



RTミドルウェア コンテスト2010

<http://www.openrtm.org/rt/RTMcontest>

SICE システムインテグレーション部門講演会(SI2010)

2010年12月24日

東北大学 川内キャンパス



エントリー

[2B1-2 RTコンポーネントとscilabを繋ぐツールボックス「RTC-scilab」の開発](#)

菅佑樹(リバスト)

[2B1-3 ロボットモデル作成ツールの開発](#)

宮本信彦(立命館大学)

[2B1-4 車輪型移動ロボットのための総合開発環境コンポーネント群](#)

高橋亮, 李周浩(立命館大学)

[2B1-5 可変構造ソフトウェアのためのノードコンポーネント](#)

小山順平, 森谷浩太, 國井康晴(中央大学)

[2B2-1 汎用データ処理のための演算コンポーネント](#)

佐々木毅, 橋本秀紀(東京大学)

[2B2-2 シミュレーションと実記制御をシームレスに実現するためのRTコンポーネント](#)

清水昌幸(静岡大学), 音田弘(産業技術総合研究所)

[2B2-3 移動ロボットのネットワーク化と制御用RTコンポーネント](#)

桑原潤一郎, 竹村憲太郎, 末永剛, 高松淳, 小笠原司(奈良先端科学技術大学)

[2B2-4 自己拡張するRTコンポーネントの実装](#)

松坂要佐(産業技術総合研究所)



スポンサー

奨励賞(特別協賛) (副賞2万円+提供製品) (2件)

- テクノロジーアート賞【提供:[株式会社 テクノロジーアート](#)】
製品版 [Pattern Weaver for RT-Middleware](#) のライセンス提供
- やっぱ、カメラたくさんで賞 【提供:[株式会社 ビュープラス](#)】
USB非圧縮カメラ [FMVU\(FMVU-03MTC-CS\)](#) 2台提供

奨励賞(個人協賛) (副賞1万円) (3件)

- RTコンポーネント再利用賞【提供:[平井成興\(千葉工業大学\)](#)】
- 便利ツール賞【提供:[末廣尚士\(電気通信大学\)](#)】
- ベストサポート賞【提供:[神徳徹雄\(産総研\)](#)】



奨励賞(団体協賛) (副賞2万円) (9件)

- 世界一軽いRTコンポーネント賞【提供:[株式会社 前川製作所](#)】
- 安川電機賞【提供:[株式会社 安川電機](#)】
- パナソニック賞【提供:[パナソニック 株式会社](#)】
- トヨタ自動車賞【提供:[トヨタ自動車 株式会社](#)】
- アドイン賞【提供:[株式会社 アドイン研究所](#)】
- 富士ソフト賞【提供:[富士ソフト 株式会社](#)】
- NTTデータ変える力を、ともに生み出す賞
【提供:[株式会社 NTTデータ](#)】
- 日本ロボット工業会賞【提供:[社団法人日本ロボット工業会](#)】
- ベストコンセプト賞【提供:[ロボットビジネス推進協議会](#)】

趣旨に賛同いただき、御協力ありがとうございました。



RTC-scilab

菅 佑樹 (株式会社リバスト)

概要:

RTC-scilabは、フリーかつオープンソースな科学技術計算支援ソフトウェア「scilab」と、OpenRTM-aistとの連携を簡単にするためのライブラリです。

特徴:

- ◆ scilab言語を使ったRTコンポーネント開発
- ◆ グラフィカルな開発ツール「xcos」を使ったRTC
- ◆ 動力学シミュレータ「OpenHRP3」との連携

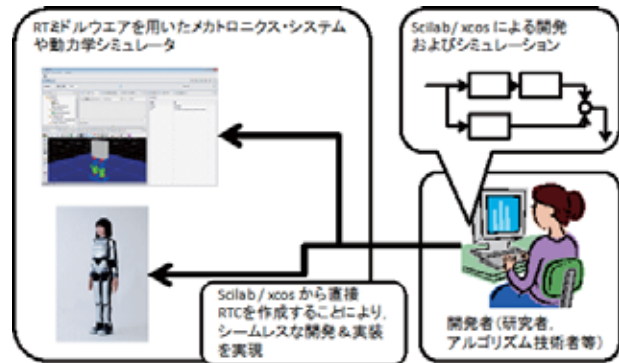
インターフェース:

◆ scilab言語によるプログラミングやxcosによるグラフィカルな開発により、自由に入出力データポートを追加出来ます (TimedLong(Seq), TimedFloat(Seq), TimedDouble(Seq)に対応。)

- ◆ 対応バージョン
 - ◆ OpenRTM-aist C++ 1.0 Release
 - ◆ scilab 5.2以降
 - ◆ OpenHRP3.1 Release
- ◆ 対応OS
 - ◆ Windows, Linux

ライセンス:

- ◆ GPL v3にて、下記URLで配布中
- ◆ http://www.ysuga.net/robot/rtm/rtc_scilab



RTC-scilabを使うことにより、数式処理やデータの視覚化、グラフィカルな制御システム開発が可能であり、またOpenHRP3と連携することにより、ロボットのダイナミカルな制御システムのプロトタイピングが可能です。RTC-scilabはロボットの制御システム開発を効率化します。

連絡先:

ysuga.net

URL: <http://www.ysuga.net/robot/rtm>

E-mail: ysuga [at] ysuga.net



RTC-scilabの機能



• scilab言語を使ったRTC開発

- 行列計算や数値解析、データの視覚化など、scilabの様々な技術計算ツールを使うことで、RTC開発効率を高めます。

```
001 sciRTC=ScilabComp_create();
002
003 sciRTC = sciRTC.registInPort(sciRTC, "TimedLong", "in");
004 sciRTC = sciRTC.registOutPort(sciRTC, "TimedLong", "out");
005
```

Scilab言語は数式処理に適した言語であり、世界中の多くの開発者が数式処理のためのライブラリを無償で提供しています。

• グラフィカルなツール「xcos」を使ったRTC開発

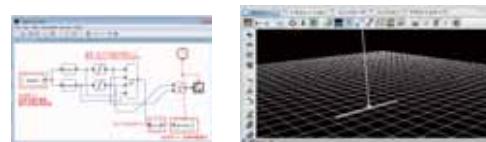
- ブロック線図を描くように開発が出来るので、より直感的なRTC開発が可能になります。



