



OSSライセンスと ロボット用OSS 2015

2015年11月18日
OSS活用WG/技術本部副本部長/
技術高度化委員長
竹岡尚三 (株)アックス



今、OSS (オープンソース・ソフトウェア) なのか？

自動運転もOSS!「Autoware」



- ・ 名古屋大学 加藤真平先生 日本で最も進んだ自動運転 研究
- ・ 名古屋大学 開発 自動運転ソフトウェア「Autoware」サポート
- ・ 名古屋大学ら、開発済み自動運転システム一式をオープンソース化
…加藤准教授「時間をジャンプ」
 - ・ Autowareは、オープンソース・ソフトウェアとして無償配布されている

<http://response.jp/article/2015/08/26/258648.html>



大人のOSS使用は、すでに十数年の歴史あり



■1980年代中,後期～

- ・ Ciscoは、BSD UNIXベースのルータ

■1990年代末期～

- ・ Linuxが、家電各社(SONY,パナソニック,シャープなど)にて採用
- ・ 半導体企業も独自CPUに、Linuxを移植

■ナウ

- ・ Android, Linux, BSDが無ければ、IT機器を作るのが大変

※JASA会員企業 いわく:

お客様の指定で、OSS採用を行わねばならない!

■これから

- ・ ロボット用 OSSが、続々と出現中



■Arduino

- オープンソース・ソフトウェア (OSS)
+
- オープンソース・ハードウェア (OSHW)
- OSS
 - 開発環境 IDE
 - ライブラリ: 膨大な量のライブラリ
 - OSSで、世界中のみんなが開発
- OSHW
 - 回路図
 - 基板レイアウト
- お手軽試作にバッチリ
 - ハードウェアは、アマチュア品質だが…

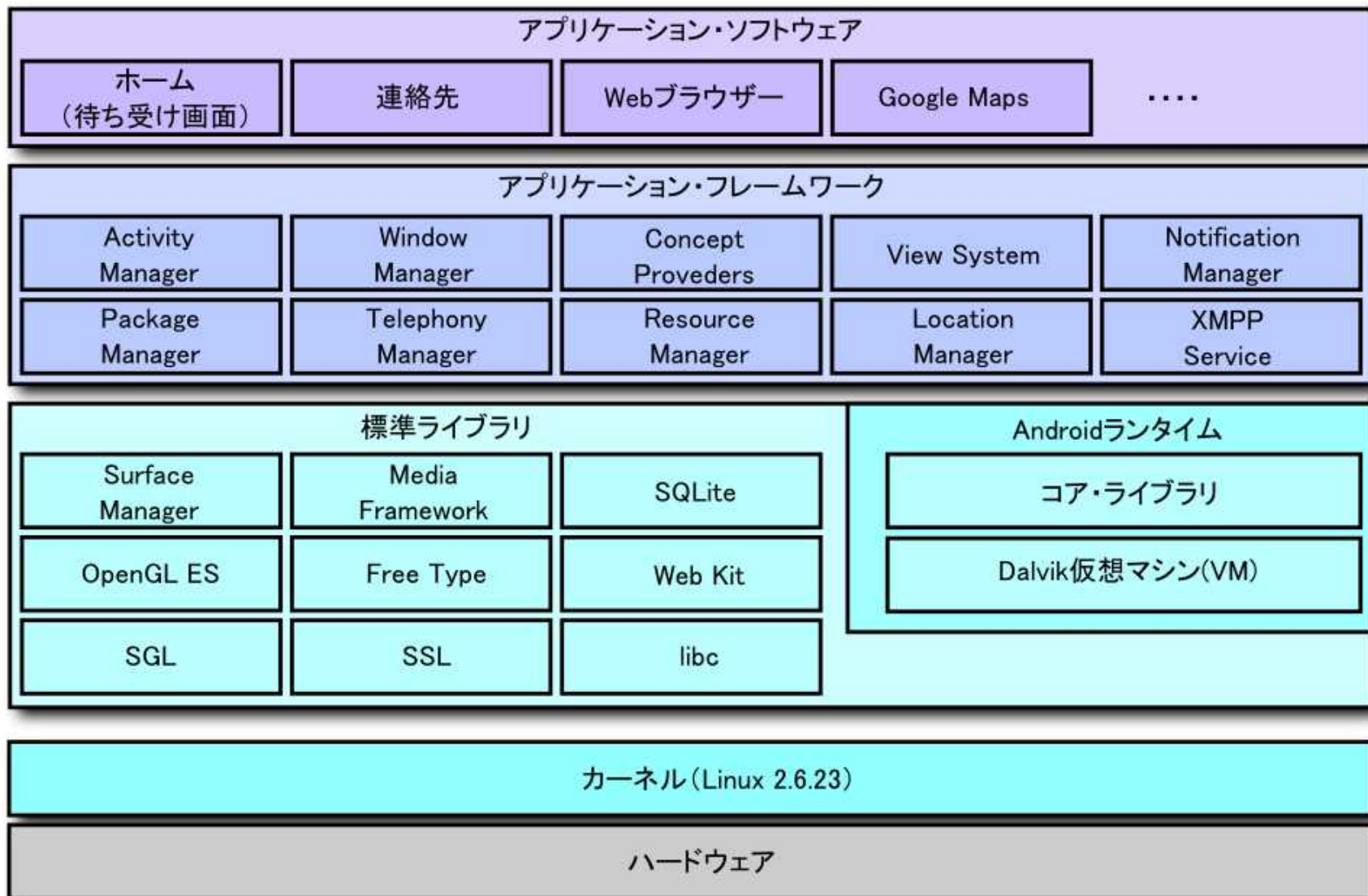
■Raspberry Pi

- Linux搭載、ハードウェア情報公開



- OS(Linux, BSD, OpenSolaris)
- インターネット接続
- Windowシステム(窓、描画管理)
- 文字レンダリング
- 3Dグラフィックス
- 仮名漢字変換
- ブラウザ、メーラ
- マルチメディア(音楽、動画)の記録/再生
- プログラミング言語 Ruby,Perl,PHP…
- オフィススイート

Androidのアーキテクチャ



- 蓬田宏樹、他著「Androidの野望」日経エレクトロニクス 2007年12月17日号 p.47-69 より引用



- 「無料だから嬉しい」とか言っていると、負ける
- 特定企業のOSとは違い、ソースがあるので、
- 理解し、独自の改良が可能
- デファクト・スタンダードがOSSなら、特定ベンダに
囲い込まれない
- 特定のソフトウェアのリリース(バージョンアップ)に
引っ張られない
 - ・独自に品質を上げたソフトウェアを、自由に維持できる

OSSって面倒くさい !?



- ライセンス問題
 - ソースコード公開義務(!?)
 - 誤解が多い
- 品質問題
 - 適当に作られた無料のソフトウェアの品質は?
 - どの誰だか、知らない人が作ったものでしょ…
 - タダのものがそんなに良ければ、プロは商売あがったりだよ
 - タダのものが、かなり良い、ということは、非常に頻繁にある
 - 品質の押さえ方は、大事
- 実は、タダ乗りも多い
 - タダ乗りは、悪いことではない
- OSS共通の問題をみんなで考えよう
→ JASA OSS活用委員会

OSSのせいで、仕事が減る?!



