情報処理学会論文誌 Vol. 52 No. 4 1425–1437 (Apr. 2011)

Andy：俯瞰カメラとマーカを用いた移動ロボットアプリケーション開発用ツールキット

加藤 淳[†1,†2] 坂本大介[†2]
稲見昌彦[†2,†3] 五十嵐健夫[†1,†2]

小型の移動ロボットは家庭での日常的なタスクを支援するアプリケーションが期待されており，タスクの指示において洗練されたユーザインタフェースが必要となる．しかし，Human-Computer Interaction 研究者の多くを含むロボット工学の知識を持たないソフトウェアプログラマにとって，ロボットを用いたアプリケーションをプロトタイピングすることはいまだ容易とはいえない．そこで我々は，Graphical User Interface のように 1API コールで平面上のロボットや物体を移動させ，リスナで二次元座標値の変化イベントを取得することのできるツールキット Andy を開発した．Andy はロボットや物体の上面にビジュアルマーカを貼り付け，俯瞰カメラの撮像からマーカ検出することで作業空間床面上の二次元座標系を取得する．本稿では，ツールキット Andy の狙い，API と実装の概要，ユーザスタディの方法と結果および関連研究について報告する．

Andy: A Toolkit for Developing Mobile Robot Applications with a Ceiling-mounted Camera and Visual Markers

Jun Kato,[†1,†2] Daisuke Sakamoto,[†2]
Masahiko Inami[†2,†3] and Takeo Igarashi[†1,†2]

Small mobile robots are expected to be utilized for helping daily tasks at home. We need sophisticated user interfaces for them. However, prototyping of robot applications is still difficult for software programmers without prior knowledge of robotics including many researchers in the field of Human-Computer Interaction. We developed a software toolkit called “Andy”, with which programmers can make robots move and push objects on a flat surface with one API call and receive their two-dimensional motion events by registering listeners. Design of the APIs is influenced by programming style of Graphical User Interface. Andy provides two-dimensional absolute coordinates on the surface by detecting visual markers attached to top surfaces of robots and objects from captured images of a ceiling-mounted camera. We will report the aim of the toolkit, summary of its APIs and implementation, method and results of user studies and related work.

1. はじめに

ハードウェアの製造技術の発達により，比較的安価な小型の移動ロボットが一般家庭向けに購入できるようになってきた．たとえば，掃除ロボットの Roomba，教育と趣味用途のネットタンサーや LEGO Mindstorms，踊る音楽プレイヤ Rolly などがあげられる．小型の移動ロボットは，将来的に掃除や物運び，服畳みや料理など，家庭での日常的なタスクを支援する使途が期待されている．そういったタスクを指示する際に必要とされるのが，洗練されたユーザインタフェースである．ユーザインタフェースは実働するアプリケーションの一部として開発するものだが，ロボットに関していえば，アプリケーションの開発はロボット工学の知識を持たないソフトウェアプログラマにとって容易ではない．これにはいくつかの問題が存在する．まず，実世界には Graphical User Interface（GUI）のように自明な絶対座標系が存在しないため，興味の範囲をフレーミングし，ロボットや物体など興味の対象をモデル化して，それらの位置を定期的に計測する必要がある．また，実世界のタスクは実行に時間がかかるため，その間はモデル上の状況の変化を監視し，必要に応じてアクチュエータなどへの出力を制御することになる．さらに，ユーザインタフェース開発においては試行錯誤を重ねながら改良を行うプロトタイピングが重要となるため，ハードウェアのセットアップも簡単であることが望ましい．

我々は，ロボットアプリケーションの開発者層を厚くすることでロボット用ユーザインタフェースの開発が促進され，ひいてはロボットの実用化に貢献できるのではないかと考えている．そこで，前述の要求に応えるツールキット Andy を提案する．Andy は，与えられた範囲内の平面を動き回るような移動ロボットを利用したアプリケーションの開発を対象とする．Andy を用いる際は，ロボットと物体の上面にビジュアルマーカを貼り付け，俯

†1 東京大学
The University of Tokyo
†2 科学技術振興機構 ERATO 五十嵐デザインインタフェースプロジェクト
Japan Science and Technology Agency, ERATO IGARASHI Design Interface Project
†3 慶應義塾大学
Keio University

瞰視点の天井カメラ（俯瞰カメラ）でその様子を撮影する．ロボットとカメラはパーソナルコンピュータ（PC）に接続する．本ツールキットの想定ユーザ層は，Human-Computer Interaction（HCI）研究者の多くを含む，ロボット工学の知識を持たないソフトウェアプログラマである．したがって，API 設計に際しては彼らが慣れ親しんだ GUI のプログラミング手法を参考にした．プログラマは，ロボットに対して，カメラが見ている範囲の平面上の二次元絶対座標系を用いて指定位置への移動や物体の運搬を指示できる．また，ロボットや物体の位置変化イベントをリスナによって取得できる．これらの API は Java 用のライブラリとして提供され，プログラマは PC 上の一般的な Java 開発環境だけでアプリケーションを実装できる．

本稿では，まずツールキット Andy のデザイン指針と実装の概要を説明する．そして，大学院の講義におけるユーザスタディとロボット用ユーザインタフェース研究における実用例を紹介し，Andy の限界および今後について議論して関連研究を紹介する．

2. ツールキット Andy のデザイン指針

本章では Andy を設計する際に我々がとった基本方針を説明する．

2.1 Human-Computer-Robot Interaction のプロトタイピング

ユーザインタフェースに関する研究ではプロトタイピングの重要性が認識され，GUI[1]，Web デザイン[2]，Augmented Reality[3] など，さまざまな分野でプロトタイピング用ツールが研究開発されている．これらのツールは最終成果物を開発するためのツールとは明確に区別され，それぞれ相補的な役割を担っている．プロトタイピングではさまざまなアイデアを試作，テストし，改良を重ねる開発過程を繰り返す．アイデアが固まったら時間をかけて開発を行い，最終成果物としての完成度を追及する．したがって，プロトタイピング用ツールには成果物の完成度を向上させることよりもアイデアを素早く試作品として実装できることが求められる．我々は，ロボットが研究段階から実用になるにつれて，HCI の場合と同様に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が必要になり，重要性を増すと考えている．

Andy は PC をロボットやカメラなどのネットワークのハブとして用いる．プログラマは PC 上でロボットを用いるアプリケーションを記述，実行，デバッグする．HCI 分野のユーザインタフェースは一般的な PC で開発されているため，Andy を介してロボットに接続し，ロボット用のユーザインタフェースに援用することができるだろう．Andy はこのような「Human-Computer-Robot Interaction」ともいうべきインタラクションスタイルを支援，実現することを目指している．この PC 中心のミニマルなシステム構成は Andy の大きな特徴である．実環境でロボットを用いる既存研究には，ロボット単体のセンサで環境を認識するものと，ロボティックルーム[4] のように環境側にもセンサが備わっており，センサとロボットがネットワークでつながって全体として 1 つのシステムになっているものがある．我々は環境側にカメラというセンサを設置するため，後者のアプローチをとっているといえる．一般に後者のアプローチではネットワーク上で複数のノードが並列に動作してやりとりをするため，これを実現する基盤技術として RT-Middleware[5] のようにミドルウェアの開発が研究課題となる．しかし我々は，PC をすべてのノードのハブとして用いる中央集権的なネットワークを用いることでミドルウェアを不要にした．PC は演算装置として高性能でありながらコモディティ化して安価なため，導入が容易かつビジュアルマーカ検出など比較的高負荷な処理を任せられる．また，システム全体が PC 上で稼働するため，ソフトウェアプログラマが馴染んだ開発環境を用いることができ，HCI と同様の作法でロボットのユーザインタフェースのプロトタイピング開発に取り組める．

