matereal

m ə t i ˈ (ə) r i ə l

インタラクティブなロボットアプリケーションのプロトタイピング用ツールキット

http://bit.ly/wiss2010-matereal

加藤 淳, 坂本 大介, 五十嵐 健夫

JST ERAO 五十嵐デザインインターフェースプロジェクト

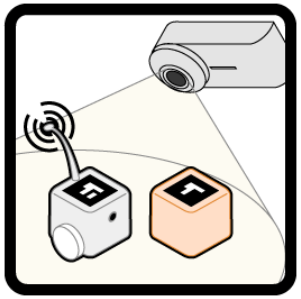
東京大学情報理工学系研究科 五十嵐研究室

matereal とは…

- ユーザの指示に従って実世界のタスクをこなすアプリケーションを楽に開発する（＝real と mate になる）ためのツールキット

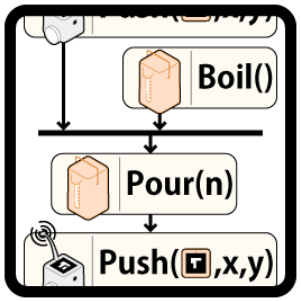
- 対象とするロボット:
  - 遠隔操作できる
  - 床面や机上など、平面上を動き回れる

matereal の機能



天井カメラとマーカーで
位置認識

平面上での**移動・物押しタスク**を指示できるＡＰＩ



指示のシーケンスを組み立てられる**タスク管理**ＡＰＩ

関連研究

ロボットの単体開発 → 環境側センサも含めた開発

プロトタイピング適性・高 → 低

Carmen
Robot Navigation Toolkit

Player

PYRO

LEGO mindstorms education

RT MIDDLEWARE

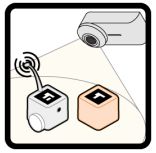
ミドルウェア

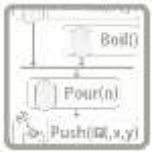
ROS.org

Microsoft Robotics Developer Studio 2008 R3

matereal
m ə t i ˈ (ə) r i ə l

位置認識

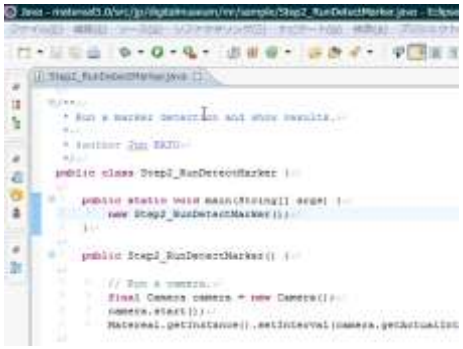




- 天井カメラとARToolKitマーカー
  - ロボットと物体の二次元座標を一挙に取得
  - 安価

- 既存手法
  - ロボットのみ測位（物体認識は別）
    - Simultaneous Localization and Mapping
    - 事前に環境地図を提供
  - 高価
    - モーションキャプチャ

位置認識



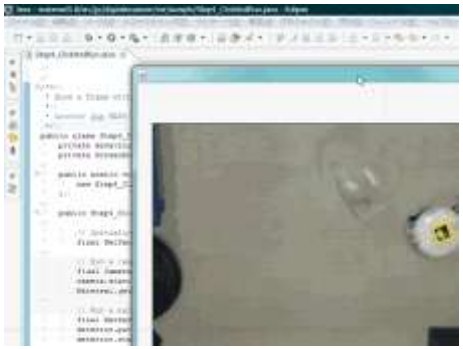
// マーカー検出と座標計算

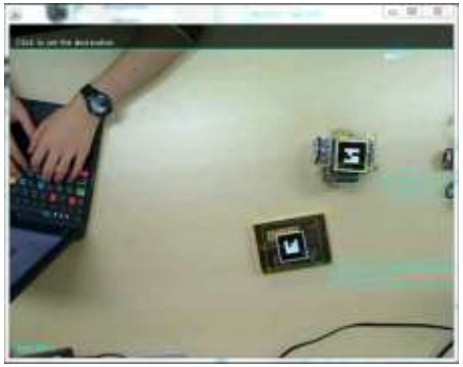
```
MarkerDetector md = new MarkerDetector();
md.put(new NapMarker("hiro.patt"), robot);
md.start();
```

