

matereal

m ə t i ' (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

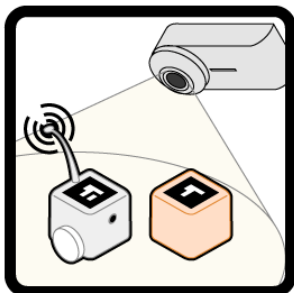
materealとは...

- 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



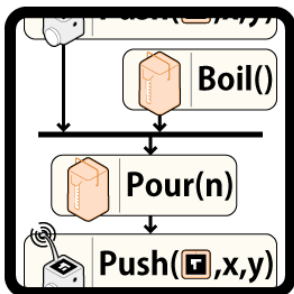
materealの機能



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

低

プロトタイプ
適性・高



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();
```

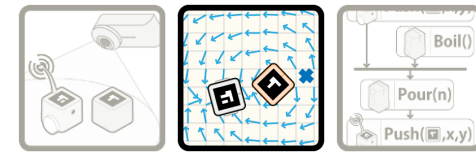
// 位置情報の取得

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

Screenshot of an Eclipse IDE showing Java code for Step2_RunDetectMarker.java. The code includes a main method that creates a new Step2_RunDetectMarker object and a constructor that initializes a camera and starts it.

```
Java - matereal5.0/src/jp/digitalmuseum/mr/sample/Step2_RunDetectMarker.java - Eclipse  
ファイル(F) 編集(E) ソース(S) リファクタリング(R) ナビゲート(N) 検索(A) プロジェクト  
Step2_RunDetectMarker.java  
/**  
 * Run a marker detection and show results.  
 *  
 * @author Jun KATO  
 */  
public class Step2_RunDetectMarker {  
    public static void main(String[] args) {  
        new Step2_RunDetectMarker();  
    }  
