

コミュニケーションモデルに基づいたパーソナルロボットのアプリケーション調停機構

小林 亮博^{†*} 久米 出[†] 上野 敦志^{††} 河野 恭之[†]
木戸出正継[†]

Mediation Architecture of Personal Robot's Applications
Based on Communications Model

Akihiro KOBAYASHI^{†*}, Izuru KUME[†], Atsushi UENO^{††}, Yasuyuki KONO[†],
and Masatsugu KIDODE[†]

あらまし 本論文はユビキタス環境に適応するパーソナルロボットのためのミドルウェアアーキテクチャを提供する。本アーキテクチャは、ロボットを通じて環境に特化した情報サービスを行う Environment-oriented Applications (EA) を環境ごとにロードし、ロボットごとに備わった特有の動作を表出する Familiarity-oriented Application (FA) と共存させる。独立に開発されたこれら二つのアプリケーションはセンサやアクチュエータを共有するため、ロボット出力の連続性や応答性が低下しユーザ-ロボット間のコミュニケーションを阻害する。そこで本研究では、1 台のロボット上で二つのアプリケーションを同時に利用するユーザを想定し、Information Unit (IU), Communication Stream (CS), Communication Channel (CC) の 3 種類のコミュニケーションモデルを構築した。本研究はモデルに基づき二つのアプリケーションの出力を調停するミドルウェアを設計し PRIMA と名づけた。本論文では調停結果がユーザに与える影響を検証し、PRIMA の性能を評価した。

キーワード パーソナルロボット, 人間-ロボットインタラクション, ロボットミドルウェア, 対話戦略

1. ま え が き

本論文は、パーソナルロボットが個々にもつ親和的な挙動を損なうことなく様々な環境に特化したサービスを提供することを目指し、パーソナルロボットが訪問先の環境からアプリケーションをロードし、ロボット自身もつアプリケーションと共存させるソフトウェアアーキテクチャを提案する。近年のロボット研究では家庭やオフィスといった日常環境への応用が着目されており、図 1 に例示する親しみやすい外見をもったロボットが開発されている [1] ~ [5]。これらのロボットはパーソナルロボットやホームロボットと呼ば

れており、主に (1) 来客の案内や家電の管理等の情報サービスを行う情報化環境における知的インタフェースの分野や [6], [7], (2) 親しみやすいペットやパートナーとしてユーザとコミュニケーションを行うエンターテインメント分野での需要が見込まれている。そのためパーソナルロボットは厳密な制御や認識能力よりも、特徴的な外見や振舞い等の人に親しみやすい身体表現に重きをおいて開発されている [3] ~ [5]。

近い将来環境の情報化が進み、公の環境に訪れた

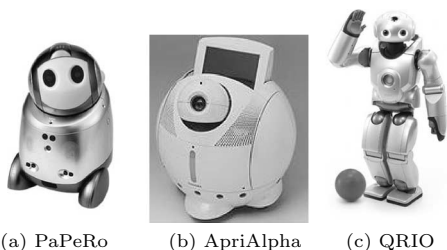


図 1 パーソナルロボットの具体例
Fig. 1 Examples of personal robots.

[†] 奈良先端科学技術大学院大学情報科学研究科, 生駒市
Graduate School of Information Science (NAIST), Nara Institute of Science and Technology, Keihanna Science City, Ikoma-shi, 630-0192 Japan

^{††} 大阪市立大学工学部, 大阪市
Faculty of Engineering, Osaka City University, 3-3-138 Sugimoto, Sumiyoshi-ku, Osaka-shi, 558-8585 Japan

* 現在, NICT 独立行政法人情報通信研究機構

ユーザに対し携帯端末へ環境ごとの情報サービスを配信する、ユビキタス情報化社会が到来すると考えられる。そのとき、ユーザ個人のパーソナルロボットが携帯端末の代わりにサービスを提供することで、ユーザは慣れ親しんだロボットから身体を伴った情報サービスを受けることが可能となる。例えば図 2 は、雑談や、写真撮影や、ダンス等の機能をもったパーソナルロボットが、ユーザに同伴し展覧会の会場に訪れた様子を示している。このロボットは、先導による会場案内や、身振り手振りを交えた展示の説明といった情報サービスを、展示会場からロードし実行する。

一方で、パーソナルロボットの親和性はロボット固有の振舞いや会話に依存するため、環境からロードしたサービスを実行中にも、ロボットが本来もつ動作を挟むことで、ロボットの親和性を保つことが望ましい。例えば図 2 においては、(i)~(iii) と (iv)~(vi) を適切に統合することにより、写真撮影や情動表現を合間に挟んだ展示案内や解説といった、ロボットの個性を保った情報サービスの実現を、本研究は目指している。本研究はこれらを実現するためサービスの開発者に対し、パーソナルロボットがもつデバイスの多様性や、独特の仕草や言葉のやり取りの多様性を隠すミドルウェア Personal Robots' Intermediating Mediator for Adaptation (PRIMA) を提案する [8], [9]。

2. では、PRIMA の調停におけるコミュニケーションモデルの必要性を述べ、アーキテクチャの概要とモデルが満たすべき目標を挙げる。3. では、複数のアプリケーションを同時に利用するユーザのコミュニケーションモデルを構築し、4. では、モデルの実現方法と

