

## MODELING AND OPERATING ROBOTIC ENVIRONMENTS USING GAZEBO/ROS

Thomas Linner<sup>1\*</sup>, Alaguraj Shrikathiresan<sup>2</sup>, Maxim Vetrenko<sup>1\*</sup>, Bernhard Ellmann<sup>1</sup>, and Thomas Bock<sup>1</sup>

<sup>1</sup> *Chair for Building Realization and Robotics, Technische Universität München, Germany*

<sup>2</sup> *Building Technology and Construction Management, IIT Madras, India*

\* *Corresponding author ([vetrenko.maxim@gmail.com](mailto:vetrenko.maxim@gmail.com))*

**ABSTRACT:** Tools like ROS and Gazebo have become more important in light of new building concepts with embedded robotic systems. Over centuries buildings have always been viewed as static structures resistant to changes. However, today they are also viewed as kinetic structures capable of change and adaptation. The concept of kinetic architecture or dynamic buildings is currently being exploited by many architects around the globe. Some projects use these concepts to make the building more sustainable and intelligent while other projects employ these concepts for assistance. However, complex kinetic architectures have to be modeled, simulated and operated. As in architecture such a program currently is not available we deploy knowledge and tools from robotics. For this purpose, the best suited open source system Robot Operating System (ROS) enables you to create a virtual model of the world and integrate any types of sensors and robots. With the help of additional tools ROS allows to program or to perform manual management of behavior of individual objects or their parts. To simulate a kinetic, dynamic or assistive environment in ROS requires creating a 3D model of the environment. Next, the 3D model is divided into small parts, connecting with joints that have different properties in mobility. Furthermore, it is possible to create a model of human beings (set a certain degree of mobility for him, for example, a person in a wheelchair) to check overall performance of a robot assisted man-machine environment.

**Keywords:** *Kinetic Architecture, Building Information Modeling, Middleware, Robot Operating System (ROS)*

### 1. INTRODUCTION

“Miniaturization” and “Downscale” as basic forces of our technological development today enable a seamless integration of sensors, actors, control components and microelectronic systems into all subsystems, components and appliances [1]. At the same time, new approaches in computation link them to co-operate distributed (robotic) systems, which can respond actively or even proactive on certain use cases. Furthermore, experts and masterminds as for example Bill Gates announce the era of service robotics and estimate that service robotics as part of assisted environments will undergo a similar fast and rigid development as the spread of personal computers in private and economic areas since the nineties. Already in 1961 Joe Engelberger wondered if relegating robotic technologies to only industrial applications makes any sense. "The biggest market will be service robots," asserted Engelberger [2],

who started the industrial robotics era when his firm (Unimation) delivered GM's first robot. Today, robots and distributed robotic sub-systems start to permeate our everyday surrounding home/town environment enhancing it with services and additional performance. At the same time, this permeation is on the way to transform our perception of what robots are, robot technology, robots' possibilities and the environment they are merged with. This transformation which has to be understood as a natural part of the evolution of robotics will especially become visible when robots enter the field of service and assistance. [3]

Today the independent living and care systems in a conventionally designed and low-tech home is often unpleasant, and in many cases not suiting to counterbalance or the ease of disabilities. Especially when entering a later stage of life, changes of habits and lifestyle