Xcosを使えば、パレットから制御ブロックをドラッグ&ドロップで追加して、制御アルゴリズムとデータポートを追加出来ます。必要な設定は専用の設定ダイアログに記入するだけです。

• OpenHRP3との連携

- 動力学シミュレータのOpenHRP3と連携することで、ロボット用コントローラ開発を支援します。



OpenHRP3との連携が可能です。双方のシミュレーション時間を同期できるので、ロボットのダイナミカルな制御システムのプロトタイピングが出来ます。



ロボットモデル作成ツール

宮本 信彦(立命館大学)



概要:

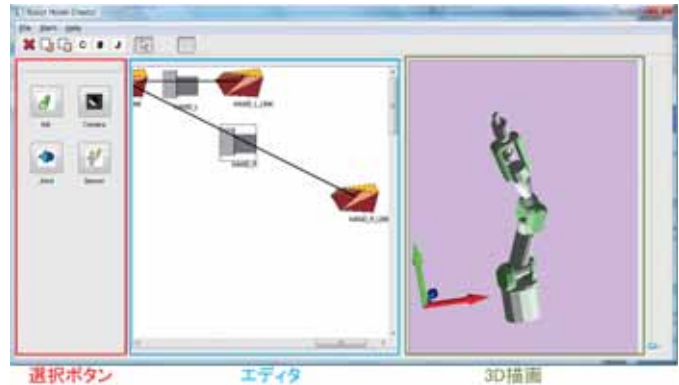
多関節ロボットの構造を簡単に設計するためのツールであり、他のRTコンポーネントからの入力によりロボットの動作確認等ができる。

特徴:

- ◆エディタに図形を並べるだけ
- ◆他のRTCからの入力により関節角度、並進距離を操作可能
- ◆センサ情報を表示
- ◆作成したモデルはOpenHRP3で使用可能

インタフェース:

Inport: Angle(TimedFloatSeq)
 ロボットの関節角度[rad], 並進距離[m]
 Force(TimedFloatSeq)
 カセンサの情報
 Gyro(TimedFloatSeq)
 3軸角速度センサの情報
 Range(TimedFloatSeq)
 距離センサの情報
 Acceleration(TimedFloatSeq)
 3軸加速度センサの情報
 Torque(TimedFloatSeq)
 トルクセンサの情報

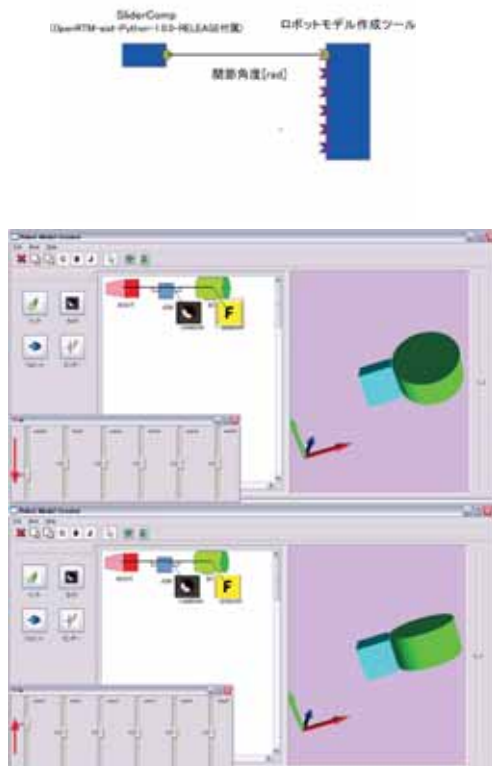


連絡先:

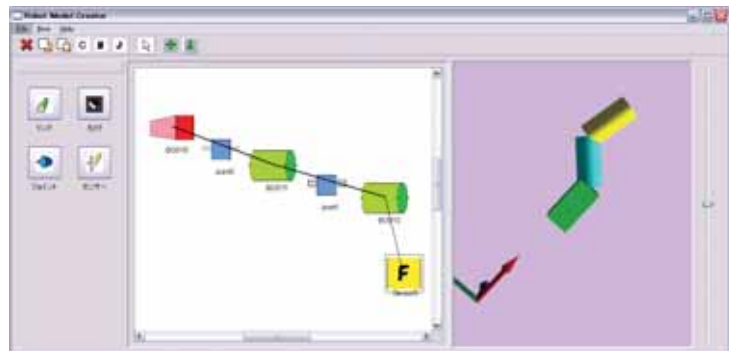
〒525-8577 滋賀県草津市野路東1丁目1-1-1
 立命館大学 イーストウイング5階 生物知能機械学研究室
 E-mail rr008072@ed.ritsumei.ac.jp
 URL: <http://www.malab.se.ritsumei.ac.jp/>



SliderCompによる操作例



手先にカセンサがある2自由度マニピュレータの作成例



開発環境:

OS: Ubuntu-10.04
 Windows Vista, XP
 RTミドルウェア: OpenRTM-aist-1.0.0-RELEASE
 コンパイラ: gcc4.4.3
 Visual C++ 2008 Express Edition
 CORBA: omniORB4.1.4
 Qt: Qt-4.7.1

ライセンス:

LGPL



車輪型移動ロボットのための総合開発環境(群)

高橋 亮(立命館大学)

AIS Lab.
Applied Interaction System Laboratory

概要:

車輪型移動ロボットの開発時に必要な総合開発環境のRTコンポーネント群。移動ロボットを初めて学習・開発を行う人を対象とし、シミュレータにより機器を導入する前から学習を始めることができ、人のコミュニケーションロボットにおける移動制御モデルの開発を行うことができる。

特徴:

- ◆人と移動ロボットがコミュニケーションを行う際の立ち位置評価シミュレーション
- ◆仮想ばねを用いた人物追従とポテンシャル法による障害物回避の同時移動制御機能

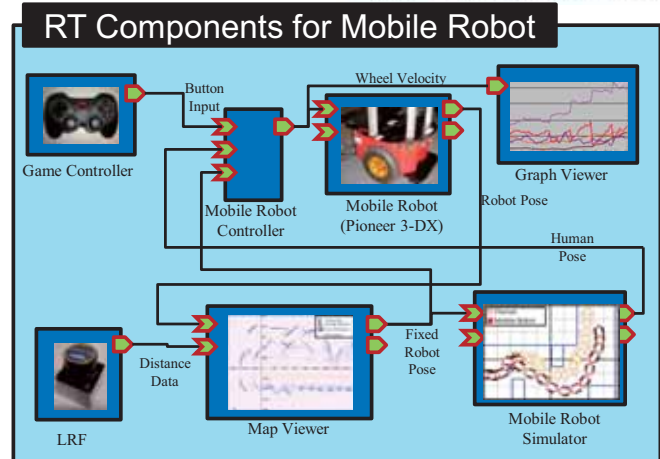
インタフェース:

右図のようにコンポーネント間で必要なデータの入出力を行う。具体的には、制御車輪速度、移動ロボットの推定位置・LRFデータ・人物位置・ポテンシャル法における制御ベクトルなど。

(OpenRTM-aist-1.0.0-RC1)

ライセンス(公開条件):

MOBILE ROBOTIS社のAriaライブラリを除き、非商用利用であれば自由に利用可能。



連絡先:

〒525-8857 滋賀県草津市野路東1-1-1 立命館大学
クリエイションコア7F アプライドインタラクションシステム(李)研究室
E-mail:takahashi@ais.ics.ritsumeai.ac.jp

URL:

<http://www.ais.ics.ritsumeai.ac.jp>

RT
middleware

開発したコンポーネント群

①移動制御(MobileRobotController)

対向二輪型ロボットの指定座標への走行・人物追従走行・障害物回避走行などの移動制御

②移動ロボットシミュレータ(MobileRobotSimulator)

障害物や人物を配置可能な環境・LRFのシミュレーションや、ロボットの立ち位置評価・ポテンシャル法の制御ベクトル表示

③ゲームコントローラ (GameController)

ゲームコントローラのキー入力の認識

④地図構築・表示 (MapBuilder)

移動ロボットに搭載したLRF・Sonarによる距離データとロボットの自己位置推定を基に環境地図の構築・表示

⑤移動ロボットPioneer 3-DX制御 (Pioneer 3-DX)

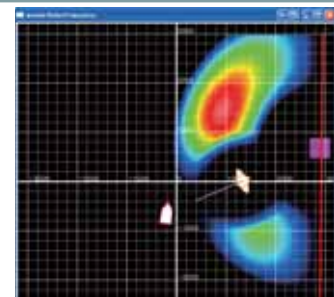
MobileRobots社のPioneer 3-DXの制御

⑥グラフ表示 (GraphViwer)

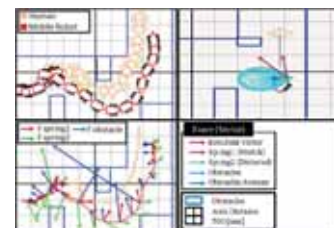
センサデータなどの数値データをリアルタイムに出力

⑦仮想対向二輪型移動ロボット制御 (MobileRobot)

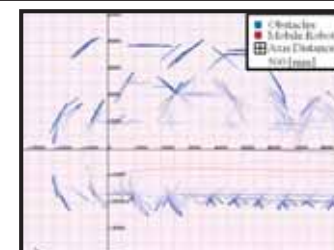
仮想的な対向二輪型移動ロボット制御



コミュニケーションロボットの立ち位置評価



人物追従+障害物回避(仮想ばね+ポテンシャル法)



LRF・ロボットの自己位置推定による地図構築・表示

RT
middleware

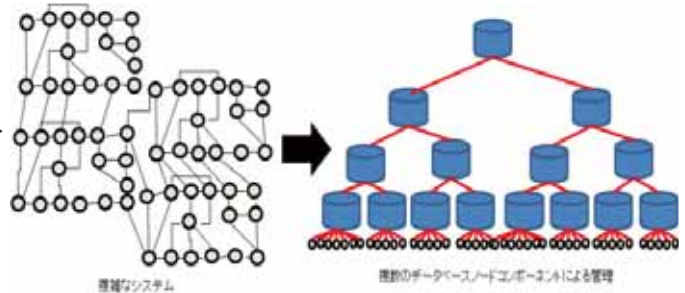
可変構造階層型ソフトウェアアーキテクチャ

小山順平(中央大学)

◆概要:

複数のデータベースでコンポーネントを管理する、階層型ソフトウェアアーキテクチャで 사용되는データベースノードコンポーネント(DNC)を構築。

DNCによる一括管理でもタスクシーケンスを簡単に変更できる



◆特徴:

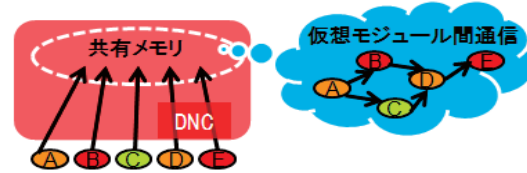
- ◆高速通信の共有メモリを介したデータ共有
- ◆インターフェイスの共通化
- ◆モジュールの連携を構築でき、タスクシーケンスを設計することができる。

◆開発環境

- ◆OS: Ubuntu8.10
- ◆Qt3
- ◆OpenGL
- ◆OpenRTM-aist-1.0.0-RELEASE

◆ライセンス（公開条件）:

- ◆Qtの公開条件及びRTミドルウェアの公開条件に準ずる



仮想モジュール間通信によるタスクシーケンスの構築

連絡先:

中央大学 国井研究室 URL: <http://www.elect.chuou.ac.jp/kunii/index.html>

データベースノードコンポーネントによる一括管理

大規模なシステムになるにつれて、モジュール間の構成も複雑になります。それを引き起こしている原因はモジュール同士のデータの受け渡しです。よって、モジュール間のデータ受け渡しの問題を解消するために、モジュールを一つのデータベースノードコンポーネントによる管理を提案します。

可変構造ソフトウェアノードコンポーネントを開発

ノードコンポーネントの一括管理における問題点として、配置されているモジュールをどのように連携させて一つの仕事(タスクシーケンス)を構築するかということが挙げられます。

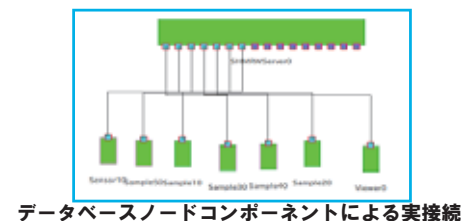
本コンポーネント及びGUIツールを用いて各モジュール間を仮想的につなぎ換えたりすることで、モジュールの連携を作り出すことができます。そのため、ロボットのタスクシーケンスの構築や切り替え(システム変更)を簡単にを行うことができます。



簡易制御
ツール

タスクシーケンス設計

タスク選択



イメージ図





汎用データ処理のための 演算コンポーネント

佐々木 毅(東京大学)

知的制御システム 橋本研究室

Intelligent Control System Laboratory – Hashimoto Lab.

