

Virtual RobotX Challengeへ向けたオープン ソース自律航行システムの開発

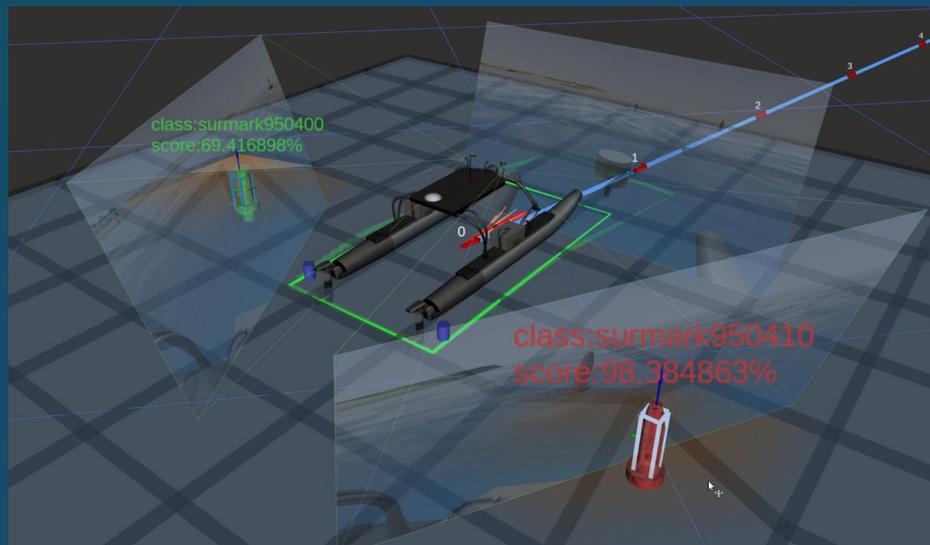
OUCT POLARIS

片岡大哉 / 増岡伸哉 / 和田翠星 / 田中良道



コンテンツ

1. 大会, チーム紹介
2. OUXT Polarisの開発方針
3. 開発過程で生まれた各種成果物の紹介
4. 今後の開発計画



大会，チーム紹介

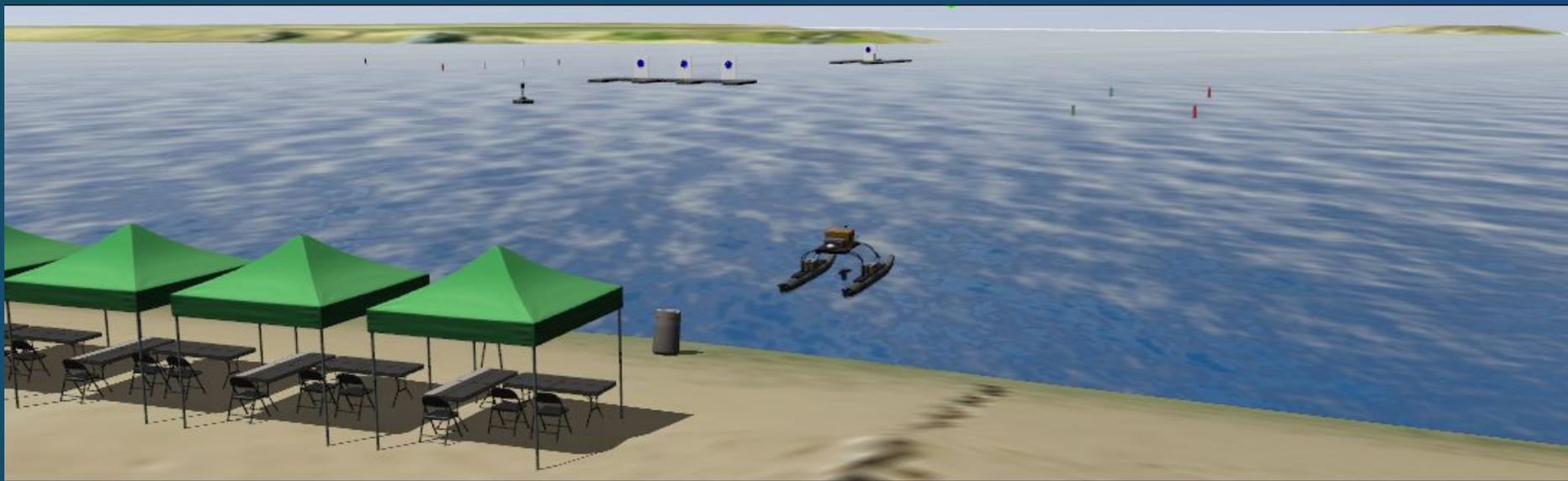
Maritime RobotX Challenge

- 完全自律走行船の**国際コンテスト**
- 世界中から全12チーム参加
- **日本からはOUXT Polarisのみ**参加
- 水上/水中タスク+広報+論文により採点
- 船体はWAM-Vを使用
- 隔年12月, ハワイで開催(次回は2020年)



Virtual RobotX Challenge

- **Gazebo**上で行われるMaritime RobotX Challenge
- Docker ImageをPushすると自動的に採点が行われる
- 全てのタスクは全自動で行い、人の介入は禁止
- 指定座標に向かう、光るブイの点滅パターンの報告等のタスクを実施