- これまで
 - 受託して、仕様を起こし、一品ずつ作っていた
 - ライブラリ/モジュール(サブルーチン)を作って販売していた
- しかし、現在
 - JPEG デコード・ルーチンの開発を委託する顧客がいるだろうか?
→OSSに乗るしかない
- 顧客が、OSS前提で仕様を決めている
 - サービス主導の時代→OSSで素早くシステム構築
- Windows+ブラウザで、できていることは、できて欲しい
- 高度なロボットを、すぐに作りたい
- OSSを活用する世界をみんなで考えよう
→ JASA OSS活用委員会

■ GPL

- ・ ソースコードは、そのバイナリを持っている人には、公開されるべき。という思想
- ・ 改変したソースコードも、公開されなければならない
 - 独自に改良したコードも公開しなければならない
 - (組み込みの世界では、厳しいと言えるだろう)
- ・ GPLのソフトウェアにリンクしたソフトウェアのソースコードも、公開しなければならない
 - 独自に作ったソフトウェアも、GPLソフトウェアにリンクしたら、自動的に公開義務が生ずる
 - (極めて厳しい、と言えるだろう)

■ LGPL

- ・ ライブラリのために作られたライセンス
- ・ LGPLライブラリは、ダイナミック・リンクであれば、リンクを行っても、他のソフトウェアに、GPL (LGPL)が伝播することは無い
- ・ 小さな組み込みシステムでは、ダイナミック・リンクが現実的ではないかも
- ・ → 結局、小規模 組み込みシステムでは厳しい、と言えるかも

■ GPL v3問題

- ・ ソースコードを寄付した人が持っている特許が、そのソースコードに含まれていた場合、当該 特許を無償で無制限に許諾しなければならない
- ・ 大企業の多くが、GPL v3のソフトウェアの開発には関わっていない
 - Linuxは、GPLv2 維持宣言をしている(2014年秋 現在)

GPL:よくある誤謬



- ソースはインターネットで配布しなければならない
 - ・ 嘘です
- ソースを誰にでも上げなければいけない
 - ・ 嘘です
 - ・ バイナリを持っている人にだけ、ソース入手の権利があります
- 誰とライセンス契約をすればいいのか
 - ・ する必要はありません
- GPL違反をするとストールマンに訴えられる
 - ・ 嘘です
 - ・ 日本では、プログラムの著作権者しか訴える権利がありません
 - ・ ストールマンが怒る可能性はありますが、それには法的な意味はありません

BSD/MIT風ライセンス



■BSDライセンス

- ・ MITライセンスは、BSDと同じ
- ・ 何をしても、ソースコードの開示義務は無い
- ・ 古いBSDライセンスは「宣伝条項」があったが…
 - 宣伝条項: すべての広告にライセンス表示を入れる
- ・ 現在のBSDライセンスは、「宣伝条項」は無い

■Apacheライセンス

- ・ Apache Licenseのコードが使われていることを知らせる文言を入れる。という義務がある程度
- ・ 何をしても、ソースコードの開示義務は無い
- ・ ライセンスを変更してもよい
- ・ ライセンスされたファイルに元々ある著作権と特許権の記述はそのまま保持
- ・ Apache Licenseのコード中に特許を含んでいる場合、特許を無償で無限に許諾しなければならない

ライセンスの知識は必要



■正しい知識があれば、問題は出ない

- ・ GPLであっても、ソフトウェア作成者は、
- ・ みんなに使って欲しいと考えている
 - ユーザが困るようなことはしていない(はず)

■ズルは止めよう

- ・ GPLのコードの断片を、独自開発のソフトウェアに混ぜると、公開義務が生じる
 - BSDライセンスのものは、ライセンス表示をソースコードに入れば、問題無い



ロボット用 OSS



■英語版Wikipedia「Open-source robotics」の項

http://en.wikipedia.org/wiki/Open-source_robotics

■フル・ロボット・プロジェクト32個

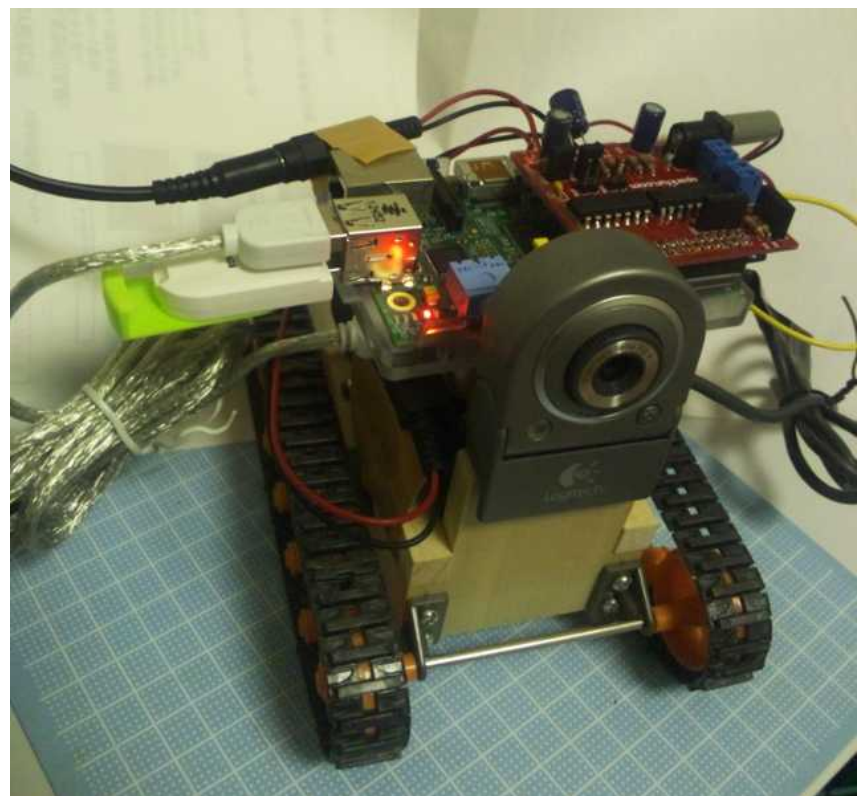
■ソフトウェアのみのプロジェクト 28 個

■オープン・ハードウェア・プロジェクト:3個

OSSロボット例 (たけおか謹製)



- 人間の顔を見つけたら、追尾する
 - ・ 顔の位置を判断して、右へ行くか、左へ行くかを定める
- Linux+OpenCV+OpenEL
- OpenCVは顔認識などを含む
- 総合的な画像処理ライブラリ
- DCモータの制御はOpenEL
- OpenELはJASAなどが推進している
- ロボット用の下位ハードウェアの
- 抽象化層の規格
- ハードウェア
 - ・ RaspberryPi
 - ARM11@700MHz
 - 512MBytesRAM
 - ・ USBカメラ
 - ・ DCモータ



■画像処理のミドルウェア

- ・ Intel, Willow Garage が開発
- ・ いろんなものが入っている
- ・ 極めて便利
 - 画像処理 (Image Processing)
 - 構造解析 (Structural Analysis)
 - モーション解析と物体追跡 (Motion Analysis and Object Tracking)
 - パターン認識 (Pattern Recognition)
 - カメラキャリブレーションと3次元再構成 (Camera Calibration and 3D Reconstruction)
 - 機械学習
 - ユーザインタフェース