    public Step2_RunDetectMarker() {  
        // Run a camera.  
        final Camera camera = new Camera();  
        camera.start();  
        Matereal.getInstance().setInterval(camera.getActualInt
```

移動・物押しタスク

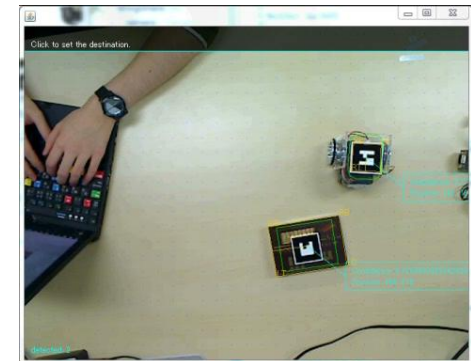
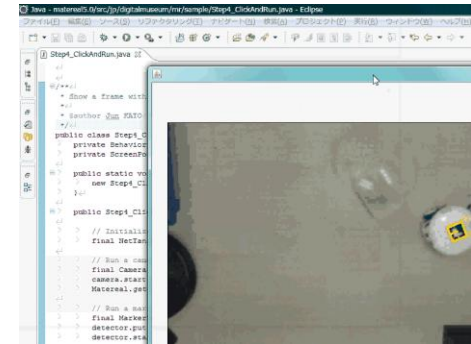


// 移動

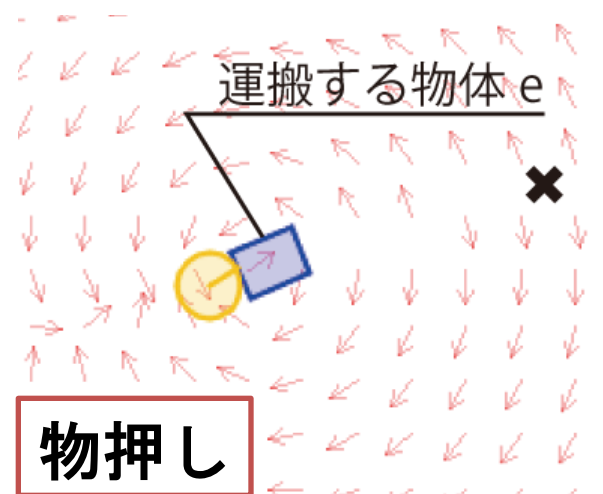
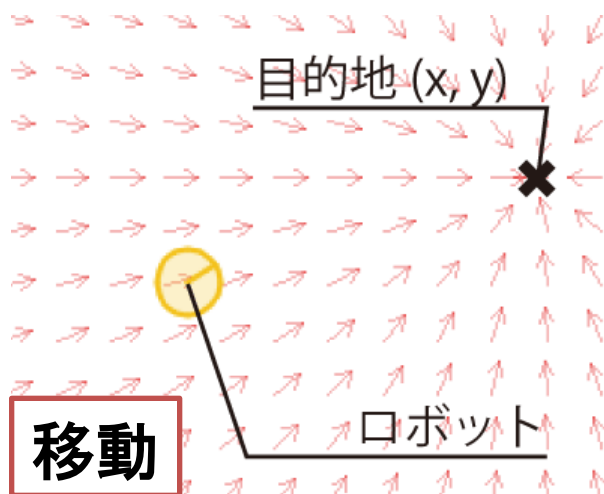
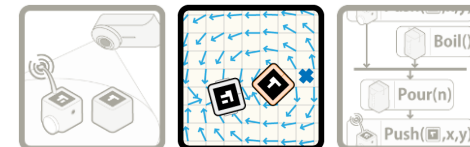
```
Move move = new Move(x, y);  
if (move.assign(robot)) {  
    move.start();  
}
```

// 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



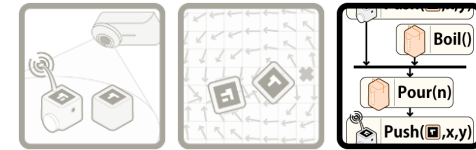
[Igarashi et al. 2010]

移動以外のタスク

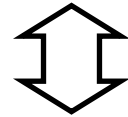
- 筆の上げ下ろし (LEGO Shodo)
- カメラ撮影 (ネットタンサー)
- 掃除 (ルンバ)
- etc.



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。



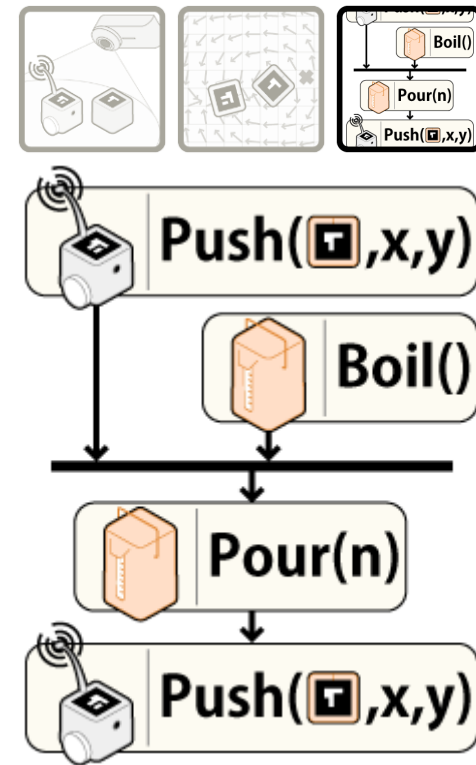
- エンドユーザの指示に従ってインタラクティブにタスクの動作順序を変えたい。
 - 料理、服たたみ、etc.
- 複数台のロボットに同時に作業させたい。
 - 味噌汁を作りながらお茶を淹れる、etc.

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);
```

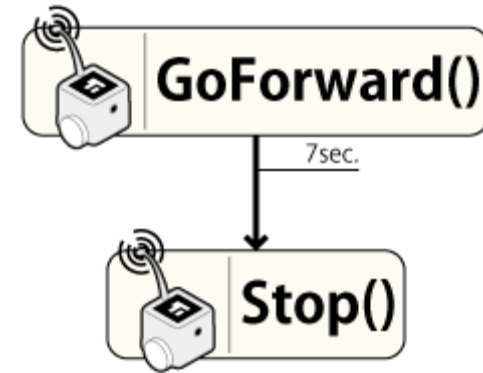
// エッジを追加

```
ad.addTransition(new TimeoutTransition(a, b, 7000));
```