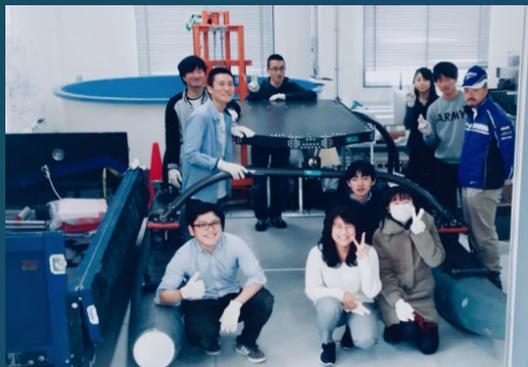


OUCT Polarix



2014年
@Singapore

6位 **入賞**



2018年
@Hawaii

日本唯一
の出場



- 大阪府立大学
- 大阪大学
- 東京大学
- 筑波大学
- 神戸大学
- 富山高専
- 九州工業大学

競技会経験者
約20名
+
社会人数名

- 現在の拠点は**大阪府立大学**
- 2018年度大会ではタスクを1つクリア
- 開発費はスポンサー様からいただいた資金で運営

OUCT Polarisの開発方針

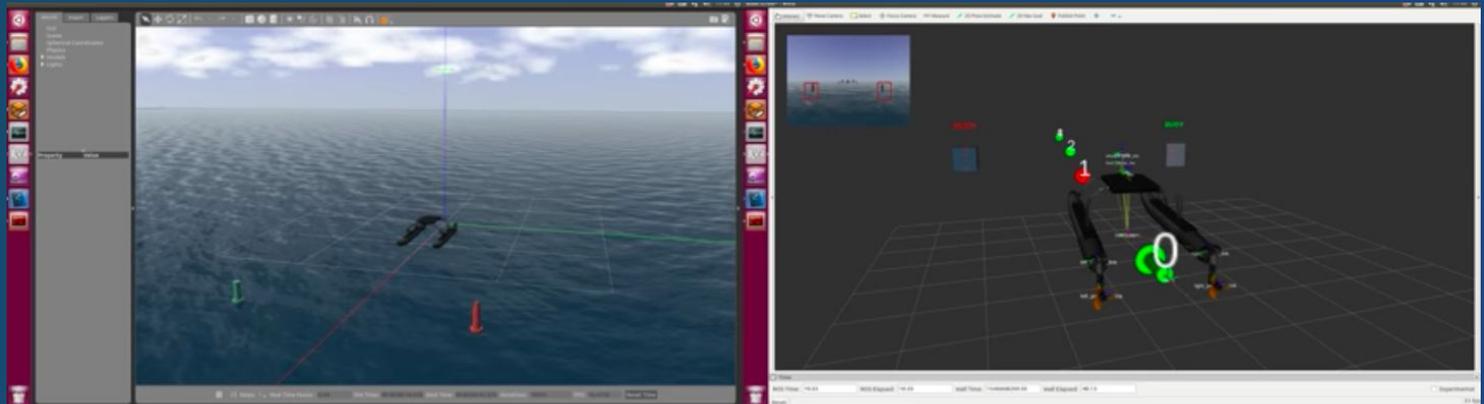
OUCT Polarisの開発方針

- 成果物を全てOSSとしてROSコミュニティに還元する
 - 非常に高い本大会の参加ハードルを少しでも下げる
 - ライセンスは全てApache 2.0で統一し、わかりやすく
- 可能な限りパッケージとしてリリースする
 - 極力標準仕様に準拠し、他のROSパッケージとも連携を容易に
 - すでに7個のリリース手続きを完了、今後もリリース対応パッケージを増やしていく予定
 - 自律航行以外でも使用可能なROSパッケージ群を目指す

上記の方針を踏まえ、
前回大会からパッケージ構成を見直し、
アーキテクチャ再設計を実施

前回大会時のパッケージ構成

- 2つだけのパッケージ分割
 - 実機で動作するプログラムをまとめたrobotx_core
 - 周辺ツールをまとめたrobotx_packages
- 問題点
 - プログラムの単位が巨大で船体のコンピュータではビルド時間が長い
 - リリースするのにパッケージの抽出が必要になる



今大会からのパッケージ構成

- vrx_packagesの中には一つ.rosinstallファイルが入っている
 - この.rosinstallファイルを用いてワークスペースの構成を管理
 - 各種プランナーや数学計算ライブラリはパッケージごとに分離され、テスト/CI/リリースが簡単に
 - 実機デプロイの際にはシミュレーション用のパッケージを取り除いたrosinstallファイルを用意し、デプロイを高速化

Why GitHub? Enterprise Explore Marketplace Pricing Search Sign in Sign up

OUXT-Polaris / vrx_packages Watch 7 Star 2 Fork 0

Code Issues 4 Pull requests 0 Projects 0 Security Insights

Join GitHub today
Dismiss
GitHub is home to over 40 million developers working together to host and review code, manage projects, and build software together.
Sign up