OpenCVによる顔認識



■OpenCV

- ・ 一般的画像処理
- ・ 画像認識
 - 学習器
 - 認識器
- ・ カメラ入力部

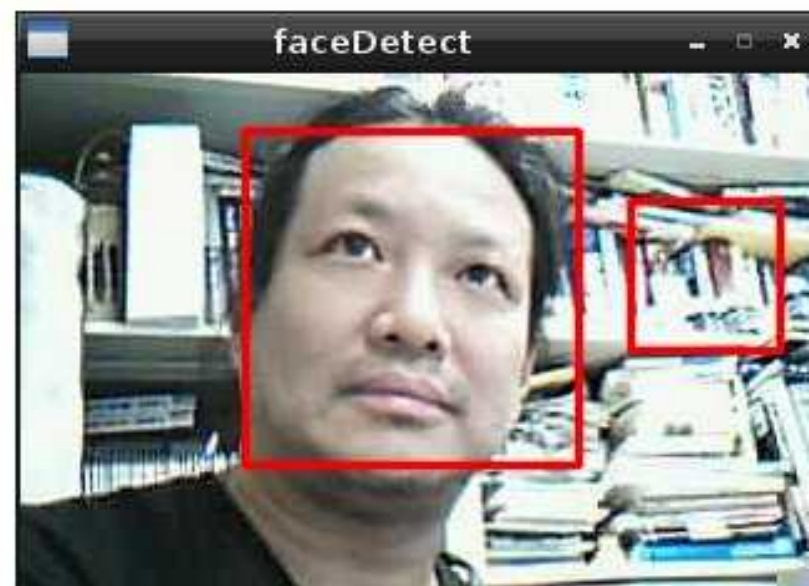
■USBカメラを接続

- ・ ロジクール QCAM-E2500, 30万画素
- ・ Logicool 861200
- ・ カメラ入力

■顔認識

- ・ 学習済みデータが用意されている
- ・ 簡単に使用できる
- ・ ロボットの目玉にでも

「たけおか opencv raspberrypi」でググる





自動運転 ミドルウェア OSS

自動運転もOSS!「Autoware」



- 名古屋大学 加藤真平先生のAutoware
 - 日本で最も自動運転の研究が進んでいる
 - 名古屋 守山市で、公道を自動運転走行
 - 自動車会社もスポンサー



<http://www.pdsl.jp/fot/autoware/>

- 愛知県知事が名古屋大学自動車運転カーに試乗

<http://www.aisantec.co.jp/ir/library/as20150618.pdf>

- 名古屋大学 開発 自動運転ソフトウェア「Autoware」サポート
- 名古屋大学ら、開発済み自動運転システム一式をオープンソース化
…加藤准教授「時間をジャンプ」
 - Autowareは、オープンソース・ソフトウェアとして無償配布されている

<http://response.jp/article/2015/08/26/258648.html>

- <http://www.pdsl.jp/%E6%97%A5%E6%9C%AC%E8%AA%9E%E3%83%88%E3%83%83%E3%83%97/>
- http://news.mynavi.jp/series/coolchips18_auto_car/003/



基本機能

- ・ 3次元自己位置推定
- ・ 3次元地図生成
- ・ ナビアプリ
- ・ 経路生成
- ・ 経路追従 (0~60km/h)
- ・ 交差点右左折／一旦停止
- ・ 自動停止
- ・ 自動駐車
- ・ 車両認識
- ・ 歩行者認識
- ・ レーン認識
- ・ 標識認識
- ・ 路上サイン認識
- ・ 信号認識
- ・ 移動体追跡
- ・ レーンチェンジ

<http://www.pdsl.jp/fot/autoware/> から引用

自動運転OSS Autoware 構成・特徴



- ・ ZMP社のロボカー製品を車両として利用可能
- ・ アイサンテクノロジー社の高精度3次元地図を利用可能
- ・ 測位衛星技術社、Javad社のGNSS測位機を利用可能
- ・ 北陽電機社、Velodyne社の3次元スキャナを利用可能
- ・ インクリメントP社のAndroid端末用ナビアプリを利用可能
- ・ インテル社のチップほか, Linuxが稼働する環境で利用可能
- ・ NVIDIA社のGPGPU技術 (GeForce, Tegra) を利用可能
- ・ イーソル社の車載向け組込みシステムに対応
- ・ インターネット上のデータベース接続などもサポート
- ・ アックスが有償サポート提供

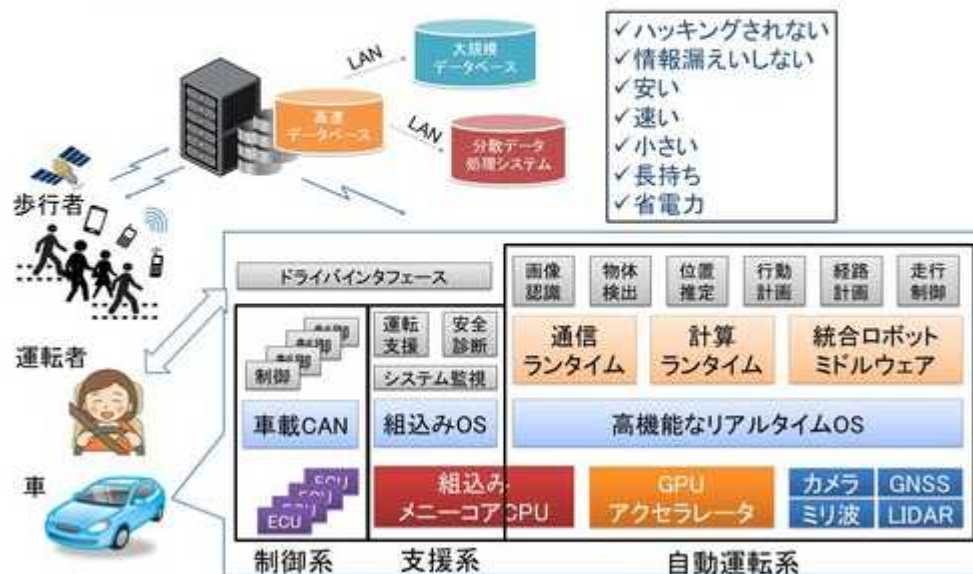


<http://www.pdsl.jp/fot/autoware/> から引用。一部、付加

写真は <http://www.aisantec.co.jp/ir/library/as20150618.pdf> から抜粋

自動運転知能の基盤ソフトウェア

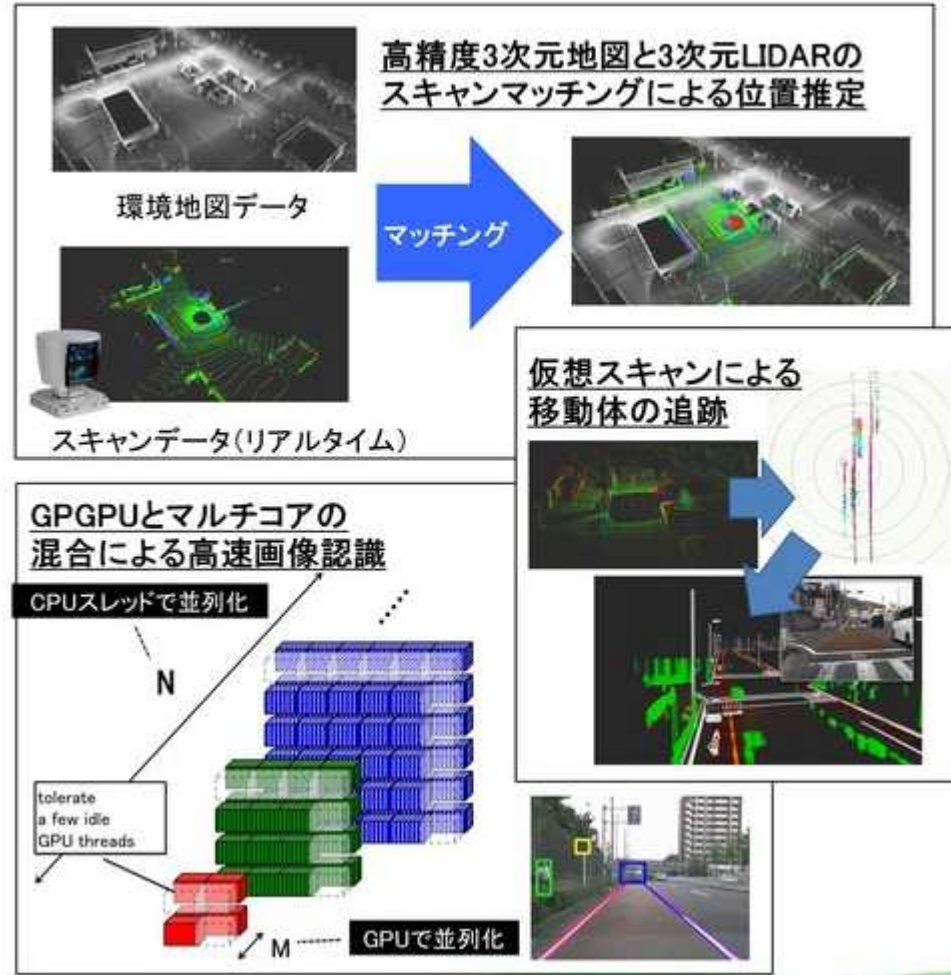
- ・オープンソース開発
- ・多様なプロセッサ、多様なセンサに対応
- ・リアルタイム性の確保