2.2 二次元の絶対座標系上での座標操作

ロボットの動作は，大まかに環境内での場所の移動と局所的なタスクの実行に分けることができる．産業用ロボットであれば環境内での位置が固定されていることも多いが，人と作業空間を共有する家庭用ロボットは机上や床の上などを動き回れることが必要である．このような動作は Localization（ロボットの測位）と Navigation（環境内での移動指示）によって実現される．しかし，Localization はロボットやロボットが動作する環境に備わったセンサの種類によって違った実装が必要になる．また，Navigation は Localization によって与えられた地図の種類や位置情報の確実性に応じて適した手法が異なる．したがって，Localization と Navigation を担当するプログラムのモジュールは通常ロボット研究ごと，あるいは研究室など動作環境ごとに開発されており，汎用的な機能を提供するツールキットは少ない．ロボット工学に関する前提知識を持たないソフトウェアプログラマがロボットを用いたアプリケーションを開発しようとする場合，これらの手法から適したものを選んで実装することは難しい．また，CARMEN[6] や Player[7] といった既存のツールキットは環境の地図を与えないと動作しないため，測位と地図作成を同時に行う Simultaneous Localization and Mapping（SLAM）と呼ばれる手法の実装を OpenSLAM[8] などから入手して組み合わせる必要がある．

Andy は，俯瞰カメラとビジュアルマーカを用いた Localization によって得られる二次元座標系上で，ベクトル場を利用した Navigation を行うツールキットである．カメラの撮

像上のスクリーン座標系がそのまま実世界座標系となるため，ツールキットを使い始める際の座標系のキャリブレーションは不要である．ロボット側に要求される機能は前後進およびその場での方向転換だけであり，本ツールキットは差動車輪を備えた移動ロボットや小型二足歩行ロボットなど多くの市販ロボットに対して用いることができる．本ツールキットのAPIは，GUIツールキットが提供する標準的なスクリーン座標系を用いたAPIを参考に設計されている．プログラマは，GUIにおけるマウスリスナのように，ロボットや物体を表すオブジェクトに位置変化を取得できるイベントリスナを登録できる．また，指定位置への移動や物体の運搬といったロボットへの指示出しを1APIコールで行える．本来このようなタスクはGUIで画像の表示位置を変えたりするのと異なり実行に時間がかかるため，状況を監視しながら必要に応じて何度も低レベルの移動指示を出す必要がある．Andyはこの処理を隠蔽し，絶対座標系上での座標操作のみをAPIとして公開している．

3. ツールキット Andy の概要

本章ではAndyの動作環境，APIと実装の概要を説明する．

3.1 ハードウェアのセットアップ

Andyは，**図1**のようにビジュアルマーカが貼られたロボットなどの物体が床などの平らな面上に置かれ，平面を垂直に見下ろす俯瞰視点のカメラが設置された環境を前提としている．ビジュアルマーカは，十分な性能を持ち検出用ライブラリがオープンソースで入手できるARToolKit[9]用のものを採用した．ユーザは，互いに異なるビジュアルマーカを操作対象のロボットおよび物体の数だけ用意する．普通の紙に印刷したものは光の反射で白飛びし，検出しにくいことがあったため，後述するユーザスタディでは白いアクリル板に黒いベロア生地の布を切り貼りしたものを用いた．ロボットは標準で**図2**に示す4種類に対応しており，複数種複数台を混在して利用できる．3種類のメーカ製ロボットは入手が簡単で，小型かつ扱いやすく，家庭の一般的な環境で動き回れる安定した性能を備えている．また，図2右端のminiは我々が開発したロボットである．アクリルの外装にマイクロコントローラ，単3電池6本，サーボモータ2つ，Bluetooth接続モジュールを組み込んだもので，2万円弱で製作できる．

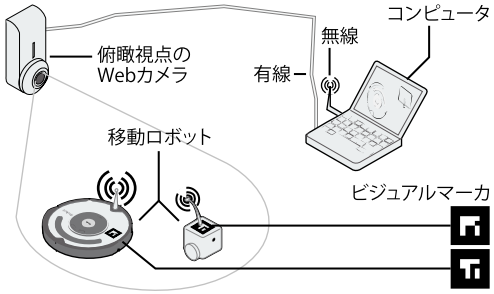

図1　Andyの動作環境の概要
Fig. 1　Overview of the environment setup of Andy.

ロボットは種類ごとにJavaのクラスとして記述されており，独自のロボットを開発した場合は3.3.4項で示すように対応するクラスを作成すればAndyから利用できる．現状，Bluetooth，TCP/IP，COMポート（USBの仮想COMポート含む）で接続するロボットに対応できる．カメラは，IEEE1394接続の高価なカメラだけでなく，USB接続可能なWebカメラと呼ばれる安価な商品を用いることができる．

アプリケーションで利用したいハードウェアについての情報は，事前にカタログと呼ばれるXMLファイルに記述しておく．カタログでは，カメラ，物体とロボット1台ごとに1ノードで初期パラメータを記述する．初期パラメータは，物体なら貼りつけたビジュアルマーカを表すファイル名，ロボットなら型（クラス）名や接続に用いるMACアドレスである．また，カメラなら名前や解像度のほかに，撮影範囲の床面の縦または横の大きさをcm単位で記述する．この情報をもとにピクセル単位のスクリーン座標とcm単位の実世界座標が変換される．

3.2 ソフトウェアの動作環境

Andyは Java 用のライブラリとして実装されており，Java VMが動作するWindows，Mac OS Xおよび Linux 上で利用できる．Javaは GUI 関連の APIが標準化されており，

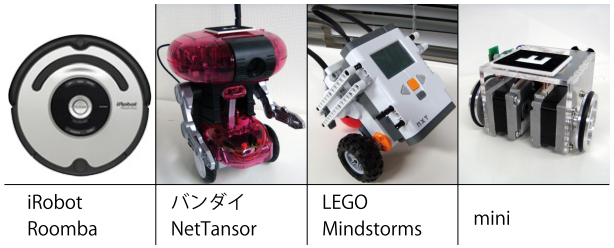

図2　Andyが初期状態で対応しているロボット4種
Fig. 2　Four types of default supported robots.

さまざまな OS 上で動作し，HCI およびロボット分野で使用実績が多かったため採用した．

3.3　API の概要

Andy の API はオブジェクト指向で設計されており，ロボットや物体，カメラを表すクラスのインスタンスが実ロボットや実物体，実カメラに対応する．ツールキットはアプリケーションの起動時にカタログを読み込んでロボットや物体，カメラのインスタンスを生成，保持する．プログラマはツールキットの Singleton インスタンスを通してロボットなどの必要なインスタンスを取得し，ツールキットの組み込み機能を利用できる．また，用意されたインタフェースを実装したり既存のクラスを継承したりしてツールキットの機能を拡張できる．以降 API の概要を説明するので，必要に応じて次項の**表 1** を参照されたい．

3.3.1　移動と物体の運搬 API

ロボットの低レベルの移動機能は移動ロボットを表すインタフェースでメソッドとして定義されており，すべてのロボットを表すクラスが実装している．現状，前後進および左右回転に相当する機能が用意されている．これらの機能は呼び出しと同時にロボットへ対応するコマンドが送られる．

高レベルの移動機能，すなわち絶対座標系上の指定した位置へ移動する機能と物体を指定した位置へ運搬する機能は移動ロボットを表すインタフェースでメソッドとして定義され，抽象基底クラスで実装されている．これらの動作は内部では次項の**図 3** のようなベクトル場として表現されている．所定位置への移動は蟻地獄のようにゴールの 1 点へ向かうベクトル場であり，物体の運搬は Igarashi らのダイポール場[10] を用いている．

高レベルな移動機能が呼び出されたら，位置情報が更新されるたび 3.4.2 項で詳述するような処理が行われてロボットが移動する．ロボットが移動中に新たに高レベルなメソッドが呼び出されたら指示内容が上書きされる．

3.3.2　位置情報を取得する API

ロボットや物体の位置情報を取得するための API は物体を表すインタフェースでメソッドとして定義されており，すべてのロボットおよび物体クラスが実装している．プログラマはメソッドを呼び出してそのときどきの位置情報を取得できるほか，リスナを登録して位置情報が更新されるたびに情報の配信を受けることができる．本 API は Java 言語におけるマウスやキーボードの状態取得を行うイベントリスナを模して作成されており，ソフトウェアプログラマにとって分かりやすい実装となっている．

3.3.3　局所的なタスクに関する API

ロボットごとに持っている特殊な機能は各ロボットを表すクラスに直接実装されている．

表 1　主要なクラスとメソッドの一覧
Table 1　Set of major classes and methods of Andy.