ROS package for virtual maritime robotx challenge

55 commits 2 branches 0 releases 1 contributor

Branch: master New pull request Find File Clone or download

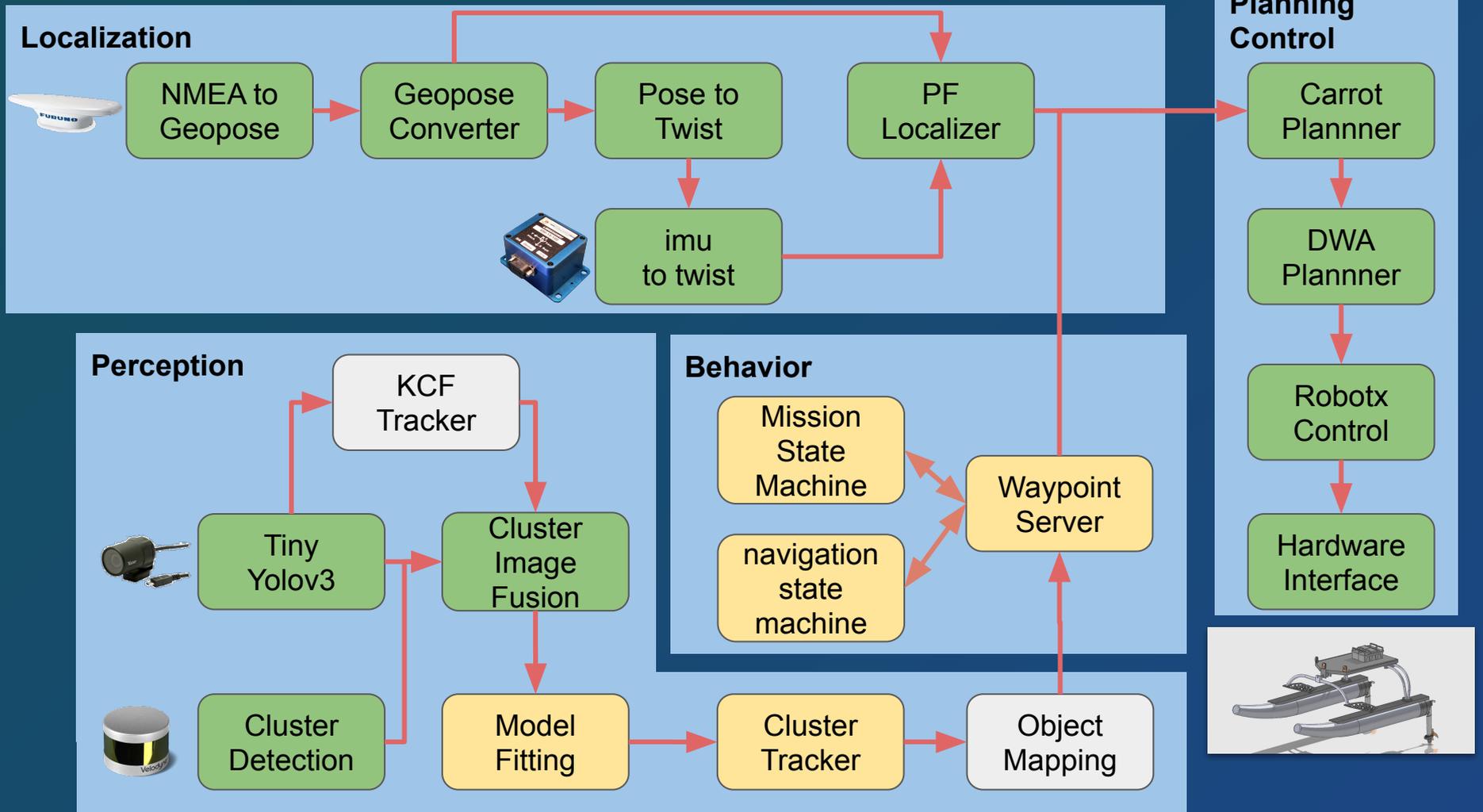
Commit	Message	Time
hakuturu583	update dependency	Latest commit 3cae21a 4 hours ago
	update .gitignore	5 months ago
	update dependency	2 months ago
	update dependency	4 hours ago
	add industrial_ci	last month

README.md

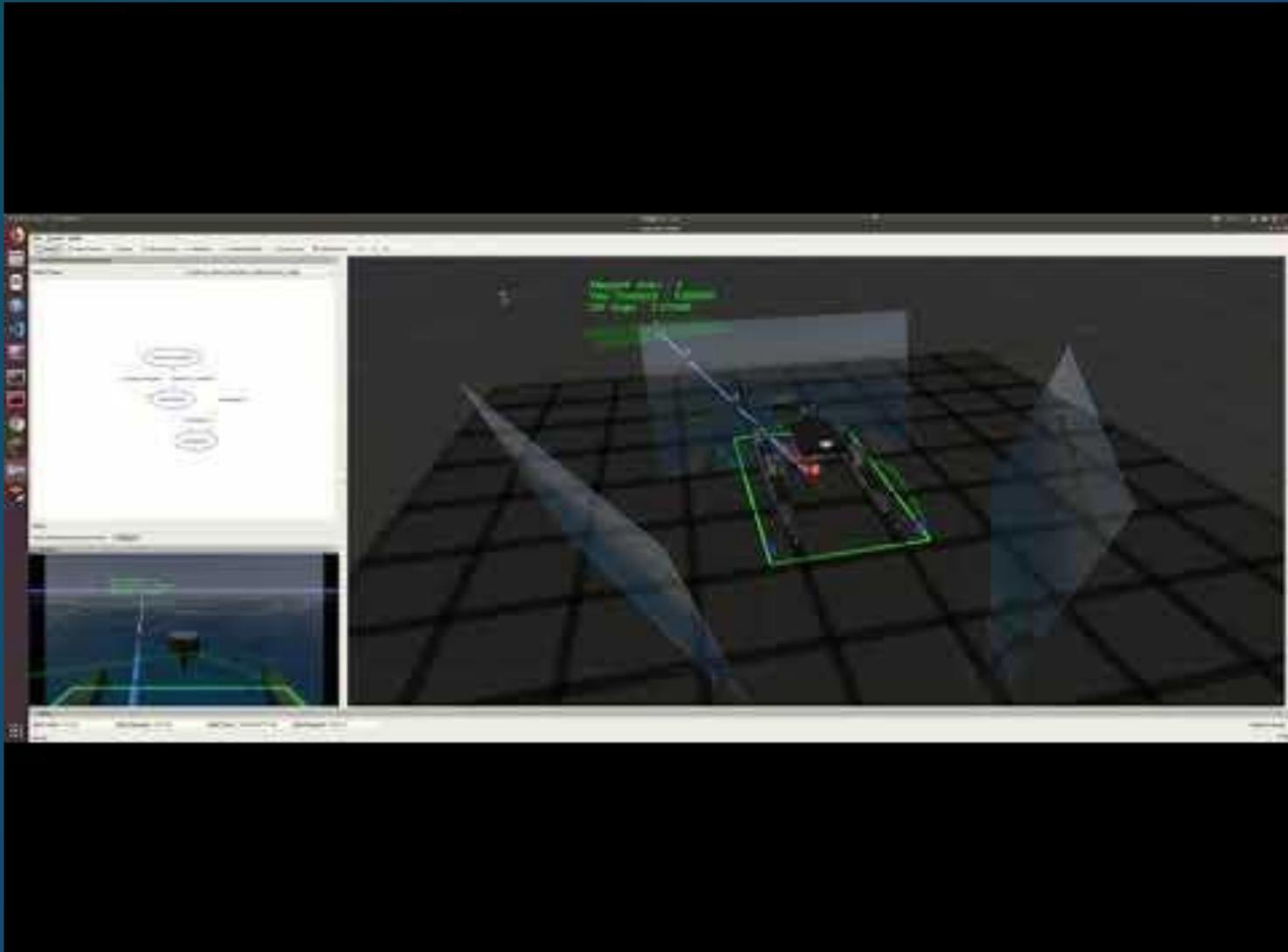
ROS packages for Virtual Maritime RobotX Challenge