車載システムとクラウドを一体とした基盤ソフトウェア

<http://www.pdsl.jp/fot/autoware/> から引用

要素技術



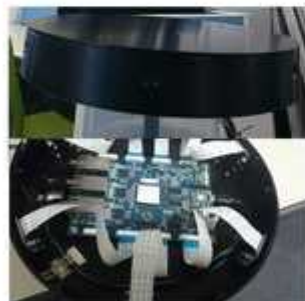
<http://www.pdsl.jp/fot/autoware/> から引用

コンピューティングハードウェア



ハイエンドGPU搭載PC
(10TFlops@1000W)

組み込みGPU搭載Drive PX
(1TFlops@10W)



移動体データ解析用
Hadoop PCクラスター
&
ディープラーニング用
超並列GPGPUクラスター



全方位カメラ用FPGA

車両および環境センサ

- ・リファレンスカー
- ・多様なセンサを搭載して性能検証



Velodyne HDL-64/32
(3D LIDAR)
Point Grey Ladybug 5
(Camera)



JAVAD RTK-GNSS
(GNSS/GPS)



Point Grey Grasshopper3



IBEO LUX 8L (3D LIDAR)

<http://www.pdsl.jp/fot/autoware/> から引用



ドローン用 OSS



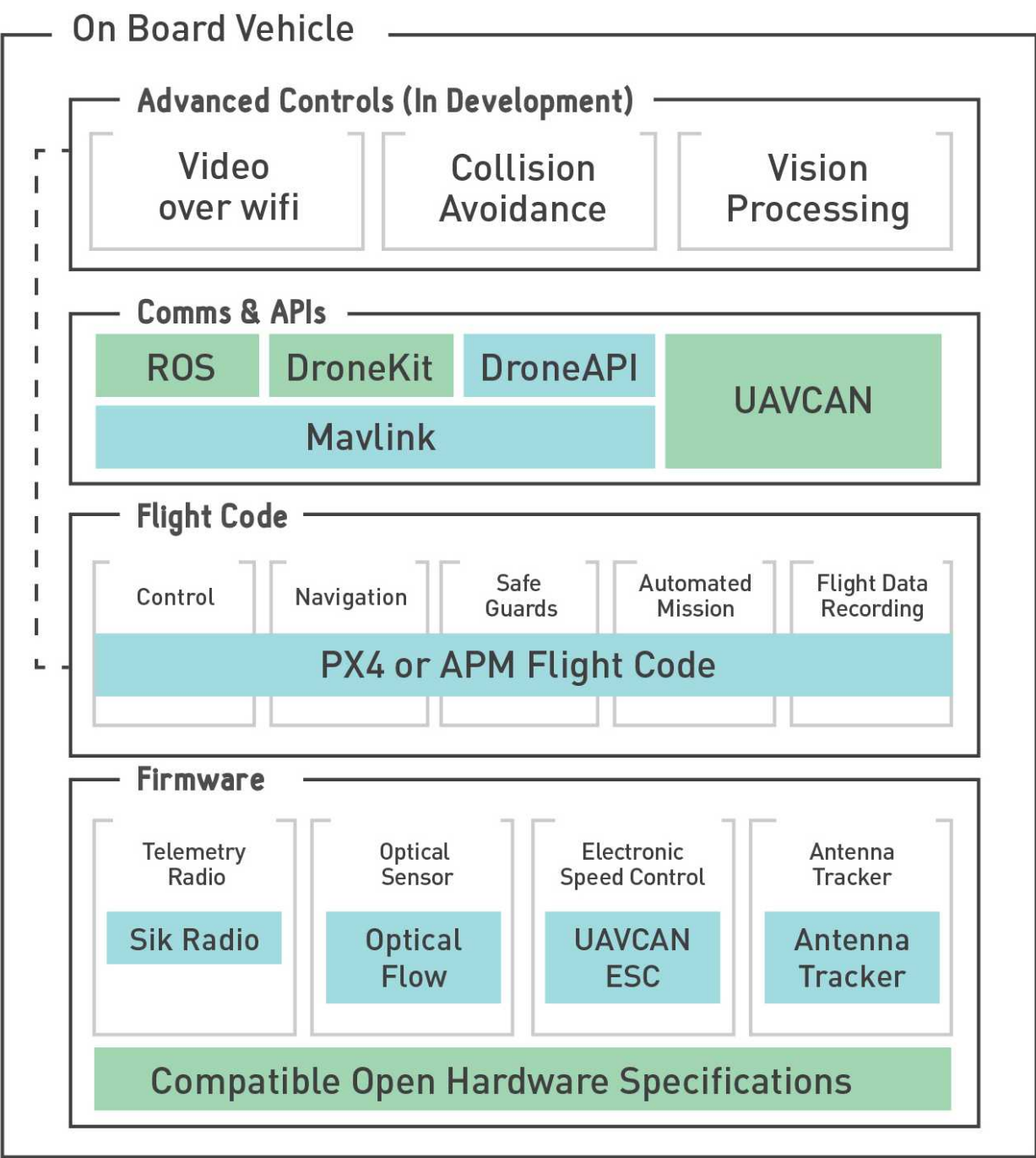
<https://www.dronecode.org/>

- Linux Foundationの取りまとめで
- 無人飛行体 Unmanned Aerial Vehicles (UAVs)を開発
- オープンソース・ソフトウェア + オープンソース・ハードウェア

- オン・ボード(機体上)と、オフ・ボードのソフトウェアがある
 - オンボード: 機体の制御、センサー制御
 - オフボード: リモコン、データ収集/記録

写真は

https://www.dronecode.org/sites/dronecode/files/styles/dronecode_header/public/front_page_slides/images/drone_video_slide.png?itok=7C7IFYIP より引用



Off Vehicle

Simulators

JMavSim

SITL

ROS & Gazebo

Flight Planning (GCS)

