The ACROBA project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017284.

Task planner

Thomas Ribeaud, Berner Fachhochschule

ResearchXchange, 13. December 2021
Plan

- ACROBA Architecture
- Plansys2
- Use cases
- Task Planner architecture
- Questions/Inputs
ACROBA Architecture

• AI-Driven Cognitive Robotic Platform for Agile Production Environments
• Flexible production solution for mid-range companies
• Modular collaborative or standard platform
ACROBA Cognitive module
ACROBA Task Planner

- Executor of the task
- Help to design the task
- Task optimization
- Online re-planning
- High level of automation
BehaviorTree.cpp

- Library to handle behavior trees from MOOD2Be project: Models and Tools to design Robotic Behaviors (2019)

Figure 8: Check if robot is localized and emergency stop is pressed.

Figure 2: Relation between an Action of the BT and a service-oriented component.

Figure 10: Relation between Groot and the C++ Executor
Plansys2

- ROSPlan for ROS2
- Can generate, execute and display Behaviour trees
- Generation from PDDL with several solvers available
PDDL

- Planning Domain Definition Language
  - Domain -> Predicates (properties) and operators (Actions) = Robots/Tools/Parts and Robot skills
  - Problem -> Goal = Domain state to reach by using actions

- Evolution of PDDL over time

(define (domain factory)
  (:requirements :strips :typing :adl :fluents :durative-actions)

;; Types;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(:types
gripper piece zone
);; end Types;;;;;;;;;;;;;;;;;;;;;;;

;; Predicates;;;;;;;;;;;;;;;;;;;;;;;
(:predicates
(piece_at ?p - piece ?pz - zone) = true if part p is in zone pz
(gripper_at ?g - gripper ?z - zone) = true if gripper g is in zone z
(gripper_free ?g - gripper) = true if gripper g doesn't hold a part
(gripper_has ?g - gripper ?p - piece) = true if gripper g holds part p
);; end Predicates;;;;;;;;;;;;;;;;;;;

;; Actions;;;;;;;;;;;;;;;;;;;;;;;;;;
(:durative-action pick
 :parameters (?g - gripper ?z1 - zone ?p - piece)
 :duration ( = ?duration 4)
 :condition (and
   (at start(gripper_at ?g ?z1))
   (at start(piece_at ?p ?z1))
   (at start(gripper_free ?g))
 )
 :effect (and
   (at start(not (piece_at ?p ?z1)))
   (at start(not (gripper_free ?g)))
   (at end(gripper_has ?g ?p))
 )
)
Use case example

- 5 use cases
  - Collaborative assembly lines
    - IKOR: PTHs on PCBs
    - ICPE: Electric motors parts
  - Light out manufacturing
    - STER: Processing of medical 3D printed parts
    - CABKA/MOSES: Defects removal and QC
  - Let's use IKOR as support for presentation
Task planner at engineering time

- GUI planning layer
- 3D simulation
- Edit in a user-friendly tool
- Edit the new sub-problems (plug-ins) like CAD-based task planner or kCOR assembly order
- Domain definition
- Skills definition
- Problem(s) definition
- Domain conversion in PDDL
- Skills conversion in PDDL (actions)
- Problem(s) definition in PDDL (goal)
- Behaviors tree
- Task behavior tree
- Visualization and check

Launch
Task planner at engineering time
Task planner at runtime
Task planner at runtime

• Online re-planning in case of human operator perturbation
Task planner overview

- **GUI planning layer**
  - Domain definition
  - Skills definition
  - Problem(s) definition

- **GUI HMI layer**
  - Task Planner
  - Behavior tree
  - Visualization and Check
  - BehaviorTree.cpp Executor in ROS2

- **3D simulation**

- **Offline design**
  - Domain definition
  - Domain conversion in PDDL
  - Skills conversion in PDDL (actions)
  - Problem(s) definition in PDDL (goal)

- **Runtime**
  - New problem definition in PDDL
  - Sub-trees dynamically loaded
  - Skills execution
  - Sometimes new problem
  - New problem definition in PDDL
  - Skills unchanged

- **Launch**
  - Edit/execute/simulate BT
  - Launch and control execution

- **Bridge to ROS1**

- **Edit the new sub-problems (plugins) to be dynamically solved (possibilities: IKOR assembly order check, CABRA best detours trajectory, STER optimization)**

- **Runtime optimization of task/problem sometimes needed**
  - Must be brought back to ROS2

- **Edit in a user friendly nice tool**
CAD based task planning

- Plug-in for the task planner
- Step files
- Info needed:
  - Relative positions of the parts
  - Relative direction of the parts
  - Movement type of assembly
  - Assembly order
  - Where to find the parts
- GUI or other way for the designer to give order of assembly
Questions/ Remarks /Inputs
The ACROBA project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017284.

Thank you for your attention!

Visit our website www.acrobaproject.eu
To get up-to-date information on demo sessions, business coaching, trainings, design contests, etc.

Follow us on our social networks:

The ACROBA project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101017284.