Requirements

vrx_packagesのアーキテクチャ



自律ナビゲーションの様子



開発過程で生まれた各種成果物の紹介

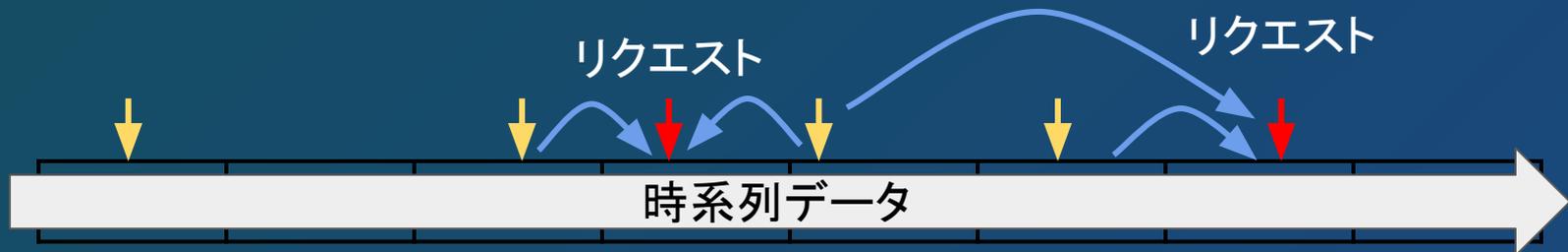
Simulatorに対する成果物

- nmea_gps_plugin
 - nmea_msgs/Sentence形式で現在の姿勢をpublishするgazeboプラグイン
 - hectorのGPSプラグインはsensor_msgs/NavSatFixでそこから自機の方位が取れなかったため開発

