

---

## Overview

The workshop will be held on November 4, 2019, in The Venetian Macao Resort Hotel, Macau, China in the context of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), one of the largest and highest impact robotics research conferences worldwide.

The workshop will be the first behavior tree-related workshop within the robotics community. We aim to gather all the outstanding researchers interested in BTs to share ideas and lesson learned. For that reason, the workshop format will include invited talks as well as short spotlight presentations followed by interactive sessions.

## Objectives

Behavior Trees is a new powerful tool for task switching and decision making that is receiving an increasing amount of attention in robotics. Recent research is beginning to explore areas such as learning, planning, and control in domains ranging from Micro Air Vehicles to Assembly robotics.

The objectives of this workshop are as follows:

- For the first time, gather everyone within the robotics community that is interested in Behavior Trees at the same place, and give them a chance to discuss and interact.
- Provide attendees with an updated picture of where behavior tree research within robotics stands today.

Behavior Trees were developed in the video game industry as a tool to achieve modular, reusable, and flexible behaviors for Non-Player Characters (NPCs), and are now an established tool to the point that BTs are integral parts of game AI textbooks as well as major game engines such as Unity3d, and the Unreal Engine. In the last decade, Behavior Trees have received an increasing amount of attention in robotic as well, with applications in both academia and industry. The main advantages of Behavior Trees with respect to conventional execution architectures, like FSMs, are their ability to combine reactivity with deliberation in a very modular way.

## Topics of interest

The following list provides a set of topics (keywords) addressed in the workshop.

- Behavior Design
- System Integration
- Execution Architecture
- AI Decision Making
- Task Planning
- Formal Methods for Robotics
- Resource-aware Robot Programming
- Behavior-Based Systems
- Autonomous Agents
- Planning, Scheduling, and Coordination

## Workshop Program

Time	Activity
09:00 - 09:15	Welcoming Words
09:15 - 09:45	Invited Talk by Simon Jones: <i>Onboard Evolution of Understandable Swarm Behaviours</i>
09:45 - 10:15	Oral Presentation: <i>Node Templates to improve Reusability and Modularity of Behavior Trees</i> - E. Safronov.
10:15 - 10:45	Oral Presentation: <i>Learning Behavior Trees From Demonstration</i> - K. French et al.
10:45 - 11:15	Coffee break
11:15 - 11:45	Invited Talk by Vincent Berenz: <i>Behavioral Reactivity: Using Reactive Programming to Create Behavior Trees</i>
11:45 - 12:15	Invited Talk by Chris Paxton: <i>Building Reactive Task Plans for Real-World Robot Applications</i>
12:15 - 12:45	Invited Talk by Francesco Rovida: <i>Application of Behavior trees to Industrial Robots Task Design</i>
12:45 - 14:15	Lunch Break

14:15 - 14:45	Oral Presentation: <i>A Modular Mission Management System for Micro Aerial Vehicles</i> - M. Lan et al.
14:45 - 15:15	Oral Presentation: <i>Real-Time Autonomous UAV Task Navigation using Behavior Tree</i> - R. Soong et al.
15:15 - 15:45	Invited Talk by Enrique Coronado: <i>Empowering End-Users with Behavior Trees, Development of Human-Robot Applications Performing "in the Wild"</i>
15:45 - 16:15	Coffee Break
16:15 - 16:45	Invited Talk by Petter Ögren: <i>Combining Reinforcement Learning with Behavior Trees</i>
16:45 - 18:00	Open Discussion

---

## Invited Speakers

### **Simon Jones (University of Bristol)**

#### **Onboard Evolution of Understandable Swarm Behaviours**

Designing the individual robot rules that give rise to emergent swarm behaviours is difficult. The common method of running evolutionary algorithms to automatically discover controllers in simulation suffers from a major disadvantage; the evolved controllers are often opaque and hard to understand. We use behaviour trees as the controller architecture because they are modular, hierarchical and human understandable. We evolve fit controllers in a swarm with high processing power, use automatic tools to simplify them, and then explain them, an important factor for safety and acceptance. We look at the design of a suitable behaviour tree architecture, applying the techniques of Genetic Programming to behaviour trees, the implementation of a fast parallel simulator and tree interpreter running on the GPUs of our swarm of robots, how large evolved trees can be automatically simplified to aid analysis, and then explain one of the evolved controllers.

