GRIFFITH UNIVERSITY

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

3806ICT Robots, Agents and Reasoning

Trimester 1, 2024

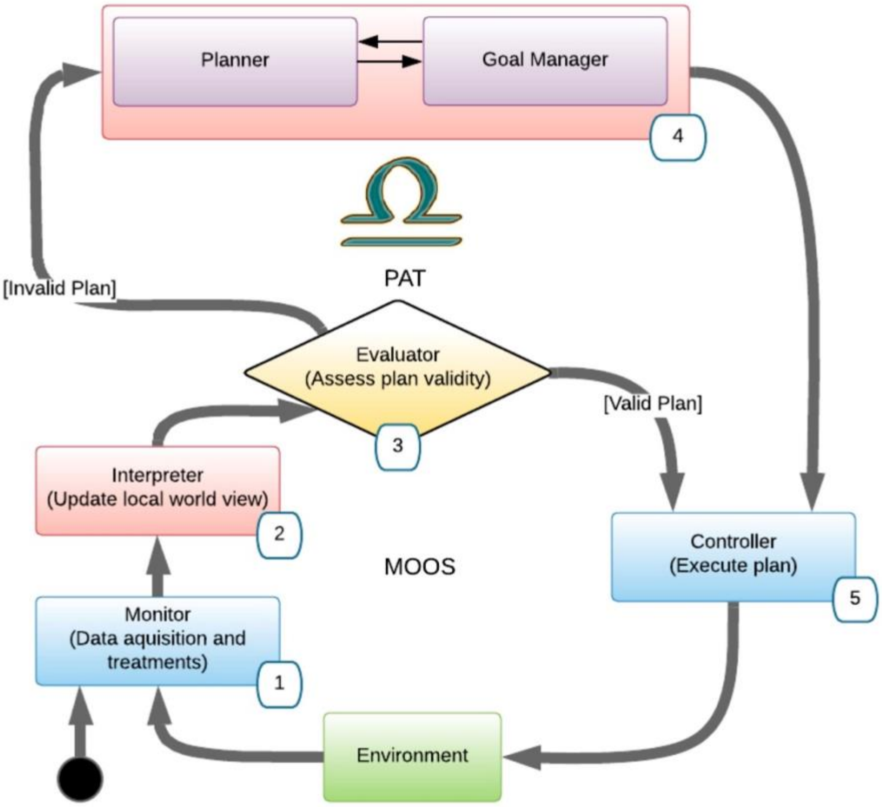
Assignment 3 (40%)

Due Time: 5pm Friday, 31st May 2024

Assignment overview and requirements

In this assignment, each group of students is expected to develop a system that combines ROS and AI decision-making to perform software robot tasks in *a team of multiple* agents setting and submit a written report that discusses the work.

The system for *individual* agents should follow the general idea (not necessarily the details; e.g., the choice of MOOS and PAT may be replaced) of Bride et al.'s work [1], which is given below.



That is, the AI decision-making component performs high-level (abstract) planning and goal reasoning, and ROS performs how-level actuation and interaction with the environment.

Your system is expected to use ROS instead of MOOS. Also, you do not have to implement all the modules in the above figure as long as it satisfies the requirements listed below.

- Your system, i.e., a (set of) program(s), should integrate AI decision-making and ROS into a holistic solution. It should contain the following modules:
  - A module interacts with the environment and obtains the readings from sensors.
  - A module processes sensor readings and generates the agent's state and the agent's understanding (e.g., a map) of the environment.

  - A module interacts with AI by passing in the agent's state and the abstract environment (e.g., an abstract/discrete representation of a map) (e.g., a CSP# model if you choose PAT) to the AI. This module also waits for the result given by the AI.
  - A module takes the AI's result and input and translates it into a low-level plan that can be executed by ROS. This module also interacts with ROS to obtain the result of the execution. The execution changes the environment, which triggers the first module to perform another interaction until the goal is achieved.
- The environment is related to the application of your work and is up to you. The simplest case could be a 2D grid world pathfinding, which is covered in the course, but feel free to be creative.
- You will need to program a "simulator" of the environment, which is necessary to run your system, but the environment (simulator) is external to your system. That is, your system should be general and works for different applications.
- You may build upon the work from previous years' students assignments (provided in a separate link).