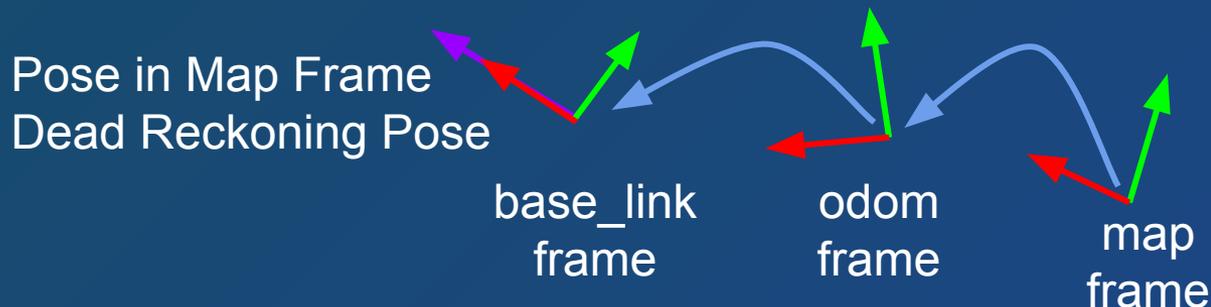


Localizationでの成果物1

- data_buffer
 - タイムスタンプ付きデータの内挿・外挿関数を書くと指定のタイムスタンプの時の値が取ってこれるバッファ

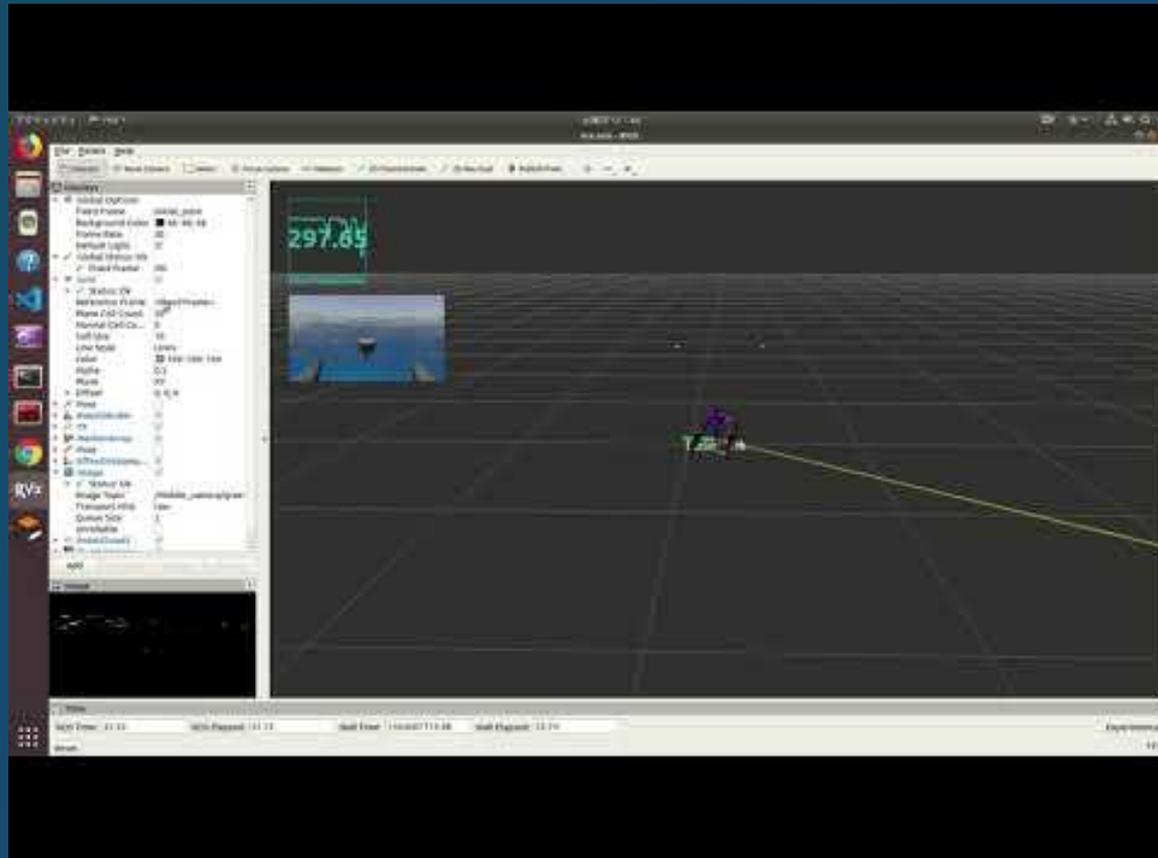


- odom_frame_publisher
 - twistをデッドレコニングした結果とposeを比較し map->odom, odom->base_linkフレームをpublish
 - 自作Localizerを簡単に[REP105](#)準拠可能