QGroundControl

Mission Planner

APM Planner

MAVProxy

Tower

Andro Pilot

UGCS

DroneDeploy

Flight/Data/ Log Analysis

Replay Tools
& Regression
Test Suites

Dronecodeのソースコード



Dronecode は、いくつかのプロジェクトとリポジトリでできている。
ソースコードは、下記から

Ardupilot <https://github.com/diydrones/ardupilot>

PX4 <https://github.com/PX4>

Pixhawk <https://github.com/Pixhawk>

MAVLink <https://github.com/mavlink>

UAVCAN <https://github.com/uavcan>

ROS <https://github.com/ros>

Other repositories <https://github.com/Dronecode>



写真は <http://diydrones.com/profiles/blogs/arp4-drone-px4-ar-drone-kits> より引用



ロボット用 OSS ミドルウェア



■NXJ

- ・ Lego NXTロボット・キット用のOSS Javaプログラミング環境

■CLARAty

- ・ JPL(ジェット推進研究所)で開発されている、Mars(火星)プログラムの一部として。

■ROS (Robot Operating System)

- ・ BSDライセンス。すでに50以上のロボットで動作

■URBI

- ・ C++の分散/組み込み コンポーネント・フレームワーク
- ・ + 並列/イベント駆動 制御スクリプト言語

■Player (robot framework)

■OpenRTM-aist (robotics technology middleware)

- ・ 日本の産総研が開発

※上記のいくつかは、ロボット・ミドルウェアである

ロボット用ミドルウェアとは



■フレームワーク

- ・コンポーネントをつなぐ
- ・コンポーネントの独立性を高めさせる
- ・オーバーヘッド無し
- ・存在を感じさせない

参考

http://en.wikipedia.org/wiki/Robotics_middleware

■ Robot Operating System

<http://wiki.ros.org/>

■ ロボットのミドルウェアと、たくさんのコンポーネント

■ BSDライセンス

■ すでに50以上のロボットで動作

■ 2007年 the Stanford Artificial Intelligence Laboratory で開発

- ・ Stanford AI Robot STAIR project

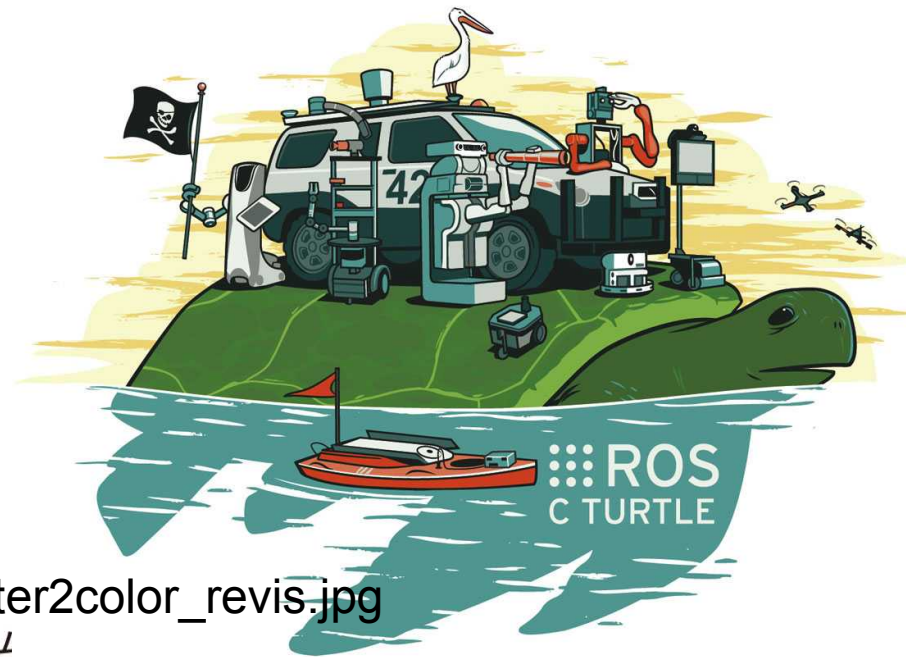
■ 2008～2013年 Willow Garageが中心に開発中

■ 2013年～ the Open Source Robotics Foundation

※参考

http://en.wikipedia.org/wiki/Robot_Operating_System

- データの流れに応じて、コンポーネントをつなぐ
- 自動車の自動運転でも採用
- OpenCVも含まれている
- 雑に言ってしまうえば…
- ロボットを作るためのソフトウェア部品の多くが含まれている