クラスまたはインタフェース	メソッド	用途
class Andy	getInstance()	Singleton インスタンスの取得．
	getImage(overlay), drawImage(g, overlay)	カメラの撮像の取得と描画．ロボットなどの情報を重畳表示可能．
	getRobot(name), getRobots(), getEntity(name), getEntities(),	ロボットや物体を表すオブジェクトの取得．
	startService(svc, t), stopService(svc, t)	間隔 t[ms]で定期的に実行するタスクの開始と停止．
interface Entity	getLocation(), addLocationListener(), removeLocationlistener()	位置情報の取得．直接取得する以外に情報の更新を受け取るリスナを登録できる．
interface MobileRobot extends Entity	forward(), backward(), spinLeft(), spinRight()	低レベルの移動指示．前進，後進，左回転，右回転．
	moveTo(x, y), pushTo(e, x, y), followVectorField(vf)	高レベルの移動指示．順に，指定位置への移動，物体の運搬，ベクトル場に沿った移動．
class Roomba extends MobileRobotAbstractImpl	clean()	Roomba 固有の機能．掃除．
class NetTansor extends MobileRobotAbstractImpl	getImage() addImageListener() removeImageListener()	ネットタンサー固有の機能．頭部カメラの撮像取得．
class Location	getX(), getY(), getScreenX(), getScreenY(), getRotation()	位置情報を表すクラス．向きの情報[rad]と座標値[cm またはピクセル]を取得できる．
interface Service	run()	定期的に実行するタスクを記述．

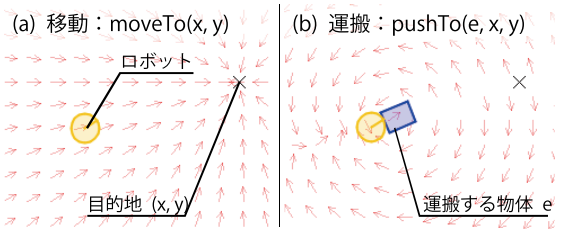

図 3　ロボットの（a）移動および（b）物体の運搬時のベクトル場の様子
Fig. 3　Vector field for a robot while (a) moving and (b) pushing an object.

たとえば Roomba は掃除機なので，その場を掃除するためのメソッドを持つ．ネットタンサーは頭部にカメラがあるため，撮像を取得するためのメソッドや，撮像が定期的に配信されるリスナを登録，解除するメソッドも持つ．

3.3.4　ツールキットの機能を拡張できる API

ロボット開発者が自作ロボットを Andy で使えるようにするには，ロボットと PC を 3.1 節で示したいずれかの手段で通信できるようにしたうえで，適切なプロトコルを定め，移動ロボットを表す抽象クラスを継承，実装するクラスを記述するだけでよい．Andy には，プロトコルの定義が面倒な場合のためにサンプル実装が用意されている．このクラスに対応するロボットは f，b，l，r，s の ASCII コードを受信したら前・後進，左・右回転，停止するように実装する．カメラやロボットハンドなどの追加機能をロボットに実装した場合は，クラスを継承して適宜メソッドを追加すればよい．

また，ツールキットには登録されたタスクを指定された頻度で定期的に実行するサービスと呼ばれる仕組みがある．たとえば 3.3.2 項の位置情報を取得する API は，撮像を取得してビジュアルマーカを検出するタスクを表すクラスがあって，このタスクのインスタンスがサービスとして定期的に実行されることで実現されている．ロボット開発者は，ロボットに搭載されたセンサの情報を定期的に取得する新たなサービスを実装できる．3.3.3 項で紹介したネットタンサーの撮像を得るための機能もサービスとして実装されている．

3.4　実　　装

本節では Andy の実装の概要を述べる．

3.4.1　ビジュアルマーカ検出

ビジュアルマーカの検出にはオープンソースで Java 版ライブラリが開発されている ARToolKit を用いた．ARToolKit は事前にビジュアルマーカのパターンデータを読み込んでおき，撮像中で矩形を検出して矩形内でパターンマッチングを行う．ARToolKit の API は矩形ごとに最も近いパターン，パターンの向きおよび検出結果の確度を返す．Andy は XML ファイルで事前に与えられた物体とビジュアルマーカの対応情報をハッシュテーブルで保持している．そして，撮像が更新されるたびにマーカ検出を行い，現在撮像中で見えている物体を取得して物体を表すオブジェクトの位置フィールドをセットし，登録されているリスナに通知する．なお，ビジュアルマーカのパターンデータについては認識しやすいものをツール[⋆1]で機械的に生成できる．

⋆1 ARToolkit Patternmaker，http://www.cs.utah.edu/gdc/projects/augmentedreality/

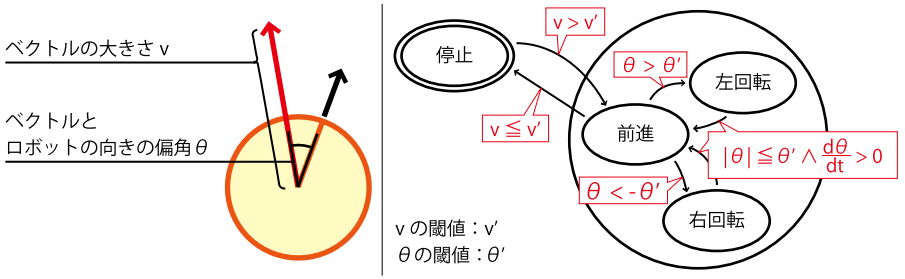

図 4　ベクトル場上のロボットの状態遷移図
Fig. 4　State diagram of a robot on a vector field.

Andy は現状 1 台のカメラを 1 座標系と紐付けするため，カメラの解像度が利用可能な実空間の広さを規定する．800 × 600 ピクセルの映像を 30 fps で伝送可能な Web カメラを用いたところ，床では高さ 2.4 m にカメラを置いたところ 2.8 × 2.1 m 程度の広さを確保でき，12 cm 四方のビジュアルマーカをロバストに検出できた．机上では高さ約 70 cm のカメラで 80 × 60 cm 程度となり，5 cm 四方のマーカをロバストに検出できた．

3.4.2　ベクトル場によるロボットの移動制御

Andy はロボットの移動をベクトル場によって制御している．ロボットの移動が指示されるとロボットに対してベクトル場が割り当てられ，位置情報が更新されるたびにその位置でのベクトル値が読み出されて，**図 4** のようにベクトルの偏角と大きさに応じて前進，回転，または停止する．回転については，偏角の変位が小さくなる限り回転を続け，大きくなった瞬間に停止して前進を始める．ベクトルの大きさと偏角の閾値はプログラマが動的に変更できる．

3.4.3　ハードウェアとの接続

Andy はカメラやロボットとの接続に外部ライブラリを用いている．カメラについては Windows 環境で DirectShow のラッパ，Mac OSX 環境で QuickTime for Java，Linux 環境で Java Media Framework を使用している．ロボットとの Bluetooth 接続には JSR-82 で定められた仕様の一部を実装する BlueCove ライブラリ[⋆2]を，TCP/IP 接続には JRE の標準ライブラリを，COM ポート接続には RXTX ライブラリ[⋆3]を使用している．

⋆2 BlueCove JSR-82 project，http://bluecove.org/
⋆3 Rxtx，http://rxtx.qbang.org/

4. ユーザスタディ

本章では Andy の α 版を大学院生およびロボット用ユーザインタフェースの研究者に使用してもらった結果を報告し，Andy の有用性を検証する．本稿で提案している Andy とユーザスタディで用いた α 版の違いは，物体を押して運搬する機能の有無である．この機能はユーザスタディの結果ニーズが大きいと判断したために追加したものであり，ユーザスタディ内で実装されている運搬機能はそれぞれ被験者のプログラマが自力で実装したものである点に留意されたい．