それに伴うアプリケーション開発者への制限を述べ、5. ではモデルの効果を検証する。

2. アプリケーションの調停

2.1 PRIMA 上で実行されるアプリケーション

図 3 は、PRIMA を利用する 3 種類の開発者 (EA 開発者, FA 開発者, ハードウェア開発者) の関係を示す。以降本論文では、単に「ユーザ」と記してある場合は、ロボットの所有者でハードウェア・ソフトウェアを利用するがプログラム開発等には関与しない人間を指す。Environment-oriented Application (EA) は、各環境サーバ上あるいはロボット上にロードされ実行されるプロセスであり、例えば図 2 の (iv)~(vi) のように、環境に特有の情報をロボットの身体を通じてユーザに提供する。PRIMA は、ハードウェアの詳細な知識をもたない EA 開発者に対し、ハードウェアアクセスのための共通インタフェースを提供することで、EA の開発を可能にする。

本研究は、EA に対し共通インタフェースの出力結果を保証するため、パーソナルロボットがもつべき最低限の機能を定義し、ハードウェアの仕様を設計した。設計時には、パーソナルロボットの機能を、ロボット間で共通の部分と固体に特有の部分とに分離した。パーソナルロボットは、家庭向けに小型で廉価になるよう設計されている。またその用途は、主にコミュニケーションに絞られており、物体の操作は重要視されていない。その意味では、パーソナルロボットが使用するデバイスや機能は図 4 のように標準化すべきである。図 4 は、EA の代表例として建物内の案内を取り上げ機能ごとに分類し、各機能が利用するデバイスを示す。本研究は、デバイスの詳細を知らずとも利用可能な最小限の単位を共通インタフェースとし、移動・対話・人間発見・自己位置同定・顔の動作の機能をパーソナ

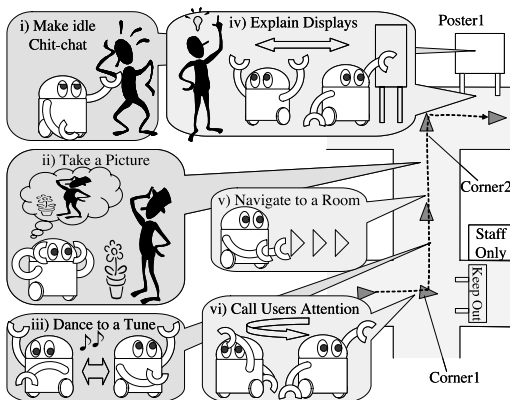


図 2 ユビキタス環境中のパーソナルロボットの行動制御
Fig. 2 Interaction control of a personal robot in an ubiquitous environment.

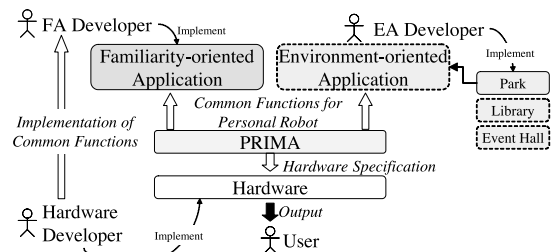


図 3 3 種類の開発者と PRIMA との関係
Fig. 3 Relations between developers with PRIMA.

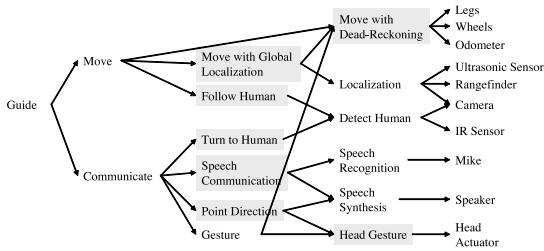


図 4 パーソナルロボットの機能分類
Fig. 4 Functions of personal robots.

ルロボットが最低限もつべきハードウェアの仕様とした。ハードウェア開発者は、それらの機能を実装することで共通インタフェースの出力を担保する。例えば「10 cm 前進」といった共通インタフェースによる要求を、車輪で移動するか歩行するかはハードウェア開発者が自由に決定できる。そのため PRIMA は、ハードウェアの詳細を知らない EA 開発者に対し、抽象度の高いインタフェースの使用を推奨している。

一方で Familiarity-oriented Application (FA) は、パーソナルロボットに固有のインタラクションや家庭内で日常的に用いるサービスを提供するプロセスであり、EA と同じく共通インタフェースで PRIMA と通信を行い、図 2 の (i) ~ (iii) に示す例題を実現する。EA 開発者と異なり、FA 開発者はハードウェアの詳細を知ることが可能である。その情報を利用し、FA が PRIMA を介さずにハードウェアへの特権的なアクセスを行った場合、EA の挙動がユーザに対して不自然になる可能性が高い。そのため本研究では、FA が必ず PRIMA を介してハードウェアにアクセスすると仮定する。しかし、FA が独特のインタラクションを実装する際、共通の機能では不十分な可能性がある。そこで PRIMA はハードウェア開発者に対し、「10 cm 前進」等の共通インタフェースへ速度・加速度のオプション追加をサポートする。