画像は下記より引用

http://www.ros.org/news/resources/2010/poster2color_revis.jpg

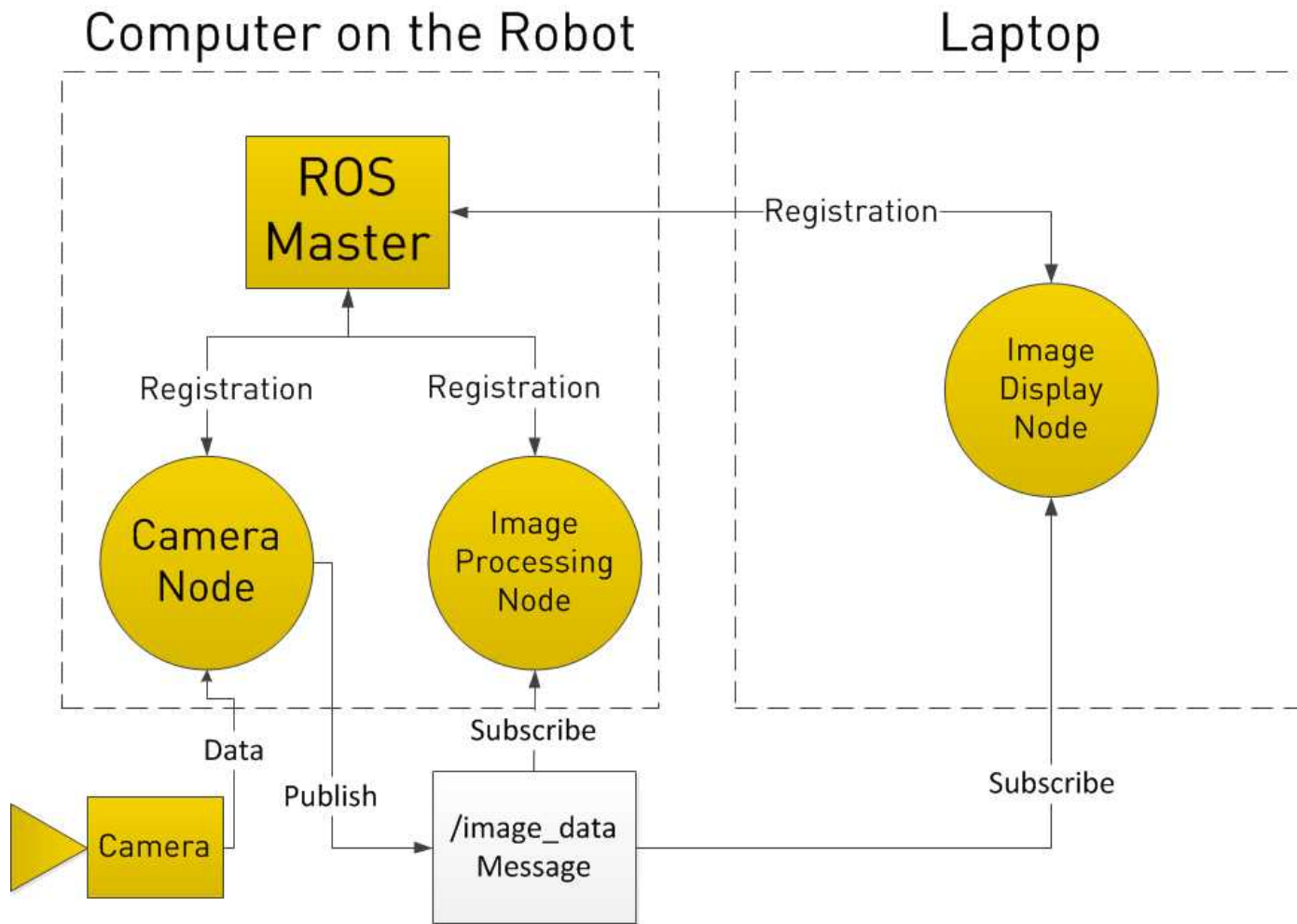
ROSパッケージの適用分野



- 知覚 (Perception)
- 物体認知 (Object Identification)
- セグメンテーションと認識 (Segmentation and recognition)
- 顔認識 (Face recognition)
- ジェスチャ認識 (Gesture recognition)
- 動作追跡 (Motion tracking)
- エゴモーション (Egomotion)
- 運動理解 (Motion understanding)
- 運動からの構造 (Structure from motion (SFM))
- ステレオビジョン:2台のカメラを通して、奥行き知覚
 - ・ (Stereo vision: depth perception via two cameras)
- モーション (Motion)
- 移動ロボット工学 (Mobile robotics)
- 制御 (Control)
- 計画 (Planning)
- 把握 (Grasping)

※http://en.wikipedia.org/wiki/Robot_Operating_Systemから引用、和訳

■データの流れに応じて、コンポーネントをつなぐ



<http://www.clearpathrobotics.com/blog/how-to-guide-ros-101/> より引用

■ROSエコ・システムのソフトウェアは3つのグループに分けられる

- ・ (1) 言語やプラットフォームから独立したツール
 - ROSベースのソフトウェアを作ったり分散させるために使用
- ・ (2) ROSクライアント・ライブラリ実装
 - roscpp, rospy, roslispなど
- ・ (3) アプリケーションに関連するコードを含んだパッケージ
 - ひとつ以上のROSクライアント・ライブラリが使用

※http://en.wikipedia.org/wiki/Robot_Operating_Systemから引用、和訳

RTミドルウェア

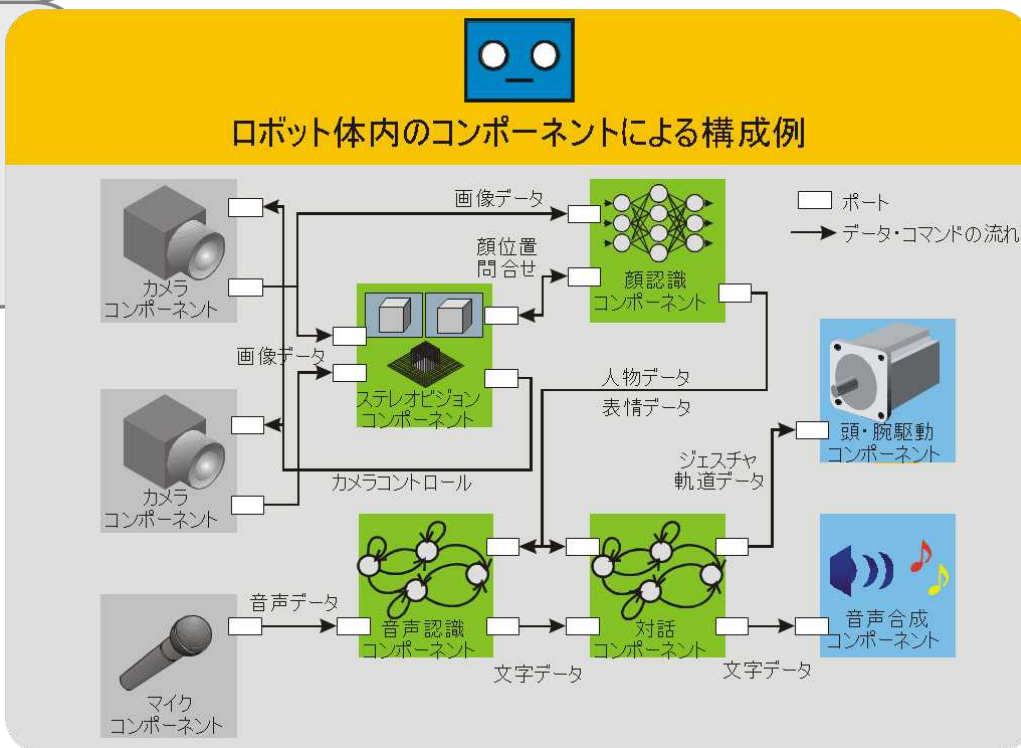
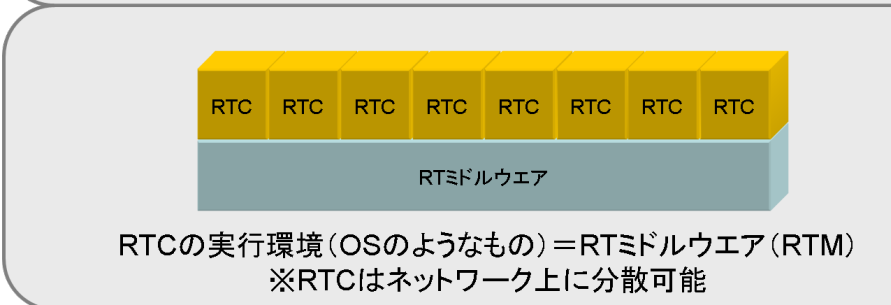
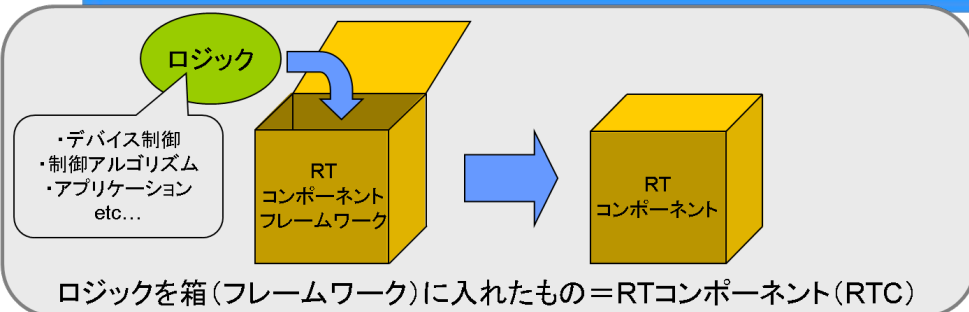


- 産総研などが開発しているロボット用ミドルウェア
 - ・ RTコンポーネントは、OMGにて、国際標準化
- RT (Robot Technology/Robotic Technology)とは
 - ・ ロボット機能要素
- RTミドルウェアとは
 - ・ 様々な機能要素をモジュール化 (RTコンポーネント)
 - ・ RTCを、ソフトウェア的に統合するためのプラットフォーム
- RTミドルウェアの目的
 - ・ 仕様をオープンにする
 - ・ 様々な実装同士が相互接続できるようにする
 - ・ オープンアーキテクチャのプラットフォームを確立する
- 通信はCORBAベース
- 「OpenRTM-aist」は、RTミドルウェアの産総研による実現
 - ・ ライセンスは、LGPLおよび産総研と個別に契約するライセンスのデュアルライセンス方式



画像引用元,参考 <http://www.openrtm.org/openrtm/ja/node/161#toc5>
<http://www.openrtm.org/openrtm/ja/content/openrtm-aist-official-website>

