Examples explanation

Eric Bernd Gil - ericbgil@gmail.com
August 24, 2021

Abstract

This document aims at providing explanation on the examples to be completed by users in order to answer the form available at `https://docs.google.com/forms/d/e/1FAIpQLSen3_kr3OSRajoFQjiA4cKIFfm6y2U9HP3hTaKfVBCo595Njw/viewform?usp=sf_link`. After completing the examples, one should answer the form in order to give its thoughts about the MutRoSe framework.

1 Example 1 - "Keeping Clean"

1.1 Introduction

The "Keeping Clean" example is taken from the RoboMax [1] list of exemplars. For this work, some aspects were removed since they are not modelled in any of the models used in the framework. The description of this example is given below.

Example: Keeping Clean

Every room that reaches more than 1 cfu/cm^2 of Staphylococcus aureus must be cleaned. The assigned robot must check if it has all the resources to fulfill the tasks. In the case of missing equipment, the robot must go to the storage room to collect the equipment or assign the FindEquipment task to a colleague. As the robot reaches the room, it must check if it is occupied. If the room is occupied, a message should be sent to the manager and the mission aborted. Otherwise, the robot should enter the room and mark it as occupied. The cleaning task must be performed in order: (i) change the furniture's covers, towels, and clothes; (ii) vacuum the floor, moving furniture when necessary; (iii) wipe the floor, and (iv) sterilize all furniture and equipment in the room.

1.2 Further Explanation

For this example, one is required to model the Goal Model and to complete variable mappings in the configuration file. The user is given: (i) the HDDL Domain definition, (ii) the partial configuration file with semantic mappings defined and (iii) the world knowledge. It is important to note that OCL types must be the same as the ones present in the world knowledge, which is basically the *Room* type. Listings 1, 2 and 3 show the given structures, which can be found in `https://drive.google.com/drive/folders/1c5AtnBTNXXa_GhsPxCZzPG8XmTr7n8zz?usp=sharing`.

Listing 1: HDDL Domain definition for the "Keeping Clean" example

```
(define (domain hospital)
    (:types room - object)
    (:predicates
        (equipmentok ?r - robot)
        (occupied ?rm - room)
    )
    (:functions
        (staphylococcus ?rm - room)
    )
    (:capabilities organization equipmentsearch sndmessage)

    (:task AbortMission :parameters (?r - robot ?rm - room))
    (:method abort-mission
       :parameters (?r - robot ?rm - room)
       :task (AbortMission ?r ?rm)
       :precondition (and
                  (occupied ?rm)
       )
       :subtasks (and
                (msg-to-mgr ?r ?rm)
```

```
            )
    )
    (:action msg-to-mgr
        :parameters (?r - robot ?rm - room)
        :required-capabilities (sndmessage)
        :effect ()
    )

    (:task FindEquipment :parameters (?r - robot ?eqrm - room))
    (:method find-equipment
        :parameters (?r - robot ?eqrm - room)
        :task (FindEquipment ?r ?eqrm)
        :precondition (and
            (not (equipmentok ?r))
        )
        :subtasks (and
            (get-equipment ?r ?eqrm)
        )
    )
    (:action get-equipment
        :parameters (?r - robot ?eqrm - room)
        :required-capabilities (equipmentsearch)
        :effect (and
            (equipmentok ?r)
        )
    )

    (:task EnterRoom :parameters (?r - robot ?rm - room))
    (:method enter-room
       :parameters (?r - robot ?rm - room)
       :task (EnterRoom ?r ?rm)
       :precondition (and
                  (not (occupied ?rm))
       )
       :subtasks (and
                (mark-room ?r ?rm)
            )
    )
    (:action mark-room
        :parameters (?r - robot ?rm - room)
        :required-capabilities (sndmessage)
        :effect (and
            (occupied ?rm)
        )
    )

    (:task ChangeCovers :parameters (?r - robot ?rm - room))
    (:method covers-change
        :parameters (?r - robot ?rm - room)
        :task (ChangeCovers ?r ?rm)
        :precondition ()
        :subtasks (and
            (change-covers ?r ?rm)
        )
    )
    (:action change-covers
        :parameters (?r - robot ?rm - room)
        :required-capabilities (organization)
    )

    (:task VacuumFloor :parameters (?r - robot ?rm - room))
    (:method floor-vacuum
        :parameters (?r - robot ?rm - room)
        :task (VacuumFloor ?r ?rm)
        :subtasks (and
            (vacuum-floor ?r ?rm)
        )
    )
    (:action vacuum-floor
        :parameters (?r - robot ?rm - room)
        :required-capabilities (organization)
    )

    (:task WipeFloor :parameters (?r - robot ?rm - room))
    (:method floor-wiping
        :parameters (?r - robot ?rm - room)
        :task (WipeFloor ?r ?rm)
        :subtasks (and
            (wipe-floor ?r ?rm)
        )
    )
    (:action wipe-floor
        :parameters (?r - robot ?rm - room)
        :required-capabilities (organization)
    )

    (:task SterilizeRoom :parameters (?r - robot ?rm - room))
    (:method room-sterilization
        :parameters (?r - robot ?rm - room)
        :task (SterilizeRoom ?r ?rm)
        :subtasks (and
            (sterilize-room ?r ?rm)
        )
    )
    (:action sterilize-room
        :parameters (?r - robot ?rm - room)
        :required-capabilities (organization)
        :effect (and
            (assign (staphylococcus ?rm) 0)
        )
    )
)
```

