



# Making tea at home

Felix Hampe  
Daniel Pyka

Planning and Scheduling WS2015/2016

Prof. Dr. Gerhard K. Kraetzschmar  
M.Sc. Iman Awaad



# Content

- Problem
- Planner
- Structure
- Strategy
- Tests
- Planning / acting
- Conclusion

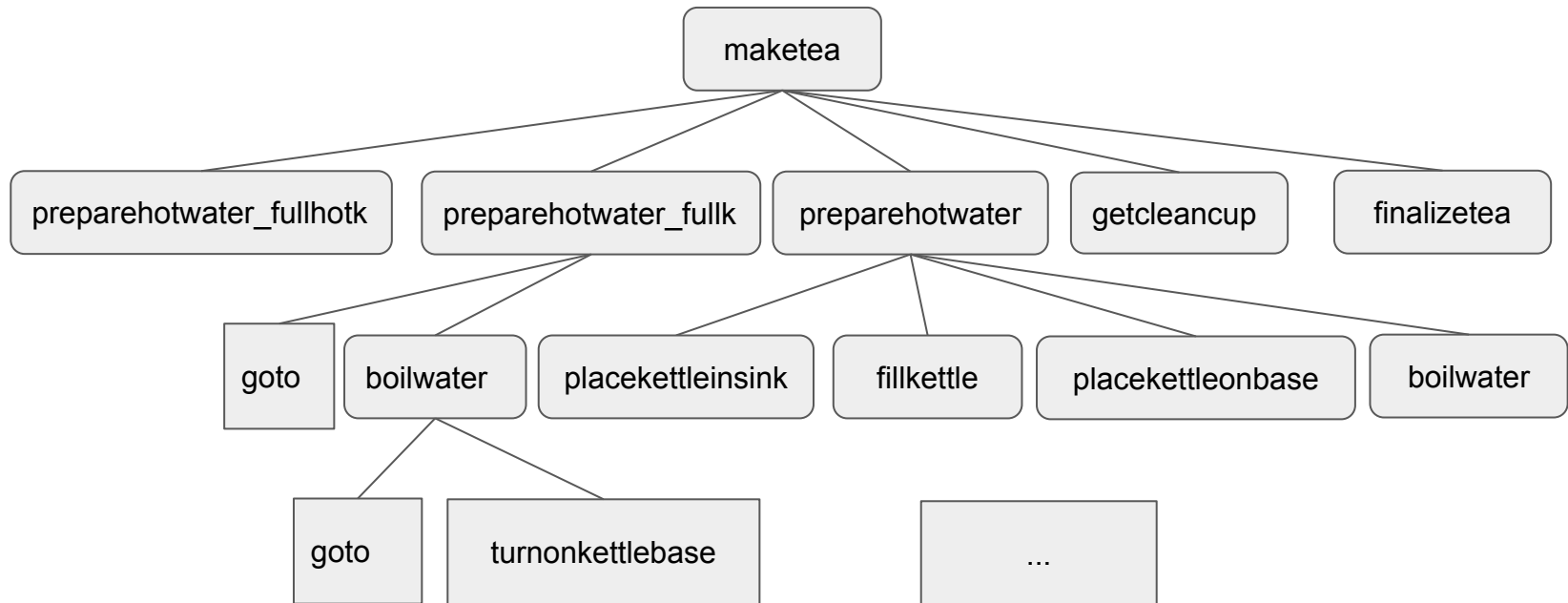
# Problem

- Partially observable domain
- Incomplete information
- HTN Planner
- Domain: kitchen room with different locations
- Either clean or dirty teacups (with unknown state)
- Robot has to discover teacups

# Planner

- Pyhop HTN Planner
- Easy to install
- Portable (only python 2.7 needed)
- JSHOP difficulties to setup
- Variables instead of logical propositions
- No planning language, functions for HTN operators and methods

# Structure





# Strategy

- Checking of all relevant preconditions in operators - failure if not fulfilled
- Alternative methods for tasks which could fail due to preconditions
- Some checking in top-methods to select correct sub-method
- Domain knowledge: state (e.g. teacup clean/dirty)
- Teacups: randomize cup locations, state (dirty / clean)

# Tests

- Test 1 / 3: “Standard recipe”
- Alternate methods for failing preconditions (kettle full, kettle hot, ...)
- domain state fully known
- Test 2: “Standard recipe” with “cup-finder”:
- limited knowledge of teacups (dirty/ unknown)
- robot discovers teacups (clean/ dirty)

## Planning / acting

- Incomplete domain knowledge issue
- Idea: Execute plan until plan fails. Update domain knowledge, then replan
- For dirty cups: Execute “cup finder” first - repeat in case of dirty cup until all cups tested or clean cup found
- Stop when domain knowledge unchanged / plan finished



# Conclusion

- Pyhop easy to install and easy to use
- Bigger domain (e.g.  $|\text{teacups}| > 10,000$ ) already takes minutes
- More refinement means bigger domain
- High amount of methods for different domain states



## Online resources

- <https://bitbucket.org/dananaau/pyhop>
- <https://github.com/dpyka/TeaAtHomeHTN>