



2008 年度下期未踏 IT 人材発掘・育成事業 採択案件評価書

1. 担当PM

石川 裕 PM (東京大学大学院 情報理工学系研究科 教授)

2. 採択者氏名

チーフクリエイター: 加藤 淳(東京大学理学部情報科学科 学部学生)

コクリエイター: なし

3. プロジェクト管理組織

株式会社メルコホールディングス

4. 委託金支払額

6,496,266 円

5. テーマ名

matereal:小型ロボットの簡単な行動デザイン用ツールキット

6. 関連Webサイト

<http://matereal.sourceforge.jp/>

<http://digitalmuseum.jp/software/matereal/>

7. テーマ概要

本提案では、小型ロボットの操作アプリケーションを、個人が簡単な手続きでプログラミングできるツールキットの開発を行う。物理的な存在(material)であるロボットをリアル(real)と仲の良いソフトウェア(mate)としてとらえ、ロボット工学にはそれほど詳しく

ないけれど、リアルに飛び出たプログラミングがしたい人たちの要求に応えることを目指す。

世界におけるロボット工学の発展は、とくにハードウェア開発の面において、日本が牽引してきたと言える。ロボットのソフトウェア開発競争についても同様の期待があるだろう。しかし、ソフトウェア開発に関する専門知識を持たないロボット工学者が主としてロボットの制御用ソフトウェアを開発しているなど状況は芳しくないようだ。

一方、近年になってハードウェアの価格が下がり、家庭向けロボットをはじめとする小型ロボットも、現実的な価格で複数種、市販されるようになってきた。これらのロボットを、メーカーや種類に依らず操作できる一般ユーザ向けのソフトウェア環境があれば、今後のロボットの一般化に大きく貢献するはずだ。学術領域においても、ロボットのプロトタイピング環境があれば、これまでボトルネックだったソフトウェア開発の多くを省略できて、研究開発を迅速化できるだろう。

これらの現状を踏まえ、ハードウェアとしてのロボット工学ではなく、ソフトウェアとしてのロボット工学の観点から開発を進められるツールキット「matereal」を提案する。

期間内に、

1. 近年になって普及した安価な PC 用カメラ(Web カメラ)を用いて環境認識を行い、
 2. ロボットの行動プログラミングを容易にするツールキットを開発し、開発者向けに API を公開する。
 3. その際、特定・一台のロボットのみならず、多種・複数のロボットが混合する環境においても同じ動作を実現する。
 4. かつ、これを簡単な設定で安定して動作させる。
- 以上を満足する水準まで開発を進める。

8. 採択理由

LEGO Mindstorm NXT, NetTansor, Roomba など複数のロボットを制御するためのプログラミングインターフェイスを開発しようという提案である。

WEB カメラによるロボットの位置情報も取得するためのライブラリも附属される。

適切なライブラリ群が提供されることにより、新たなプログラミング環境が構築され、利用者市場も開拓される可能性を秘めている。

9. 開発目標

本プロジェクトでは、近年登場した比較的安価な家庭用の小型ロボットを簡単に遠隔操作できる Java 及び Processing 用ツールキットを開発し、ソースプログラムを公開すると共に API を規定する。以下のようなプログラミングを想定する。

安価なPC用カメラ(Webカメラ)を用いて環境認識を行い、ロボットの位置座標からロボットの遠隔操作を行なう。

同じプログラムで多種の小型ロボットを同様に操作できるようにロボットの遠隔操作を抽象化し、市販ロボット固有のコマンド体系を隠ぺいする。

10. 進捗概要

プロジェクト開始当初より、ユーザがオブジェクト指向プログラミングの利点を生かした直感的なロボットプログラミングおよび再利用が可能となるようなAPI設計に時間を費やした。また、既存の同様のロボットプログラミング環境との違いを明確化した。サンプルプログラミングを充実させ、提供APIのプログラマビリティの検証も行った。ほぼ期待通りの成果を上げたといえる。

11. 成果

Materialの動作環境を下表に示す。

OS	Windows 2000/xp/Vista/7(RC1), Mac OSX Tiger/Leopard, 他、Processingが動作するLinux
開発環境	Java, Processing
実行環境	Java VMとJRE1.5以降の標準ライブラリ
その他要件	ロボットとの接続に用いる物理インターフェース (Bluetooth, Ethernetポート, Serial/Parallelポートなど)
模式図	<p>The diagram illustrates the system architecture. It shows a small robot (3) with a marker (4) on top. The robot is connected via a webcam (2) to a computer (1). The robot communicates wirelessly with the computer. The components are labeled: 4. マーカー (Marker), 2. Webカメラ (Webcam), 3. 小型ロボット (Small Robot), and 1. 計算用PC (Computing PC).</p>