<http://dfs.iis.u-tokyo.ac.jp/>

概要:

様々な入力データに対し、必要に応じた処理を行うことが可能な汎用演算コンポーネント

特徴:

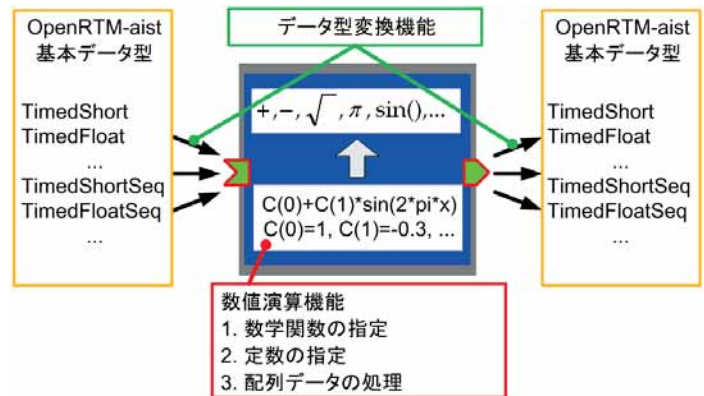
- ◆ 数学関数による処理内容の記述が可能
- ◆ OpenRTM-aistの基本データ型間のデータ型変換が可能

インタフェース:

- ◆ 入力: 処理を行う数値もしくは数値列データ
- ◆ 出力: 処理後の数値または数値列データ
- ◆ データ型は各々下記のOpenRTM-aistの基本データ型の1つから選択可能
 - ◆ Timed{Octet, Short, UShort, Long, ULong, Float, Double, OctetSeq, ShortSeq, UShortSeq, LongSeq, ULongSeq, FloatSeq, DoubleSeq}
- ◆ RTミドルウェアのバージョン:
OpenRTM-aist-1.0.0-RELEASE

ライセンス(公開条件):

著作権は作者に帰属しますが、非商用利用であれば自由にご利用ください



連絡先:

〒153-8505 東京都目黒区駒場4-6-1 東京大学生産技術研究所
Ew201 橋本研究室

佐々木毅

email: sasaki-t <at> ieee. org

URL: <http://dfs.iis.u-tokyo.ac.jp/~sasaki/>

<http://openrtm.sakura.ne.jp/cgi-bin/wiki/wiki.cgi/2010>



演算コンポーネントの用途

RTコンポーネントを用いたロボットシステムにおける様々なデータ処理に利用可能

- 単位換算
- 座標変換
- データ型変換
- など

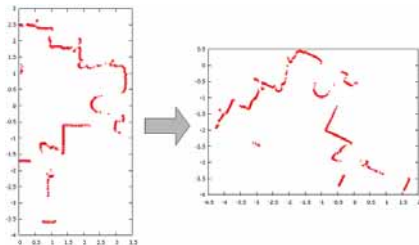
開発環境

LinuxおよびWindowsにて動作確認

- OS: Ubuntu Linux 9.04 / Windows XP Service Pack 3
- RTミドルウェア: OpenRTM-aist-1.0.0-RELEASE (C++版)
- コンパイラ: gcc 4.3.3 / Microsoft Visual C++ 2008 Express Edition
- CORBA: omniORB 4.1.2-1 / 4.1.4
- Eclipse: Eclipse 3.4.2 + OpenRTM Eclipse tools 1.0-RELEASE
- Java実行環境: Sun Java Version 6 Update 22

演算コンポーネントの利用例

– 座標変換 –



回転・並進変換

– 指令生成 –



カメラ画像上でクリックされた位置が画像の中心となるようにパン-チルトカメラへの指令角を生成

– 画像処理 –



原画像

上下左右反転

RGB値の割合の変更

グレースケール画像

二値画像



シミュレーションと実機制御のシームレス化

清水 昌幸(静岡大)

音田 弘(産総研)



概要:

OpenHRP3で開発したロボット制御コンポーネントを用いて、実ロボットマニピュレータPA10を制御するためのRTコンポーネント。シミュレーションと実機制御との間の移行を完全シームレス化することが可能。

特徴:

- ◆ ポートの繋ぎ替えだけでシミュレーションと実機制御を簡単に切り替え可能。再コンパイル不要。
- ◆ OpenHRP3用の制御RTCをそのまま再利用可能。
- ◆ コンポーネント間の完全同期制御を実現。

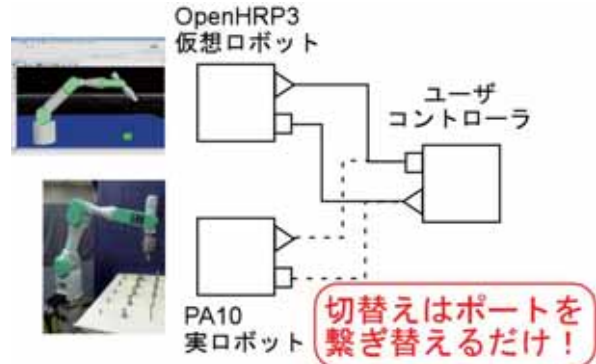
インタフェース(または、使用環境など):

入力ポート(排他利用): 関節速度制御指令値
関節トルク制御指令値

出力ポート: 関節角度 / 関節速度 / 関節トルク
(開発プラットフォーム: OpenRTM-aist-1.0.0)

ライセンス(公開条件):

著作権は開発者に帰属します。大学・公的研究機関における教育・研究目的には無償で利用を許可します。商用利用についてはお問い合わせ下さい。



連絡先:

国立大学法人 静岡大学

工学部 機械工学科

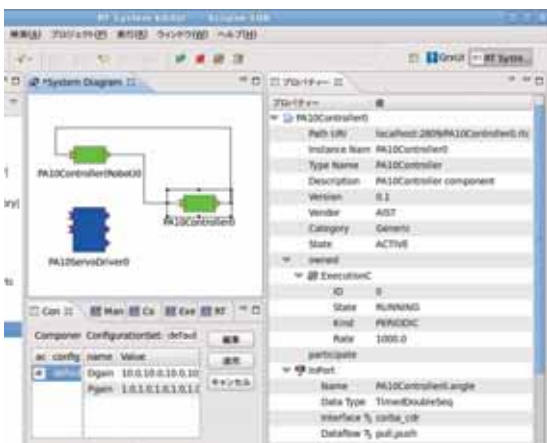
清水 昌幸

〒432-8561 静岡県浜松市中区城北3-5-1

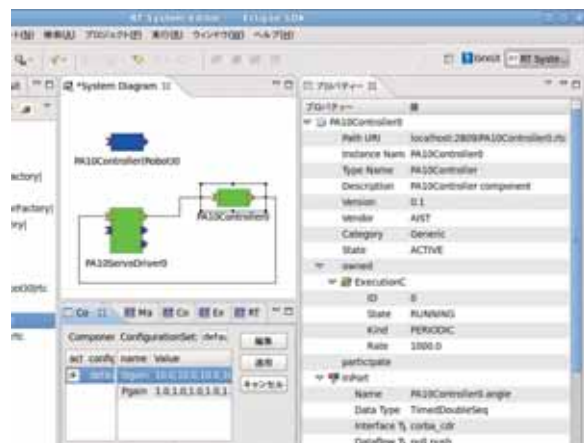
Emai: tmsimiz <at> ipc.shizuoka.ac.jp



- シミュレーションモード時のコンポーネント接続



- 実機制御モード時のコンポーネント接続



- モード切替はポートの繋ぎ替えのみで簡単に実現。専門知識は不要。
- (参考)モード切替に要する時間: 30秒以下
- ART LinuxやPreemptive Kernelを用いることにより、実時間制御が可能。

* 実現できる最短制御周期は環境に依存します。



移動ロボットのネットワーク化



桑原 潤一郎 竹村 憲太郎 末永 剛 高松 淳 小笠原 司(奈良先端大)

■概要:

「ロボットの状態の配信」、「制御命令の受信」、「車輪エンコーダの値の送信」を行うソフトウェアと、それと接続することでUDPでブロードキャストされている環境中のロボットの状況を把握し、接続するロボットを選択、制御することが可能なネットワークコントローラコンポーネント

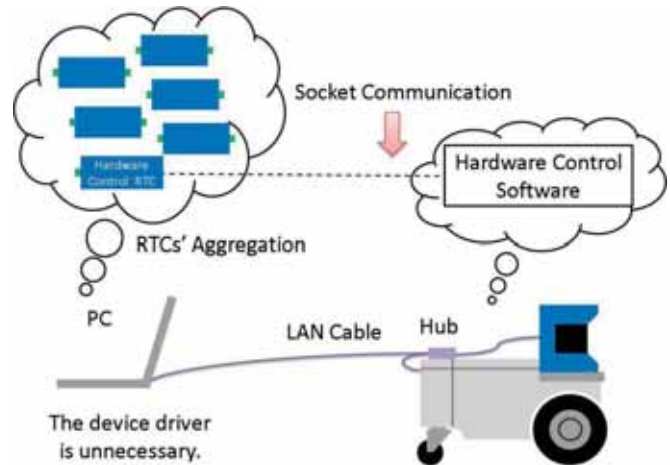
■特徴:

- ・コンポーネントを起動する場所の制限解除
- ・コンポーネントの集中管理の簡易化
- ・ロボットのドライバ開発の簡易化

■インターフェース

・制御用RTC

- InPort: TargetVelocity(TimedVelocity2D)
ロボットに送信する目標速度
- OutPort: VelocityData(TimedVelocity2D)
ロボットから受信した速度
- OutPort: OdometryData(TimedPose2D)
ロボットから受信したオドメトリデータ



連絡先:

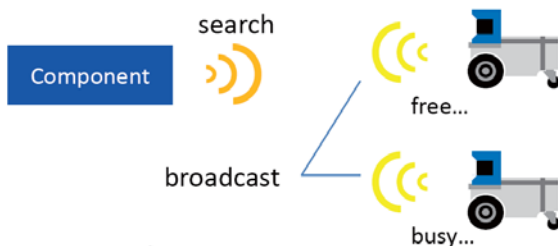
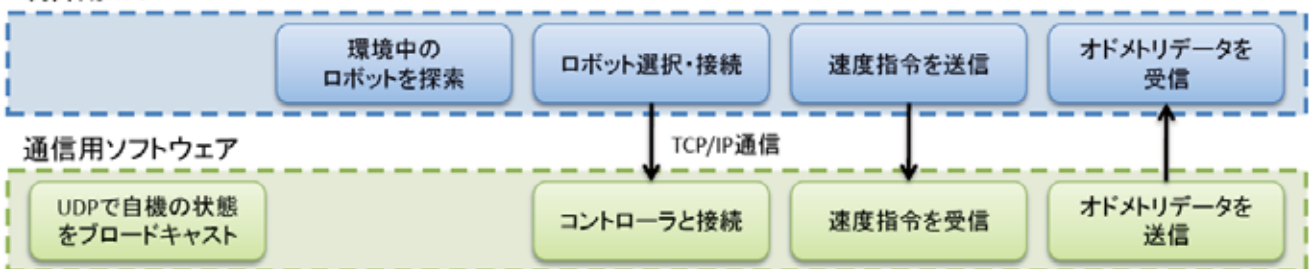
連絡先 〒630-0101

奈良県生駒市高山町8916-5 奈良先端科学技術大学院大学
情報科学研究科情報システム学専攻ロボティクス講座

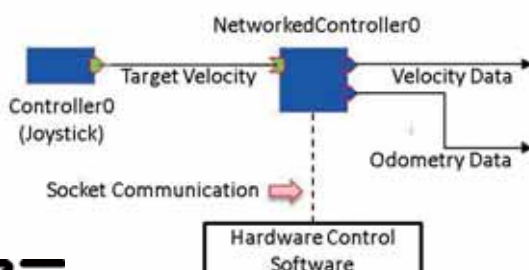
URL: <http://openrtm.sakura.ne.jp/cgi-bin/wiki/wiki.cgi/2010/2B23?page=FrontPage>



制御用RTC



■RTC配線の例



■ライセンス(公開条件)

BSD License

■説明

・通信用ソフトウェア

- UDPで自機の状態をブロードキャスト
- RTCからのTCP/IP接続を受け付け情報をやりとり

■開発環境:

- ・OS: UbuntuLinux 10.04
- ・RTミドルウェア: OpenRTM-aist-1.0.0-RELEASE
- ・コンパイラ: gcc4.4.3
- ・CORBA: omniORB4.1.2-1
- ・Eclipse: Eclipse 3.4.2
- ・Java 実行環境: Sun Java 6-22-0