RTミドルウェア

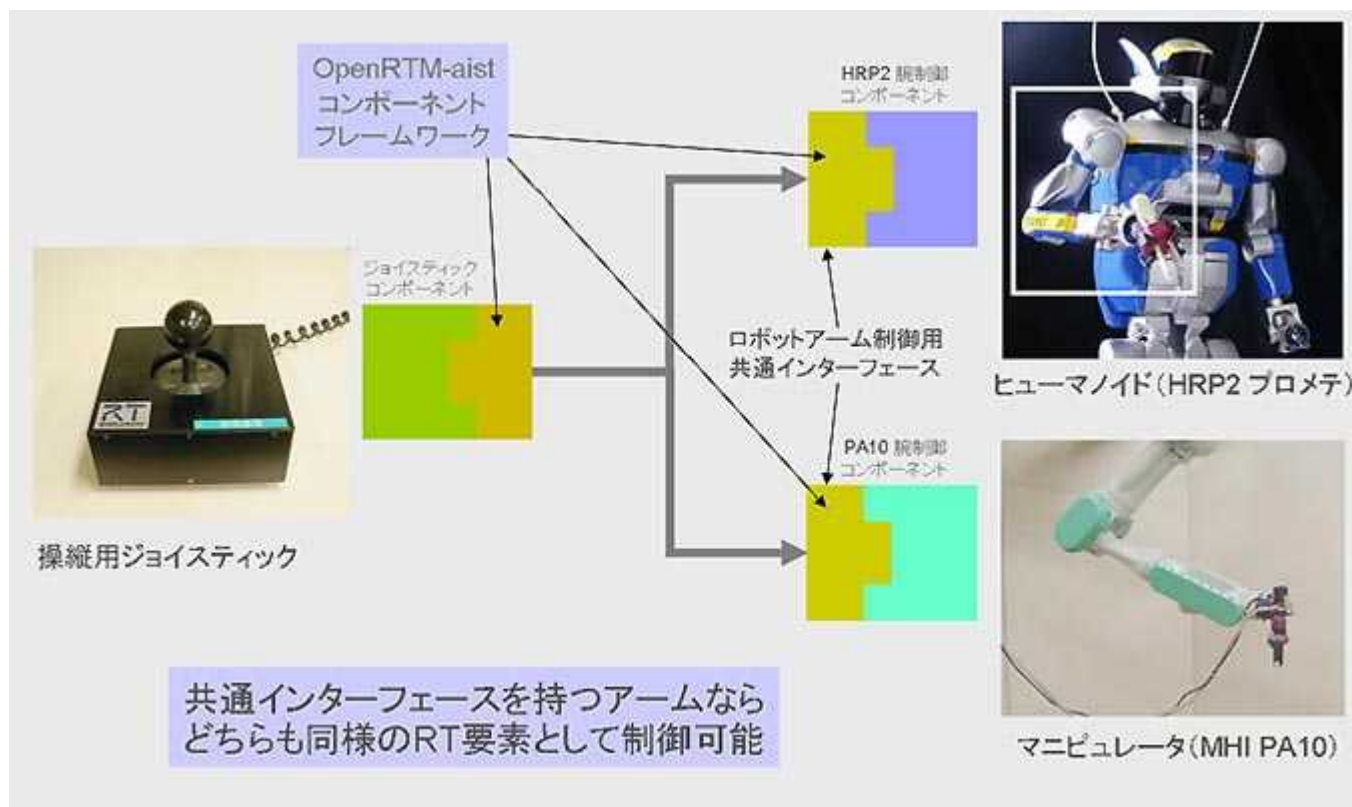


<http://www.openrtm.org/openrtm/ja/content/rt%E3%83%9F>

<http://www.openrtm.org/openrtm/ja/content/rt%E3%83%9F%E3%83%89%E3%83%AB>

<http://www.openrtm.org/openrtm/ja/content/rt%E3%83%9F%E3%83%89%E3%83%AB%E3%82%A6%E3%82%A8%E3%82%A2%E3%81%A8%E3%81%AF%E3%81%BC>

%9F-0 よりコピー



引用元

http://www.aist.go.jp/aist_j/press_release/pr2005/pr20050224/pr20050224.html

RTC(RTコンポーネント) OMG標準



- RTM / OpenRTM-aist は
 - ・ コンポーネントモデル
 - ・ そのインターフェース
 - ・ である「RTC」が OMG 国際標準
- OMG (Object Management Group)
 - ・ 1989年に設立されたソフトウェア標準化団体
 - ・ CORBA (Common Object Request Broker Architecture)分散オブジェクトミドルウェア:
 - ・ UML (Unified Modeling Language) ソフトウェアモデリング言語
 - ・ などを策定・管理している組織
- RTCのインターフェース仕様
 - ・ OMG において, 産総研と米国ミドルウェアベンダ RTI (Real Time Innovations) により標準化
 - ・ RTC (Robotic Technology Component) Specification
(<http://www.omg.org/spec/RTC/1.0/>) として2008年4月に公式リリース

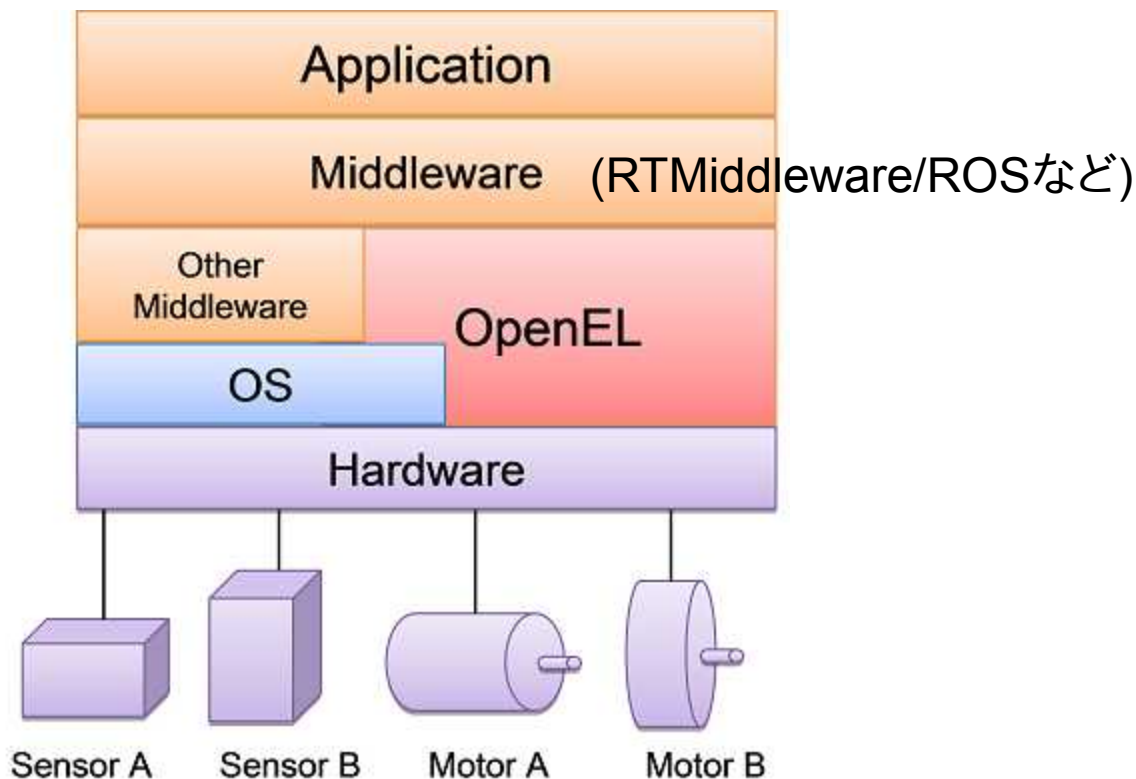
<http://www.openrtm.org/openrtm/ja/content/rt%E3%83%9F%E3%83%89%E3%83%AB>

