

NHK学生ロボコンに向けた ソフトウェア開発と教育から 得られた知見

京都工芸繊維大学ロボコン挑戦プロジェクト
ForteFibre
@ ROSConJP 2025



今日話すこと

- ForteFibreの紹介
- 部内ライブラリ”fibril”の紹介
- 教育プランの紹介・変化



ForteFibreについて

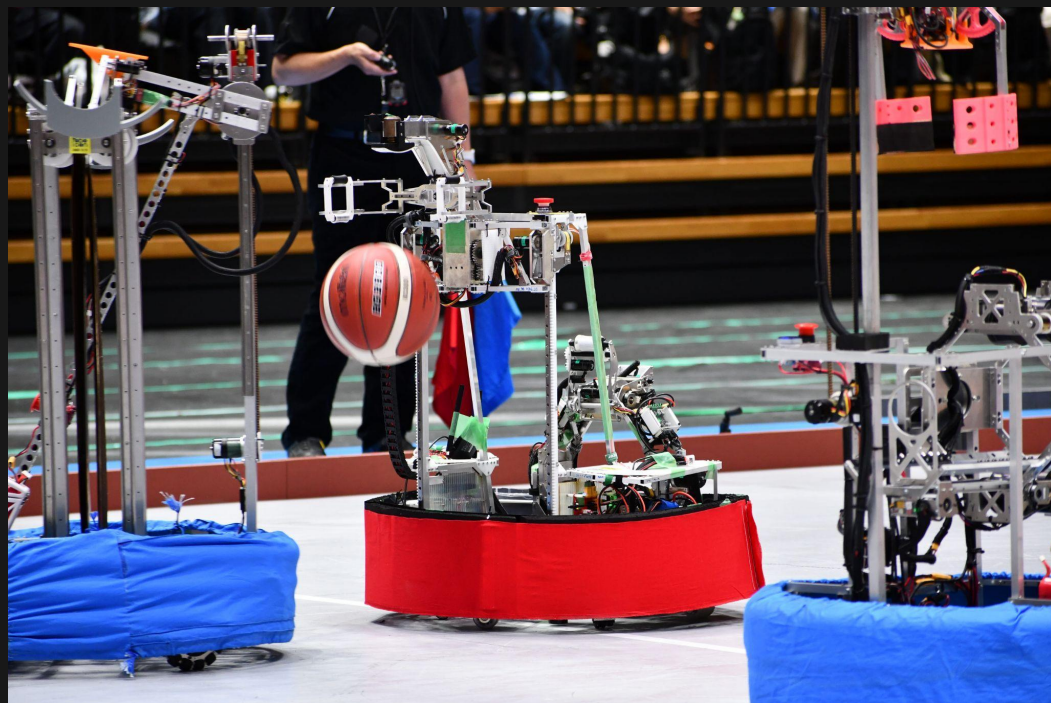
京都工芸繊維大学の
ロボコンチーム

「NHK学生ロボコン」
を主に対象

部員数およそ60名

直近の実績

- NHK学生ロボコン2024
技術賞



NHK学生ロボコン2025における様子



部内ライブラリ “fibril”

- 2023年末から開発が始まったROS 2向けライブラリ
- 自己位置推定・自動走行・各種ユーティリティ
- ROS 2の既存ライブラリを多く使用
→ メンテナンスコストを下げる狙い
- 機能ごとにNodeを細分化