FA と EA の開発者は互いのアプリケーションを把握できない。そこで本研究は、FA と EA は独立したプロセスで動作するエージェントとし、FA と EA が互いに隠れいされ内部情報を共有せずにデバイスへアクセスするアーキテクチャを提案する。PRIMA はこれらのアクセスをマルチタスク OS のように実行時にスケジューリングする。通常の OS が競合解消の対象としているリソースは、抽象化されておりユーザに対し競合を容易に隠れいできる。しかし、本ミドルウェアが競合解消を行う対象は身体をもったロボットの動

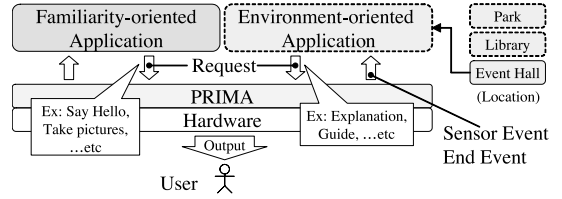


図 5 提案するパーソナルロボットの調停アーキテクチャ
Fig. 5 A Mediation architecture for personal robots.

作であるため、単純なプロセススケジューリングでは人間とロボットのインタラクションを阻害することになる。そのため PRIMA は、2.2 で述べるポイントを踏まえた上で、アプリケーションとユーザ間のインタラクションを制御する。これを本研究は調停と呼ぶ。

2.2 ロボットのインタラクション制御

多くの対話ロボットは、有限状態機械 (FSM) でインタラクションを管理している。FSM は、現在の状態とセンサ入力やアクチュエータの出力結果により次の状態を決定する。ロボットは通常、FSM の状態からとるべき行動を決定しており、プログラムによる明示的な記述若しくは学習により FSM を獲得する。対話制御の FSM を人手で記述する場合は、コードのメンテナンス性を高めるため対話状態を階層化する場合が多い。Expo.02 の RoboX [10] は、線形的に動作するメインの案内シナリオと、任意のタイミングで発生するイベント駆動のサブシナリオを分けている。HERMES は「サービス受付」や「案内」等をタスクとして大きな対話状態とした上で、タスク内で現れる「確認」「つづり」「オブジェクト特徴」等の認識文法が再利用可能な状況をサブタスクとして対話状態に定義している [11]。Nursebots は、部分観測マルコフ過程で対話状態をモデル化しており、認識が不確実な環境下においても、頑健に終了状態へ向かう対話が可能である [12]。また、ロボットへは未適用であるが、人間同士の対話コーパスを正事例として学習し、対話制御の FSM を抽出する研究も行われている [13]。

[10] ~ [13] の対話制御手法は、プログラムが対話状態を有限個にモデル化する、若しくは学習データ集合が実行時の入出力を包含することを前提としている。しかし、FA と EA は独立に開発されるため、両者を含み図 2 に示すような、フレキシブルなインタラクションを実現する FSM を生成することが困難である。そこで本研究は独立な FSM をもつ FA と EA を調停可能なアーキテクチャを提案する。図 5 に示すように

本アーキテクチャ上で EA と FA は、互いの情報共有なしに入力イベントや出力の終了イベントのみで状態変化を行う。ただし前提条件として、それぞれの開発者は他アプリケーションによる割り込みの可能性を認識した上で開発するものとする。

ここで各環境はある単位ごとに区切られた場であり、その場で必要なサービスを来訪したロボットに提供する唯一の EA とする。すなわち、図 2 の (iv) ~ (vi) のような機能を内包した EA が、1 個の建物のようなある程度の広さをもった単位ごとに一つ存在し、図 2 の (iv) ~ (vi) のどれが動作するかは EA が決定する。(iv) の開発者と (vi) の開発者は情報の共有が可能であり、タスク依存の知識を用いたインタラクション制御が可能である。一方で FA と EA は互いの内部状態が不明であるため、ユーザにとって一貫性がある出力を PRIMA が保証する。このとき FA と EA の開発者は、トレードオフの関係にある要素、連続性と応答性に着目してアプリケーションを開発する必要がある。

連続性 FA と EA が共存する環境では、アプリケーションの連続実行は制限される。本研究では、一方のアプリケーションの要求を途切れずに実行することでユーザが感じるロボット動作の一貫性を連続性と呼ぶ。連続性が低下すると、ロボットの状況認識性能やプログラムの演出意図の表出、ユーザによるロボット動作の理解を阻害する。

応答性 上記想定環境では、ロボットの反射的な行動は制限される。本研究では、ユーザ入力や環境変化に対する反応の速さを応答性と呼ぶ。応答性には、アクセスがあったことを検知したという情報をユーザに返す acknowledge の側面と、実際に処置をして内容的応答を返す側面がある。対話ロボットがもつ問題の一つとして内部状態の分かりにくさがあり [14], acknowledge の遅れはこの問題を助長するため、本研究では特に acknowledge の側面をもつ出力の応答性の向上を目指す。

FA と EA はそれぞれ自分が必要とする連続性と応答性を要求し、PRIMA はアプリケーションの与えた情報に基づき、ユーザから見て連続性と応答性のバランスのとれた出力を実現する。このときアプリケーション開発者は、以下の共通認識をもつとする。

- (1) 出力タイミングは PRIMA が最終決定する。
- (2) 位置制御等の抽象度の高い行動は PRIMA による自由な軌道生成を許す。
- (3) 連続性と応答性を PRIMA に要求できる。