Listing 2: JSON configuration file for the "Keeping Clean" example

```
{
    "world_db": {
        "type": "file",
        "file_type": "xml",
        "path": "knowledge/world_knowledge.xml",
        "xml_root": "world_db"
    },

    "output": {
        "output_type": "file",
        "file_path": "output/task_output.json",
        "file_type": "json"
    },

    "location_types": ["Room"],

    "type_mapping": [
        {
            "hddl_type": "room",
            "ocl_type": "Room"
        }
    ],

    "var_mapping": [

    ],

    "semantic_mapping": [
        {
            "type": "attribute",
            "name": "staphylococcus",
            "relates_to": "Room",
            "belongs_to": "world_db",
            "mapped_type": "function",
            "map": {
                "pred": "staphylococcus",
                "arg_sorts": ["room"]
            }
        },
        {
            "type": "attribute",
            "name": "occupied",
            "relates_to": "Room",
            "belongs_to": "world_db",
            "mapped_type": "predicate",
            "map": {
                "pred": "occupied",
                "arg_sorts": ["room"]
            }
        },
        {
            "type": "attribute",
            "name": "equipment_ok",
            "relates_to": "robot",
            "belongs_to": "world_db",
            "mapped_type": "predicate",
            "map": {
                "pred": "equipmentok",
                "arg_sorts": ["robot"]
            }
        }
    ]
}
```

Listing 3: Knowledge file for the "Keeping Clean" example

```
<world_db>
    <Room>
        <name>RoomA</name>
        <staphylococcus>3</staphylococcus>
        <occupied>False</occupied>
    </Room>
    <Room>
        <name>RoomB</name>
        <staphylococcus>1.6</staphylococcus>
        <occupied>True</occupied>
    </Room>
    <Room>
        <name>EquipmentRoom</name>
        <staphylococcus>0</staphylococcus>
        <occupied>False</occupied>
    </Room>
</world_db>
```

2 Example 2 - "Room Preparation"

2.1 Introduction

The "Room Preparation" example is an example created for this work, where rooms in a hospital domain must be prepared (i.e., cleaned and having its furniture correctly placed). The description of this example is given below.

Example: Room Preparation

For every room that its not prepared, robots must be assigned in order to prepare it for incoming patients. For each room there are basically two goals to be achieved:

- The room must be cleaned, where this task must be performed by a single robot with the cleaning capability. If the door is open the robot simply enters the room to clean it but if it's not it must perform some action in order to open it before cleaning. After cleaning, the robot must go the the *SanitizationRoom* in order to be sanitized.

- Furniture inside the room must be rearranged, where this task must be performed by a group of robots from 2 to 4 robots and requires the moveobject capability. In order to achieve this goal the room must be clean, so the team of robots must wait for the cleaning robot to finish cleaning. Despite having to wait for cleaning, the team of robots that will move furniture do not have to wait for the cleaning robot to be sanitized.

In addition to these goals, a robot must be sent to assert the cleaning of rooms. This assertion of cleaning is a task that is performed in a group of a rooms by a single robot, so it receives a group of rooms as location parameter.

2.2 Further Explanation

For this example, one is required to model the Goal Model and the methods of the HDDL Domain definition. One is also required to complete variable mappings in the configuration file. The user is given: (i) the partial HDDL Domain definition, with tasks and actions already defined, (ii) the partial configuration file and (iii) the world knowledge. As is the case of the "Keeping Clean" example, OCL types must be the same as the ones in the world knowledge. Listings 4, 5 and 6 show the structures, which can be found in `https://drive.google.com/drive/folders/19Kf6nQ04cppBaeHy-8eK04_fLgqY6mN5?usp=sharing`.

