# Online Hierarchical Conformance Refinement Planning for Intelligent Autonomous Mobile Robots using Answer Set Programming

Oliver Michael Kamperis

Supervisors: Marco Castellani and Yongjing Wang

# The Objective

- Objective is to obtain a planner effective for very capable robots, that will be deployed into real-world domains, including our homes or work places, to do general tasks previously done only by humans.

- This thesis tackles the problem of plan generation in such domains,
- This needs a general method that is fast and finds good quality plans.

- Planning time and execution latency must be very small to be useful;
  - Search space explodes exponentially with plan length in exhaustive search,
  - Hard to avoid this and still get good quality plans for complex problems,
  - Even harder to find method that doesn’t use problem specific knowledge.

# ASP Based Planning

- ASP is a knowledge representation and reasoning language;
  - Highly expressive and general for modelling and solving problems,
  - Easy to write and expand encodings (very good elaboration tolerance).
- Fundamental physical laws of a system written as logical axioms;
  - Effects - “Moving changes your location and/or orientation”,
  - Conditions - “You can only move between locations that are connected”,
  - Constraints - “An object you are grasping always shares your location”.
- Performs well for problems with many complex interacting constraints and large number of relevant entity constants,
- Performs poorly for problems with long minimal plan lengths.

# Classical Heuristic Planning

- Classical heuristic planners have contrasting strengths to ASP.
- Heuristic planners use predictive model to estimate cost to goal;
  - Used to avoid searching most of the space of possible plans,
  - Resultantly can (usually) deal with long plan lengths very efficiently.
- They require ability to get accurate heuristic estimate efficiently;
  - Some trivially measured property of the domain (e.g. literal distance to goal),
  - Elaborate predictive models that will generalise to more complex problems.
- Only effective for real-world problems, if they are dealing with a limited representation, that allows the heuristic to work effectively.
- Not as expressive or general as ASP, often poor elaboration tolerance.

# Comparison

- ASP appears to have the capabilities to achieve our objective;
  - Well suited at representing complex problems with desired generality,
  - But these tend to have plan lengths that are too long for it to deal with.
- Heuristic planners beat ASP for problems that are simple enough to get accurate heuristic estimates, or when a lot of prior knowledge of the problem is available, even when plan lengths are very long.
- We can’t rely on heuristics to enhance ASP for multiple reasons;
  - Types of problems ASP used for are too complex to rely on heuristic guidance only,
  - Heuristics can make the logic program exponentially larger,
  - Even if we can find one, we might lose the desired generality.
- If we want the benefits of ASP we need some alternative way of dealing with these long plans and high planning complexity.

# Hierarchical Refinement Planning

- In place of a heuristic an abstraction hierarchy is used,
- Refines abstract plans down an abstraction hierarchy to progressively divide and sub-divide problems into smaller sub-problems;
  - Divide-and-conquer can exponentially reduce plan lengths and planning complexity,
  - Good where obtaining an accurate heuristic estimate too difficult or expensive.
- However, existing planners only support relaxed and reduced models;
  - These are not always appropriate for all domain/problems,
  - Relaxed models can make planning harder if branching factors are increased.
- Planning time and plan quality *can* be exponentially worse;
  - Not always an effective way to divide a given problem,
  - If sub-problems are not independent plans can approach exponentially longer.