パーソナルロボットと人間とのコミュニケーションが高精度の時間管理を必要とする状況は少ないため、(3) により連続性と応答性が満足できれば FA と EA の開発者は (1) と (2) の制限を許容すると期待できる。しかし FA や EA の開発者が、相手アプリケーションとのバランスをとって連続性や応答性を記述することは困難である。そこで PRIMA がコミュニケーションモデルをもち、開発者の記述が最小限で済むシステムを目指す。本研究は、パーソナルロボットが家庭外で所有者以外とコミュニケーションを行ったり、所有者から遠く離れる機会は少ないとし、ユーザとロボットが 1 対 1 のコミュニケーションモデルを構築する。

3. コミュニケーションモデル

人間-ロボット間のコミュニケーションは、グリーンサイズが小さい順に、局所的な動作、話題や話の流れ、コミュニケーションの開始と終了、の 3 種類の連続性を必要とする。本研究は、それぞれに対応するコミュニケーションの構成単位を、(3.1) Information Unit (IU), (3.2) Communication Stream (CS), (3.3) Communication Channel (CC) と呼ぶ。本章ではそれぞれの場合について応答性とのトレードオフを議論し、望ましい調停戦略を構築する。

3.1 Information Unit の導入

一般的に、ユーザがロボット行動を理解したり、ロボットがユーザ入力を認識する上で、時間軸上に分節すべきでない最小源の単位がある。これを本研究では Information Unit (IU) と呼ぶ。ロボット行動に関する IU の例として、対話中の 1 発話や一連のジェスチャ等が挙げられる。これらの行動は開始から終了まで連続的に再現されなければ、ユーザはロボットの行動を理解できない。ユーザ入力に関する IU の例として、ロボット発話に対するユーザ応答が挙げられる。ユーザ応答を待ち受ける間に他のアプリケーションによる割り込み、例えば音楽演奏やダンス動作が発生した場合、ロボットの音声認識性能は低下する。

上記のように IU は、人間-ロボット間のコミュニケーションで連続性に関する制約が最も強い単位であると仮定している。そのため PRIMA は IU 内の連続実行を保証し割り込みを許さない。大きすぎる IU は応答性に悪影響を与えるため、PRIMA は、アプリケーションが定義する IU に制限を与えている。4.1 において、IU の実装方法及び各開発者に対する制限の詳細を述べる。

3.2 Communication Stream の導入

従来のロボットアプリケーションでは他のアプリケーションが動作しないことを前提に開発されている。そういうアプリケーションでは、複数の発話や動作の間に因果関係が暗黙的に埋め込まれている。それらの連続性が維持されない場合、ユーザはロボットの行動に一貫性を感じられなくなる。そのため EA と FA が同時に出力を要求している場合は、それぞれの IU を交互に実行するような、平等にリソースを振り分けるスケジューリング手法は好ましくない。本研究では、同一アプリケーションの IU 間に含まれる因果関係や共通する話題を Communication Stream (CS) と呼び、その連続性の維持を目指す。

本研究は 2.2 の応答性の項で述べたように、acknowledge の側面をもつ出力の応答性の向上を目指す。PRIMA は、アプリケーションにユーザ入力や環境変化を伝えた後、アプリケーションが最初に要求した行動がこの側面をもつと仮定し、CS の維持より優先する。ユーザはロボットの行動の因果関係を推測できるため、環境変化によって CS が不連続になる場合は、一貫性は失われぬ。一方で、acknowledge を行った後の内容的応答の応答性は CS の維持によって保たれる。

図 6 は PRIMA が CS を単位として調停を行う際の動作を示す。横軸は時刻、黒い長円は既に行われた IU を表し、白い長円はアプリケーションの要求によりこれから実行予定の IU を表す。長方形は CS を示し、CS 内の IU の実行順序はアプリケーションが指定しているものとする。時刻 $t = T_0$ において、PRIMA は App0 の Stream0 を実行中である。 $t = T_1$ のときユーザ入力や環境変化に対し App1 は Stream1 を生成する。PRIMA は応答性を優先し、Stream0 を中断し Stream1 を開始する。PRIMA は終端まで Stream1

を維持し、 $t = T_2$ の Stream1 終了時に Stream0 を中断位置から再開する。4.3 では、CS の切れ目と、応答性が必要な CS の二つを決定する方法を述べる。

3.3 Communication Channel の導入

3.2 の戦略に従った場合、実行中の CS の優先と、ユーザ入力や環境変化に起因する割り込みが行われる。しかし、重要な対話の最中に鳴るアラーム等、ロボットの行動内容によっては、図 6 の $t = T_1$ で、Stream0 に割り込むべきでない状況や、 $T_1 < t < T_2$ で Stream1 を連続すべきでない状況が存在する。これは、ユーザにとって必要な連続性や応答性が、IU の性質やタスクの内容によって異なるためである。本研究は、ユーザごとの個人差やタスクの内容に依存しない、ロボットのコミュニケーション行動が本質的にもつ優先度を定義し、調停に利用することを目指す。ロボットは、ユーザから見て、コミュニケーションを打ち切ろうとしているように見える行動をとることがある。本研究では、そういった行動が以下二つの特徴をもつことから優先度の定義に利用した。