// 位置情報の取得

```
Location location = md.getLocation(robot);
```

移動・物押しタスク

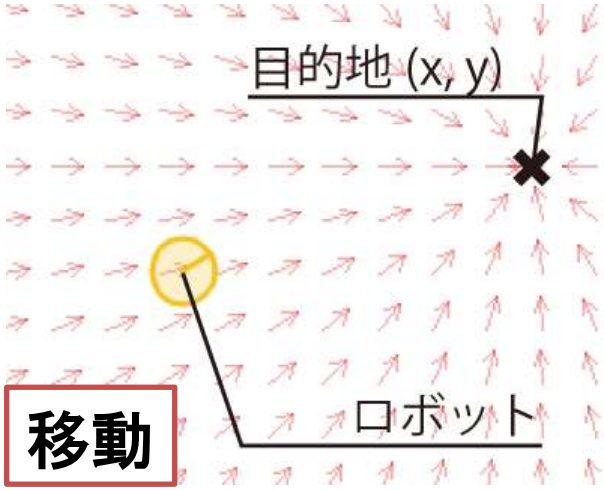


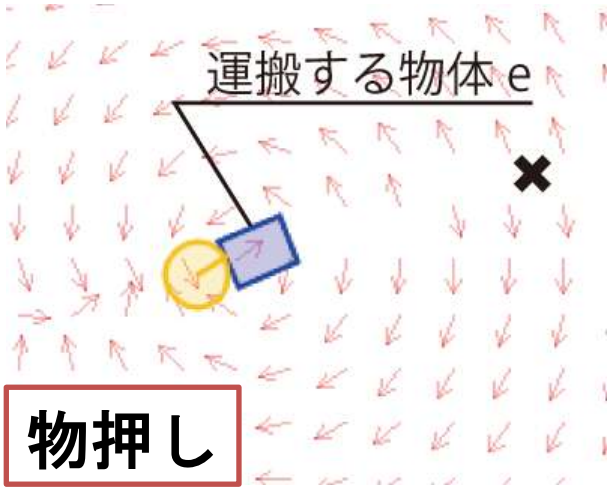


```
// 移動
Move move = new Move(x, y);
if (move.assign(robot)) {
    move.start();
}

// 物押し
Push push = new Push(e, x2, y2);
if (push.assign(robot2)) {
    push.start();
}
```

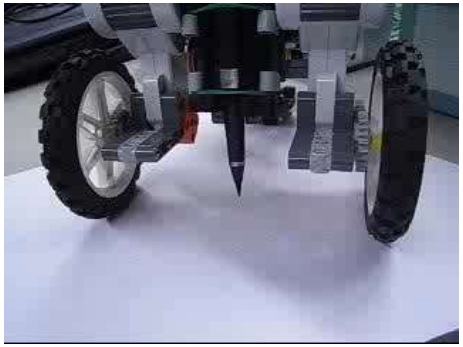
ベクトル場による移動





[Igarashi et al. 2010]

移動以外のタスク



- 筆の上げ下ろし（LEGO Shodo）
- カメラ撮影（ネットタンサー）
- 掃除（ルンバ）
- etc.