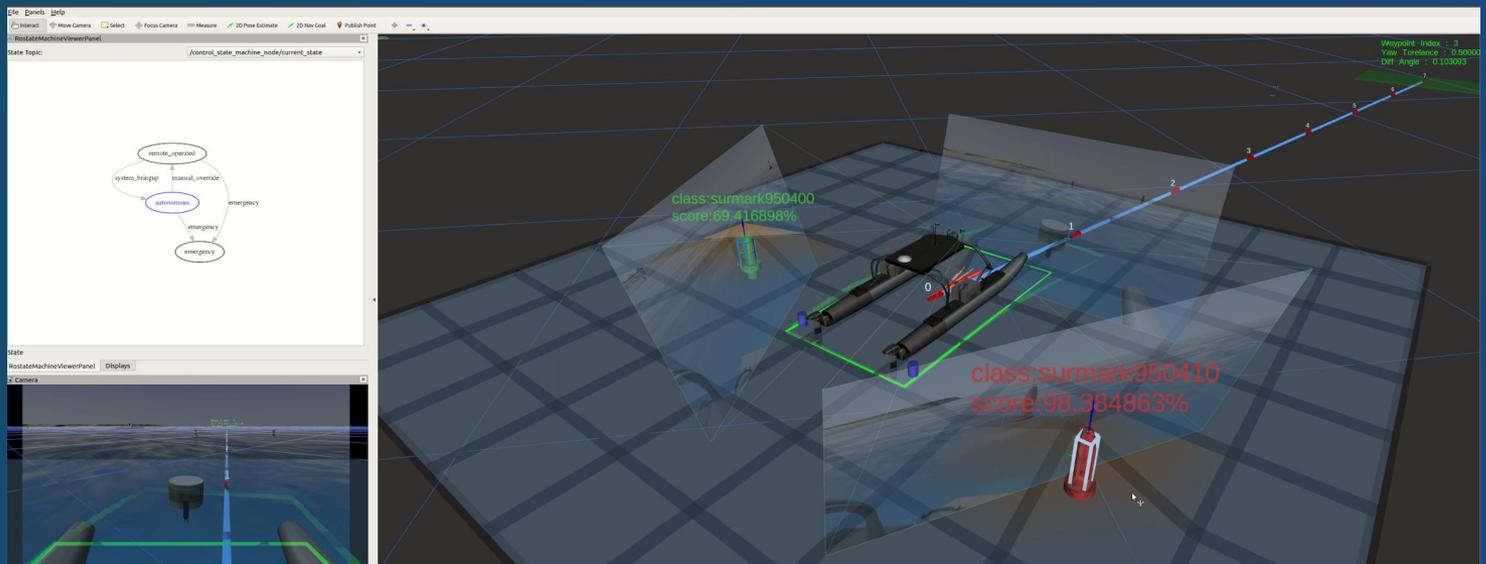
Localizationでの成果物2

- PF Localization
 - 粒子フィルタによるLocalizer
 - Pose + Twistを三次元でフュージョン



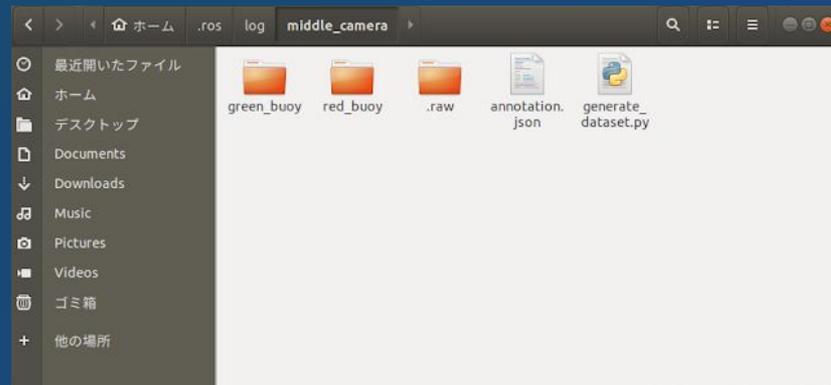
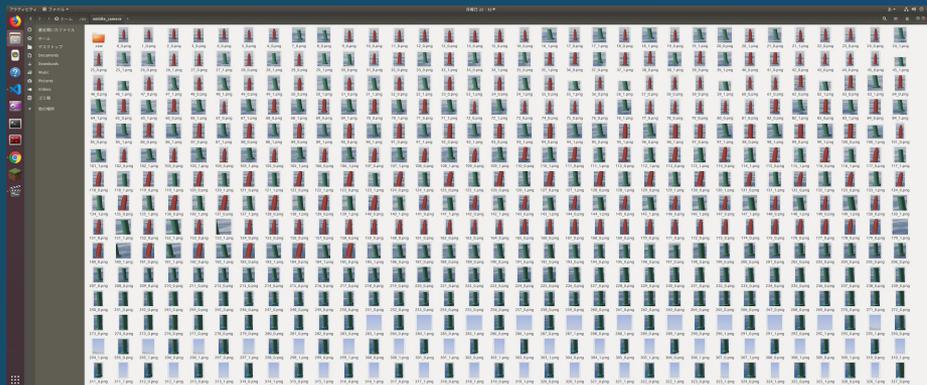
Perceptionでの成果物1

- vision_msgsに対するコミット
 - tracking_idのフィールドを追加, トラッキング込みの運用が可能になった
 - vision_info_serverを開発、物体認識データベースのバリデーション+データベースのパスを自動化
 - visualizer_nodeを開発, 認識結果の可視化に貢献



Perceptionでの成果物2

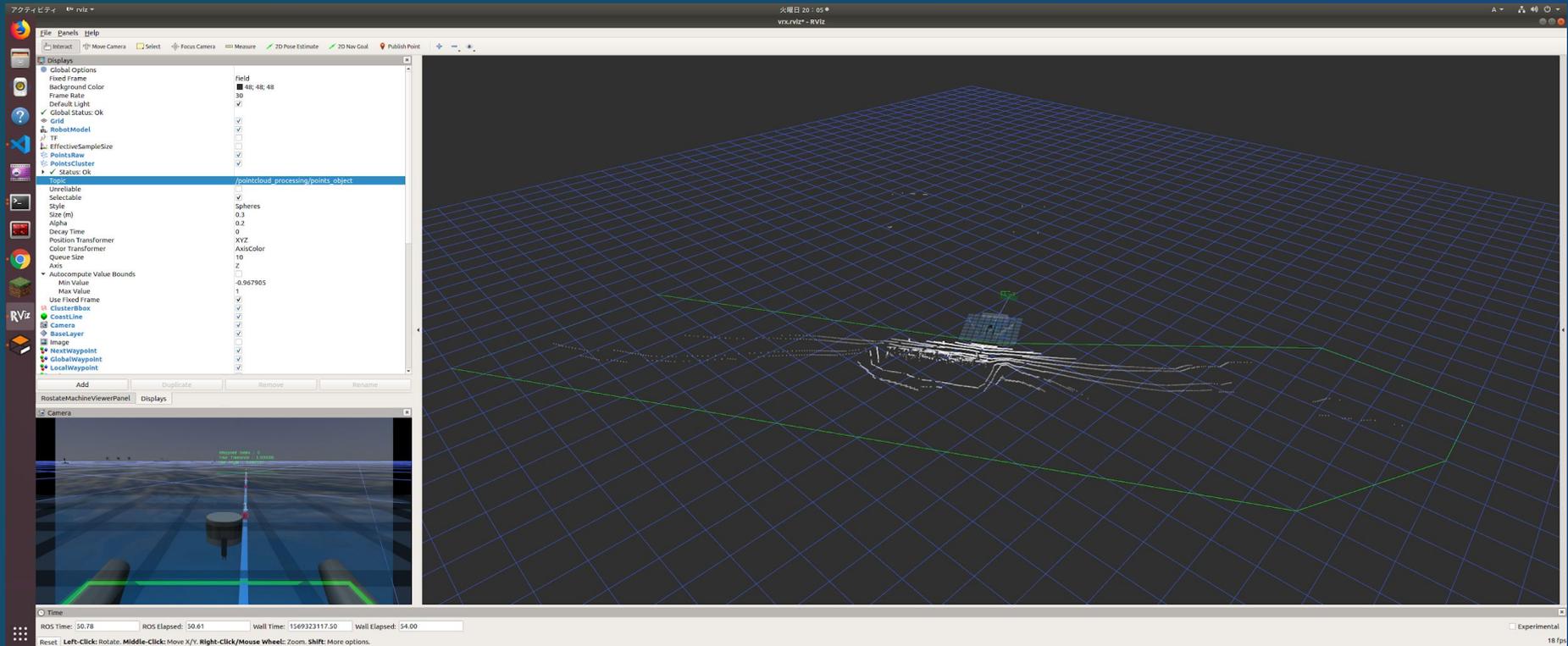
- 学習データセット作成の自動化
 - jsk_packageのクラスタリング結果からROIを生成
 - クラスごとにフォルダを作って切り抜かれた画像をクラスごとのフォルダに仕分けたあとPythonスクリプトを実行するとデータセットができる
 - 現在はdarknetのみ対応