IMPORTANT: On top of the above system for individual agents, you are expected to extend it (the previous year's assignment) to handle the interaction of *multiple* agents. This can be done in different ways. For example:

  - You may use federated reinforcement learning, in which case your AI decision-making may be driven by reinforcement learning.
  - If your AI decision-making is driven by PAT automated reasoning, then you can orchestrate the team of agents through hierarchical model checking via N-PAT.

- You will need to conduct an experiment to evaluate the performance of your system.
- Everything not specified here and in the marking rubric is up to your interpretation and implementation. These might include:
  - The programming language, libraries and packages.
  - The application and the environment specifics.
  - The type of planning and reasoning performed by AI.
  - The tasks actuated by ROS.
  - Etc.

Instructions for writing the report

- The report should be in LNCS format. Do not change font size, margin size etc.
  - Word template can be found here:

https://www.springer.com/gp/authors-editors/conference-proceedings/editors/word-template/19338734

  - LaTex template can be found on Overleaf:

https://www.overleaf.com/latex/templates/springer-lecture-notes-in-computer-science/kzwwpvhwnvfj

  - The report should be *no more than* **16 pages, including references**.

- Your report should have an abstract.
    - No more than 500 words.
    - Give an overview of the topic and what you have learned.
- Your report should have an introduction section.
    - Describe the background of the topic and why it's important.
    - Give an overview of the state-of-the-art/related methods.
    - Give an overview of your work.
- Your report should have a related work section.
    - Discuss related work in the literature.
    - Choose recent work that represents the state-of-the-art.
    - This section essentially expands the overview in introduction with more details.
    - At the end of this section, give a comparison of your work with the related methods.
- Your report should have a "proposed approach" section.
    - Describe your work in detail.
    - Focus on the conceptual design.
- Your report should have an implementation and experiment section.
    - Describe the implementation details (e.g., programming language, libraries, important programming specifics, but no need to show code unless it is significant.)
    - Describe the experiment setup (e.g., OS, hardware specs, etc.).
    - Demonstrate your system in different applications/examples. Show the results about computational time and resources needed.
    - OPTIONAL: compare with related work's implementations if available.
    - Discuss and analyse the results.
- Your report should have a discussion/conclusion section at the end.
    - Summarise the most significant ideas you have learned.
- The order of sections is up to you.
- Use figures and tables to present your ideas.
    - Vectorised images are preferred. If other images (e.g., screenshots) are used, ensure that they do not look blurry/jagged on a 200+ PPI (e.g., 4K 21-inch or 5K 27-inch) screen.
- Cite existing work, including all the libraries and code you used that are not developed by yourselves.
    - The reference should contain the authors, the paper/project title, the year of publication, and the venue of publication (name of conference/journal).
    - In the case of code or library, venue of publication is optional, but provide URL.
- Acknowledge the usage of generative AI. Document the queries and responses in an Appendix (exclude from page limit). Do not use generative AI for writing.

SUBMISSION

Each group submits ONE PDF copy of the report to L@G (Canvas) / 3806ICT / Assessments / Assignment 3 -> Group Report

Each student submits a PDF copy of the peer review form to L@G (Canvas) / 3806ICT / Assessments / Assignment 3 -> Individual Review Form

Late submissions: marks will be deducted according to university policy.

References

[1] Hadrien Bride, Jin Song Dong, Ryan Green, Zhé Hóu, Brendan Mahony, Martin Oxenham, GRAVITAS: A model checking based planning and goal reasoning framework for autonomous systems, Engineering Applications of Artificial Intelligence, Volume 97, 2021, 104091, ISSN 0952-1976, https://doi.org/10.1016/j.engappai.2020.104091.