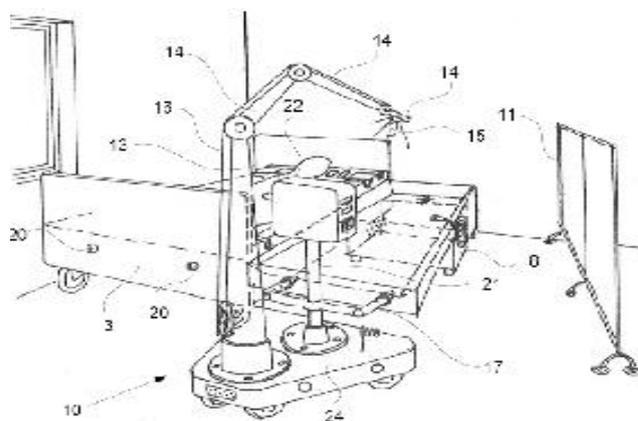
occur and unforeseen disorders, multi-morbidity and disorder progress, often make existing houses or flats inapplicable for elderly inhabitants. In many cases, a re-configuration of the existing home or the implementation of various assistive technologies as modular component systems would be needed to meet multiple needs with multiple sets of technologies [4]. Normally, the implementation of new technologies, sensors, actors, assistance devices and robotic sub-systems needed in a certain use case is a complex, costly and time consuming matter often forcing elderly people to move to a new home, or to set aside the idea of being supported by advanced assistance technology. So we present ways to model and test the assistive environments using ROS before executing them in real life. In section 2 we describe the system of robotic care environments. Section 3 describes the process of modeling and operating environments by ROS, and section 4 illustrates the concept of robotic buildings. In chapter 5 we summarize our results.

## 2. ROBOTIC CARE ENVIRONMENTS

Mechatronic systems, downscaled motors and lifting Systems, integrated sensors and acorns in combination with new technologies for human-machine-interaction can be accounted as sub-systems of immobile robotic systems. If such sub-systems are integrated and merged with everyday artefacts as furniture, appliances and whole buildings, those artefacts tend to change both, functionality and appearance. Emerging technologies are not only transforming functionality [5], but also designs and forms of our environment, following the principle of “form follows function” (Sullivan, 1896) being discovered and scientifically explained by L. Sullivan already more than one hundred years before [6].

In the future, technological complexity will be shifted from the care robot to the environment and distributed among various intelligent subsystems. The future care robot is supporting care takers or people in need. It will be complementary part of an intelligent care environment as well as sensors and actuators attached to doors, furniture, ceiling and so on, to enable an environment to support and take over robotic tasks. The future care robot can

cooperate with various subsystem components in its' surrounding melting robot and environment to an integrated and efficiently working entity. **Figure 1** shows the concept for a multi-joint robot as subsystem of a robotic care environment. Some segments include adapters that allow the attachment of different tools and connectors in order to facilitate the nurse/ care helper aid with modular equipment. All elements of the care environment from bed to the bathtub are designed as complementary part of the robotic care environment and have through plug and play interfaces the capability to couple with the robot informationally and/or physically to perform certain tasks as an entity.



**Fig. 1:** Multi-joint robot as sub-system of a robotic service/care environment [6]

## 3. ROS MODELING

Originally developed in 2007 by the Stanford Artificial Intelligence Laboratory in support of the Stanford AI Robot project [7], ROS is an open source operating system [8] for robots. It offers hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management [9]. It has enormous possibilities and allows simulation of gravity, creating a dynamic environment, different types of robots with many degrees of freedom, the creation of sensors of various kinds. The first step in modeling for an AAL in ROS requires creating a three dimensional model of the environment. The 3D model can be created on any of the modeling tools like 3dsMax, Maya or Auto Cad. Once the model is done it is exported in the

format of stl meshes to be integrated into the ROS environment. It is important to remember that all parts of the environment having a different degree of freedom of movement must be created separately. For example, in creating a room with a door, the walls and the door must be exported separately as the walls are static and the door is dynamic and has a certain degree of freedom for movement. After having created the 3D models of individual components they are combined in ROS to whole environments. This is possible with help of the package robot model, which contains the parser URDF files (Unified Robot Description Model). Also the URDF descriptions link different meshes/links through the concept of links and joints (Fig. 2). Two links connected through a joint are referred as a parent link and child link and the type of movement of each element in the model is determined by the type of joint between the two links. The joint type can be any one from the following:

- **Fixed:** As name suggests this type of joints are fixed and has no degree of freedom for movement
- **Revolute:** This type of Joint rotates around an axis and the movement is restricted within the specified upper and lower limits
- **Continuous:** This Joint is similar to revolute joint and it rotates around an axis with the exception of having no limits.
- **Prismatic:** This type of joint slides along an axis within the specified limits
- **Planar:** In this type of joint the movement is perpendicular to the specified axis
- **Floating:** This is not exactly a joint as all six degrees of freedom are free.

