

Evgenii Safronov

**Open Robotics**  
[info@openrobotics.org](mailto:info@openrobotics.org)

February 11, 2018

Dear Open Robotics,

since I use ROS (and program with its libraries) *every day* I couldn't miss opportunity to apply for this amazing internship. Now I'm a captain of Skoltech's Eurobot team (<http://www.eurobot.org/>). This contest includes full development of 2 mobile autonomous robots. Our code architecture is built around ROS (see here <https://github.com/SkoltechRobotics/ros-eurobot-2018>, in development). I'm really inspired by the work you are doing, it cuts time of development and simplify concurrent engineering inside team.

Last two years I have been coding primarily on Python. And now my code for Eurobot is also written on Python. However, I've started to learn C++ in school (7 years ago!) and continued in university. I've got bachelor degree with honors (GPA 4.92/5) from MIPT and always got excellent grades for programming subjects. MIPT is one of the best russian universities in math & programming. For example, MIPT team was awarded with Silver medal at ACM ICPC World Finals 2017. I took advanced course on C++ during my study, check <https://github.com/safoex/nixcpp>.

Eurobot Finals are scheduled to the middle of May, so I'm available since June. I looked through the list of potential ideas for Google Summer of Code. I chose most interesting and relevant for me:

- **Robot Work Cell Discovery** This project can improve many of my skills. I'd like to mention, that I have a small experience with UR3 robot from Robotics course at Skoltech. Project goal is actual for such collaborative robot. Also, experience gained can be applied for other scene building tasks (e.g. for indoor mobile assistants). My future master thesis will be connected with mobile robots or with collaborative robots such as UR3. So, this is a great opportunity to work on this topic and finish it as part of my master thesis.
- **Simulate Radar sensor** I've got an excellent background in physics (even won over MIT team in International Theoretical Physics Olympiad <http://thworldcup.com/contest2017> and attached NagaokaBags.pdf), so this project involves an efficient fusion of my skills! Also, it again gives me possible mobile robotics connected master thesis.
- **Create Default Grasping Library** Grasping is another *hot* area of robotics. Experience in grasping is required for developing a good collaborative robot.

Personally I'm working on decision-making for Eurobot competition. One of subtasks is a strategy optimization and execution. As for optimization part, that's individual for each project. For execution we decided to use behavior trees for their reactivity and human readability. There are several realisations of Behavior Tree concept in ROS:

- **behavior\_tree** This one doesn't work on ROS Kinetic by default. I made few corrections in code (and in 2 depended packages, see again <https://github.com/SkoltechRobotics/ros-eurobot-2018> ROS-Behavior-tree, xdot and rqt\_dot packages). However, their approach didn't match our goals for Eurobot. It's

not light-weight and huge BT initialisation and re-initialisation takes a long time. Majority of leaf nodes for mobile robots are 'moving' nodes and in my opinion this package isn't optimized for fast-execution of similar (but not exactly the same) tasks.

- **decision\_making** pure C++ approach.
- **pi\_trees** I discovered it very recently and like their approach.

However, it still lacks some desired features. I found all them miss certain desirable features:

- Save and load BT-structure, construct tree from text/xml/json/etc description. This feature simplifies debug, work and online reconstruction of BT.
- Multi-agent BT-execution. Nowadays there is a many research and state-of-the-art work in robotics in the area of multi-agent cooperation.
- Optimized work in terms of service/topics/actionlib re-use. ROS is executed on Linux, but in many projects robot's actuators are not connected directly to Linux-carrying board. Usually, people use STM32/Arduino/etc board for low-level electronics. So, many leaf actions are just communication with this boards.

These features improves the value and actuality of BT framework. I've started to work on it and wrote a simple Python class for BT (<https://goo.gl/37b1hU>, Github). To finish this job in proper way will be very beneficial and joyful work for me.

Of course, I'm open to any other suggested projects! I'm on the beginning of my engineering career and don't have any prejudices against any robotics/software engineering areas. Hope I delivered my sincere interest in your internship.

**Evgenii Safronov**