**・Ros node構造**

**・topicの命名**

**・動作全体像**

**・制御node内部のstatus定義**

**・topic messageの構造と定義**

**・サンプルコード**

**・デモシーンの実現（一部）**

**・役割分担**

・**Ros　node構造**

**制御node**

**機能node**

音声

アーム

Rulo

Yolo

Realsense

Top

　　　　　　　結果topic

指示topic

＃topicのmsgが共通で定義

・**topicの名前**

**結果topic：各nodeにより異なります**

Yolo: results\_YOLO, Rulo: results\_RULO, アーム: results\_ARM, 音声: results\_VOICE

**指示topic**

Command\_TOP

＃各nodeを上記のtopic名前の受信と発行を設定してください

**・動作全体像**

各機能nodeがTopからの指示topicを受信して、実行した結果を結果topicで送信

デモの動作制御（statusの遷移など）は全部制御nodeの中で処理する

各機能nodeについて、デモの動作状態を一切考慮せず制御nodeからの指示topicを待ち

機能nodeは下記の２種類状態しかない

実行：実行→結果topic送信→次の指示topic待ち

待ち：次の指示topic待ち

＃詳細が以降のpageのサンプルコードを参照

・**status 定義**

＃制御nodeしか使わない。機能nodeに関係ない

**Status0: 初期**　　必要な定義やcallback関数の登録とか

**Status1: 待機**

**Status2: 探索**

**Status3: 移動**

**Status4: 微調整**

**Status5: 把持**

**Status6: PR**

＃上記のstatusがTOPの作成により変更する可能性があります

＃機能nodeに影響をしないので、各機能nodeの実装には上記のstatusに依存しない

**・結果topic/指示topicのmsg構造**

**../msg/common.msg**

TopicのMsg構造

|  |
| --- |
| Common.msg |
| Int topic\_id |
| NODE node |
| RESULTS ret |
| MSG msg |

指示topicの場合は、どのnode向けの指示

結果topicの場合は、無意味（TOPがこの値を参照しない）

|  |
| --- |
| NODE |
| ALL=0 //全node |
| Yolo=1 |
| Rulo=2 |
| arm=3 |
| voice=4 |
| invalid=5 |

**Yolo関係**

Id=0 タバコ

Id=1 人

指示topicの場合は、無意味

結果topicの場合は、各nodeの実行結果の状況を設定

＃ERR\_code\_xxはdebug用　各自で必要に合わせて追加すれば

|  |
| --- |
| RESULTS |
| Ok=0 |
| Ng=1 |
| ERR\_code\_1=2 |
| ERR\_code\_2=3 |
| … |
| Invalid=.. |

**rulo関係**

Id=0 ランダム移動

Id=1 目的地に移動

Id=2 poseの微調整

|  |
| --- |
| MSG |
| Float32 x3d, y3d, z3d //三次元位置座標 |
| Int32 x2d, y2d　　　　　//二次元位置座標 |
| Int32 id　　　　　　　 //class id / rulo動作仕方 |
| Int32 key //keyboard command |

指示topicの場合は、機能nodeの動作に必要な情報を設定

＃例えば、ruloに三次元位置を渡す、ruloが三次元位置に移動する

＃例えば、yoloにidを渡す、yoloがidのclassだけを認識します

結果topicの場合は、実行結果の詳細を設定

＃例えば、yoloの場合は認識した三次元や二次元の結果を設定

・**サンプルコード**

#imageを伝えるためのサンプルなので、ビルドが通らないと思う

#同じ機能が実現できれば、別に下記の書き方じゃなくてもOK

**機能node(YOLO)**

void mainCallBack (const common::ConstPtr& msg)

{

if(msg->node == 1 || msg->node == 0) **//TOPから実行指示受ける**

{

YOLOのコア処理　　**//YOLOの認識処理　既にコードある**

　　　common results;　　　**//結果topic msgを定義**

results.topic\_id = msg->topic\_id;

results.ret = OK;　　　**//実行結果の状況を設定　認識できた→OK**

**//認識できなかった→NG**

**//何かエラーがあれば、エラーコードを設定**

results.msg.x3d=…　　**//認識結果を設定**

results.msg.y3d=…

results.msg.z3d=…

publish(results);　　　**//結果topicを発行**

return;　　　　　　　**//次の指示topicを待つ**

}

Else　　　　　　　　　**//他のnode向けの指示なので、yoloが次の指示topicを待つ**

{

return;　　　　　　**//次の指示topicを待つ**

}

}

**機能node(Rulo)**

void mainCallBack (const common::ConstPtr& msg)