- 既存のロボット IF 分類で状況の特定が可能
- 特定した状況・優先度が一般性をもつ

ロボットの視線や対人位置は「ロボットが人間に注目しているか」等のユーザの主観評価に相関関係があるといわれており [15]、ユーザの空間把握の補助 [16] や、多人数対話における話者決定 [17] に利用されている。以上の身体を伴うコミュニケーションが行われるとき、人間とロボットの間で構築される関係を、本研究では Communication Channel (CC) と呼ぶ。CC は、ロボットや人間の音声による呼びかけ・ジェスチャ・視線を合わせる等の行動で接続され、別れの挨拶や移動等で切断される。センサ情報による CC の状況推定はハードウェアの性能に大きく依存し、すべてのパーソナルロボットの調停基準として採用するには好ましくない。そこで本研究は、ロボットのとらうとする行動から CC の状態を推定する。本来 CC はアプリケーションの内容に依存する単位であるが、CC の状態をアプリケーション開発者が記述すると、アプリケーション開発者の労力が増大するため、PRIMA は、既存のロボットインタフェースの分類を用い、アプリケーションによる明示なしで CC の状態を推定する。

本研究では、大きな移動を伴う IU によって CC が切断されるとし、CC の接続状態の維持するよう調停を行う。ロボットの移動インタフェースはその移動の抽象度によって分類されることが多く、上記 IU が特定し

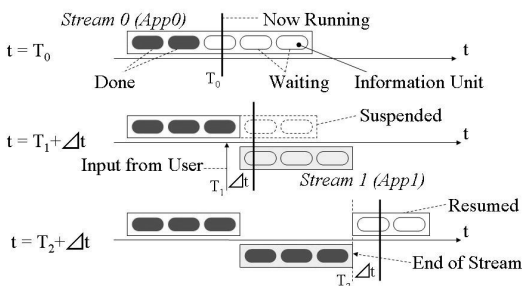


図 6 Communication Stream の調停
Fig. 6 Mediating Communication Stream.

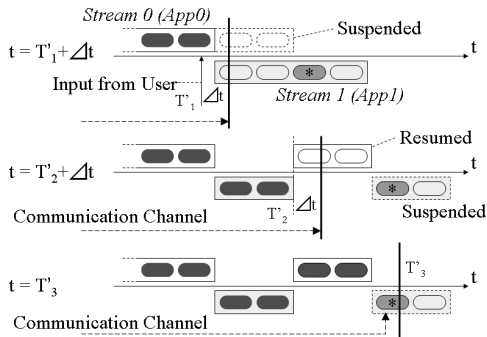


図 7 Communication Channel の調停
Fig.7 Mediating Communication Channel.

やすい, また, ユーザとロボットが遠く離れている状態では [15] ~ [17] にあるようなコミュニケーションを行うことは困難であることから, 多くのユーザ・状況に当てはまる. 例えば, 図 7 の $t = T_1 + \Delta t$ は, Stream0 の対話中にユーザ入力によって Stream1 が割り込んだ状況である. このとき Stream1 内の IU(*) が人間から遠く離れるような動作であった場合, Stream0 が終了する前に CC が切断されてしまい, 対話の終了前にロボットの側から対話を打ち切ったという印象をユーザに与える. 以上のような一貫性の欠如した出力を防ぐため, PRIMA は CC を維持する IU を優先する. 4.2 では, 具体的な分類と IU の優先順位を調停に反映する方法について述べる.

4. 調停戦略の実現

本論文は, 連続性の粒度を三つ (IU, CS, CC) に分類しそれぞれの粒度における連続性をできるだけ維持する, という調停戦略を提案する. 具体的には, 他の動作を割り込ませることにより, それぞれの各粒度における連続性が損なわれる場合には割り込みの優先度を下げる. 一方, この連続性に対してトレードオフの関係にある応答性の実現戦略がある. 応答性の維持は, 応答性が必要な状況をイベントとして定義することを義務づけ, アプリケーション開発者が宣言的に優先度を高めることで実現する. PRIMA は, EA と FA の 2 種のプログラムが投入された状態において, 並列に実行できない場合には以下の (1) ~ (4) の順序に従い動作を制御する.

- (1) 必ず連続性を維持すべき単位 (IU) は, アプリケーション開発者が一つにまとめて事前に記述する.
- (2) 必ず連続性を維持すべき単位 (CC) につい

表 1 Command 仕様
Table 1 Command specification.

Command
•Command Type
•Motion Arguments
•Before Node
•Additional Arguments

表 2 Command Type 例
Table 2 A list of Commands' Types.

Command Type	Occupying Device
Active Sensing Human	Human Detection, Locomotion, Localization
Face Human	Human Detection, Locomotion
Q&A	Speaker, Speech Recognition
Tracking Human	Human Detection, Locomotion
Face Human and Speak	Human Detection, Locomotion, Speaker
(i) Turn and Speak	Locomotion, Localization, Speaker
Set Direction	Locomotion, Localization
Turn	Locomotion
Set Speed and Time	Locomotion
Stop Moving	Locomotion
Speak	Speaker
Move Head (Pan, Tilt)	Neck
Point by Hand	Arm
Gesture	Arm Neck
Topological Move	Locomotion, Localization
(ii) Set Position	Locomotion, Localization
Set Position and Direction	Locomotion, Localization

ては, これを切断する IU は PRIMA が事前に規定しており, 実行時に PRIMA が該当 IU を優先しないことにより連続性が維持される.

