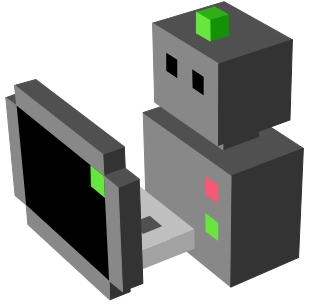
MORSE & HRI

Recent Perspectives

October 14, 2014

Séverin Lemaignan, Marc Hanheide,
Michael Karg, Harmish Khambhaita,
Lars Kunze, Florian Lier,
Ingo Lütkebohle and Grégoire Milliez



OVERVIEW

OVERVIEW
1. Brief Recap of MORSE
2. Simulating HRI
3. Less Usual Use-cases
4. What Next?

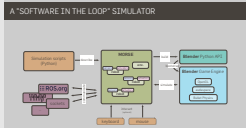
1. Brief Recap of MORSE
2. Simulating HRI
3. Less Usual Use-cases
4. What Next?

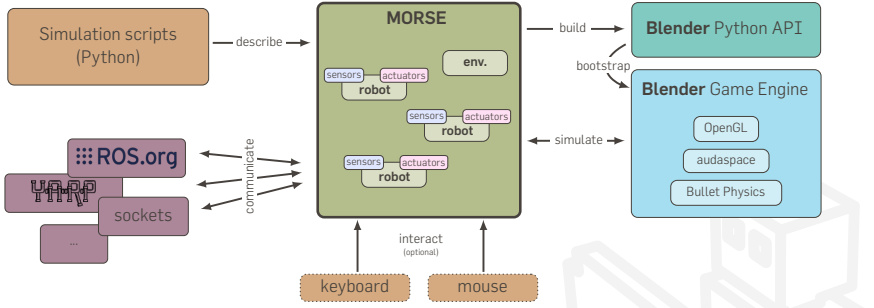
A few facts on MORSE:

- 100% open-source, permissive BSD-like license
- Latest release (1.2.1) in July 2014
- 25kLOC of Python code
- Academic project, baked by academics
- Initiated at LAAS-CNRS in 2009
- Contributions by 12 institutions worldwide

BRIEF RECAP OF MORSE

A "SOFTWARE IN THE LOOP" SIMULATOR





Describe the normal usage flow: Builder script → MORSE build the scene using Blender API, then start the BGE → further interactions via middlewares.

Humans are robots. We can optionally use keyboard + mouse to control them.

```
> morse create my_sim

> cd my_sim
> ls
default.py  scripts/  src/
```


How is it like to actually use MORSE?

Briefly explain default.py (illustrate the fact that robots are build out of actuators and sensors, and connected to one or several middlewares).

```
> morse create my_sim

> cd my_sim
> ls
default.py  scripts/  src/

> morse run my_sim
```

How is it like to actually use MORSE?

Briefly explain default.py (illustrate the fact that robots are build out of actuators and sensors, and connected to one or several middlewares).

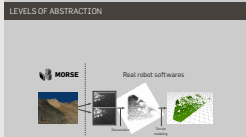
```
> morse create my_sim

> cd my_sim
> ls
default.py  scripts/  src/

> morse run my_sim

> cat default.py
```

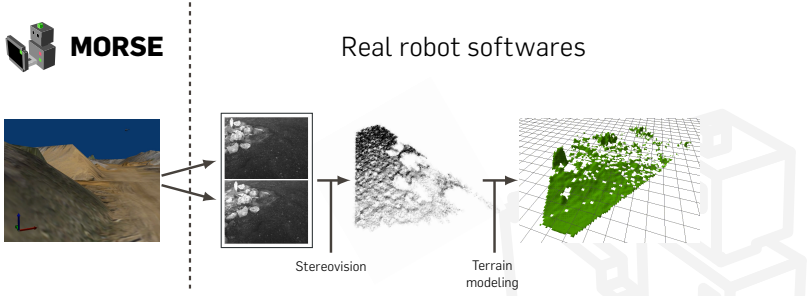
```
#! /usr/bin/env morseexec
from morse.builder import *
robot = Morsy()
robot.translate(1.0, 0.0, 0.0)

motion = MotionVW()
robot.append(motion)

keyboard = Keyboard()
robot.append(keyboard)

pose = Pose()
robot.append(pose)

robot.add_default_interface('socket')

env = Environment('sandbox')
env.set_camera_location([ 10.0,
                         -10.0,
                          10.0])
env.set_camera_rotation([1.05,
                            0,
                         0.78])
```

How is it like to actually use MORSE?

Briefly explain default.py (illustrate the fact that robots are build out of actuators and sensors, and connected to one or several middlewares).