Perceptionでの成果物3

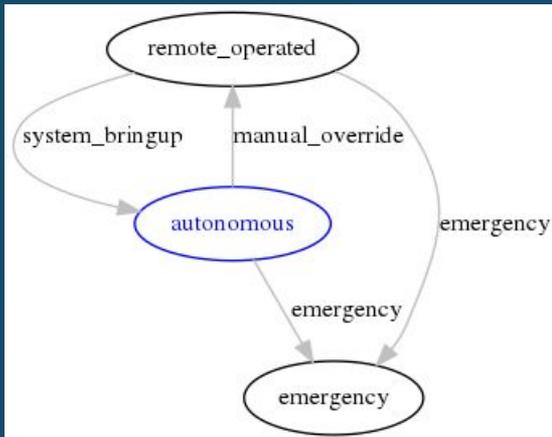
● crop_hull_filter

- convex_hull内部の点群を削除したり、内部の点群だけ残すことが可能
- 海岸線の点群を削除するため開発



Behaviorでの成果物

- rostate_machine
 - ROS用state_machineパッケージ
 - 専用クライアントライブラリを使用することで、状態定義のハードコードを防止し、ビヘイビアの肥大化を防ぐ
 - xmlフォーマットで状態機械を記述
 - rvizからリアルタイムで現在の状態を確認可能



```
<state_machine>
  <init_state name="remote_operated"/>
  <state_machine_name name="example_state_machine"/>

  <callback tag="become_autonomos" when="on_entry" states="autonomous"/>
  <callback tag="stop_motor" when="on_exit" states="autonomous,remote_operated"/>
  <callback tag="check_motor" when="always" states="autonomous,remote_operated"/>

  <transition from = "remote_operated" to="autonomous" name="system_bringup"/>
  <transition from = "remote_operated" to="emergency" name="motor_disconnected"/>
  <transition from = "autonomous" to="remote_operated" name="manual_override"/>
  <transition from = "autonomous" to="emergency" name="motor_disconnected"/>
</state_machine>
```

パッケージをリリースするメリット

- apt・rosdepでパッケージがインストールできるので**ビルド時間が短縮**できる
- 設計時から徹底的に再利用性を意識したコーディングをすることになるので**技術的負債が少なくなる**
- 過去に作ったノードを**再利用**したいときにすぐ再利用できるようになる

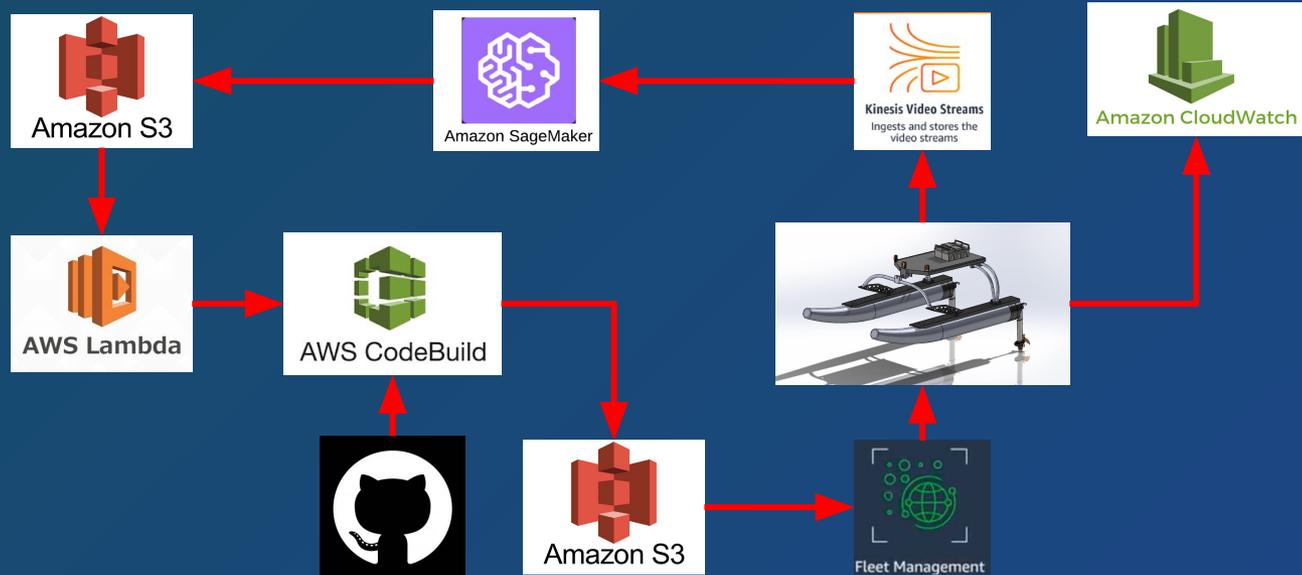
ROSパッケージは**非常に簡単**にリリースできます！
ぜひ積極的にリリースを検討されてはいかがでしょうか？
詳細は[こちら](#)をご参照ください