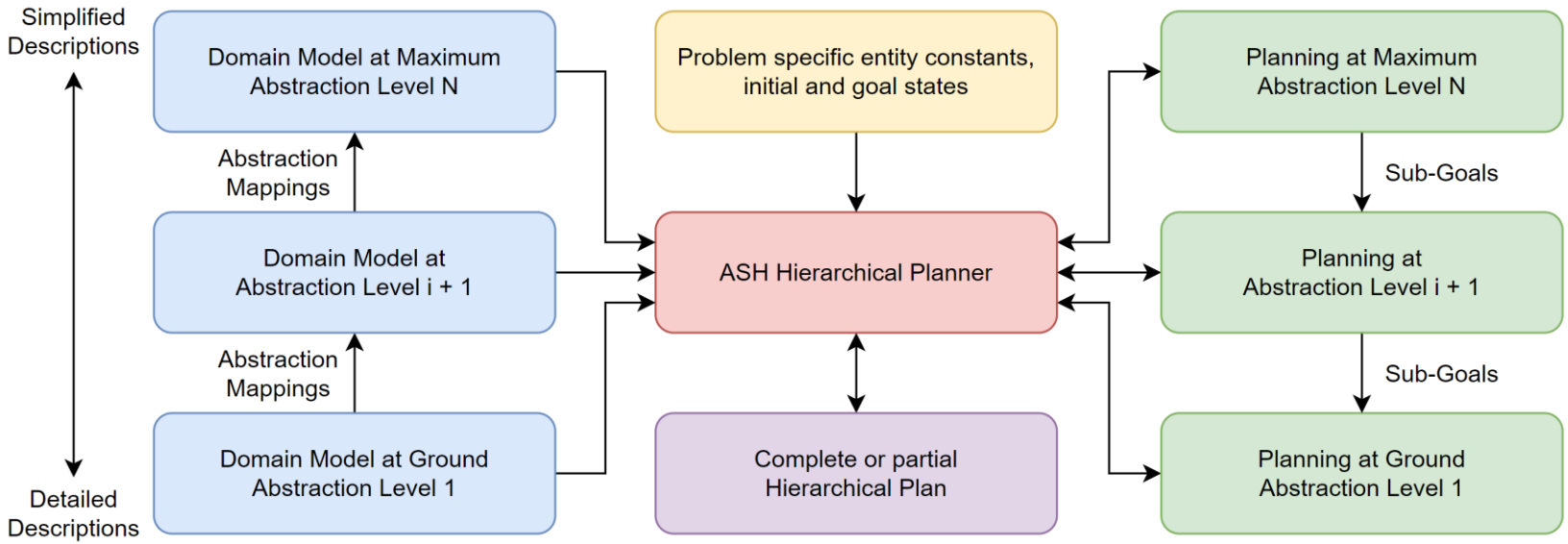
# The Proposed Approach

- ASP based and refinement planning have complimentary strengths, this thesis seeks to show they overcomes weaknesses of each other.
  - ASP very well suited to representing abstraction hierarchies,
  - Refinement very well suited to dealing with long plan lengths.

- *Hierarchical Conformance Refinement Planning*: An approach to ASP planning based on refinement which relaxes many of the limiting restrictions of existing techniques on both fronts;
  - Novel highly versatile abstraction hierarchy representation allows for much larger variety of different abstract models for different domains,
  - New flexible constraint based refinement method based on sub-goal planning alleviates the dependency problem that affect existing refinement planners.