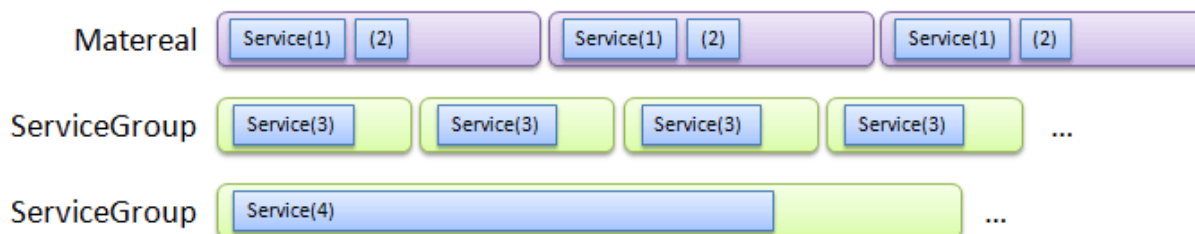
模式図にある通り、Material は、以下の 4 つのハードウェア構成を想定している。

1. 指定された OS と実行環境を備えた計算用 PC
2. Web カメラ
3. 対応するクラスの用意された小型ロボット
4. ロボットなどに貼りつけて位置検出に使う ARToolKit マーカー

ロボットの位置検出用に ARToolKit マーカーを用い、Web カメラによりロボットの位置情報を得る。ロボット位置情報を必要としないアプリケーションでは、これらは必要としない。

Service

material において、ロボットの挙動は Service インターフェイスの実装クラスとして定義される。Service インターフェイスは start(), stop(), pause(), resume()などのメソッドを持っている。material システムにより Service インターフェイスを持つインスタンスの start メソッドが周期的に呼び出される。複数の Service を同一周期で実行するのではなく、グループ化して異なる周期で Service の run メソッドが呼び出されるように ServiceGroup というインターフェイスを提供している(下図)。



Service インターフェイスの組み込み実装クラスとして以下の 2 つを紹介する。

(1) Camera

Camera は動画像のキャプチャを行い画像を配信するサービスである。画像を配信するインターフェイス ImageProvider に加え、スクリーン座標系と世界座標系の変換関数を与えるインターフェイス CoordProvider を実装している。このサービスを用いると簡単にカメラ映像を PC 上のウィンドウに出せるほか、次に説明するサービス MarkerDetector のソースとして割り当ててマーカーを検出することもできる。また、クリックされた位置の世界座標系における座標値なども簡単に取得できる。