(3) 応答性が要求される場合はアプリケーション開発者がイベントを定義し, 実行時に PRIMA はこれを (4) よりも優先して割り込み処理を行う.

(4) 応答性を優先することで, 場合によっては連続性が損なわれてもかまわない単位 (CS) は, アプリケーション開発者が各 IU に分割して記述し, 各 IU の系列を親子関係として事前に指定する.

4.1 Information Unit の実装

アプリケーション開発者による IU 定義は, 記述負荷が大きく, また応答性の低下を招きやすい. そこで本研究では, PRIMA は IU のひな形として IU の機能をもつ Atomic Behavior (AB) を用意し, アプリケーションは AB を生成する共通インタフェース Command に与える引数で IU の範囲を特定する.

アプリケーションは, 表 1 に示す要素をもった構造体を PRIMA に送信する. Command Type は動作の種類を示し, いくつかの例を表 2 に掲載した. Motion Argument は, Command Type ごとに定義された動作のパラメータであり, 例えば移動の Command では距離や速度の値が代入される. Additional Argument は FA のみが用いる拡張的なオプションである. Command の実装を知っている FA 開発者は, この値を利

用してハードウェアの状態を細かく指定し、ハードウェアが本来もつ性能を引き出す。

本研究は、アプリケーション開発者が理解しやすいよう、基本的に1個のCommandが1個のABを生成するものとして、各Command Typeの設計を行った。2.1で述べたように、本研究はデバイスの詳細に関する情報なしに利用可能で最小限の単位をCommandに定義した。「一発話」等の意味の区切りごとにCommandを呼び出すという方法は、アプリケーション開発者がIUを記述しやすく、Commandが発行したABが必要十分な粒度をもつと期待できるため、1ABを1IUとして扱う。

一方で内部にフィードバックループをもつCommandは、ループ1回ごとにABを生成するよう義務づけている。PRIMAは、EAが多様なロボット上で動作するよう、固定座標による移動先指定といった抽象度が高いCommandを用意している。一方でABは応答性に対し強い制限をかけるため、IUの定義に反しない限り小さい方が望ましい。そのためPRIMAでは、ミドルウェアが軌道生成の自由度をもつ位置制御等のCommandは、ループ1回の移動ごとにABを生成する。この場合も1AB=1IUとする。

例外的に、複数のABを一つのIUとして扱う場合がある。例として、複数のCommandを組み合わせたジェスチャ表現がそれにあたる。ABはPRIMAが汎用性を重視して定義した粒度の小さい単位であるため、Commandの引数記述だけではアプリケーションが望むIUを定義できない。PRIMAは、アプリケーションが複数のCommandをリストでくりPRIMAに一括要求する場合にのみ、複数のCommandをIUに定義できるとした。この記述方法は、状況に応じて異なるCommandを要求するような一連の行動全体をIUに定義できないといった制限をもち、応答性に悪影響を与えかねない大きすぎるIUが定義されることを防いでいる。ジェスチャの多くは決まりきった動作を要求するので、上記の制限の中でもIU記述は十分可能であると期待できる。実際には、複数のABへの参照をもつ仮想的なABを生成することで、PRIMAはこの定義を実装している。4.2, 4.3においては、ABはすべてIUとしての粒度をもつとする。

4.2 Communication Channelの実現

図2の行動(iv)をABに分割したものを図8に示す。各ノードはABを表し、有向リンクはAB間の実行順序を表し、分岐は並列実行を表す。このようにAB

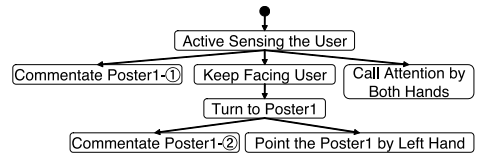


図8 Atomic Behavior 木
Fig. 8 Atomic Behavior tree.

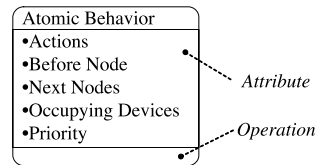


図9 Atomic Behavior クラスの属性
Fig. 9 Attributes of Atomic Behavior class.

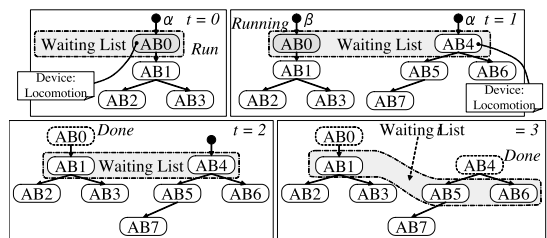


図10 ワーキングメモリの動き
Fig. 10 State of working memory.