LEVELS OF ABSTRACTION

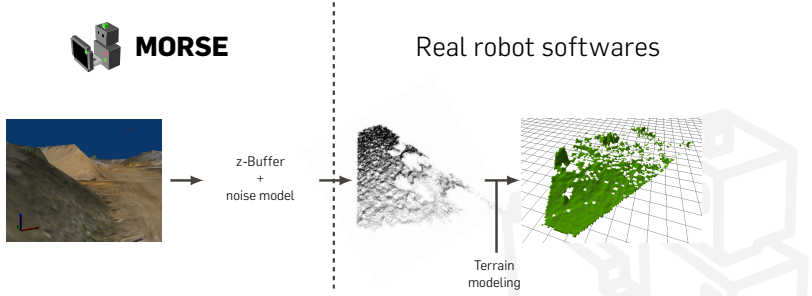




One of the strenghts of MORSE: the concept of 'abstraction level'.

- Fig1: MORSE only simulates the environment + cameras, remaining is done with the robot software stack
- Fig2: MORSE provides a depth image: no stereovision required
- Fig3: MORSE provides a terrain model: no reconstruction needed
- Fig4: semantic camera: MORSE provides the list of visible abstract objects.

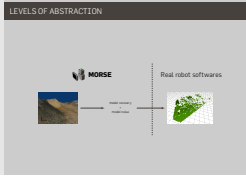
LEVELS OF ABSTRACTION

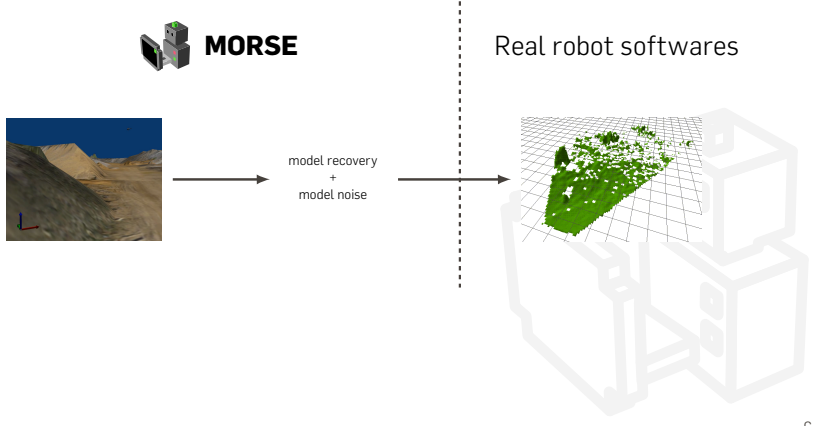


One of the strenghts of MORSE: the concept of 'abstraction level'.

- Fig1: MORSE only simulates the environment + cameras, remaining is done with the robot software stack
- Fig2: MORSE provides a depth image: no stereovision required
- Fig3: MORSE provides a terrain model: no reconstruction needed
- Fig4: semantic camera: MORSE provides the list of visible abstract objects.

LEVELS OF ABSTRACTION

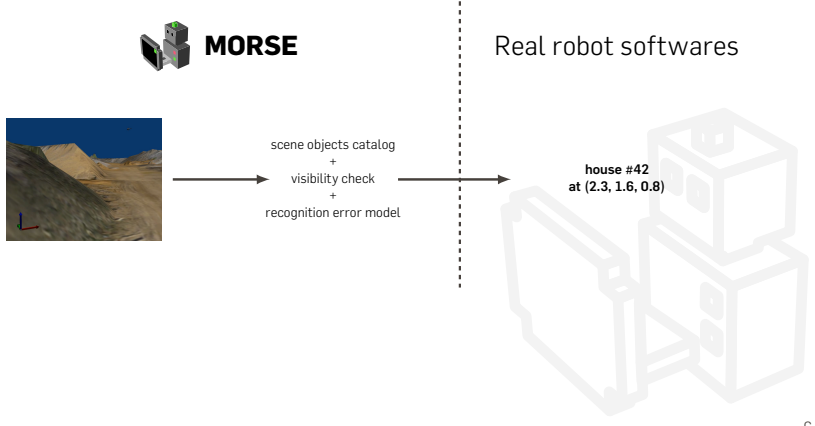




One of the strenghts of MORSE: the concept of 'abstraction level'.

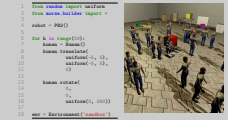
- Fig1: MORSE only simulates the environment + cameras, remaining is done with the robot software stack
- Fig2: MORSE provides a depth image: no stereovision required
- Fig3: MORSE provides a terrain model: no reconstruction needed
- Fig4: semantic camera: MORSE provides the list of visible abstract objects.