That URDF files contain a description of models and robots for import into the ROS. Such description includes the name of the object, its mesh-structure, the shell for the calculation of the collision, additional information about the visual components of an object, such as color, texture, and others. Consider a more detailed composition urdf-file

- **<robot>** tag contains the name of the robot, which will be displayed in all subsystems of ROS. This tag

contains a description of the whole AAL environment or part thereof (for example, a robot, an object or the sensor).

- **<link>** tag also contains the name and represents a visual component of the simulated environment, or an object that is connected to other objects with help of joints (<joint> tags). Inside tag <link> you can see a description of the visual part of the model (<visual> tag), which contains information about the geometry of the object <geometry>. Possible values - <cylinder>, <box>, <sphere> and <mesh>.

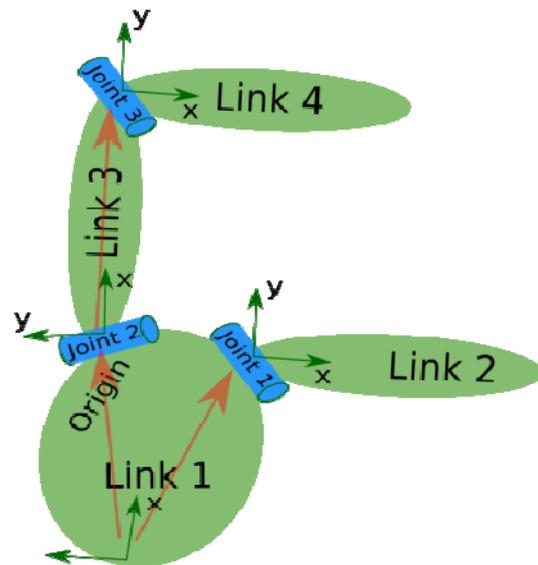


Fig. 2: Robotics structure using concept of links and joints

- Tag **<origin>** adjusts the initial position of an object in space.
- Tag **<inertial>** defines mass of the object and the offset center of mass within the object. It is necessary to simulate gravity.
- Tag **<collision>** initiates the parameters of collision. This is possible due to the fact that ROS creates an invisible layer on top of the visual object that is a shell which can collide with other objects.

- Tag `<joint>` is a connection between two parts of an object `<link>`. Tag `<joint>` has description of a parent-object and a child-object (their names), connection type, that is type of movement (revolute - rotation around some axis with upper and lower limit, continuous - an endless rotation around defined axis, without limitation, and other types of movement); also it has an information about location of the joint; information about controllers.

Thus URDF file allows you to make a fully functional model of any AAL environment. URDF file is usually created in a text editor. Of course in this case it is impossible to monitor in real time the changes in 3D space. However, ROS has a special package Rviz, which allows you to import URDF files and shows the model in three dimensions. In addition, Rviz gives an opportunity to manipulate any objects that are included in the environment. For example, by moving the slider, you can move robot's hands or open the doors. Rviz can analyze collisions and represent them as a three-dimensional model, show the information about joints, mass and center of mass. Consider the example of the work in Rviz with help of the robot Kuka KR360. To run the package Rviz needs to create a launch-file, this describes all the nodes and their parameters.

#### Code Example:

```
<? xml version="1.0"?>
<launch>
<paramname="robot_description" textfile="$(find
kukakr360)/urdf/kr360.urdf" />
<nodename="robot_state_publisher"
pkg="robot_state_publisher" type="state_publisher" />
  <arg name="gui" default="True" />
  <param name="use_gui" value="$(arg gui)"/>
<nodename="joint_state_publisher"
pkg="joint_state_publisher" type="joint_state_publisher"
></node>
<nodename="rviz" pkg="rviz" type="rviz" args="$(find
kukakr360)/config/rviz.vcg" />
</launch>
```