4.1 大学院講義における試用

我々は，ユーザスタディとして HCI に興味のある大学院生に Andy を提供し，実際にロボットを用いたアプリケーションを開発してもらった．本節ではその手法と結果に続き，学生の作例を一部紹介する．

4.1.1 手　　法

我々は，コンピュータ科学専攻の HCI に関する講義を履修している大学院生に，ソフトウェアのツールキット Andy とハードウェアの “ロボットキット” を与えた．Andy は，使い方の説明やリファレンスとともに，講義の Web サイトを通じて配布された．ロボットキットには，移動ロボット mini 1 台，Web カメラ 1 台（Logitech 社製 Logicool Qcam Pro for Notebooks），Bluetooth アダプタ 1 台（Princeton 社製 PTM-UBT5），ビジュアルマーカ 3 つと，ロボットの駆動に必要な充電池および充電器が含まれていた．

講義は 15 名の学生が履修しており，キットは数に限りがあったため，11 グループに分かれて課題にあたった．10 グループの学生はロボットを扱うプログラムを書いた経験がなく，後に紹介するプロジェクト “Grab That Robot” を実装した 1 名の学生に限り LEGO Mindstorms のプログラムを書いたことがあった．

我々は学生に対し，Andy の狙いと概要を説明してからロボットキットを配布し，Andy と移動ロボットを用いた新しい HCI や HRI のアプリケーションを考えて開発するという課題を出題した．課題終了時には，ソースコード，デモビデオ，アンケートへの回答とプロジェクトについての報告書の提出を求めた．また，ロボットキット以外に必要なハードウェアがあれば配布する旨を伝え，実際に要求があったグループに追加のカメラやロボットを与えた．なお，良い成績を得るためにアンケートにおいてツールキットに対して不当に高い評価を与えるようなプレッシャを与えないよう，課題は講義の成績とは無関係であることを明示した．

4.1.2 結　　果

課題に取り組んだすべてのグループの学生が実動するアプリケーションの開発に成功した．アンケート結果では，11 グループ中 6 グループの学生が Andy を使う前に “ロボットを用いたアプリケーション開発は困難だと思う” と答えていたのに対し，プロジェクトを開発したあとは 8 グループが実際には “困難ではなかった” と答えている．他の 3 グループに関しては，困難だった理由について “Andy が依存する他ライブラリのインストールが難しかった”，“XML 形式の設定ファイルを書くのが難しかった”，“ビジュアルマーカを検出する過程で撮像を二値化する閾値をうまく設定するのが大変だった” と答えている．他ライブラリについてはライセンスの関係で生徒が各自ダウンロードしてパスを通すよう指示されており，パスがうまく通せないのは API の使い勝手と無関係な Java の一般的な問題である．設定ファイルについては書式を Web 上で公開していたが，たとえ仕様が十分公開されていたとしても，テキストを直打ちする際には人為的なミスが容易に起きうる．また，撮像を二値化する閾値については，現状では試行錯誤を重ねるしかない．課題遂行途中でこれをサポートする独自の GUI ツールを配布したが，対応が不十分だったと考えられる．

以下，生徒の作例の一部を**図 5** にあげ，アプリケーションと実装の概要を紹介する．なお，その他の作例については講義の Web サイト[⋆1]を参照されたい．

- Moving Speaker

人が一番音楽を聴きやすいように，スピーカを載せたロボットが人にあわせて適切な位置へ動く，ユーザの明示的な操作を必要としないアプリケーション．人の両肩にビジュアルマーカを置く．両肩のマーカ位置の平均がある程度以上移動したのを検知するとロボットを移動させる API が呼ばれる．

- Grab That Robot

コンピュータのスクリーン上に表示されたロボットを指でつまんで移動させるユーザインタフェース．指の動きはロボットの座標系を見ているものとは別の Web カメラで画像右のようにキャプチャしており，コンピュータのスクリーンに画像左のようにオーバレイ表示されている．ロボットまたは既存の目的地を指でつまんで離す動作が完了するたび次の目的地が設定される．目的地が初めて設定されたとき，および現在の目的地に到達したとき，次の目的地へロボットを移動する API が呼ばれる．

⋆1 東京大学大学院 2008 年度メディア情報学, http://mr.digitalmuseum.jp/projects/workshop/media2008/

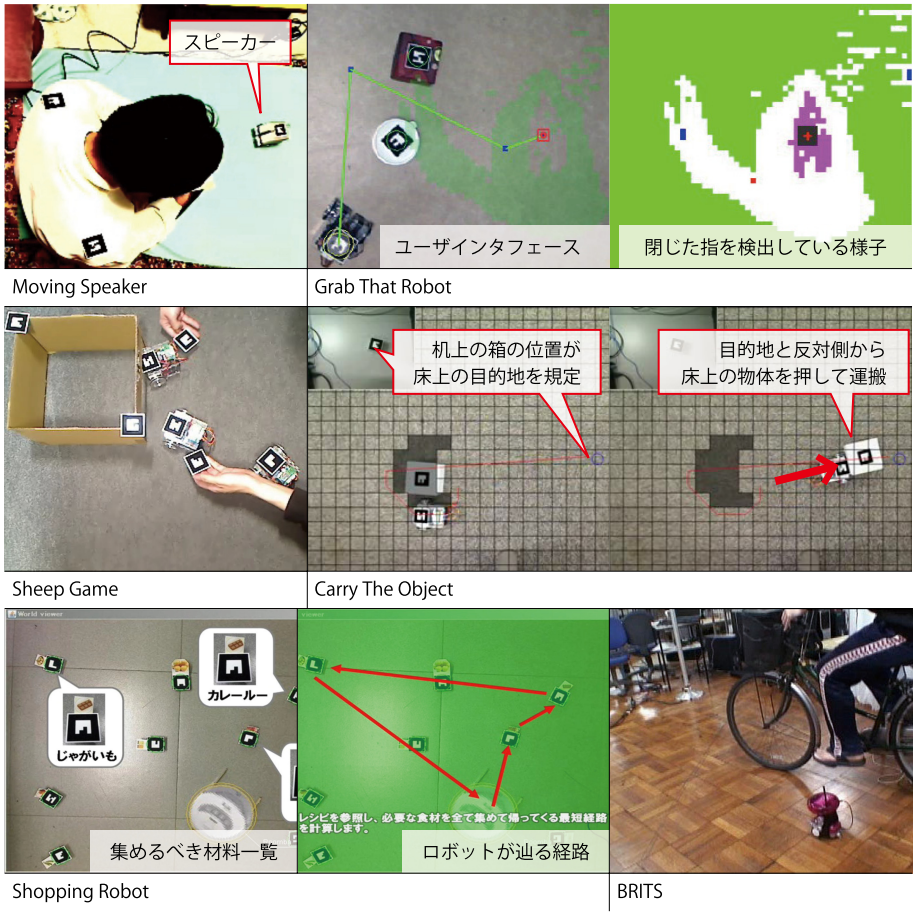

図 5　生徒が開発したアプリケーション例
Fig. 5　Example applications developed by students.