### **Vincent Berenz (Max Planck Institute for Intelligent Systems)**

#### **Behavioral Reactivity: Using Reactive Programming to Create Behavior Trees**

We will present how to apply reactive programming to the design of robot behavior tree. Using this paradigm, the user defines branches and then associate them with rules of activation. Contrary to other approaches, the behavior tree is not traversed at runtime. Rather the branches compete continuously one with another for activation. And because branches can be attached to targets (discretized high level sensor entities), the activation of a branch will correspond to the activation of a specific set of sensory motor couplings that will shape the behavior of the robot. As various videos will show, this programming paradigm supports rich and complex behaviors characterized

by high behavioral reactivity. Playful, our open-source reactive programming interpreter supporting behavior trees, will be introduced. Our more recent work exploring the automated generation of reactive programming scripts from human demonstration will also be discussed.

## **Chris Paxton (NVIDIA)**

### **Building Reactive Task Plans for Real-World Robot Applications**

Robots are increasingly an important part of our world, from working in factories and hospitals to driving on city streets. As robots move into more unstructured environments such as homes, however, we must be able to create complex, reactive task plans that can deal with stochastic actions, unreliable sensors, and that above all are intuitive and easy to build. To this end, we created the Behavior Tree-based CoSTAR system -- which allows novice end users to create task plans for industrial robot task plans, shown in a 35-person user study to be highly user friendly and offer a useful set of tools for creating task plans. We also describe a variant on Behavior Trees, the Robust Logical-Dynamical System (RLDS), which supports symbolic task planning and guarantees on performance. Finally, we describe a manipulation case study on an example of an unstructured household manipulation task.

## **Francesco Rovida (Aalborg University)**

### **Application of Behavior trees to Industrial Robots Task Design**

Reconfigure collaborative robots on new tasks quickly and efficiently is today one of the great challenges for manufacturing industries. In this respect, behavior trees already proved to be a great tool to design complex coordination schemes with important required characteristics, such as high modularity, predictability and reactivity. Nevertheless, BTs are also affected by some limitations that are preventing their widespread adoption. For example, being a state-less logic in their classical form, BTs have more problems representing sequences w.r.t. other approaches such as e.g. FSMs. We identified this and other issues while applying BTs to industrial robots task design. In this presentation we will discuss our experience and look at the solutions implemented to make the classical BT a more complete task programming framework for industrial applications.

## **Enrique Coronado (Tokyo University of Agriculture and Technology)**

### **Empowering End-Users with Behavior Trees, Development of Human-Robot Applications Performing "in the Wild"**

In many cases, the creation of robot applications performed "in the wild" (i.e., moving from the laboratory to natural and in-situ scenarios) require to include end-users (e.g. therapists, psychologists, teachers, and artists) for their suitable design and testing. In order to support end-user goals, we developed RIZE (Robot Interface from Zero Experience) a cross-platform End-User Development tool back-ended by Component-Based Distributed Robot Frameworks. This interface enables intuitive robot programming by the use of block-based Visual Programming Languages, which translate the user code to Behavior Trees for its posterior execution. In this presentation, we will show how this interface has been used in the development of short-term and long-term Human-Robot Interaction applications performed "in the wild".

## **Petter Ögren (Kungliga Tekniska Högskolan)**

## Combining Reinforcement Learning with Behavior Trees

Reinforcement Learning (RL) has received a lot of attention in recent years for its ability to generate efficient controllers for complex problems without the need for a system model. However, the resulting designs often lack both modularity and transparency. It is hard to replace, remove or add different components of an RL policy, and it is hard to know why a given action is taken, beyond the fact that it is believed to maximize the expected long term reward. In this talk, we explore how a combination of RL and BTs can provide the best of both worlds. The modularity and transparency of BTs with the model free performance of RL.