自己拡張するRTコンポーネント

松坂要佐(産業技術総合研究所)



概要:

OpenRTM-aistにおいて定義されるあらゆるデータ型に接続することができるコンポーネントを実現するための基底クラスをオープンソースで公開します。

特徴:

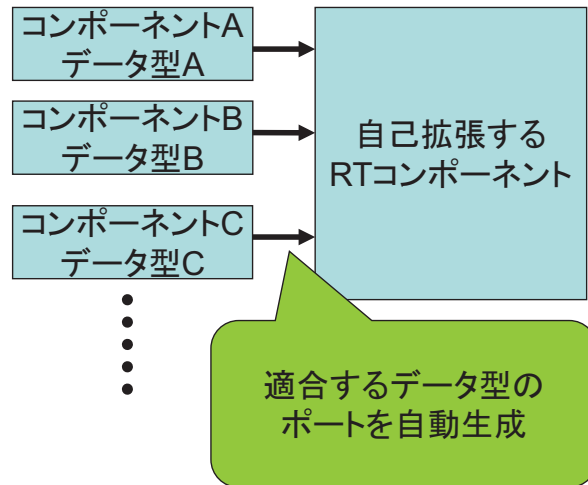
- 以下の機能を持ったコンポーネントを簡単に実現することができます。
- ◆柔軟に自己の形状を拡張することで、どんな数の接続でも受け入れることができる
 - ◆ポート型を相手に合わせることで、どんなデータ型でも受け入れることができる
 - ◆コンポーネントは自動で変形するので、複雑な設定は不要

インタフェース:

OpenRTM-aist-1.0.0がサポートするあらゆるデータ型に対応します。

ライセンス(公開条件):

基底クラスを使った人工知能コンポーネントについてはEPLライセンスで公開します。基底クラス本体についてはより自由に改変し利用していただけるライセンスにできるよう検討中です。



連絡先:

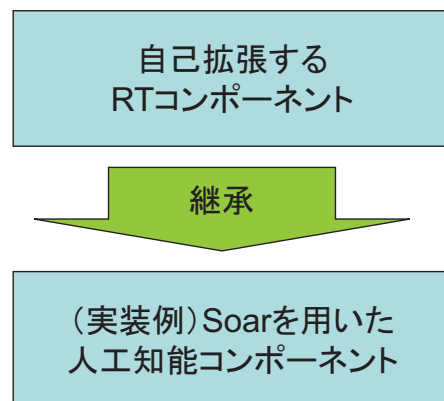
独立行政法人産業技術総合研究所
 知能システム研究部門 インタラクションモデリング研究グループ

〒305-8568 茨城県つくば市梅園1-1-1 中央第2
 email: yosuke.matsusaka@aist.go.jp

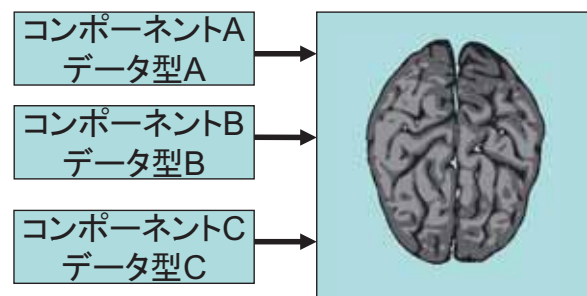
URL: <http://www.openhri.net/>



今回公開する基底クラスを用いることであらゆるデータ型の入力をいくらでも接続することが出来るコンポーネントを簡単に作成することができます。



基底クラスを用いた実装例としてSoarを用いた人工知能コンポーネントを公開します。入力の数によらない再利用可能なロジックの作り方についても合わせて議論します。



RTコンポーネントはどのように 作ればよいか？

安藤慶昭(産総研), ○栗原真二(産総研),
ビグズジェフ(産総研), 神徳徹雄(産総研)

目次

- コンポーネント開発
 - コールバックを実装する際のルール
 - データポートを利用する際に注意すべき点
 - サービスポートとそのインターフェースを実装する際の注意
 - コンフィギュレーション機能を利用する際に注意すべき点
- システム開発
 - システム全体をモジュール分割する際の粒度は？
 - モジュール間の依存性をどのように整理する？
- RTC作成に関する心得
- まとめ

コンポーネント開発(1)

コールバックを実装する際のルール

- 初期化・終了に関するルール
 - 動的リソースの確保をコールバックで行う場合は、確保をonInitialize, 解放をonFinalize をできるだけ対にして行う。
 - デバイスのオープン, その他リソースの確保は可能な限りonInitialize で処理する。
- 実行に関するルール
 - 通常のコンポーネント(Dataow 型) では, 主たるロジックはonExecute に実装する。データポートのコールバック等に主ロジックを実装することは推奨されない。
 - コンポーネントアクティビティ型(Component's activity type) は通常複合化が容易なPERIODIC型が推奨される。その際onExecute の実行は十分短い時間かつ一定時間以内に終了するロジックとして実装する。
onExecute 内では入力待ちなど, 完全に停止するようなロジックを実装することは可能な限り避ける!
- エラーに関するルール
 - onExecute 中で回復可能なエラーは, エラー状態にはせずできるだけエラーが出た時点で回復するようなロジックにする。
 - onExecute に対してはonError, onActivated/onDeactivated に対してはonAbortingを対応させて実装する。

コンポーネント開発(2)

データポートを利用する際に注意すべき点

- データポートにはできるだけ既存の型を利用する。
OpenRTM-aist-1.0 以降は, ExtendedDataTypes.idl, InterfaceDataTypes.idl 内で定義される型を利用することが推奨される
- InPort からは常にデータが取得できるとは限らない前提でロジックを実装する。(InPortのisNew()を使うなど。)
- onWriteやonReadなど, データポートのコールバックは, 補助的に利用し主たるロジックを実行しない。
- 一つのポートからは一つの意味をもったデータのみ入出力するようにする。コンフィギュレーションなどでデータポートの意味を変えることは推奨されない。

コンポーネント開発(3)