Launch file contains the path to URDF file as a parameter (a description of the robot `robot_description`) and the parameter for using the GUI for manual control of the AAL environment with help of the sliders. Furthermore, the launch file runs the nodes "robot\_state\_publisher" and "joint\_state\_publisher", which necessary for the processing of information on the joints, as well as "package\_Rviz", which shows us a three dimensional representation of the model. To run this file, you need to active directory where the file resides, and then at the command prompt enter "roslaunch `pwd`/kr360.launch". In the Rviz window, you can see the model of the robot Kuka KR360. In addition Rviz allows adding axes, joints and their names to help an engineer with construction of the robot. It is very simple to add this information into the Rviz: just click "add button" at the bottom of the Rviz window and choose "TF". Actually, you can add any information, such as view from the camera, laser data, odometer information and even motion planning if you already programmed the robot.

Unfortunately, Rviz can only show us the three dimensional representation of the robot or an object of AAL environment, but not able to create the emulation of the world. This is a complex problem is solved by a package Gazebo, simulating gravity, which allows you to import many models and gives them an opportunity to communicate with each other. Gazebo allows you to experiment with the AAL environment. To import models in Gazebo, we need to add some lines into the launch file:

```
<param name="table_description" command="$(find
xacro)/xacro.py $(find kukakr360)/urdf/kr360.urdf" />
<node name="spawn_table" pkg="gazebo"
type="spawn_model" args="-urdf -param table_description
-z 0.01 -model kr360_model" respawn="false"
output="screen" />
```

Now when you run launch file the model will open in two packages - Rviz and Gazebo. The user can manually control the robot for making experiments. (**Fig. 3**, **Fig. 4**) It is important to remember that you can add an unlimited number of models in the launch file, setting their initial location in the Gazebo virtual world. In the end, you will

get AAL environment emulation with the ability to manually manage all its objects.

sustainable and intelligent while other projects employ these concepts for assistance.

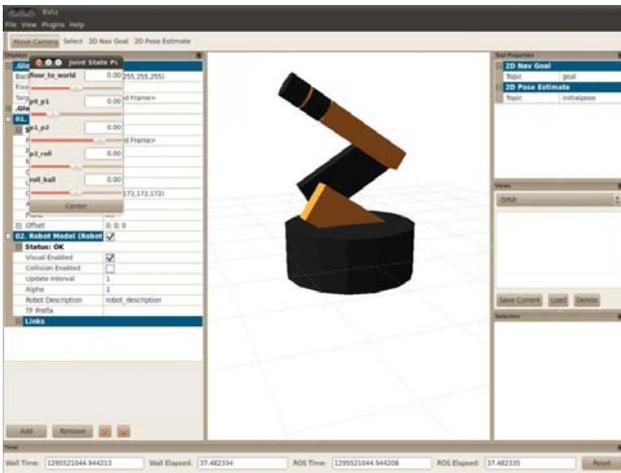


Fig. 3: Multi-joint robot created with urdf notification in Rviz.

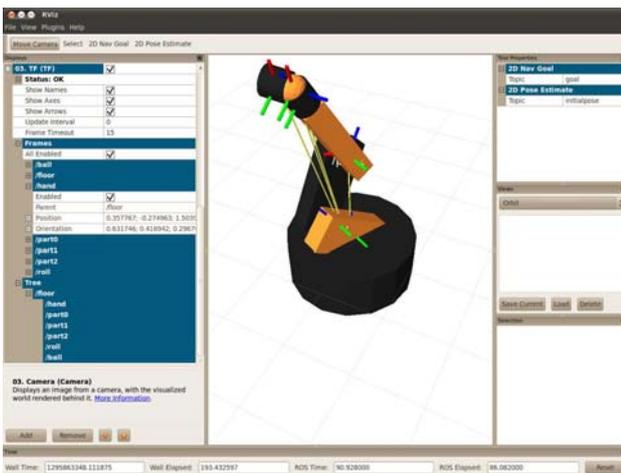


Fig. 4: Rviz model of the Kuka Robot showing Joints and Moving Axes

#### 4. ROBOTIC BUILDINGS

Tools like ROS and Gazebo have become more important in light of new building concepts with embedded robotic systems. Over centuries buildings have always been viewed as static structures resistant to changes. However, more recently there have been a different viewpoint about buildings, and they are often viewed as kinetic structures capable of changing its structure or profile to suit the various environmental or user conditions. The concept of kinetic architecture or dynamic buildings is currently being exploited by many architects around the globe. Some projects use these concepts to make the building more