タスク管理

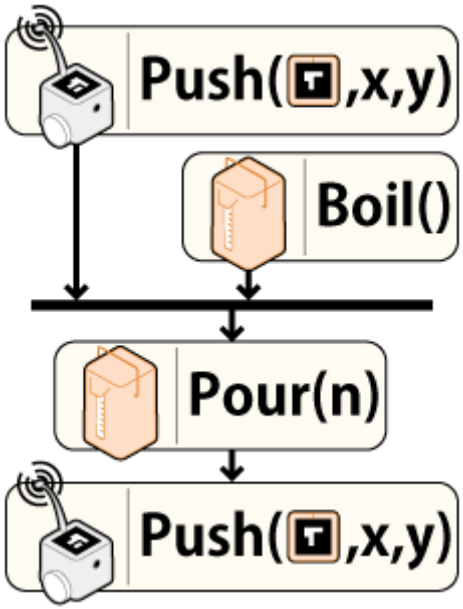
- これまでに挙げた機能だけで、ユーザの指示に従ってロボットに一つのタスクを実行させるプログラムは書ける。

⇕

- エンドユーザの指示に従ってインタラクティブにタスクの動作順序を変えたい。
  - 料理、服たたみ、etc.
- 複数台のロボットに同時に作業させたい。
  - 味噌汁を作りながらお茶を淹れる、etc.

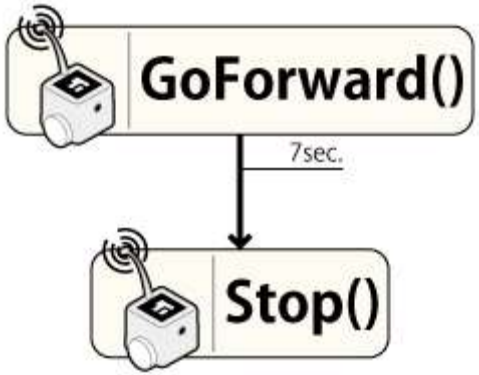
タスク管理

- 動作主とタスクのペアをノードとする図式表現を生成、実行
- フローコントロールも可能



- 既存手法
  - イベントドリブンプログラミング
    - タスクの実行順序は自己管理
  - 並列処理プログラミング
    - デッドロックなど多くの面倒事

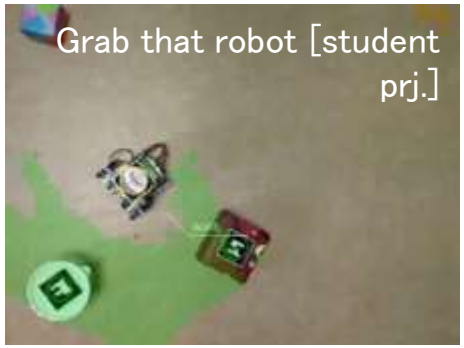
タスク管理



```
// アクティビティ図を作成
ActivityDiagram ad = new ActivityDiagram();
// ノードを追加
Action a = new Action(robot, GoForward());
Action b = new Action(robot, Stop());
ad.add(a);
ad.add(b);
// エッジを追加
ad.addTransition(new TimeoutTransition(a, b, 7000));
// 実行
ad.setInitialNode(a);
ad.start();
```

matereal の活用例

自転車で走るロボ



ピンチジェスチャ
による移動指示UI

スケッチによる
タスク指示UI

マルチタッチ
マルチロボ操作

移動スピーカー

LEGO書道

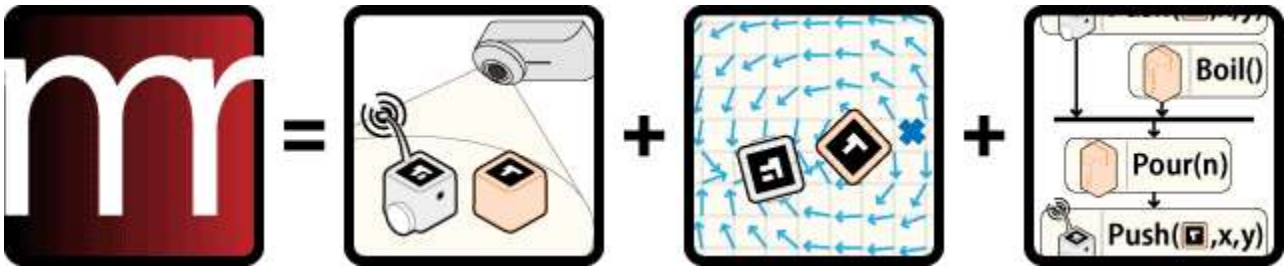
調理手順指示UI

And
it's your turn!