- Sheep Game

人の手でロボットを追い立てるように移動させられるユーザインタフェース．牧羊犬が羊の群れを追い立てる様を人の手とロボットで模したゲーム仕立てになっている．人の手にビジュアルマーカを置き，リスナで動きを検知するたび，現在見えているすべての手の位置から遠ざかる方向に仮の目的地が設定され，ロボットをそこへ向けて移動する API が呼ばれる．複数プレイヤで複数台を用いて遊べる．

- Carry The Object

机上の箱のミニチュアを動かすと，ロボットが実環境中の物体を床上で対応する位置まで運搬してくれる．ロボットは，まず，物体の周りで目的地の反対側となる位置へ物体とぶつからない経路を通って移動する．次に，ロボットが目的地の方角へ前進すると，物体が押されて目的地まで運ばれる．

- Shopping Robot

夕食を作るための食材をロボットに集めさせるシステム．食材にはビジュアルマーカが付してある．まず，ユーザが夕飯のメニューを引数に渡してシステムを起動する．次に，システムが必要な食材を通り，不要な食材を避けて通る経路をサブゴールのリストとして算出する．そして，ロボットが各サブゴールを巡回する．リスナでロボットの移動を検知し，サブゴールに到達するたび次のサブゴールへの移動を指示している．ロボットには簡単なアームが取り付けてあり，実際に食材を集めることが可能となっている．

- BRITS（Bicycle-Robot Interactive Tele-operation System）

自転車にロータリエンコーダを取り付けて車輪の回転数を得て，ハンドルの回転の情報とあわせてロボットの動作を指示できるユーザインタフェースである．低レベルの移動 API のみ使用している．低レベル API の使用例としては，他に Wii リモコンや音声認識でロボットを操作できるようにしたグループが観察された．

4.2　研究における実用

我々はロボット用ユーザインタフェースの研究者に Andy を提供し，実際に研究で使ってもらった．本節では**図 6** に示す 3 つのユーザインタフェースを例にあげ，実装時にツールキットの API がどう活用，拡張されたか説明する．

4.2.1　複数台ロボットの同時制御のためのマルチタッチインタフェース[11)]

マルチタッチディスプレイ上に俯瞰カメラの映像とベクトル場を可視化した様子が重畳表示されており，ユーザはディスプレイに触ってベクトル場を直感的に編集できる．複数台のロボットがすべて同一のベクトル場に沿って動くようにプログラムされているため，ユーザはベクトル場の編集を通してフィールド上の全ロボットの動きを同時にコントロールできる．

本研究の実装では独自のベクトル場を設計する API が使われている．すなわち，ユーザがマルチタッチインタフェースを介して編集できるグローバルなベクトル場を表すインスタンスを 1 つ用意し，すべてのロボットがそのベクトル場を見て動くような指示を出してい

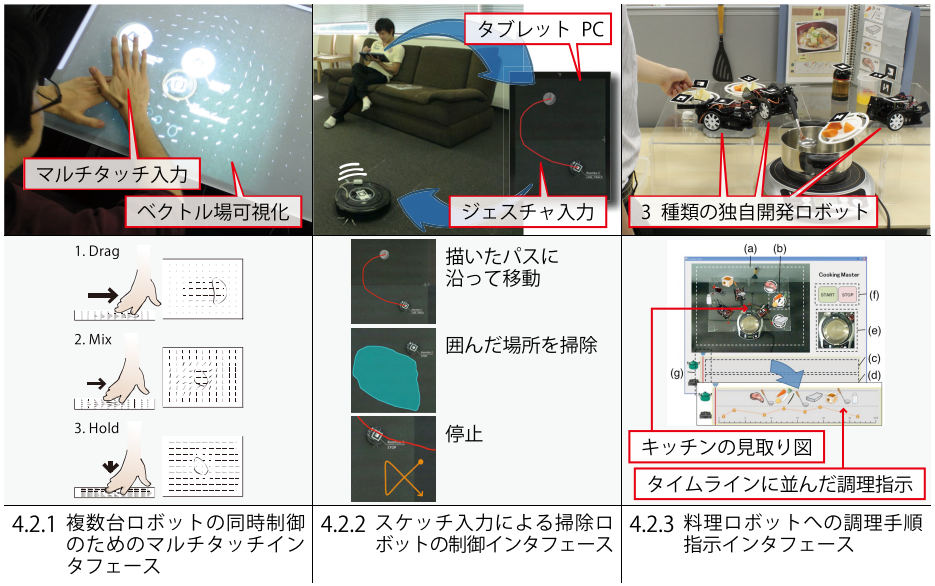

図 6　研究で開発されたユーザインタフェース例
Fig. 6　Example user interfaces developed by researchers.

る．ベクトル場を表すインスタンスでは，作業空間床面を一定の大きさの正方形格子に切り分け，格子の頂点ごとにベクトルを保存している．ある点におけるベクトル値は，近接する頂点のデータを補間して得るような実装になっていた．

4.2.2　スケッチ入力による掃除ロボットの制御インタフェース[12)]

ロボットがいる環境の俯瞰映像が投影されたタブレット PC 上で特定のジェスチャを描くと，入力された場所でロボットが掃除などのタスクをこなしてくれる．また，画面上にパスを描き，ロボットにパスをたどらせることができる．

本研究では，指定した場所へロボットを移動させる API と，移動を検知するリスナ，その場で掃除をさせる Roomba 固有の API が使われている．また，描いたパスをたどるためにパスを点列に分解して順にたどらせている．パスをたどらせる方法として点列への分解は非常にシンプルで分かりやすい．しかし，現実のロボットはセンサ情報に遅延があると目的地を行き過ぎることがあり，その際ロボットは前後を逆転して元いた方向へ引き返す．本研究の実装では，点列を適当に間引き，行き過ぎと判断する閾値（図 4 の v’）を大きくとって逆進がなるべく起こりにくいようにしていた．

4.2.3　料理ロボットへの調理手順指示インタフェース[13)]

味噌汁を作る際の具材の投入タイミングや鍋の温度の推移を GUI で指示し，具材を準備して置いておくとロボットが調理してくれる．

本研究では．食材の載った皿や調味料の入ったコップにビジュアルマーカを付し，鍋の撹拌，食材の投入，調味料の投入という 3 つの機能を持った 3 台のロボットを自作した．自作ロボットについてはそれぞれの独自機能を使えるように新たにロボットのクラスを定義している．皿を傾けるために特定の角度を向かせることができるサーボモータを用いているため，とくにロボット側のセンサ情報を PC 側で処理することなく現実に意味のあるタスクを任せることができている．

5.　議　　論

本章では 2 章であげたデザイン指針を検証し，ユーザスタディの結果をふまえて Andy の特徴について考察したうえで今後についての議論を展開する．

5.1　デザイン指針の検証

Andy の Human-Computer-Robot Interaction を容易にするシステム構成に関しては，ユーザスタディの結果，Andy の動作環境である Java 上で PC 向けのユーザインタフェースをロボットに接続する例が複数観察されており，ロボット用ユーザインタフェース開発を支援する目的での有用性が検証されたといえる．たとえば，Grab That Robot の指で輪を作って離す動作を検出するインタフェースは元々HCI 分野で提案されたものである．Carry The Object ではカメラ映像を投影したウィンドウのクリックを検出してロボットへの移動指示を発行している．また，Wii リモコンや音声認識でロボットを操作するインタフェースは PC 上で動作するライブラリを用いている．床面の二次元絶対座標系を提供したことについては，ユーザスタディのアンケートで 11 グループ中 8 グループがアプリケーション制作は困難ではなかったと答え，3 グループは API 以外の点で使い辛さを指摘していたことから，このデザイン指針が抵抗なく受け入れられたものと考えられる．ユーザスタディの多彩な実例から，Andy はソフトウェアプログラマが開発できるアプリケーションの幅を広げることに成功したといえるだろう．

5.2　Andy の応用範囲と拡張性についての議論

5.2.1　ロボットの開発とロボットのユーザインタフェース開発に関する議論

Andy はロボットを PC とつないで「使う」ためのツールキットであり，ロボット自体を

「作りこむ」ためのツールキットではない．すなわち，ロボット本体にコマンドを送信して利用することが不可能な機能を用いることはできない．たとえば，ロボットにレーザレンジファインダや超音波センサが積載されており，ロボットのプロセッサ上でセンサ値をモニタすることで衝突回避することが可能だったとしても，そのようなプログラミングはロボット用のファームウェアとして開発するほかなく，Andy がそれを支援することはできない．ただし，ロボットがセンサの生データをホスト PC に送信する機能を持っている場合は，プログラマはロボットクラスを拡張してセンサ値を処理するような Java プログラムを書き，ロボットの動作にフィードバックすることができる．その際，通信による遅延が無視できないほど大きくプログラムが正しく動作しない場合は，やはりロボット用のファームウェア開発が必要になるだろう．