Listing 4: Partial HDDL Domain definition for the "Room Preparation" example

```
(define (domain hospital)
    (:types room - object
            MoveRobot CleanerRobot - robot)
    (:predicates
        (clean ?rm - room)
        (sanitized ?r - robot)
        (prepared ?rm - room)
        (door-open ?rm - room)
        (clean ?rms - rooms)
    )
    (:capabilities moveobject cleaning sanitize door-opening)

    (:task CheckCleanRooms :parameters (?r - robot ?rms - rooms))

    (:action clean-rooms-check
        :parameters (?r - robot ?rms - rooms)
        :precondition (clean ?rms)
        :effect ()
    )

    (:task RoomCleaning :parameters (?r - CleanerRobot ?rm - room))

    (:action clean-room
        :parameters (?r - CleanerRobot ?rm - room)
        :required-capabilities (cleaning)
        :precondition (and
            (not (clean ?rm))
        )
        :effect (and
            (clean ?rm)
            (not (sanitized ?r))
        )
    )
    (:action open-door
        :parameters (?r - CleanerRobot ?rm - room)
        :required-capabilities (door-opening)
        :precondition (and
            (not (door-open ?rm))
        )
        :effect (and
            (door-open ?rm)
        )
    )

    (:task FurnitureMoving :parameters (?rt - robotteam ?rm - room))

    (:action move-furniture
        :parameters (?rt - robotteam ?rm - room)
        :required-capabilities (moveobject)
        :precondition ()
        :effect (and
            (prepared ?rm)
        )
    )

    (:task RobotSanitization :parameters (?r - CleanerRobot ?srm - room))

    (:action sanitize-robot
        :parameters (?r - CleanerRobot ?srm - room)
        :required-capabilities (sanitize)
        :precondition (and
            (not (sanitized ?r))
        )
        :effect (and
            (sanitized ?r)
        )
    )
)
```

Listing 5: JSON configuration file for the "Room Preparation" example

```
{
    "world_db": {
        "type": "file",
        "file_type": "xml",
        "path": "knowledge/World_db.xml",
        "xml_root": "world_db"
    },

    "output": {
        "output_type": "file",
        "file_path": "output/task_output.json",
        "file_type": "json"
    },

    "location_types": ["Room"],
    "type_mapping": [
        {
            "hddl_type": "room",
            "ocl_type": "Room"
        },
        {
            "hddl_type": "rooms",
            "ocl_type": "Sequence(Room)"
        }
    ],

    "var_mapping": [

    ],

    "semantic_mapping": [
        {
            "type": "attribute",
            "name": "is_clean",
            "relates_to": "Room",
            "belongs_to": "world_db",
            "mapped_type": "predicate",
            "map": {
                "pred": "clean",
                "arg_sorts": ["room"]
            }
        },
        {
            "type": "attribute",
            "name": "is_prepared",
            "relates_to": "Room",
            "belongs_to": "world_db",
            "mapped_type": "predicate",
            "map": {
                "pred": "prepared",
                "arg_sorts": ["room"]
            }
        },
        {
            "type": "attribute",
            "name": "is_sanitized",
            "relates_to": "robot",
            "belongs_to": "robots_db",
            "mapped_type": "predicate",
            "map": {
                "pred": "sanitized",
                "arg_sorts": ["robot"]
            }
        },
        {
            "type": "attribute",
            "name": "door_open",
            "relates_to": "Room",
            "belongs_to": "world_db",
            "mapped_type": "predicate",
            "map": {
                "pred": "door-open",
                "arg_sorts": ["room"]
            }
        },
        {
            "type": "attribute",
            "name": "is_clean",
                "relates_to": "Sequence(Room)",
                "belongs_to": "world_db",
                "mapped_type": "predicate",
                "predicate_type": "Universal",
                "map": {
                    "pred": "clean",
                    "arg_sorts": ["rooms"]
                }
            }
    ]
}
```

Listing 6: Knowledge file for the "Room Preparation" example

```
<world_db>
    <Room>
        <name>RoomA</name>
        <is_clean>False</is_clean>
        <is_prepared>False</is_prepared>
        <door_open>False</door_open>
    </Room>
    <Room>
        <name>RoomB</name>
        <is_clean>False</is_clean>
        <is_prepared>False</is_prepared>
        <door_open>True</door_open>
    </Room>
    <Room>
        <name>RoomC</name>
        <is_clean>True</is_clean>
        <is_prepared>True</is_prepared>
        <door_open>True</door_open>
    </Room>
    <Room>
        <name>SanitizationRoom</name>
        <is_clean>True</is_clean>
        <is_prepared>True</is_prepared>
        <door_open>True</door_open>
    </Room>
</world_db>
```

3 Additional Notes

There are some additional notes to be given for users who will implement the examples:

- In case of any questions/difficulties, the authors email is at the beginning of the document. He will answer as soon as possible
- The reader of this document must read the document called "Instructions for MRS mission specification in the MutRoSe framework" for instructions on how to model what is necessary
- It is recommended to keep everything in the models very high-level, since this is the aim of the specification framework. It is not required though, if one wants to use lower level abstractions it is possible.

References

[1] M. Askarpour, C. Tsigkanos, C. Menghi, R. Calinescu, P. Pelliccione, S. García, R. Caldas, T. J. von Oertzen, M. Wimmer, L. Berardinelli, *et al.*, “Robomax: Robotic mission adaptation exemplars,” in *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, pp. 245–251, IEEE, 2021.