間の関係を木構造で表しノード単位で実行制御を行えば、行動内に存在する順序関係を守りつつ調停を行いやすい。本研究はこの木をAB木と呼び、PRIMAの内部表現に用いる。

各ABは図9に示す属性値をもつ。Actionは、ハードウェアインタフェースの識別子と引数を格納しており、各TypeのCommandがMotion Argumentをもとに決定する。PRIMAはCommand Typeごとに占有デバイスと優先度を定義しており、各Commandは発行するABにその値をコピーする。Before NodeはAB木の親ノードを指し、アプリケーションがCommandを通して指定する。Next NodesはAB木の子ノードを指し、PRIMAが探索して決定する。

図10は時刻t=0にEAがAB木(α)を要求し、時刻t=1にFAがAB木(β)を要求した場合での、PRIMAのワーキングメモリ(WM)の様子を表している。PRIMAはAB木の受理時とABの実行終了時にWMの更新を行う。PRIMAはAB木を受理すると根ノードの子ノードを待機リストに加え、実行中の他のABと競合がない限り実行を開始する。例え

ば $t = 1$ では, AB0 と AB4 は占有デバイスと同じデバイスをもつことから競合が検知され, AB4 は実行されない. AB の実行終了時は, 終了した AB の子ノードを待機リストに加え競合判定を行う. 例えば時刻 $t = 2$ では, AB1 を待機リストに加え AB1 と AB4 の間で競合判定を行い, 競合がない場合は両方を並列実行する. AB の優先度に基づいて実行順序を決定する. $t = 2$ は AB4 を優先した例であり, AB4 終了後その子ノードを待機リストに加え, そこでまた判定を行う.

3.3 で述べたように, PRIMA は CC を維持する IU を優先する. PRIMA の Command は表 2 に示すように, CC を切断するもの (i) と, それ以外 (ii) の 2 種類に分類されている. 本研究ではユーザの位置を考慮しない移動が CC を切断すると仮定し, 絶対座標移動, トポロジカル移動を (i) に分類した. Command の優先度は 2 段階あり, (i) の優先度を低, (ii) の優先度を高とする. PRIMA は, 発行元 Command の優先度を継承した AB を以上のように調停することで, 3.3 のモデル実現を目指す.

4.3 Communication Stream モデルの実現

3.2 で述べた調停戦略を実現するために, PRIMA は, CS の終端判断と終端までの連続性保証と各 AB の応答性判断を行う必要がある. PRIMA は, AB 木の葉ノードを CS の終端とみなす. 4.2 で述べた処理において, 優先度が等しい AB に対しては親ノードの終了時刻がより新しい AB を優先することで, CS の連続性を保証する. 以上の終端を定義する理由は以下の 2 点である.

- PRIMA は AB 木のノード間の連続性のみ考慮すればよく, ルールを簡単にできる
- 順序関係や依存関係をもっている行動ごとに木を生成するという表現方法がプログラマに理解しやすい

PRIMA は, 親ノードの実行終了までに要求された AB に対してのみ, 親とのリンクを形成し木の同一性を保証する. PRIMA が WM 上にない未知の AB に対し連続性を保証する場合, 新たな AB が要求されるまで他方のアプリケーション要求はすべて中断しておく必要がある. 以上の手段をとると応答性を著しく阻害しユーザ評価を下げるため, PRIMA は, 実行終了時に子ノードをもたない AB を AB 木の葉ノードとし, 以降の要求を別の木とする.

4.2 で述べたように, 各 AB の親ノード項はアプリ

ケーションが決定する. PRIMA はアプリケーションに対し以下 2 種類の親ノード記述法を提供する. 経路に沿った展示紹介等の現在の状態から将来にとる行動を決定可能な場合は前者を用い, センサ情報からリアルタイムに動作を変更する場合に後者を用いる.

- 木構造をリストで表現した複数の Command を, 一括して PRIMA に送信する
- 過去に送信した Command オブジェクトへの参照を親ノードとして, 子ノードの引数に与える

伝統的なロボットプログラミングでは, ユーザ入力や環境変化に対する反射的な行動を多用する. 本研究ではそのような反射的行動が応答性を最も必要とするとして仮定する. ユーザ入力や環境変化のタイミングをとらえる手段として, JAVA [18] や CORBA [19] 等で用いられているイベント通信の利用が一般的であるため, PRIMA はプログラミング上の制限として, 応答性が必要な状況をイベントとして定義することを義務づける. 各アプリケーションは, 自身のイベントハンドラ上で呼び出した Command に対しフラグを立てる機能をもち, RPIMA は最新のイベントハンドラ上で最初に要求された AB 木の根ノードを優先する. 整理すると, 図 10 の $t = 2$ のとき以下の順に判定を行う.

- (1) CC 維持のため優先度が高い AB を優先する.
- (2) 双方の優先度が等しい場合, 最新のイベントハンドラ上で最初に要求された AB 木の根ノードであるかどうかを判定する. AB4 が上記の条件に当てはまる場合, 応答性を高めるため AB4 を優先する.
- (3) 双方が (2) の条件に当てはまらない場合, 親ノードの終了時刻が新しい AB0 を優先し CS の連続性を保つ.