？

まとめ

- ユーザの指示に従って実世界のタスクを実行できるアプリケーションを簡単にプロトタイピングできるツールキットを開発した。



※本プロジェクトは2008年12月から2009年8月まで、情報処理推進機構2008年度下期未踏IT人材育成発掘事業（未踏本体）の支援を受けて開発が進められました。

未来ビジョン

- ツールキットを多くの人に使ってもらって、色んなロボットアプリケーションを作ってほしいです。
- それが、「ロボット」が身近になる第一歩だと思います。

未来ビジョン（裏）

- 実用的なロボットアプリケーションとは:

人の「実世界に関して**こうなってほしい**」という意志を、コンピュータの力を借りて実行するアプリケーション

- 例:
  - 掃除: 部屋は**こう掃除してほしい**
  - 料理: 素材は**こう料理してほしい**
  - 洗濯物畳み: 洗濯物は**こう畳んでほしい**

思考実験

- Amazonに商品を発注した。数日後、商品がポストに入っていた。
- 誰がポストに商品を入れた？

思考実験の結果？

- Amazonに商品を発注した。数日後、商品がポストに入っていた。
- 誰がポストに商品を入れた？

- 「佐川の人が届けてくれたに違いない」

思考実験の結果

- Amazonの入力インタフェースはHuman-Computer Interactionのみで、実装レイヤーはユーザから隠されている。
- 常識の枠を取り払えばAmazonはロボットアプリケーションと見分けがつかない。

- ~~「佐川の人が届けてくれたに違いない」~~
- 「ロボットが配達したのかもしれない」

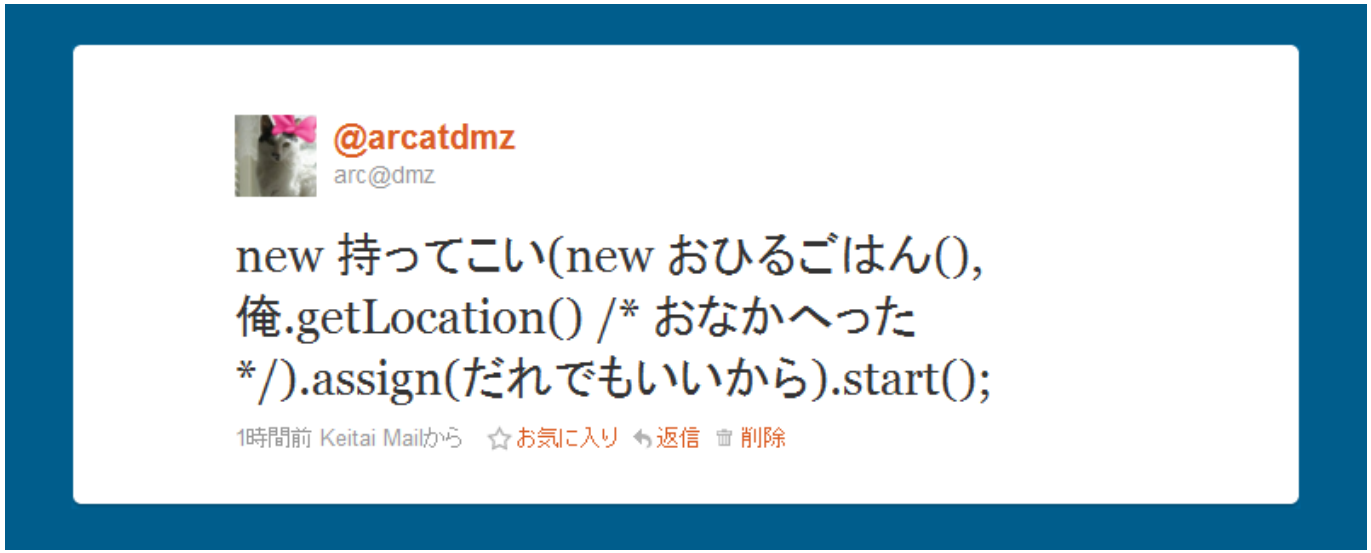
逆転の発想

- ロボットとは:

自分の意志を代わりに実行してくれる実装

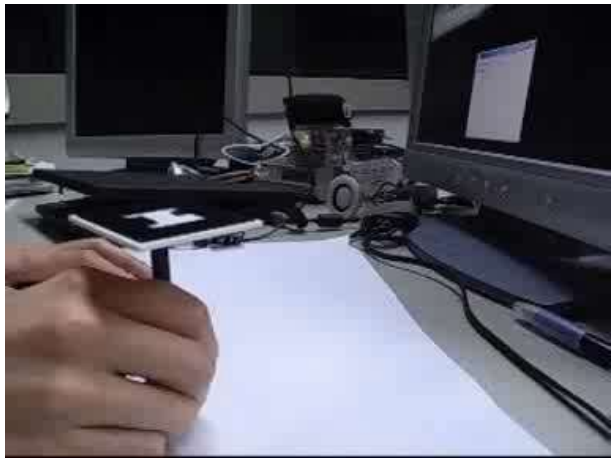
⇩

- 人間も実装の一種と考えることができる。



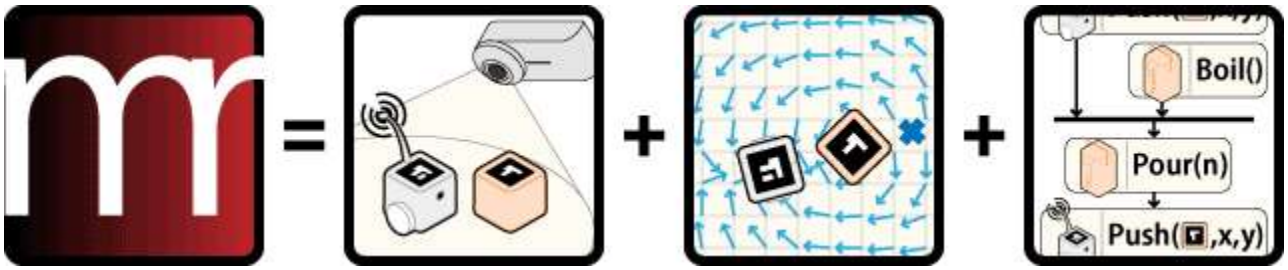
未来ビジョン

- matereal**は人とロボットを区別しない。**
  - **人とロボットを統一的に扱うツールキットへ。**
- **応用例**:
  - Human-Robot Collaboration
    - **調理の加減を判断できる人間クラス**
  - Human-Controlling Interface
    - ~~LEGO書道~~**俺書道**
    - **校庭に大きな絵を描く**
      - **耳を引っ張る歩行誘導デバイス** [Yuichiro et al. 2009]**との連携？**



まとめ（再掲）

- ユーザの指示に従って実世界のタスクを実行できるアプリケーションを簡単にプロトタイピングできるツールキットを開発した。



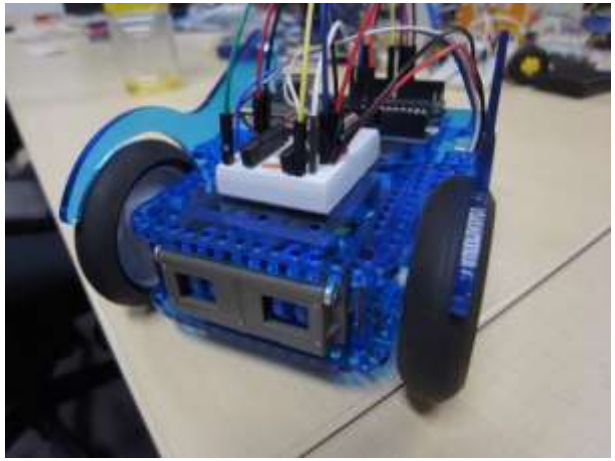
※本プロジェクトは2008年12月から2009年8月まで、情報処理推進機構2008年度下期未踏IT人材育成発掘事業（未踏本体）の支援を受けて開発が進められました。