サービスポートとそのインターフェースを実装する際の注意点

- サービスポートのインターフェースにはできる限り既存のものを利用する。
OpenRTM-aist-1.0 以降は、ExtendedDataTypes.idl, InterfaceDataTypes.idl 内で定義される型を利用することが推奨される
- コンシューマを利用する場合は常に、プロバイダが対応付けられているとは限らない前提でロジックを記述し、利用する個所には必ず例外処理(C++ではtry-catch 節)を設ける。
- コンシューマからプロバイダを利用する場合、オペレーション呼び出しの実行時間が長くなる場合または不確定な場合、可能な限り非同期呼び出し(coil::Async を利用する等)を行う。
- サービスポートのコンシューマ、プロバイダはCORBA のオペレーション呼び出し規則に従い、C++等ではvar 型を用いるなどしてメモリリークが起こらないよう注意し実装する。

コンポーネント開発(4)

コンフィギュレーション機能を利用する際に注意すべき点

- コンポーネント内の変更される可能性のあるパラメータは、できるだけコンフィギュレーションパラメータにする。
- 型がある言語の場合、パラメータには適切な型を選択する。連続値にはdouble, 数・個数・順序を表現する場合はint, 配列・行列には配列コンテナ型(vector)等を利用する。
- enum に相当する列挙型はstring 型を利用し、string からenum (int) への変換関数を定義したうえで、(C++の場合) bindParameter 関数の第4引数にこれを与える。

システム開発(1)

システム全体をモジュール分割する際の粒度は？

- 既存のシステムがある場合、最初は無理に分割せずに2つのコンポーネントから始める。その後は拡張が必要な部分から徐々にモジュール化していく。
- 細粒度のモジュールは一般に再利用性が高い。しかし、分割したモジュール間に強固な依存性(例えばエンコーダ(とカウンタ)とモータ(とドライバ)は物理的に結合している等)がある場合には分割しても再利用できる可能性は低くなるので注意すべきである。
- システムをモジュール分割していくと、異なるレイヤにまたがる比較的大きなモジュールがどうしても必要になる場合がある。これを無理に分割しようとするのは、その部分はシステムの本質(デザインルール)である可能性がある。

モジュール間の依存関係ができるだけ少なくなるようにモジュールを分割した方がよい

システム開発(2)

モジュール間の依存性をどのように整理する？

- 密結合が必要な場合、データの流れとロジックの実行順序を検討のうえで、複合コンポーネント、実行コンテキストの共有による同期実行が必要ないか検討する。
- リアルタイム実行が必要な部分では、予め各コンポーネントの実行時間と周期を勘案したうえで、適切なリアルタイム実行コンテキストを利用する。
- より複雑な実行方法が必要な場合、実行コンテキストの拡張を検討する。

モジュール間の結合度が強い場合は、適切な実行コンテキストの選定が必要！

RTC作成に関する心得


- コンポーネントを作る前に似たような機能のコンポーネントがないかどうか調べ、可能な限り再利用する。問題があれば作者にフィードバックをする。
- 一から実装せずに既存のライブラリをできるだけ活用する。すでに公開され、使用されているコードは、何度も実行されているはずで、それだけでこれから書こうとするコードより信頼性は高い。
- OpenRTM-aist のクラスリファレンスマニュアルを活用する。サンプルコンポーネント以上のことを行うには、クラスリファレンスを参考にする。
RTObject, InPort, OutPort クラスとそのスーパークラスのマニュアルを読めば何ができて何ができないか分かる。

サンプルコンポーネント以上のことを行うには、クラスリファレンスを参考にする！

まとめ

- RT コンポーネントでロボットシステムを作る、かつ作成されたモジュールをユーザ間で共有し、再利用するといった目的のためには、ユーザ間で共通したルールに従ってコンポーネントを作成する必要がある。
- ここに示したルールは、適用アプリケーション、ミドルウェアの機能や、その他の外的要因により変わることがあり、盲目的に従うのではなく常に検討を繰り返していく必要がある。
- ユーザコミュニティでこうした議論についても継続的に行われることを期待したい。

RTコンポーネント再利用に関する報告と提案



○小島幸也(富士ソフト(株)), 二宮恒樹,
津幡善信, 高橋秀行, 伊藤晴康,
小笠原哲也, 平井成興(千葉工業大学)

Report and proposal concerning reuse of RT components

○Yukiya KOJIMA (FUJISOFT INC.), Tsuneki NINOMIYA,
Yoshinobu TSUBATA, Hideyuki TAKAHASHI, Haruyasu ITOH,
Tetsuya OGASAWARA, and Shigeoki HIRAI(CIT)