なお，衝突を回避しながら目的地へ移動するなど，複数の目的を同時に達するようにロボットをプログラムする際，既存研究では Subsumption Architecture[14] のように目的ごとに低レイヤから高レイヤまでアルゴリズムを並列して走らせるなど，さまざまなソフトウェアアーキテクチャが提案されている．しかしながら我々は，ユーザインタフェース研究が黎明期にある現状において，ロボットの同じ部品を複数のアルゴリズムが制御する必要が生じるほど複雑なプログラミングが必要であるとは考えなかったため，特別なアーキテクチャを提供してプログラミングの様式を必要以上に縛ることをあえて避けてきた．また，単に衝突を回避しながら目的地へ移動するだけであれば，ロボット以外の物体の周りに反発するようなベクトルが生じる衝突回避用のベクトル場と目的地に向かうベクトル場を重ね合わせることによって，現在の Andy の枠組みの中で実現できる．さらに複雑な動作を設計するようになればロボット工学で提案されてきたさまざまなソフトウェアアーキテクチャが有用になることが考えられるが，そのようなアーキテクチャに基づく自律動作プログラムはロボット工学者がロボット本体に実装し，ツールキットはあくまで高レベルな機能を外側から「使う」ための API を提供するように役割分担したほうがよいのではないだろうか．この役割分担においてツールキットがどの程度の抽象化を施し，ユーザインタフェース研究者がどの程度まで低レイヤのプログラミングをするべきかという問いは，今後ツールキットを拡張していくなかで，また，ロボット工学者が多くのユーザインタフェース研究者と協働するなかで明らかにしていくべきオープンクエスチョンであると考えている．

5.2.2　実時間と実空間を扱うことについての議論

Andy はプログラマが二次元座標系をシンプルに扱えるようにするために，実世界アプリケーションに固有の問題に取り組んでいる．まず，ロボットの動作には時間がかかり，その過程でセンサの情報を取得して指示を出し直すフィードバックループを回す必要があるが，Andy を用いればロボットへの指示出しが基本的に 1API コールで済む．そして，GUI であればオブジェクトの移動指示は指定した場所へ正確に実行されるが，実世界ではセンサ情報の遅延やノイズによって正確な実行結果は期待できない．Andy はこの問題をベクトル場によって解決している．ベクトル場を用いればロボットの行動可能範囲全域においてどの方向へ向かえばいいかという情報が与えられるため，正確な実行結果が得られなかった場合のエラー処理を織り込み済みの移動指示を書くことができるのである．このように，実時間と実空間を扱う際に現出する問題の多くは座標系を二次元に限っていても直面するものであり，現状の Andy はソフトウェアプログラマが実世界アプリケーションを実装する初めの一歩を支援するツールキットとして十分有意義であると考えられる．

5.2.3　ツールキットの三次元への拡張についての議論

Andy はロボットの床面上での動作にフォーカスしたツールキットであり，現状ツールキットの組み込み機能で高さ方向のパラメータを扱うことはできない．しかしながら，三次元方向のパラメータを扱うアプリケーションは，ロボット側に必要な機能を実装すれば，3.3.4 項のようにツールキットを拡張して作ることができる．最も単純な例が，高さ方向の可動範囲が決まっている 4.2.3 項の具材投入，撹拌ロボットである．

三次元座標系が必要になるのは前例のようにロボットが目の前にある何かを認識したり操作したりするユースケースだろう．この場合，高度なタスクを指示するためにはロボットのローカル座標系を扱う必要が出てくる．したがって，ツールキットとして現在の二次元絶対座標系を単純に高さ方向に拡張した三次元絶対座標系を提供するだけでは実用性に欠ける可能性が高い．我々は，三次元座標系上でのロボットのタスクのプロトタイピングを積極的にサポートするためには，実際にアプリケーションを作成しながら，別途目的に合ったツールキットを設計してみる必要があると考えている．とくに，Andy の開発を通して GUI ライクなカメラの撮像上のスクリーン座標系で対象を指示する API がプログラマにとって直感的なものとして受け容れられたことは 1 つの方向性を示唆している．たとえば頭部カメラとアームのついたロボットであれば，カメラの撮像を取得する API と，カメラ座標系で物体の位置を指定し，その把持を試みさせるといった指示を出せる API をロボットクラスに用意することが考えられる．この際，撮像中のビジュアルマーカ検出処理を PC 側のタスクキューに登録して定期的に実行し，ロボットが見ているビジュアルマーカを貼り付けた物体を認識させることで，物体の三次元位置を計算でき，それを把持するために必要なアームへの出力を得られるだろう．

5.3　今後についての議論

5.3.1　実世界エラーハンドリングのサポート

Andy には実世界固有のエラーに関するハンドリングを支援するための機能が不足している．現状，ロボットがコマンドを受け付けなかった際に，そのイベントはアプリケーション側に通知されない．ロボット側の都合（センサで衝突を検知したなど）でコマンドが拒否されたのか，コマンドを送りすぎて通信が不安定なのか，ロボット側のバッテリが切れたのか，ハードウェアが破損したのかなど，原因の切り分けができないとデバッグに支障をきたすため，改善の必要がある．そのためには，API でロボットに指示を出したあと非同期で結果を返せる仕組みを実装する必要があるだろう．

5.3.2　ロボットの移動指示アルゴリズムの拡張

Andy は移動指示に際して目的地へ落ち込む非常に単純なベクトル場を用いている．これを 5.2.1 項で触れたように衝突回避のベクトル場と重ね合わせると，目的地に到達できない場合が生じうる．このような問題を避けるためには，さまざまに提案されている経路探索のアルゴリズムから適したものを選択して実装するか，目的地に到達できないようなベクトル場を生成しない，特殊なポテンシャル場を計算するアルゴリズム[15] などを実装する必要があるだろう．ベクトル場によって移動指示を出せるインタフェース自体は非常に直感的なものであり残すべきものと考えているが，ツールキットの組み込み機能として，より複雑な目的を達することのできるベクトル場を実装していく必要があるだろう．

5.3.3　二次元座標系に関する機能の拡張

Andy は現状鉛直見下ろし方向で床を撮影した状態でしか正しく動作しない．そして，1 つの座標系を 1 台のカメラでしか提供できない．今後，カメラのパラメータを計算して斜め見下ろし方向でも床面の座標系を取得できるようにしたり，複数のカメラで座標系を共有してより広い範囲を見られるようにしたりすることが考えられる．

また，現在の座標系関係の API はロボットから見たローカル座標系をサポートしていない．たとえばロボットの前方に物体が現れたらそれを指定位置へ運搬するといった動作を設計しやすくするためには，5.2.3 項でも触れたようにロボットのローカル座標系をサポートする機能を実装する必要があるだろう．

5.3.4　エンドユーザ向けプログラミング環境の開発

ユーザスタディでツールキットが提供する二次元の環境の見取り図をエンドユーザ向けのインタフェースにそのまま活用する例が多く見られたことは，プログラマにとって分かりやすい GUI のアナロジがエンドユーザにも同様に受け入れられる可能性を示している．今後，ツールキットをインタプリタと接続したり VPL を実装したりして，エンドユーザ向けプログラミング環境を開発して検証することが考えられる．

6. 関 連 研 究

6.1　ロボットや物体の測位に関する研究

ロボットが実世界で行動するためには自己位置を同定する必要があり，これまでにさまざまな測位手法が提案されてきた．屋外では Global Positioning System を用いるのが一般的である．精度が必要な場合にはロボット自身の走行距離センサを用いるオドメトリを併用することもある．屋内では，レーザーレンジファインダや超音波によって周囲の障害物との距離を測定して自己位置推定を行ったり，ロボットに設置したカメラの撮像から環境中の目印を検出したりする手法が提案されている．ただし，これらの手法は事前に環境の地図が必要か，あるいは自己位置推定と同時に環境の地図を構築する SLAM を用いる必要があり，物体の配置が動的に変わりうるアプリケーションに対応することが難しい場合が多い．そこで，実用的なロボットアプリケーションの研究では，多くの場合に市販のモーションキャプチャシステムが用いられる．また，環境に設置した複数のカメラの撮像からロボットに搭載した赤外 LED によるマーカを検出する手法[16] が提案されている．本研究は床面を向いた 1 台のカメラのみで物体やロボットの位置検出を行うものであり，文献 16) と比べてロボットや物体が物陰に隠れやすい一方，カメラの撮像上のスクリーン座標がそのままロボットや物体の動く実世界座標系となるため，カメラ間の相対位置を検出するキャリブレーションが不要であり，また，プログラマにとって座標系を直感的に把握しやすいという利点がある．