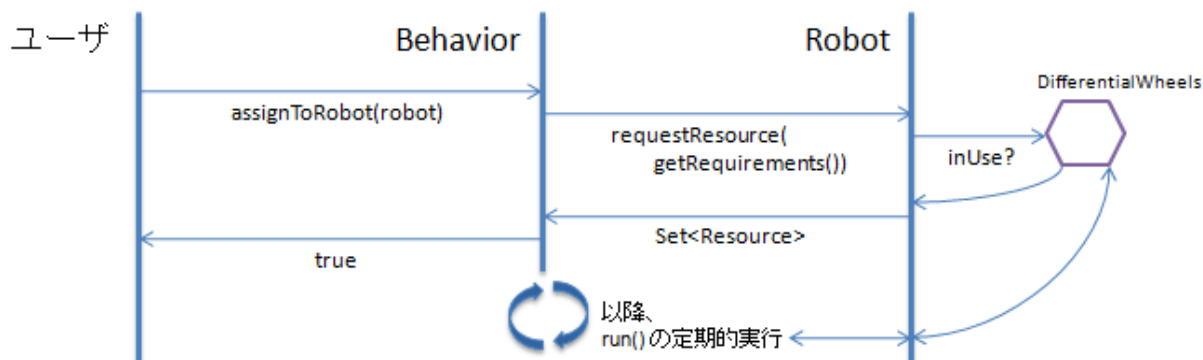
(2) MarkerDetector

MarkerDetector は、ARToolKit を用いたマーカー検出器を用い、Camera など

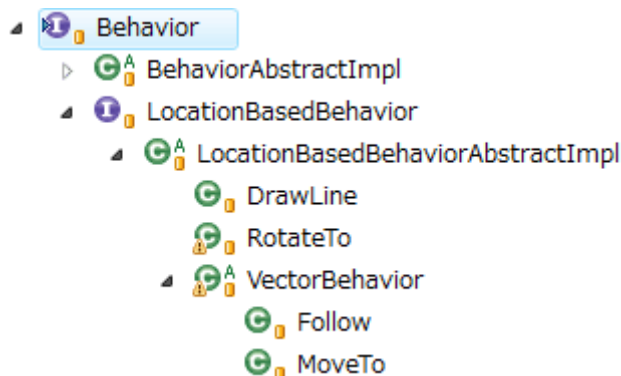
ImageProvider の実装クラスが提供する画像の中でマーカ-の位置と傾きを取得するサービスである。後述する Entity とマーカ-の対を登録すると、スクリーン座標系で Entity の検出位置を教えてくれる。マーカ-の検出結果を描画する補助機能もある。

Behavior

ロボットの振る舞いを決めるのが Behavior インターフェイスであり、Service インターフェイスを継承する。Behavior とロボットの実体との関連づけは動的に行なわれる。Behavior には assignToRobot というメソッドが用意されている。ユーザは、Behavior の assignToRobot を呼び出すときにロボットオブジェクトを引数に渡す。

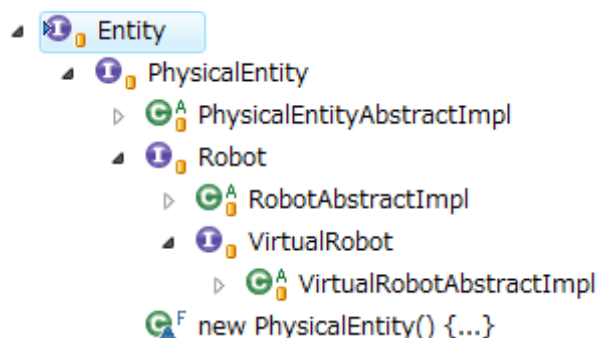


Behavior は、ロボットオブジェクトとの間で上記図に示すようにロボットが持つ可動部分を制御する権利を要求し(requestResource)、以降、定期的にロボットの可動部分を制御する。ロボットの振る舞い記述を容易にするために下図に示す通りの Behavior クラス階層を提供している。下図に示す通り、ロボットの座標を取得するメソッドが定義される LocationBasedBehavior インターフェイスおよびその抽象実装クラスである LocationBasedBehaviorAbstractImpl を用意している。また、ロボットの移動として、DrawLine、RotateTo、VectorBehavior などの抽象クラスを提供している。

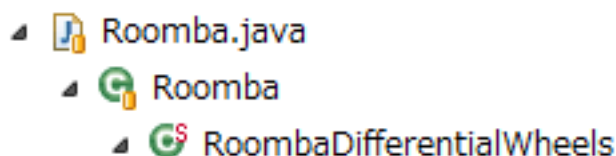


Robot

ロボットやマーカーを付した物体を表す Entity とそれを保持する EntityHolder が提供されている。Entity の階層を以下に示す。



ロボットは Robot インターフェイスの実装として、ロボットの種類ごとに個々のクラスで定義される。ロボットの部品は Resource というインターフェイスの実装として、ロボットの内部クラスで定義される。例えば iRobot 社の Roomba というロボットは、下図に示す通り差動車輪を表すインターフェイス DifferentialWheels の独自実装 RoombaDifferentialWheels を内部クラスとして保持する Roomba クラスとして定義される。



プロジェクト期間中に開発した市販ロボットのためのクラスは、NetTansor、LEGO Mindstorms NXT、Roomba である。

12. プロジェクト評価

当初目標であった家庭用の小型ロボットを簡単に遠隔操作できる Java 及び Processing 用ツールキットである Material が実現された。Web カメラ画像によるロボットの位置検出と、その情報によるロボットの移動を制御するデモプログラムを作成し最終成果報告会でデモすることができた。ほぼ期待通りの成果を上げたといえる。プログラミングの容易性、拡張性、記述能力などを評価して、より記述性の高いシステムに育てていくことを期待する。

13. 今後の課題

まず、ソフトウェアならびに API 仕様書を公開し、一般ユーザに使用してもらうことが直近の課題である。そして、ユーザからのフィードバックを得ながら、システムの不具合ならびに API の問題点を洗い出す必要がある。さらに、Material を使ったアプリケーションの充実とロボット動作の高度化を支援するクラス群の整備が必要であろう。