詳しくは休憩時間などで



“fibril”の一部OSS化

ロボコン向けライブラリ: 秘匿することが多い

→ 独自技術の流出を避けたい・優位性を保ちたい

“fibril”は既存OSSの活用により独自技術が少ない

→ 公開しやすい

BehaviorTree関連や
ユーティリティ等の公開を進めている

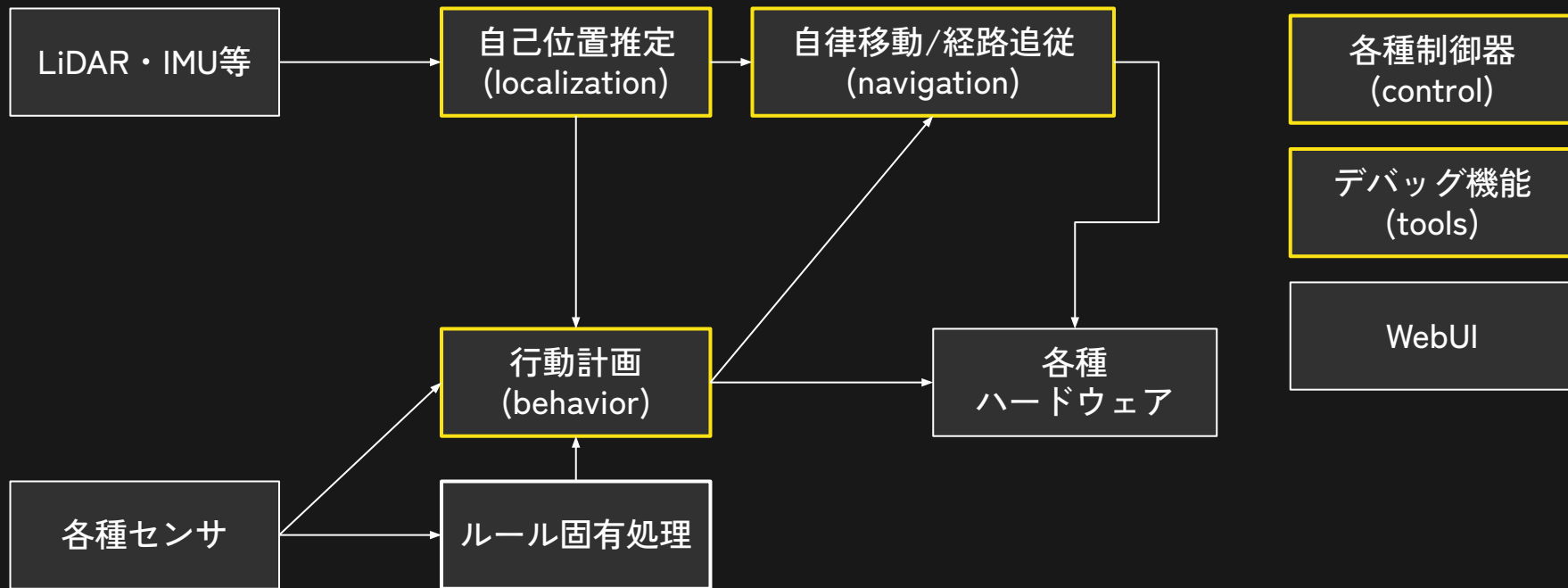


fibrilの構成

- common: ROS・組み込みファームウェア向け
- fibril_ros: ROS Node用monorepo
- envとactions: ROS環境構築・CI/CD環境
 - apt: Binary Packagesの配布
- firmware群: 組み込みファームウェア



fibrilを使うロボットの構成



通信: Ethernet / USB(CAN bus)
ROS Node数: 20程度



各種機能の特色: localization

ロボコンのフィールド $\equiv$ 直線が多く直角の組み合わせ
に最適化された軽量な点群処理

- 点群同士のマッチングではなく、
直線の検出 + 直線のマッチング



各種機能の特色: ハードウェアとの通信

- CANバスによる通信
- PCとの接続はUSB接続
- enumベースで定義されたプロトコル
 - 拡張性に限界
 - CANopenライクなプロトコルへの移行中



教育について



なぜ教育をするのか

- 新入部員は基本的に初心者
 - 場合によってはPC操作すらも得意ではない
 - 適切な知識なしではロボコンはできない → 退部等に繋がる
 - 有志の団体と大きく異なる点
- 学年制限や卒業 → 次に託せる人材の育成が必須
- 教育をすることで教える側の知識も深くなる