{

if(msg->node == 2 || msg->node == 0) **//TOPから実行指示受ける**

{

Switch(msg->msg.id)　**//動き方がmsg.idの値を参照**

case 0:

ランダム移動　　　　**//既にコードある？**

　　common results;　　　**//結果topic msgを定義**

results.topic\_id = msg->topic\_id;

　　results.ret = OK;　　　**//実行結果の状況を設定**

break;

case 1:

目的地への移動　　　**//msg.x3d などにより目的地を設定既にコードある？**

　　　　　common results;　　　**//結果topic msgを定義**

　　results.topic\_id = msg->topic\_id;

results.ret = OK;　　　**//実行結果の状況を設定**

　　　　　break;

case 2:

pose微調整　　　　　**//msg.x2d y2dによりposeを微調整　既にコードある？**

　　　　　common results;　　　**//結果topic msgを定義**

　　results.topic\_id = msg->topic\_id;

results.ret = OK;　　　**//実行結果の状況を設定**

　　　　　break;

　　　publish(results);

return;　　　　　　　**//次の指示topicを待つ**

}

Else　　　　　　　　　**//他のnode向けの指示なので、次の指示topicを待つ**

{

return;　　　　　　**//次の指示topicを待つ**

}

}

**機能node(arm)**

void mainCallBack (const common::ConstPtr& msg)

{

if(msg->node == 3 || msg->node == 0) **//TOPから実行指示受ける**

{

アームのコア処理　　**//把持処理　＃既にコードある？**

　　　common results;　　　**//結果topic msgを定義**

results.topic\_id = msg->topic\_id;

results.ret = OK;　　　**//実行結果の状況を設定　動作完了→OK**

publish(results);　　　**//結果topicを発行**

return;　　　　　　　**//次の指示topicを待つ**

}

Else　　　　　　　　　**//他のnode向けの指示なので、次の指示topicを待つ**

{

return;　　　　　　**//次の指示topicを待つ**

}

}

**制御node(TOP)**

void mainCallBack (const common::ConstPtr& msg)

{

State machine (省略)

　common msg

msg.node = 1; **//yoloを動かせ**

msg.msg.id = 1; **//人を検知して**

pusblish(msg);

}

・**デモシーンの一部**

**TIME**

**動作：タバコを見つけて、近づいて隠す**

Msg.node=1

Msg.msg.id=0

Yolo

Top

Msg.results=ng

「タバコを認識する」指示をyoloに出す

Yoloが認識できなかった結果topicを発行

Msg.node=2

Msg.msg.id=0

Rulo

Top

Msg.results=ok

「ランダム移動」指示をruloに出す

Ruloが移動完了した結果topicを発行

Msg.node=1

Msg.msg.id=0

Top

Yolo

Msg.results=ok

「タバコを認識する」指示をyoloに出す

Yoloが認識できた結果をtopic発行

Msg.node=2

Msg.msg.id=1

Rulo

Top

Msg.results=ok

「目的地に移動」指示をruloに出す

Ruloが移動完了した結果topicを発行

**TIME**

Msg.node=1

Msg.msg.id=0

Yolo

Top

Msg.results=ok

「タバコを認識する」指示をyoloに出す

Yoloが認識できた結果topicを発行

Msg.node=2

Msg.msg.id=2

Rulo

Top

Msg.results=ok

「pose微調整」指示をruloに出す

Ruloが動作完了した結果topicを発行

Msg.node=3

アーム

Top

Msg.results=ok

「把持」指示をアームに出す

アームが同さん完了した結果topicを発行

Msg.node=2

Msg.msg.id=1

Rulo

Top

Msg.results=ok

「目的地に移動」指示をruloに出す

Ruloが移動完了した結果topicを発行

・**役割分担**

TOP（制御node）　 → 居

Yolo　　　　　　　 →　居

アーム　　　　　　　→　長田さん　濱田さん

音声　 　　　　　　→　長田さん　濱田さん

Rulo　　　　　　　　→　長田さん

という感じかな？