LEVELS OF ABSTRACTION



One of the strenghts of MORSE: the concept of 'abstraction level'.

- Fig1: MORSE only simulates the environment + cameras, remaining is done with the robot software stack
- Fig2: MORSE provides a depth image: no stereovision required
- Fig3: MORSE provides a terrain model: no reconstruction needed
- Fig4: semantic camera: MORSE provides the list of visible abstract objects.



```python
from random import uniform
from morse.builder import *

robot = PR2()

for h in range(30):
    human = Human()
    human.translate(
            uniform(-5, 5),
            uniform(-5, 5),
            0)

    human.rotate(
            0,
            0,
            uniform(0, 360))

env = Environment('sandbox')
```

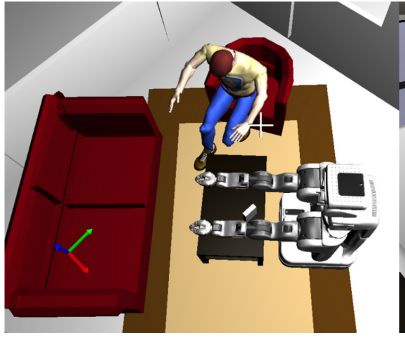
Describe how MORSE can easily be programmed to generate complex scenes.

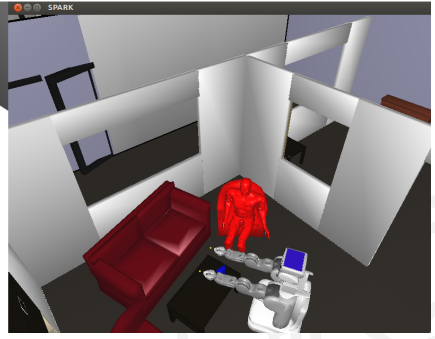
Note that in this example, humans won't move by themselves. For that, one need to add a motion actuator, and write and external script to control the crowd.


Not there yet...

SIMULATING HRI

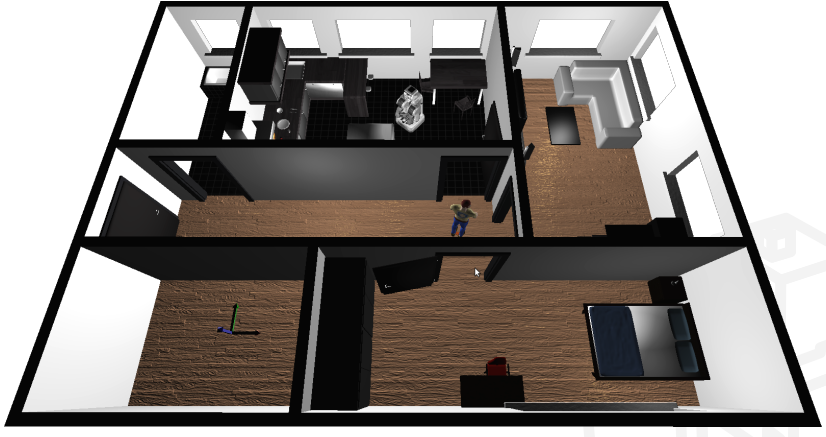
SITUATION ASSESSMENT WITH HUMAN





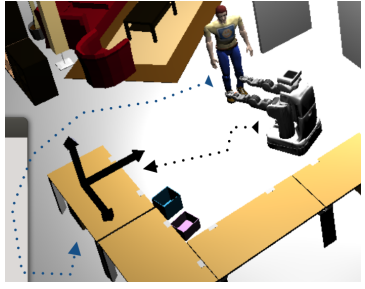
MORSE provides a virtual environment that we use to harness situation assessment algorithms that also include human-centered perspective taking. The robot updates its knowledge using its own position, human position and objects seen through abstracted, symbolic cameras provided by MORSE (so-called semantic cameras). In this particular scenario the human is sitting in a couch and ask the robot to bring specific objects that may be in another room (Pick-Place-Carry task).

LARGE SCENARII

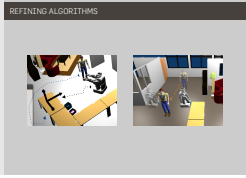


An apartment is simulated in which a domestic service robot is living together with a person. A PR2 robot is controlled via ROS and the Cram reactive plan language, which is used on several other real robots. The robots' duty is to observe the person performing different activities and detect unexpected situations based on the validation of different types of expectations. The detection of such unexpected behavior can help future domestic service robots to better assess situations and adapt their actions to human behavior.

