Robot Model Development on Steroids

by Kai von Szadkowski
<?xml version="1.0"?>
<!-- created with Phobos 0.7 -->
<robot name="mantis">

<link name="body/0">
  <visual name="visual_torso_left_socket">
    <origin xyz="0.0198386 0.082563 0.0100624" rpy="0 0 5e-07"/>
    <geometry>
      <mesh filename="./meshes/torso_left_socket.bobj" scale="1.0 1.0 1.0"/>
    </geometry>
    <material name="Joint"/>
  </visual>
  <collision name="collision_torso_left_socket">
    <origin xyz="0.0198387 0.082563 0.0100624" rpy="0 0 5e-07"/>
    <geometry>
      <box size="0.1397003 0.2348451 0.1699998"/>
    </geometry>
  </collision>
</link>

<link name="body/1">
  <visual name="visual_torso_right_socket">
    <origin xyz="0.0198383 -0.0825922 -0.0099376" rpy="0 0 5e-07"/>
    <geometry>
      <mesh filename="./meshes/torso_right_socket.bobj" scale="1.0 1.0 1.0"/>
    </geometry>
    <material name="Joint"/>
  </visual>
  <collision name="collision_torso_right_socket">
    <origin xyz="0.0192199 -0.0990479 -0.01" rpy="0 0 5e-07"/>
    <geometry>
      <box size="0.1397003 0.2348451 0.1699998"/>
    </geometry>
  </collision>
</link>

<link name="body/2">
  <inertial>
    <origin xyz="0.0099378 -1.14e-05 -0.1320033" rpy="1e-07 0 0"/>
    <mass value="2.3"/>
    <inertia ixx="0.1088414" ixy="0" ixz="0" iyy="0.1248771" izy="1e-07" izz="0.0271677"/>
  </inertial>
  <visual name="visual_hip_left">
    <origin xyz="0.0099378 -1.14e-05 -0.1320033" rpy="1.5707977 8e-07 1.5707963"/>
    <geometry>
      <mesh filename="./meshes/hip_right.bobj" scale="1.0 1.0 1.0"/>
    </geometry>
    <material name="limb"/>
  </visual>
</link>

</robot>
What is Phobos?

- Blender AddOn
- free & open source

https://github.com/rock-simulation/phobos

- created as a tool for MARS

https://github.com/rock-simulation/mars
Blender

- comprehensive 3D modelling and rendering tool
- editing tools for:
  - geometry
  - materials
  - animation "armatures"
  - various file formats
  - rendering and video rendering
- free & open source
- all major platforms
- Python scripting API
Object Tree

- allows matching of URDF and Blender model structures
WYSIWYG
Importing from CAD
import bpy

for obj in bpy.context.selected_objects:
    print(obj.name)
import bpy

for obj in bpy.context.selected_objects:
    print(obj.name)
Importing from CAD

Object Tree
- allows real-time OpenGL
- Blender mode streams

WYSIWYG

URDF vs. Blender

Blender
- comprehensive 3D modeling
- and rendering tool
- editing tools for
- geometry
- materials
- animation "smoothers"
- various file formats
- modeling and visualization rendering
- free & open-source
- all major platforms
- Python scripting API
Robot dictionary

- Python dictionary
- decouples Blender model from import and export
- allows generic definitions of components
- simplifies model debugging
SMURF = Supplementable Mostly Universal Robot Format

- URDF + YAML
- hierarchical structure
Models and scenes

- SMURF scenes:
  - multiple *smurfs* get exported as individual smurfs
  - URDF joints as scene anchors
Phobos as an upstream tool

Workflow:
1) create models in Phobos
2) export model to URDF & SMURF

Pro:
• all information in one place

Con:
• version control becomes difficult
Models and scenes

- **SMURF scenes**:
  - multiple smurfs get exported as individual smurfs
  - URDF joints as scene anchors

---

**SMURF**

- Supplementable
- Modifiable
- Universal
- Robust
- Extensible
- URDF + YAML
- Hierarchical structure

---

**Phobos as an upstream tool**

- **Workflow**:
  1. Create models in Phobos
  2. Export model to URDF & SMURF

- **Pros**:
  - All information in one place

- **Cons**:
  - Version control becomes difficult
Development

The road ahead...

- URF not yet supported 100%
- e.g.: mirror parts
- dictionary and custom properties not fully specified depend on SMURP’s current status
- code refactoring and debugging
Github

https://github.com/rock-simulation/phobos
Documentation

Robot Dictionary

To facilitate simple im- and export of robot models in various formats, Phobos uses a file-format-independent Python dictionary representation to store robot data. This dictionary representation is loosely based on URDF/SDF, with some added MARS' and Blender's own naming conventions. It is laid out in the following in YAML notation:

```
model:
  - (joint):
    - (joint1):
      - (name): str
      - (type): str
      - (lower): d
      - (upper): d
      - (effort): d
      - (limits):
        - (min): d
        - (max): d
    - (joint2):
      ...
      ...
  - (link):
    - (name): str
    - (pose):
      - (x): d
      - (y): d
      - (z): d
    - (visual):
      - (name): str
      - (diffuseColor):
      - (ambientColor):
      - (emissionColor):
      - (specularColor):
      - (material):
    ...
```

Github Pages: Sphinx

Github Wiki
The road ahead...

- URDF not yet supported 100%
  - e.g.: mimic joints
- dictionary and custom properties not fully specified (depend on SMURF's current status)
- code refactoring and debugging

(contributors welcome)
Acknowledgements

- Contributors:
  - Stefan Rahms
  - Ole Schwiegert

Grant number: 5ORA1218 / 5ORA1219