必然的に対象がソフトウェア開発の
経験を持たない



教育内容の紹介



1年生

出場大会

- Summer Robot Contest
- 関西春ロボコン

2年生以降

出場大会

- 関西夏ロボ
- NHK学生ロボコン



適切な教育が無い場合に起きること

- ライブラリはあるが、使われない
- 適切な使用方法で使ってもらえない
- 独学に基づく車輪の再発明で時間が溶ける

→ チームの衰退を招く



教育内容の紹介



環境構築が容易・教えやすいマイコンボード中心
C++の練習・簡単な制御・少し複雑な制御まで行う

ライントレース

小型のロボット



教育内容の紹介



環境構築が容易・教えやすいマイコンボード中心
C++の練習・簡単な制御・少し複雑な制御まで行う

2022年前後から行っていて、実績がある



教育内容の紹介

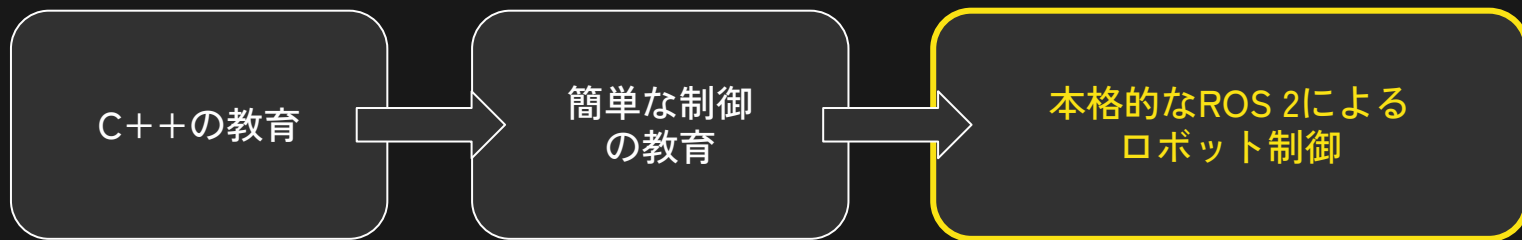


特徴

- RTOSスレッドを避けイベントループを採用
→ ROS 2 Executorへのスムーズな導入に繋がった？
- PlatformIOの採用
→ Windows PCで完結する



教育内容の紹介



ROS 2について教育: 年度ごとに大きく変化している
→ つまづく人が多発していたため

今回の発表ではこの部分の変化を主に取り上げる



ROS 2教育 今までやったこと

- 公式チュートリアルベースの自習
 - 読み合い（輪読）
- 公式チュートリアルの簡易版（日本語）の作成
 - 教材の試行錯誤
- 「ロボットを動かす」を中心とした講習
- ROS 2ハンズオン（OUXT 片岡さん）



公式チュートリアルベースの自習・勉強会

docs.ros.orgにあるチュートリアルを各自やってもらった

- 言語の壁と「面白くない」 → 苦手意識を持つ人
🐢 とリアルなロボットが紐付かない
- 部のロボットを使う知識 ≪ チュートリアル知識
- 各自やる形式だと躓いた際にやる気を無くしたり...

ROSを学習する時期に動かせるロボットが
無かったことも面白くないに繋がった



公式チュートリアル of 簡易版（日本語）の作成

日本語の資料として以下のような内容を作成

- 基礎的なCLIの使い方
- C++によるトピック通信の方法と実践
- rosbagの使用方法

→ 公式チュートリアルよりは導入として良さそう



（その後あった）教材の試行錯誤

色々な視点から教材（記事）が作成された

- ros2cliの解説を通じて「使う」ことに特化する記事
- 「モータを動かす」のような物理重視
- C++でノード作成として簡易シミュレータ作成

内容の重複込で
60記事くらいあった



(その後あった) 教材の試行錯誤

色々な視点から教材（記事）が作成された

- ros2cliの解説を通じて「使う」ことに特化する記事
- 「モータを動かす」のような物理重視
- C++でノード作成として簡易シ

比較的良いフィードバック

内容の重複込で
60記事くらいあった



ROS 2ハンズオン (OUXT 片岡さん)

- Dockerベースの初心者向けハンズオン
- Nav2によるデモ + 解説付きのNode開発

→ 1回実施のみに留まるも、解説付きC++コードなどは
部内でも参考になっている



現在の方針: 「ロボットを動かす」を中心とした講習

1. 過去の大会のシミュレータでイメージを持ってもらう
 - ノード図なども見せる
2. IPCの考え方やノード等基本概念を覚えてもらう
3. マイコンで動かしたロボットをROSで動かす
 - あまりコードは書かない
4. 簡単なノードを書いてもらう



つまり言いたいこととして…

学生団体（大学の団体）がROS 2教育をするなら…

- 最初に「おもしろそう」となってもらう
 - 凄そうなデモや実機を最初に持ってくる
- （ライブラリが整備されているなら）
そこから始めたほうがいい
- 日本語のほうがいい



オマケ: 誤った独学の例

- モノリスなNodeの誕生
→ マイコン時代のコードがほぼそのままやってきた形



ご清聴ありがとうございました