6.2　実世界指向のツールキット

HCI 研究において，Tangible User Interface や Physical Computing のように実世界に対して入出力のあるインタフェースの研究が急速に広まってきている．この分野では新しいデバイスをプロトタイピングして研究を効率的に進めるためのツールキットが必要とされており，はんだ付けや電子工作の知識がなくてもセンサやアクチュエータをソフトウェアから扱える Phidgets[17] やマイコンのプログラミングを簡単にする Gainer[18] や Arduino[19]，デバイスの試作とテスト，改良の反復過程を支援する d.tools[20] などが提案されている．ただし，これらのツールキットで作成した入力インタフェースで実現できるタスクはコンピュータの中の処理であることが多く，実世界への出力はディスプレイや触覚デバイス，スピーカなどを用いた情報提示やアクチュエータの単純な回転による効果にとどまっている．既存研究では机上の人形をコンピュータ制御で移動させることで可能になるインタラクション[21]

が提案されているが，既存のツールキットではこのようなアプリケーションを作成するのは難しい．我々のツールキットと移動ロボットを用いればこのようなアプリケーションを簡単に作成できる．一方，我々のツールキットは基本的にソフトウェアの技術であり，デバイス制作を支援する機能は持ち合わせていない．Phidgets を用いれば豊富なセンサを用いてロボットへの指示出しインタフェースを開発でき，また，Arduino を用いれば比較的簡単に移動ロボットを作成できる．すなわち，我々のツールキットは既存の実世界指向のツールキットと機能を補完し合う関係にあり，共用することによって互いの応用範囲を広げることができる．

6.3　ロボット用のユーザインタフェース

Human-Robot Interaction（HRI）の既存研究は，ロボットが社会に受け入れられていくためにユーザがロボットと対話する際なるべく心理的な負荷を感じないことを目標にした，社会心理学的な意味合いの強いものが多かった．また，ロボット工学の分野では，物体の把持や運搬といった具体的なタスクを実現するためにロボットを制御する研究が進められてきたが，タスクの指示自体は元よりあるものとしてハードコーディングされていることが多かった．したがって，具体的なタスクを想定したユーザインタフェースの提案は比較的少なく，かつてはキーボードやマウス，ジョイスティック入力を用いるのが一般的であった．その後，自然言語の音声認識と手のジェスチャ認識で動作モードを指示したり[22]，PDA のタッチパネルで一台のロボットの移動を指示したり[23]，タブレット PC に環境の地図の概要をスケッチして複数ロボットの移動を指示したりする[24] ユーザインタフェースの研究が提案されている．近年では，ユーザスタディで紹介した研究[11)–13)] のほか，非明示的なロボット操作のためのカードインタフェース[25] が提案されている．また，マルチタッチディスプレイ上で複数台のロボットを操作するために適したジェスチャ[26] も提案されている．これらのうち，多くのインタフェースが Andy を用いれば比較的簡単に実装できるだろう．HRI 自体新しい研究分野だが，ロボット用のユーザインタフェース研究はいっそう新しいいまだ黎明期の分野であり，我々のツールキットはこの分野への HCI 研究者の流入を後押しするものである．

6.4　ロボット用のツールキット

ロボット用のツールキットとして，既存研究では大きく分けて 3 種類が開発されてきた．以下 3 種類に関して詳述するとおり，我々のツールキットはいずれとも異なる目的を持っているが，将来的には連携することができるだろう．

まず，ハードウェアの機能を抽象化し，ネットワーク上のリソースを透過的に扱えるようにするミドルウェアがある．具体例として，Player[7]，Microsoft Robotics Developer Studio (MRDS)[27] や RT-Middleware[5]，Robot Operating System[28] などが有名である．我々の研究はこのように汎用的な分散ソフトウェアプラットフォームを目指すものではなく，あくまで最小限のハードウェアセットアップを PC 中心の小規模な中央集権型ネットワークで結ぶものである．ただし，我々のツールキットをこれらのミドルウェア上に実装することは可能である．将来的にはエンドユーザが持つ端末上でツールキットとアプリケーションが動作し，ミドルウェアに接続して家中のロボットを操作することが考えられる．

次に，ロボット工学で既知の自律アルゴリズムを実装し，新たな研究に活かせるようにした研究用ツールキット[6] がある．我々の研究の興味は自律アルゴリズムではなくユーザインタフェースにあり，これらのツールキットとは趣旨を異にする．我々の将来ビジョンは，これらの自律アルゴリズムはミドルウェア上でロボットのノード内で動作し，我々のツールキットはロボットが公開している高レベルな機能のインタフェースにアクセスするというものである．

最後に，プログラミングの導入としてロボットを用いて簡単な自律アルゴリズムを実装できる教育用ツールキットがある．たとえば，Visual Programming Language（VPL）[29] や簡単な命令セット[30] によってアプリケーションを組みやすくしているものがある．これらのツールキットは 1 台の PC 上でアプリケーションを開発，実行，デバッグでき，2.1 節デザイン指針で示したような Human-Computer-Robot Interaction のプロトタイピングに使うことができる．しかしこれらのツールキットでは，個々のロボットについてロボット本体に備わっているセンサの値に応じたローカルな動きしか設計できない．一方我々のツールキットは，床面上の二次元絶対座標系という実世界を表現する明確なモデルを提供する．MRDS のようにミドルウェアと一体となっているフルスペックの開発環境ならこのようなことが不可能ではないが，難しい．我々の研究においては，将来的に VPL を開発用インタフェースとして提供するなど，ツールキットの実用性を損なわない範囲で教育用途に応用を広げることはありうる．

7. おわりに

本稿では，俯瞰視点で環境を見るカメラとビジュアルマーカを用いて移動ロボットを使ったアプリケーションを開発できるソフトウェアプログラマ向けのツールキット Andy を提案した．プログラマはロボットの移動やロボットによる物体の運搬を GUI のオブジェクト移動と同様に座標値を引数にとる 1 回の API 呼び出しのみで指示でき，ロボットや物体の

位置変化を GUI のマウスリスナなどと同様にリスナで取得できる．ユーザスタディでは，ツールキットを使用した学生 11 グループのすべてがロボットを用いたアプリケーションの開発に成功した．また，すでに 3 つの HRI 研究がツールキットを用いて実装されており，ツールキットの有効性，実用性が確認できた．

これまでのロボット開発は，限られた専門家の手によって多機能多目的なロボットの完成を目指してきたケースが多い．一方我々のツールキットを用いれば，ロボット工学の前提知識を持たないソフトウェアプログラマが，具体的なユースケースに即してロボットアプリケーションをプロトタイピングできる．我々は，ユーザインタフェース研究で有用性が認められてきたプロトタイピングという開発手法をロボットにも適用することで，その実用化に貢献できるのではないかと考えている．Andy は，現在オープンソースプロジェクト Matereal [⋆1] の一部として公開されている．今後も，ユーザスタディで得られた知見を活かして取り組みを続けていきたい．

謝辞　本研究で実施したユーザスタディは，東京大学大学院で平成 20 年度開講された講義「メディア情報学」を履修した学生の協力のもと行われた．ここに記して感謝の意を示す．