<http://www.openrtm.org/openrtm/ja/content/rt%E3%83%9F%E3%83%89%E3%83%AB%E3%82%A6%E3%82%A8%E3%82%A2%E3%81%A8%E3%81%AF%EF%BC%9F-0#toc4> より引き写し

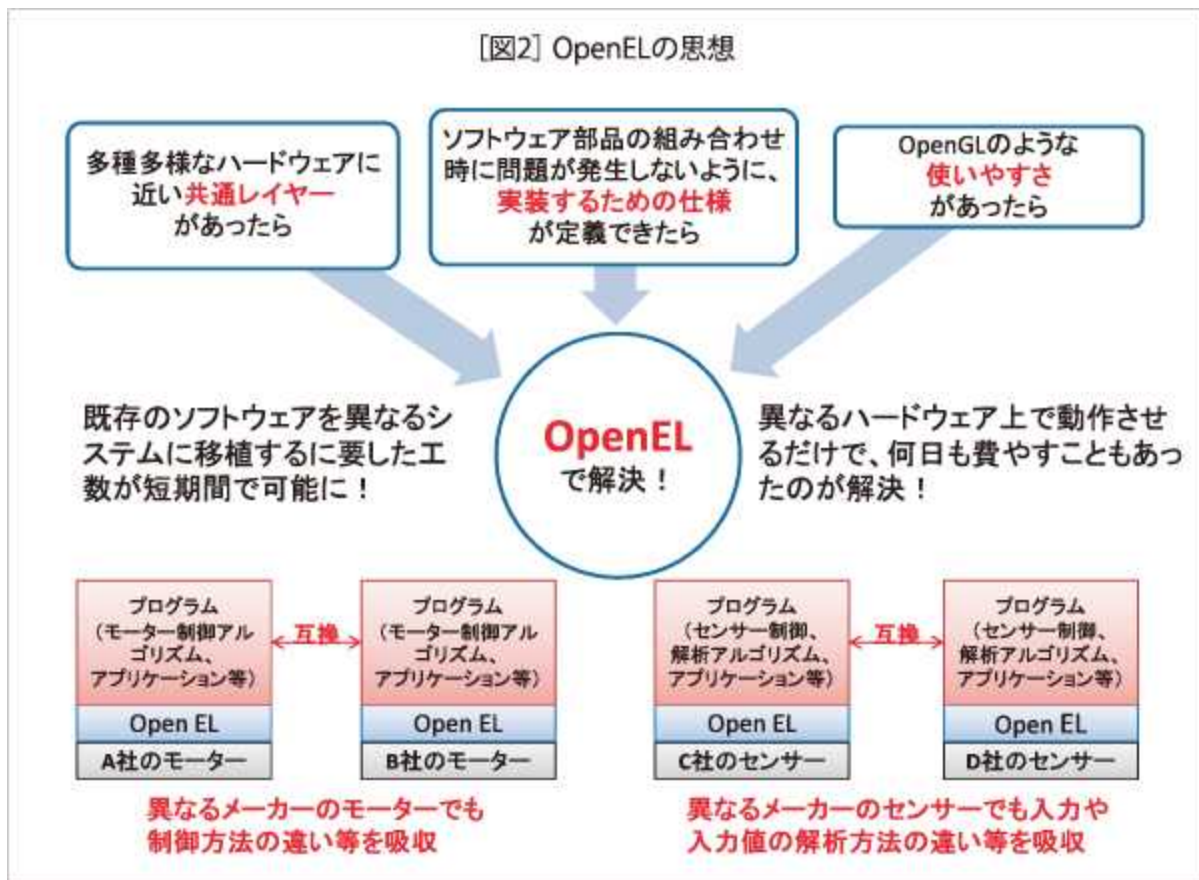
- JASA, 産総研などが推進しているフレームワーク
- 国際規格にするべく、OMGに提案中
- RTミドルウェアなどのコンポーネントの可搬性を高める
- ハードウェアに近い層を、抽象化
- OpenEL準拠で書かれたソフトウェア(RTC)は、
- ハードウェア・ドライバに依存せずに動作する

参考

http://jasa.or.jp/openel/Main_Page/ja#OpenEL.E3.81.AE.E6.AD.B4.E5.8F.B2.E3.81.A8.E6.99.AE.E5.8F.8A.E3.83.BB.E5.95.93.E7.99.BA.E6.B4.BB.E5.8B.95



引用元 http://jasa.or.jp/openel/Main_Page/ja



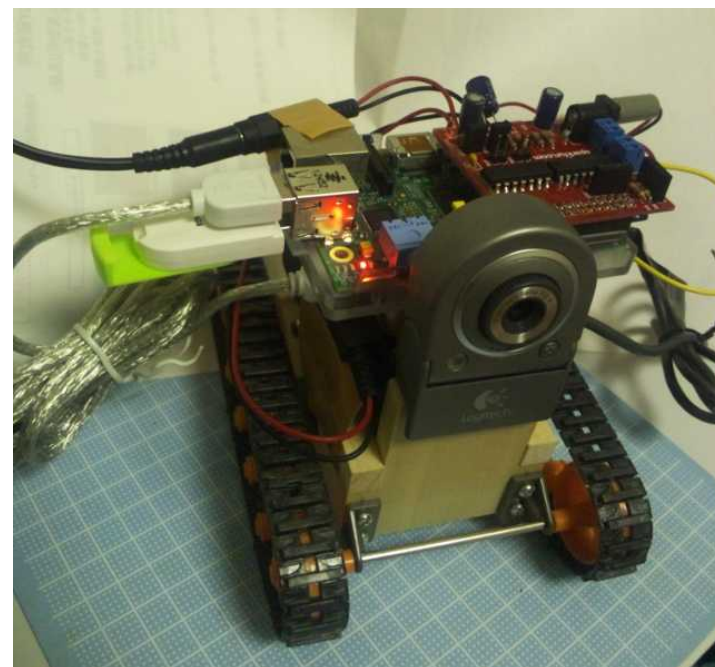
引用元 http://www.jasa.or.jp/top/activity_bulletin/bulletin041-05.html

OpenEL使用 オレオレ実例



- ・ライトレーサ
マルチスレッドBASICインタープリタ
OS無し
OpenELでDCモータ駆動

PIC32MX250
(MIPSコア@50MHz、ROM128KB,
RAM32KB)
秋月で360円



- ・顔追尾ロボット
OpenCV under Linux
OpenELで DCモータ駆動

Raspi
(ARM11@700MHz, 512MBメモリ)



「OSSライセンスとロボット用OSS 2015」

2015/11/18 発行

発行者 一般社団法人 組込みシステム技術協会
東京都中央区日本橋大伝馬町 6-7 住長第2ビル 3階
TEL: 03(5643)0211 FAX: 03(5643)0212
URL: <http://www.jasa.or.jp>

本書の著作権は一般社団法人組込みシステム技術協会(以下、JASTA)が有します。
JASTAの許可無く、本書の複製、再配布、譲渡、展示はできません。
また本書の改変、翻案、翻訳の権利はJASTAが占有します。
その他、JASTAが定めた著作権規程に準じます。