REFINING ALGORITHMS



Testing path planning algorithms (without killing humans and destroying robots...)



BENEFITS

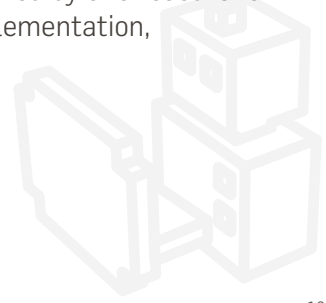
BENEFITS

Repeatable: fine for benchmarks, regression tests,
Abstraction levels: simulate only what is needed,
Computer game-style, tests can be carried by one researcher alone: support iterative design and implemantation,

Repeatable: fine for benchmarks, regression tests,

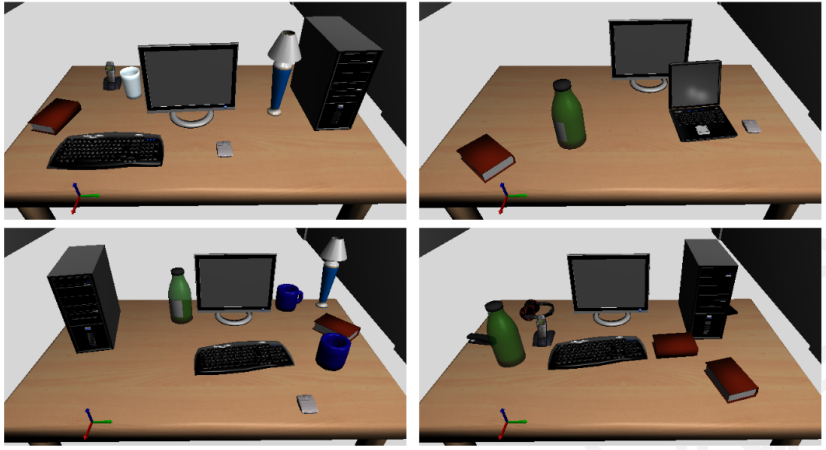
Abstraction levels: simulate only what is needed,

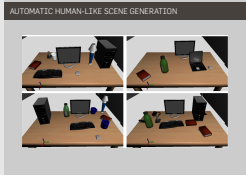
Computer game-style, tests can be carried by one researcher alone: support iterative design and implementation,



LESS USUAL USE-CASES

AUTOMATIC HUMAN-LIKE SCENE GENERATION





Aim: simulating credible human environments to train systems to appropriately react to them.

Robots need to know, when, where and how people manipulate objects and how they arrange and structure them in space → understand the long-term, spatio-temporal relationships of objects and activities of people.

We look at learning qualitative spatial relations of objects on office desks. As an accurate classification and pose estimation of objects on real-world office desks is still a challenging and difficult task for current robot perception systems we acquired a data set of object arrangements using the MORSE simulator. For this, we automatically generated a set of physically possible desktop scenes. Based on the generated data we learned relational models of object arrangements on desks. The learnt models enabled a robot to predict the position of an object given a landmark.

CONTINUOUS INTEGRATION AND HRI

In human-robot interaction studies, robots often indicate behavioral variability that may influence the experiment's final outcome. However, manual testing on physical systems is usually the only way to prevent this, but remains labour-intensive. To tackle this issue, we introduced early automated prototype testing.

The goal is to incrementally decrease the level of abstraction until a satisfactory/sufficient degree of "realism" to make an assumption about real world behavior is reached — in an integrated and continuous approach.

WHAT NEXT?

Soon there?

MAKEHUMAN INTEGRATION

More realistic human models, possibly procedurally generated.

WHAT NEXT?

Autonomous navigation of humans

WHAT NEXT?
Autonomous navigation of humans

WHAT NEXT?

Autonomous navigation of humans

Crowd simulation

WHAT NEXT?
Autonomous navigation of humans
Crowd simulation

WHAT NEXT?

Autonomous navigation of humans

Crowd simulation

Better library of motions (walk cycles, pick/place, emotions...)

WHAT NEXT?
Autonomous navigation of humans
Crowd simulation
Better library of motions (walk cycles, pick/place, emotions...)

WHAT NEXT?

- Autonomous navigation of humans
- Crowd simulation
- Better library of motions (walk cycles, pick/place, emotions...)

Open "HRI" issues on GitHub:
https://github.com/morse-simulator/morse/labels/HRI

WHAT NEXT?

Autonomous navigation of humans
Crowd simulation
Better library of motions (walk cycles, pick/place, emotions...)

Open "HRI" issues on GitHub:
https://github.com/morse-simulator/morse/labels/HRI

Thank you!
http://morse-simulator.github.io/

Thank you!

http://morse-simulator.github.io/