参 考 文 献

1) Landay, J.A. and Myers, B.A.: Interactive Sketching for the Early Stages of User Interface Design, *Proc. 1995 Conference on Human Factors in Computing Systems*, pp.43–50 (1995).
2) Newman, M.W., Lin, J., Hong, J.I. and Landay, J.A.: DENIM: An Informal Web Site Design Tool Inspired by Observations of Practice, *Human-Computer Interaction*, Vol.18, No.3, pp.259–324 (2003).
3) MacIntyre, B., Gandy, M., Dow, S. and Bolter, J.D.: DART: A Toolkit for Rapid Design Exploration of Augmented Reality Experiences, *Proc. 17th Annual ACM Symposium on User Interface Software and Technology*, pp.197–206 (2004).
4) Sato, T., Nishida, Y. and Mizoguchi, H.: Robotic Room: Symbiosis with Human through Behavior Media, *Robotics and Autonomous Systems*, Vol.18, pp.185–194 (1996).
5) Ando, N., Suehiro, T., Kitagaki, K., Kotoku, T. and Yoon, W.K.: RT-Middleware: Distributed Component Middleware for RT (Robot Technology), *Proc. 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.3555–3560 (2005).
6) Montemerlo, M., Roy, N. and Thrun, S.: Perspectives on Standardization in Mobile Robot Programming: The Carnegie Mellon Navigation (CARMEN) Toolkit, *Proc. 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.2436–2441 (2003).
7) Gerkey, B.P., Vaughan, R.T. and Howard, A.: The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems, *Proc. 11th International Conference on Advanced Robotics*, pp.317–323 (2003).
8) OpenSLAM. http://www.openslam.org/
9) Kato, H. and Billinghurst, M.: Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System, *Proc. 2nd IEEE and ACM International Workshop on Augmented Reality*, pp.85–94 (1999).
10) Igarashi, T., Kamiyama, Y. and Inami, M.: A Dipole Field for Object Delivery by Pushing on a Flat Surface, *Proc. 2010 IEEE International Conference on Robotics and Automation*, pp.228–233 (2010).
11) Kato, J., Sakamoto, D., Inami, M. and Igarashi, T.: Multi-touch Interface for Controlling Multiple Mobile Robots, *Proc. 27th International Conference Extended Abstracts on Human Factors in Computing Systems*, pp.3443–3448 (2009).
12) Sakamoto, D., Honda, K., Inami, M. and Igarashi, T.: Sketch and Run: A Stroke-based Interface for Home Robots, *Proc. 27th International Conference on Human Factors in Computing Systems*, pp.197–200 (2009).
13) Sugiura, Y., Sakamoto, D., Withana, A., Inami, M. and Igarashi, T.: Cooking with Robots: Designing a Household System Working in Open Environments, *Proc. 28th International Conference on Human Factors in Computing Systems*, pp.2427–2430 (2010).
14) Brooks, R.A.: Robust Layered Control System for a Mobile Robot, *IEEE Journal of Robotics and Automation*, Vol.2, No.1, pp.14–23 (1986).
15) Kim, J.O.: Real-time Obstacle Avoidance Using Harmonic Potential Functions, *IEEE Trans. Robotics and Automation*, Vol.8, No.3, pp.338–349 (1992).
16) Hada, Y. and Takase, K.: Multiple Mobile Robot Navigation Using The Indoor Global Positioning System (iGPS), *Proc. 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol.2, pp.1005–1010 (2001).
17) Greenberg, S. and Fitchett, C.: Phidgets: Easy Development of Physical Interfaces through Physical Widgets, *Proc. 14th Annual ACM Symposium on User Interface Software and Technology*, pp.209–218 (2001).
18) Gainer. http://gainer.cc/
19) Arduino. http://arduino.cc/
20) Hartmann, B., Klemmer, S.R., Bernstein, M., Abdulla, L., Burr, B., Robinson-Mosher, A. and Gee, J.: Reflective Physical Prototyping through Integrated Design,

⋆1 Matereal – A Toolkit for Prototyping Interactive Robot Applications, http://mr.digitalmuseum.jp/

Test, and Analysis, *Proc. 19th Annual ACM Symposium on User Interface Software and Technology*, pp.299–308 (2006).
21) Aoki, T., Matsushita, T., Iio, Y., Mitake, H., Toyama, T., Hasegawa, S., Ayukawa, R., Ichikawa, H., Sato, M., Kuriyama, T., Asano, K., Kawase, T. and Matumura, I.: Kobito -Virtual Brownies-: Virtual Creatures Interact with Real Objects and Real People, *ACM SIGGRAPH 2005 Emerging Technologies* (2005).
22) Rogalla, O., Ehrenmann, M., Zollner, R., Becher, R. and Dillmann, R.: Using Gesture and Speech Control for Command a Robot Assistant, *Proc. 11th IEEE International Symposium on Robot and Human Interactive Communication*, pp.25–27 (2002).
23) Fong, T., Thorpe, C. and Glass, B.: PdaDriver: A Handheld System for Remote Driving, *Proc. 8th International Conference on Advanced Robotics* (2003).
24) Skubic, M., Anderson, D. and Blisard, S.: Using a Hand-drawn Sketch to Control a Team of Robots, *Autonomous Robots*, Vol.22, No.4, pp.399–410 (2007).
25) Zhao, S., Nakamura, K., Ishii, K. and Igarashi, T.: Magic Cards : A Paper Tag Interface for Implicit Robot Control, *Proc. ACM Conference on Human Factors in Computing Systems*, pp.173–182 (2009).
26) Micire, M., Desai, M., Courtemanche, A., Tsui, K.M. and Yanco, H.: Analysis of Natural Gestures for Controlling Robot Teams on Multi-touch Tabletop Surfaces, *Proc. ACM International Conference on Interactive Tabletops and Surfaces*, pp.41–48 (2009).
27) Microsoft Robotics Developer Studio. http://www.microsoft.com/robotics/
28) Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R. and Ng, A.: ROS: An open-source Robot Operating System, *Proc. Open-Source Software Workshop of the 2009 International Conference on Robotics and Automation* (2009).
29) Cox, P.T. and Smedley, T.J.: Visual Programming for Robot Control, *Proc. 1998 IEEE Symposium on Visual Languages*, pp.217–224 (1998).
30) Blank, D., Yanco, H., Kumar, D. and Meeden, L.: Avoiding the Karel-the-Robot Paradox: A Framework for Making Sophisticated Robotics Accessible, *Proc. 2004 AAAI Spring Symposium on Accessible, Hands-on AI and Robotics Education* (2004).

(平成 22 年 6 月 28 日受付)
(平成 23 年 1 月 14 日採録)

加藤　　淳

2008 年より JST ERATO 五十嵐デザインインタフェースプロジェクト研究補助員．2009 年東京大学卒業後，同大学院情報理工学系研究科修士課程進学．ACM CHI'09 Student Research Competition 1st Place 等受賞．実世界タスクを指示するユーザインタフェースおよびその開発手法の研究に従事．

坂本　大介（正会員）

2008 年公立はこだて未来大学大学院システム情報科学研究科博士（後期）課程修了．博士（システム情報科学）．東京大学にて日本学術振興会特別研究員 PD を経て，2010 年 4 月より科学技術振興機構 ERATO 五十嵐デザインインタフェースプロジェクト研究員．ACM/IEEE HRI2007 Best Paper Award，情報処理学会論文賞，Laval Virtual 2010 Grand Prix du Jury，ACM ACE2010 Best Paper Silver Award 等受賞．人とロボットを含む情報環境とのインタラクション設計に関する研究に従事．

稲見　昌彦（正会員）

1999 年東京大学大学院工学系研究科博士課程修了．博士（工学）．東京大学助手，科学技術振興機構さきがけ研究者，MIT コンピュータ科学・人工知能研究所客員科学者，電気通信大学教授等を経て，2008 年 4 月より慶應義塾大学大学院メディアデザイン研究科教授．科学技術振興機構 ERATO 五十嵐デザインインタフェースプロジェクトグループリーダー，日本バーチャルリアリティ学会理事等を務める．IEEE Virtual Reality Best Paper Award，文化庁メディア芸術祭優秀賞，情報処理学会論文賞等各賞受賞．

五十嵐健夫（正会員）

2000 年東京大学大学院工学系研究科博士（工学）．2002 年同大学院情報理工学系研究科講師，2005 年助教授．2007 年より JST ERATO 研究総括．学術振興会賞，SIGGRAPH 若手科学者賞等受賞．専門はユーザインタフェースおよびグラフィクス．