---

## Oral Presentations

### Evgenii Safronov (Sberbank Robotics Laboratory)

#### Node Templates to improve Reusability and Modularity of Behavior Trees

Behavior Trees (BTs) got the robotics society attention not least thanks to their modularity and reusability. The subtrees of BTs could be treated as separate behaviors and therefore reused. We address the following research question: do we exploit the full power of BT on these properties? We suggest to generalise the idea of subtree reuse to “node templates” concept, which allows to represent an arbitrary nodes collection. In addition, previously hardcoded behaviors such as Node\* and many Decorator nodes could be implemented in a memory-based BT by node templates.

### Kevin French (The Robotics Institute)

#### Behavioral Reactivity: Using Reactive Programming to Create Behavior Trees

Robotic Learning from Demonstration (LfD) allows anyone, not just experts, to program a robot for an arbitrary task. Many LfD methods focus on low level primitive actions such as manipulator trajectories. Complex multistep task with many primitive actions must be learned from demonstration if LfD is to encompass the full range of task a user may desire. Existing methods represent the high level task in various forms including, finite state machines, decision trees, formal logic, among others. Behavior trees are proposed as an alternative representation of high level task. Behavior trees are an execution model for the control of a robot designed for real time execution, modularity, and, consequently, transparency. Real time execution allows the robot to reactively perform the task. Modularity allows the reuse of learned primitive actions and high level task in new situations, speeding up the process of learning in new scenarios. Transparency allows users to understand and interactively modify the learned model. Behavior trees are used to represent high level tasks by building on the relationship it has with decision trees. We demonstrate a human teaching our Fetch robot a household cleaning task.

### Menglu Lan (National University of Singapore)

#### A Modular Mission Management System for Micro Aerial Vehicle

In this paper, we present a modular mission management system for micro aerial vehicles (MAVs), with a successful application in real environments with the actual platform. We use the behaviour tree as the main underlying structure to represent and further execute the complex mission plan. The proposed system is highly modular, allowing the user to specify different missions by simply grouping various mission-independent sub-behaviours together. The paper also describes a set of

commonly used behaviours for MAVs. The proposed system has been applied to the 2017 International Micro Air Vehicle Competition (IMAV 2017) and has helped the team win both indoor and outdoor championships.

## **Ru-Tai Soong (National Chengchi University)**

### **Real-Time Autonomous UAV Task Navigation using Behavior Tree**

Reconfigure collaborative robots on new tasks quickly and efficiently is today one of the great challenges for manufacturing industries. In this respect, behavior trees already proved to be a great tool to design complex coordination schemes with important required characteristics, such as high modularity, predictability and reactivity. Nevertheless, BTs are also affected by some limitations that are preventing their widespread adoption. For example, being a state-less logic in their classical form, BTs have more problems representing sequences w.r.t. other approaches such as e.g. FSMs. We identified this and other issues while applying BTs to industrial robots task design. In this presentation we will discuss our experience and look at the solutions implemented to make the classical BT a more complete task programming framework for industrial applications.

---

## Call for Papers

### **Information for Authors**

Please submit either a link to an existing paper or a 2-page extended abstract, following the IEEE conference style: <http://ras.papercept.net/conferences/support/support.php>

We welcome also material already published.

All submissions will be peer-reviewed. Accepted papers will be presented as oral presentations.

Send your PDF manuscript by email, with the subject including the prefix "[BTRS IROS 2019]", to the following email: [michele.colledanchise\[at\]iit.it](mailto:michele.colledanchise@iit.it) (replace [at] with @).

### **Important Dates**

September 30, 2019 - Submission Deadline

October 15, 2019 - Acceptance/Rejection Notification

November 4, 2019 - Full-day Workshop

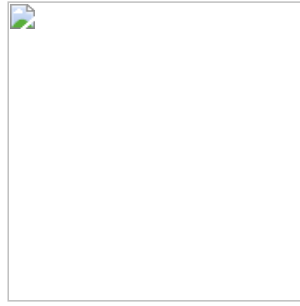
---

## Organizers

© 2019 Behavior Trees for Robotics Systems



Michele Colledanchise  
Istituto Italiano di Tecnologia (IIT)  
Genoa, Italy



Lorenzo Natale  
Istituto Italiano di Tecnologia (IIT)  
Genoa, Italy



Petter Ögren  
Kungliga Tekniska Högskolan (KTH)  
Stockholm, Sweden