Fig. 5: Dymaxion House by Buckminster Fuller

Figure 5 shows the picture of the Dymaxion house designed by Architect Buckminster Fuller in 1927. It was one of the earliest examples of Robotic Building featuring movable walls. The exterior wall can be slid over one another thereby it can open to the surroundings. Furthermore, it features a solid and a glass wall such that the glass wall can be used in winter to let the sun in and the solid wall can be slid over in summer to block (sun) radiation.



Fig. 6: Rotating Home by Johnstone [10]

Figure 6 shows the picture of the Rotating home designed by Johnstone. It can rotate 360 degrees and works on the concept of direct control where the rotation can be

controlled using an on/off switch. But the concept can be used for a more intelligent system, where sensors can be integrated within the living spaces and the control system can be planned to follow the sun so that the habitable spaces receive sun light for the maximum duration and helps to reduce the heating costs of the building.



Fig. 7: Dubai Rotating Tower [11]

**Figure 7** shows the rotating tower project designed by Architect David Fisher. Electrically powered motors assist each floor to rotate around its concrete core independent of the other floors thereby the building changes its form constantly over time. Seventy horizontal turbines designed in between each rotating floor along with solar panels generate sufficient energy to be sustainable.

## 5. CONCLUSION

Currently, ROS is predominantly used for mobile robots and to an extent for AAL and intelligent environments. However, when the developments in the field of sensors, actuators, mobile robots and Immobile systems are integrated and merged together with architectural design the buildings will become more complex and would require stimulation in an environment like ROS and Gazebo before it is executed. Furthermore, it would bring about a change in the modeling process where it demands not only the modeling of the three dimensional images but also the kinematics of the structure. So for all these purposes the open source system ROS provides a virtual testing platform and the capabilities are limited only by the imagination of the engineer.

## REFERENCES

- [1] Weiser, M., Seely-Brown, J. "The coming age of calm technology", in *Beyond Calculation: The Next Fifty Years*, Copernicus, pp. 75-85, 1997.
- [2] Englberger, J.F. "Robotics in Service" Massachusetts: MIT Press, 1989.
- [3] Linner, T., Bock, T. "Demographic Change Robotics", in: *Introduction to Modern Robotics*, Edited by N. McDowall, iConcept Press, 2011.
- [4] Linner, T., Kranz, M., Ellmann, B., Bittner, A. "Robotic Service Core for Ambient Assisted Living", 4th International Conference on Pervasive Computing Technologies for Healthcare 2010, Munich, March.
- [5] Bock, T., Linner, T. "Enhanced Industrialized Customization Performance by Building Integrated Microsystems", 27th International Symposium on Automation and Robotics in Construction (ISARC), June 2010, Bratislava.
- [6] Bock, T., Linner, T. "Mechatronic Assisted Living in intelligent Environments", 27th International Symposium on Automation and Robotics in Construction (ISARC), June 2010, Bratislava./ [www.Umweltstiftung \(Fabritz\)](http://www.Umweltstiftung(Fabritz))
- [7] Morgan Q, Eric B, and Andrew Y. Ng. "Stair: Hardware and software architecture", in *AAAI 2007 Robotics Workshop*, 2007.
- [8] Quigley, M, Conley, K, Gerkey, B, Faust, J, Foote, T, Leibs, J, Wheeler, R and Andrew Y. Ng. "Ros: an open-source robot operating system", ICRA, Workshop on Open Source Software, 2009.
- [9] ROS Website: <http://www.ros.org/wiki/ROS/Introduction>, last visited 04.04.2011
- [10] Johnstone, A., & Janet, "Maximize Your View with Rotating Home", 2006 Website <http://www.rotatinghome.com/>, last visited 04.04.2011
- [11] Fisher, D. H, "Rotating Tower Dubai", CTBUH 8th World Congress, 2008