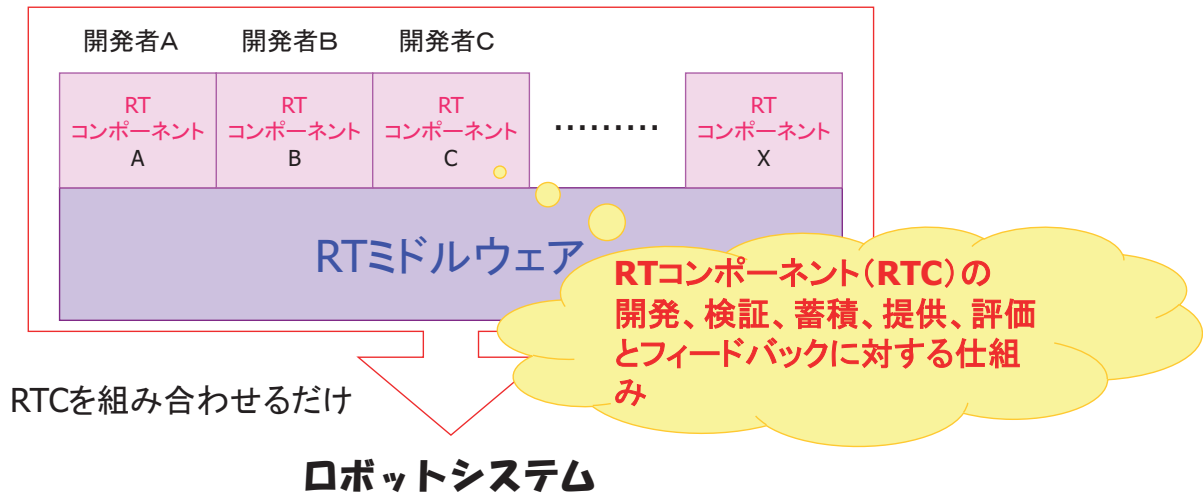


発表の流れ

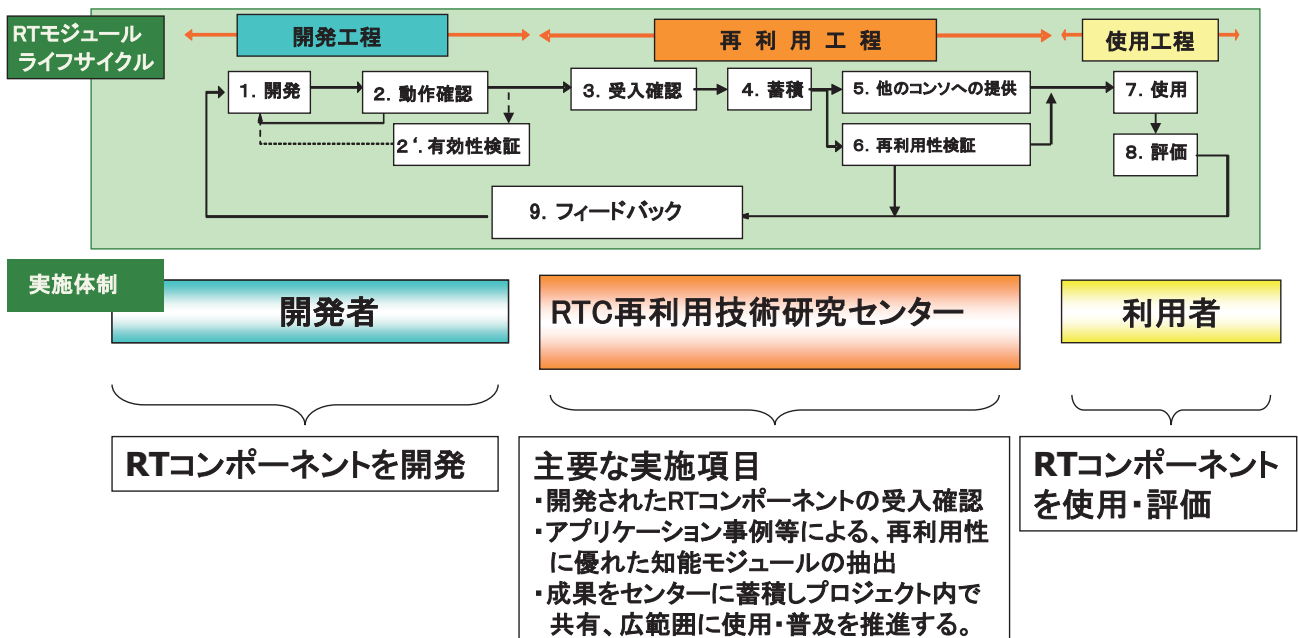
- RTC再利用技術研究センターの活動内容紹介
- RTC再利用における問題点報告
- ドキュメント作成の必要性
- 最低限必要なドキュメント
- まとめ

次世代ロボット智能化技術開発プロジェクト

■ ロボット技術開発の効率化



RTCライフサイクルと開発支援体制の構築



RTコンポーネント再利用 に関する問題点報告



Check Sheet

蓄積モジュールの 再利用性検証実施 ... **できない！！**

- ・ モジュールファイルが不足
- ・ 動作に必要なパッケージ類不足
- ・ 動作環境情報不足
- ・ 操作手順情報不足

ドキュメント不足

ドキュメント作成は当たり前

ドキュメント作成の必要性

開発した成果物を他人に使ってもらうには？

それを理解するためのドキュメントが必要！

- 企業の開発現場では工程毎に細かく作成され利用されている
- 各開発工程に対する成果物と捉えることもできる

ソフトウェア開発工程	成果物
要求分析・仕様決定	仕様書
設計	詳細設計書
実装	ソース/実行オブジェクト
試験	試験項目兼成績書, 取扱説明書
運用	運用手順書

要求分析・仕様決定:仕様書

- 開発目的の明確化
何のために？
- サービス内容の把握
何がしたい？
- 求められる要件の抽出
どんな機能が必要？
どんな見た目？
どう使う？
- 実現可能な仕様の決定



仕様書

仕様内の矛盾・不備の有無を確認

設計:詳細設計書

- システムの構造を設計(システム再構築が可能となるレベルの情報)

- ・システムの構成
 - ハードウェア構成図
 - モジュール構成図
 - モジュール一覧表
- ・動作シーケンス
 - シーケンス図
- ・インターフェース仕様
 - モジュールインターフェース詳細
 - モジュール間接続情報
- ・モジュールの機能仕様
 - 機能概要
 - 関数仕様
 - モジュール構成ファイル



詳細設計書

設計内容の矛盾・不備の有無を確認
システムの理解

実装: ソース/実行オブジェクト

- ソフトウェア開発におけるメインの成果物
- システムをプログラミング言語で記述したものという意味ではソースコードもドキュメント

- ・コード規約
- ・コメント規約

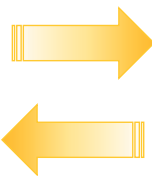
を決めるなど、より読みやすいソースとなるよう工夫をする

試験: 試験項目兼成績書

- プログラムの動作が仕様書や設計書の通りか確認



試験項目兼成績書



試験名	レベル
単体試験	モジュール
統合試験	モジュール構成
システム機能試験	システム仕様
出荷試験	開発依頼者要求



仕様書 詳細設計書

必ず作成する

試験不可能

- 不具合表の作成とフィードバック情報洗い出し
(原因究明と証拠保存) (修正情報やログ参照資料)

リリース: 取扱説明書

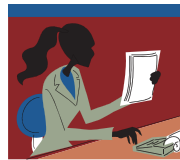
■ 使用方法の取りまとめ

- ・動作環境構築
 - 動作環境詳細
(使用する機器、ファイル、ソフトウェア等)
 - 環境構築手順
- ・操作方法
 - 起動・終了手順
 - 操作例と動作例
 - 操作機能一覧
- ・注意事項



取扱説明書

使うために最低限必要な
情報を載せる



まとめ

- ドキュメントを残すことで、成果物に対する情報を共有できる
- 情報の確認や問題点の洗い出しができ、開発者自身のためになる



開発効率の向上
技術の普及・発展



サンプルドキュメント

- ドキュメント作成促進のためにプロジェクト内サイトである再利用WEB(参考文献[4])にてドキュメントのサンプルを配布していく。

配布仕様書
仕様書
詳細設計書
試験項目兼成績書
取扱説明書



ご清聴ありがとうございました