今後の開発計画

Maritime RobotX Challenge2020への開発

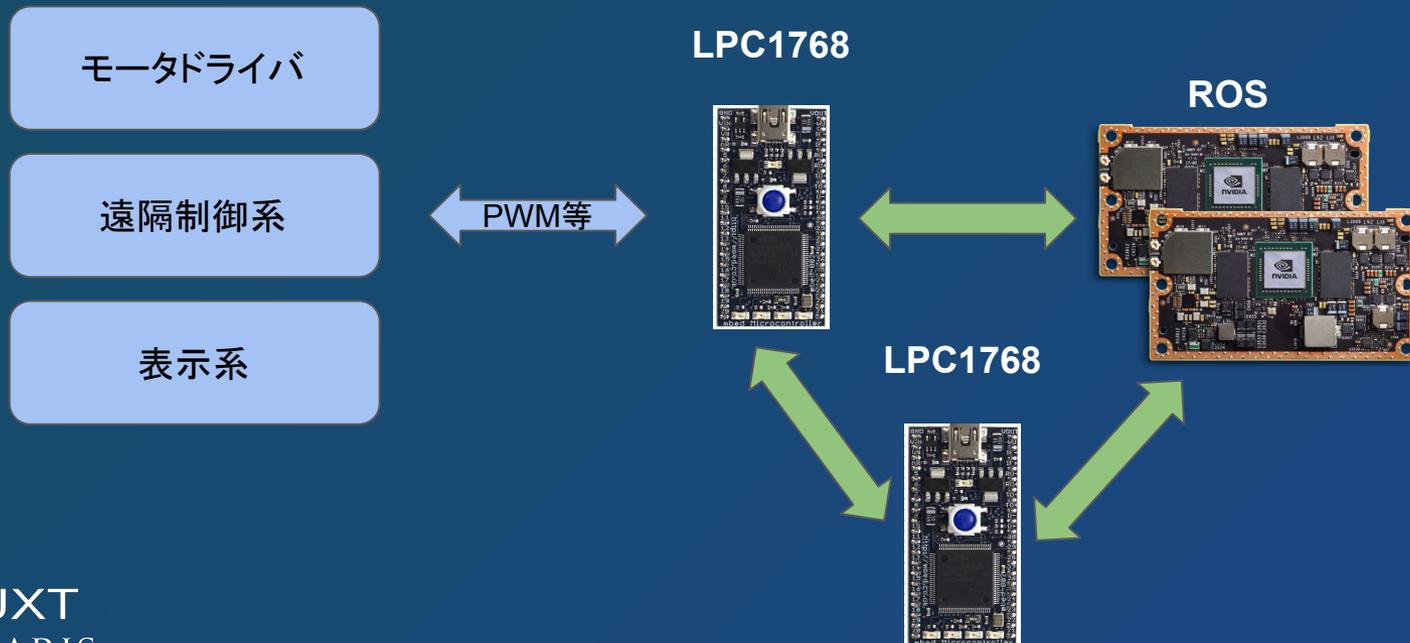
- クラウドベースのサポートツール群開発

- AWS RobomakerのFleet management + Code Buildによるデプロイパイプライン設計中
- Kinesis Video Streams + Sagemakerによりラベリングと機械学習を半自動化、現地での貴重な時間をラベリングに割くことを防止



Maritime RobotX Challenge2020への開発

- mbed用ミドルウェア(ros_mbed_bridge)の開発
 - 船体制御システムとROSをつなぎ込むためのライブラリ
 - マイコン間でUDPを使って文字列をやりとり
- 実機へのシステム適用
 - 実機はJetson TX2でクラスタを組んで運用しているため、処理を高速化して実機で動作可能にする



今後の開発計画

開発項目	2019/9	2020/3	2020/9	2020/12
トラッキング	→			
SLAM		→		
Webツール群	→			
実機対応	→			

まとめ

- Virtual RobotX Challenge/Maritime RobotX Challenge2020に向けたOSS自律航行システムのアーキテクチャを発表
- 抽象化を徹底, 複数のパッケージリリースを行い, ROSでできることの幅を広げた
- 今後はトラッキング、マッピングを導入し、より安定した自律航行を目指す
- 来年以降はWebツール群、実機適用を行いMaritime RobotX Challenge 2020での自律航行の実現を目指す

スポンサーのお願い

OUCT Polarisの開発はスポンサー様からのご支援により成り立っております。ご支援頂いたスポンサー様にはHP、船体へのロゴ掲載等を行っております。

前回大会時のスポンサー様



FURUNO



NORTHROP GRUMMAN



今後のVirtual RobotX Challengeへの開発

● トラッキング, マッピングの導入

- トラッカーを導入し, 一度認識したオブジェクトを追跡
- ランドマークベースのマッピングを導入し, 物体の位置関係に基づいてナビゲーション

