an HTN planner for robotics. *CoRR*, abs/1405.5345, 2014. URL http://arxiv.org/abs/1405.5345. 3

[12] Charles Lesire and Alexandre Albore. pyHiPOP – Hierarchical partial-order planner. In *Proceedings of 10th International Planning Competition: Planner and Domain Abstracts – Hierarchical Task Network (HTN) Planning Track (IPC 2020)*, pages 13–16, 2021. 15

[13] Maurício Cecílio Magnaguagno, Felipe Meneguzzi, and Lavindra de Silva. HyperTensioN: A three-stage compiler for planning. In *Proceedings of 10th International Planning Competition: Planner and Domain Abstracts – Hierarchical Task Network (HTN) Planning Track (IPC 2020)*, pages 5–8, 2021. 15

[14] Dana S. Nau, Tsz-Chiu Au, Okhtay Ilghami, Ugur Kuter, J. William Murdock, Dan Wu, and Fusun Yaman. SHOP2: an HTN planning system. *J. Artif. Intell. Res.*, 20:379–404, 2003. doi: 10.1613/jair.1141. URL https://doi.org/10.1613/jair.1141. 3

[15] Damien Pellier and Humbert Fiorino. Totally and partially ordered hierarchical planners in PDDL4J library. In *Proceedings of 10th International Planning Competition: Planner and Domain Abstracts – Hierarchical Task Network (HTN) Planning Track (IPC 2020)*, pages 17–18, 2021. 15

[16] Abdeldjalil Ramoul, Damien Pellier, Humbert Fiorino, and Sylvie Pesty. Grounding of HTN planning domain. *Int. J. Artif. Intell. Tools*, 26(5):1760021:1–1760021:24, 2017. doi: 10.1142/S0218213017600211. URL https://doi.org/10.1142/S0218213017600211. 3

References IV

[17] Dominik Schreiber. Lifted logic for task networks: TOHTN planner lilotane in the IPC 2020. In *Proceedings of 10th International Planning Competition: Planner and Domain Abstracts – Hierarchical Task Network (HTN) Planning Track (IPC 2020)*, pages 9–12, 2021. 15

[18] David E. Smith, Jeremy Frank, and William Cushing. The ANML language. In *The ICAPS-08 Workshop on Knowledge Engineering for Planning and Scheduling (KEPS)*, 2008. `https://ti.arc.nasa.gov/m/profile/de2smith/publications/ICAPS08-ANML.pdf`. 3