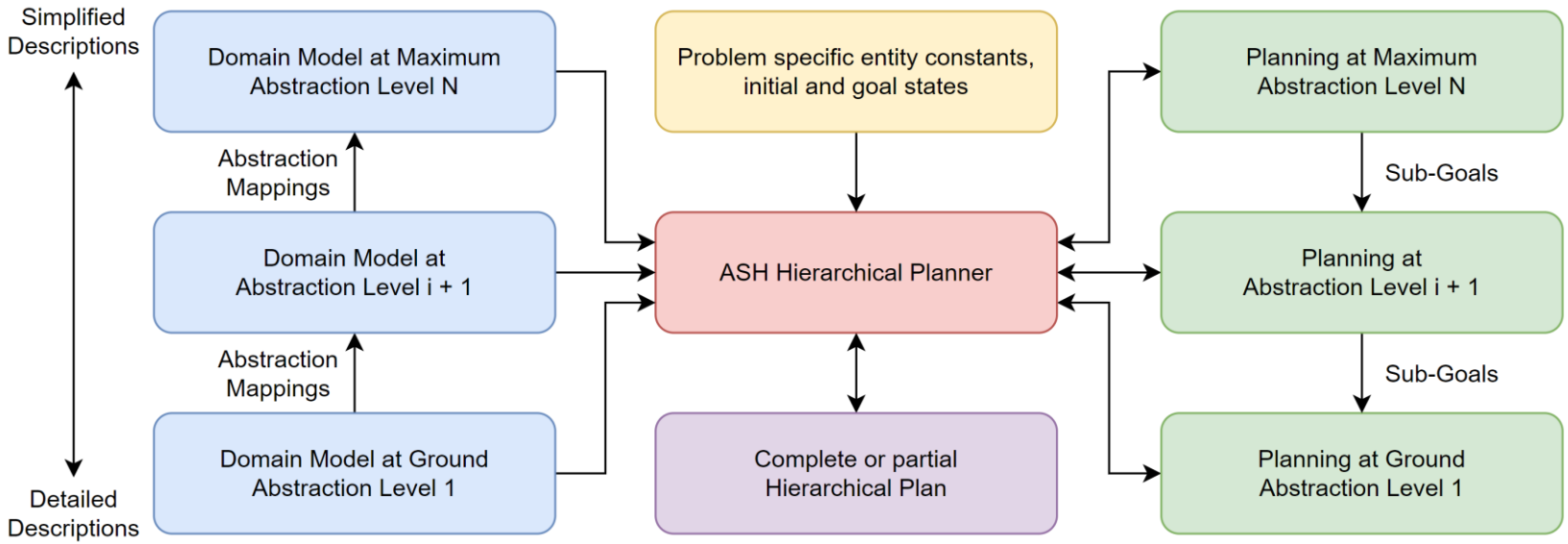
# Conformance Refinement Planning

- Abstraction hierarchy containing arbitrary number of abstract models allows for progressive simplification of a domain/problem,
- Abstract plans refined successively over an abstraction hierarchy, under simple constraint that requires that plans at different levels achieve the same effects and remain structurally similar.



# Conformance Refinement Planning

- The abstract plans produce a sequence of fundamental sub-goal stages, that provide understanding of the structure of the plan at the original level, and increase planning speed from two properties;
  - Conformance constraints formed by the sub-goal stages restrict search space,
  - Optional division of problems to exponentially simpler partial-problems.



# Abstraction Hierarchies

- Can represent of any form of abstraction that can be expressed logically and mapped to deterministically from the original model,
- Two novel model types are contributed;
  - Condensed models work by reducing size of state representation,
  - Tasking models work by achieving a level of high-level task planning.
- An abstraction hierarchy has to be designed for a specific domain,
- The generality of a hierarchy is over different problems in the domain.
- For complex domains, this hierarchy may be easier to obtain than an equivalent heuristic for classical planning, but only if there are properties of the domain that allow a hierarchy to be made.

# Conformance Refinement Planning

- Primary benefits over existing ASP based planners;
  - Maintains the expressivity and generality,
  - Finds complete plans exponentially faster than existing ASP based planning,
  - Ability to plan online reduces execution latency exponentially (to seconds).
- Rarely finds the optimal plan but does find some plan of acceptable quality.

- Most complex problem tested so far
  - Initial partial solution found in ~12 seconds,
  - Complete solution found in ~5 minutes,
  - Estimated ~16% longer than optimal plan.
- This cannot even be solved realistically by existing ASP planners, takes over an hour to reach half minimal plan length, with exponential trend.

# Progress and Changes

- Planner implementation complete,
- Currently running experiments and plan for thesis chapters/structure,
- For the problems I have tested so far, I have managed to achieve the objective, and successfully combine ASP with refinement.

- Due to time constraints I have removed;
  - Diagnostics, re-planning, and plan repair,
  - Multiple robots,
  - Informed division strategies,
  - Minimum step bound estimation,
  - Scaled back to just one planning domain, but increased variety of problems.