// 実行

```
ad.setInitialNode(a);
```

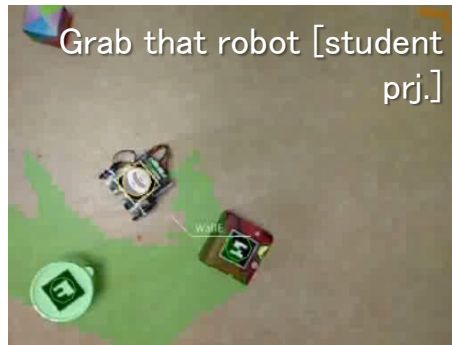
```
ad.start();
```



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



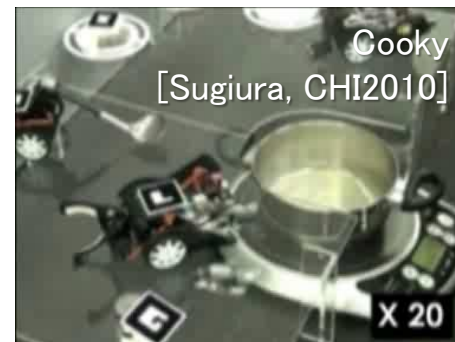
マルチタッチマルチロボ操作



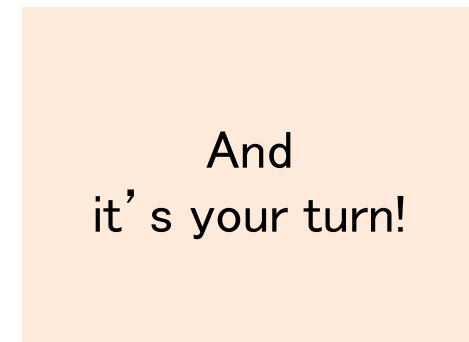
移動スピーカー



LEGO書道

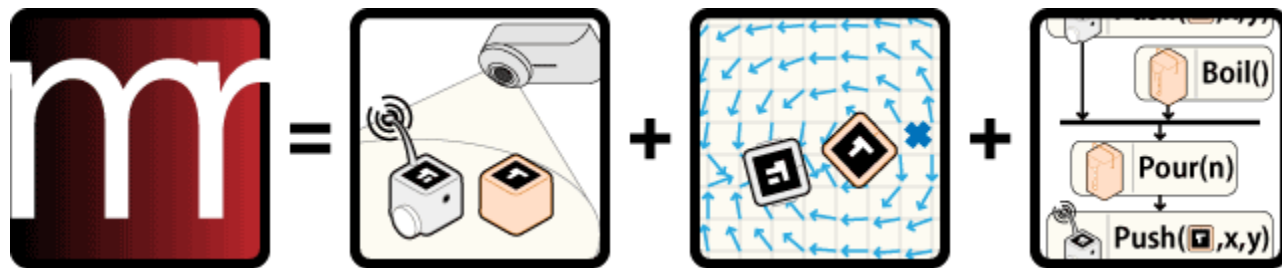


調理手順指示UI



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」

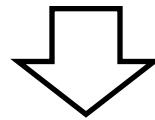
思考実験の結果

- 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。



@arcatdmz
arc@dmz

```
new 持ってこい(new おひるごはん(),  
俺.getLocation() /* おなかへった  
*/).assign(だれでもいいから).start();
```

1時間前 Keitai Mailから ☆お気に入り ↶返信 🗑️削除

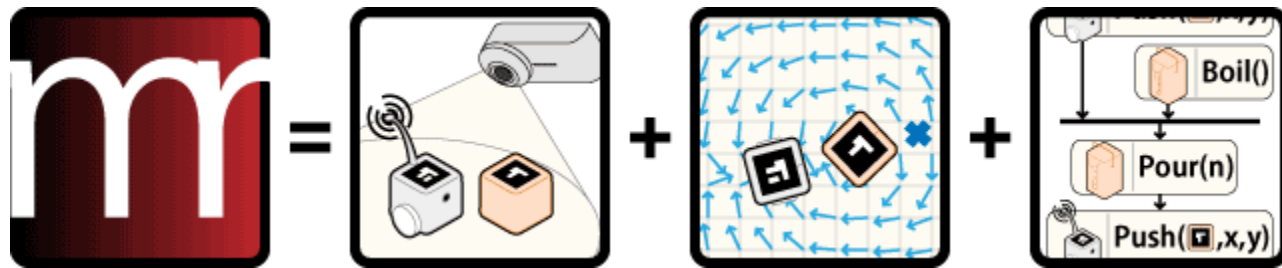
未来ビジョン

- 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
 - ~~EEG~~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。