

<http://code.google.com/p/rtm-ros-robotics/>

エージェントシステム

2011/04/27

岡田 慧

<http://code.google.com/p/rtm-ros-robotics/>

ROS 概説

- アプリケーション fetch beer, elevator
- 知能モジュール navigation, arm, grasping
- ライブラリ tf, pcl, opencv, openrave
- デバイスドライバ camera_drivers
- デバッグツール rxgraph, rosbag, rostopic
- メッセージ通信 roscore, rospy, roscpp
- 起動ツール rosrun, roslaunch
- コンパイルツール rosmake
- ファイルシステム roscd, rosls
- インストールツール rosininstall

<http://code.google.com/p/rtm-ros-robotics/>

インストールツールとファイルシステム

- `rosinstall` PATH [URI] options
- 50以上の研究室(プロジェクト, 個人含む)がリポジトリを公開. 1600位常のパッケージ.
<http://www.ros.org/wiki/Repositories>
- チェックアウトするソースコードの一覧, チェックアウト処理, 環境変数の設定を管理
- `rosllocate info [PACKAGE]` で調べる
- `.bashrc` の `source` を忘れないように!!!
- `roscd`, `rosls`などが利用できるようになる.
- <http://www.ros.org/wiki/rosinstall>
- <http://www.ros.org/wiki/rosbash>

<http://code.google.com/p/rtm-ros-robotics/>

パッケージ(スタック)

- ソースコード, ライブラリ, データ, 設定ファイルなどの集合. パッケージが集まってスタックを構成する

bin/: compiled binaries

include/package_name: C++ include headers

msg/: Message (msg) types

src/package_name/: Source files

srv/: Service (srv) types

scripts/: executable scripts

CMakeLists.txt: CMake build file (see CMakeLists)

manifest.xml: Package Manifest

mainpage.dox: many packages will often place their Doxygen mainpage documentation here

- ROSの様々なツールは“manifest.xml”を利用している(rosdep, rosmake)
- <http://www.ros.org/wiki/ROS/Concepts>

<http://code.google.com/p/rtm-ros-robotics/>

コンパイルツール

- `rosmake [options] [PACKAGE]...`
- 依存関係の存在するパッケージのコンパイルを行う.
- インクルードファイルへのパス, ライブラリのリンク等も行っている.
- <http://www.ros.org/wiki/rosmake>

<http://code.google.com/p/rtm-ros-robotics/>

起動ツール

- `roslaunch package executable`
 - package 以下の executableを起動
- `roslaunch package file.launch`
 - file.launchを使ってプログラムを起動
- <http://www.ros.org/wiki/roslaunch>
- <http://www.ros.org/wiki/roslaunch/XML>

メッセージ通信

- ノード: プロセス
 - Unixのコンセプト
 - それぞれの役割に特化したプログラムを組み合わせ、複雑な機能を実現する
 - 小さいプログラムをパイプで組み合わせてタスクをこなす
- トピック: 出版—購読型の名前付きバス
 - 非同期のメッセージング
 - 出版側は購読側を想定せずにメッセージを送信
- サービス: 要求—応答型の通信

<http://code.google.com/p/rtm-ros-robotics/>

デバッグツール

- `roscd` {info,kill,list,machine,ping}
 - ノードの情報を取得するツール
- `rostopic` {bw,echo,find,hz,info,list,pub,type}
 - トピックの情報を取得するツール
- `rosviz`
{record,info,play,check,fix,filter,compress,decompress, reindex}
 - ログファイルの生成, 取得, 操作, 修復
- `rxgraph`
 - ROSの通信状態の視覚化
- `rxplot`
 - トピックのグラフ化
- `rxviz`
 - ログファイルの視覚化
- <http://www.ros.org/wiki/>[ツール名]

<http://code.google.com/p/rtm-ros-robotics/>

より複雑な内容

- 名前空間
 - <http://www.ros.org/wiki/Names>
- パラメータサーバ
 - <http://www.ros.org/wiki/Parameter%20Server>
- YAML
 - <http://www.ros.org/wiki/ROS/YAMLCommandLine>
- WillowGarage ビジネスモデル
 - <http://www.ros.org/wiki/Events/CoTeSys-ROSSchool?action=AttachFile&do=get&target=wg-overview-rosfallschool.odp>
 - <http://pr.willowgarage.com/wiki/Talks?action=AttachFile&do=view&target=RoboDev-OpenSource-Nov08.odp>

<http://code.google.com/p/rtm-ros-robotics/>

宿題

- 基本課題
 - ホームページ>ROSサンプルプログラム>画像処理サンプル(http://code.google.com/p/rtm-ros-robotics/wiki/ROS_Example_ImageProcessing)を参考に、画面中の小さい丸で8の字を書くようにする.
- 応用課題
 - <https://code.ros.org/svn/opencv/trunk/opencv/samples/cpp/> にあるサンプルプログラムを参考に画像処理ノードを作る.
 - 顔認識, 領域分割, オプティカルフロー, ステレオ処理等
 - 結果をmpeg1にして来週発表