（以降、予備スライド）

これからの話

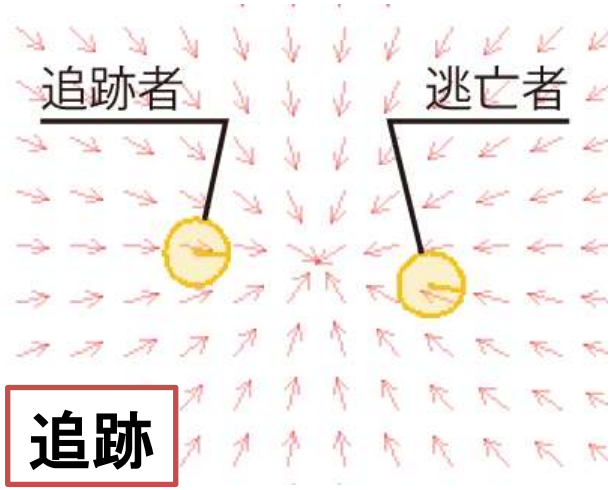
- QPToolkitとの連携

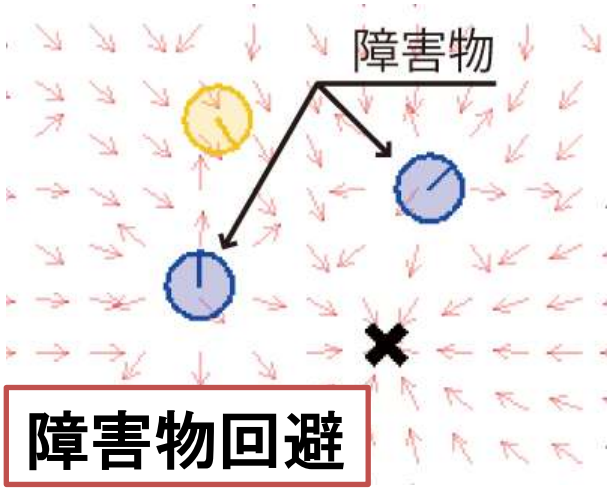
- ハードウェアの作り方公開



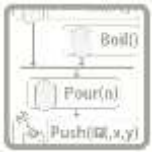
- ワークショップ？（開催地募集中！）

ベクトル場による移動





ベクトル場による移動



// ベクトル場を定義

```
VectorFieldTask vft = new VectorFieldTask() {
    public void getUniqueVectorOut(Position p, Vector2D v) {
        // 位置pにおけるベクトル値vをセットする
        v.set(1.0, 0);
        v.mul(getMinimalNorm());
    }
};
```

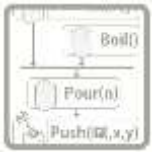
// ベクトル場を重ね合わせ

```
vft.add(new CollisionAvoidance(robot));
if (vft.assign(robot)) vft.start();
```

matereal の欠点

- マーカーが物体の陰に隠れると位置が取れない。
  - 対応策:
    - 複数カメラ・斜め視点
    - モーションキャプチャ
    - QPToolkit?

matereal の欠点



- 複雑なパスプランニングの支援機能がない？
  - 対応策:
    - 自前で実装
    - さしあたり必要なかったので…

matereal の欠点

- リアルタイムのHuman-Robot Interactionを支援する機能がない？
  - 対応策:
    - アクティビティ図を用いてシナリオを記述
    - イベントドリブンなプログラム

    - 依然、一緒に物を持って運ぶなど、リアルタイム性が重要な共同作業は設計しづらい。
    - オープンクエスチョン。Future work…