5. 検証

5.1 ビデオを用いたアンケート

本章は, 3. で設計した戦略に沿って調停したロボット出力を被験者がユーザの観点から観察することにより, 戦略のユーザに対する有効性を評価した. 検証用に表 3 に示すアプリケーション群を設計した. FA はあるパーソナルロボットの FA がもつ六つの機能を示し, 便宜上 FA1, FA2 と呼称する. EA はそれぞれ異なる環境でロードする EA であり, 独立したアプリケーションである. 本研究では, 上記の組合せやユーザの応答の変化を想定して表 4 に示す調停例 10 例を作成し, NEC 製パーソナルロボット PaPeRo (図 1(a)) を用いて動作の様子をビデオに撮影した.

表 3 アプリケーション例
Table 3 Application example.

FA	1	Express Emotions
	2	Walk Following User
	3	Dance to a Tune
	4	Take a Picture
	5	Set Video Timer
	6	Scheduler
EA	1	Guide a Museum and Explain Displays
	2	Search Books in a Library
	3	Navigate to a Room & Tell Messages from Hosts

表 4 被験者が検証した調停例
Table 4 Test sample.

EA	FA	内容
1	2	道案内中にユーザの要望がありユーザの追跡を行ったが、追跡中に交差点を通過した
1	2	道案内中にユーザの要望がありユーザの追跡を行ったが、追跡中にメッセージ通信を受けた
1	2	道案内中に人間の追跡をしようとしたが、その前処理の人間を発見する最中に館内メッセージの通信が入った
1	5	道案内中にビデオの録画予約を行ったが、予約の最中に館内メッセージの通信が入った
2	2	展示解説の最中に、ユーザがロボットに追従を要求した
2	1	展示解説の最中にロボットをなでてやった
2	3	展示解説の最中にダンスを開始させるイベントが発生した
2	4	写真を撮るため解説1の最中にロボットのタッチセンサに触れた
3	5	図書検索の途中でビデオ予約を入れた
3	6	図書検索の途中でバス乗車の予定時刻15分前であることを知らされた

0 U	スリープ中にタッチセンサに触れる	<イベント>
1 R	「何か御用ですか？」	
2 U	「ついてきて」	
3 R	「どちらにいらっしゃいますか」と言いながら、特殊行動「回転+観察」を繰り返す	
4 R	人間を発見する	<イベント>
5 R	「見つけました。行きましょう。ついて行きます」と言って、追跡を開始する	
6 R	人間の位置に対して適切な距離と方向を維持しながらついていく	
7 U	タッチセンサに触れる	<イベント>
8 R	「ついて行くのはもういいですか」	
9 U	「はい」	
10 R	スリープ状態になる	

特殊行動「回転+観察」:
人間が発見できるまで静止する。一定時間内に発見できない場合は回転して視野を変更してからもう一度人間の発見を試みる

図 11 FA2 の出力例
Fig. 11 Output example of FA2.

被験者は、個々のアプリケーションが単独動作する様子とその調停例を示す映像と、映像を説明するテキストを閲覧し、被験者の主観に基づき調停例の行動順を評価した。図 11~図 14 は被験者が閲覧した資料の一部である。図 11 は FA2 (ユーザを発見した後追跡)、図 12 は EA4 (来客を目的地まで案内・来客に対しメッセージがあれば再生) が単独で動作する様子を示す。被験者は同時に、以上のテキストに示される内容のビデオを閲覧する。図 14 は、FA2 と EA4 の調停例として被験者に与えたビデオの一部であり、図 13 は、そのビデオを説明するテキストである。それぞれの写真番号は図 13 の行に対応する^(注1)。

表 5、表 6 は、図 13 の例題において、各アプリケーションが PRIMA に要求した Command を示す。

0 R	建物内に入り、アプリケーションが起動	
1 R	「どちらに御用ですか？」	
2 U	「木戸出先生」	
3 R	「こちらです。参りましょう」	
4 R	ゴールに向かってトポロジカル移動を開始する	
5 R	管理者からのソケット通信を受信する	<イベント>
6 R	いったん停止して発話「メッセージを受け取りました」	
7 R	「木戸出です。会議が長引いたので5分ほど遅れます。」	
8 R	「返信はありますか」	
9 U	「はい。木戸出先生の部屋でお待ちしています。」	
10 R	「分かりました。お伝えします」と言って返信内容を管理用サーバに送信する	
11 R	ゴールに向かってトポロジカル移動を再開する	
12 R	交差点に差し掛かる	<イベント>
13 R	交差点で曲がる必要がある場合は「木戸出先生の部屋はこちらです」と言いながら、顔を進行方向を指し、進行方向に回転	
14 R	ゴールに到着する	<イベント>
15 R	「到着しました」	
16 R	アプリケーションが終了する	

図 12 EA3 の出力
Fig. 12 Output example of EA3.

0 R	建物内に入りアプリケーションが起動	
1 R	「どちらに御用ですか？」	
2 U	「木戸出先生」	
3 R	「こちらです。参りましょう」	
4 R	ゴールに向かってトポロジカル移動する	
5 U	道中のポスターに気を引かれる	
6 U	廊下を移動中、ロボットの頭をふれる	<イベント>
7 R	ロボットは停止する「何か御用ですか？」	
8 U	「ついてきて」	
9 R	「どちらにいらっしゃいますか」と言いながら、特殊行動「回転+観察」をm回繰り返す	<イベント>
10 R	人間を探す間「管理者からのソケット通信を受信する	
11 R	立ち止まり発話「メッセージを受け取りました」	
12 R	「木戸出です。会議が長引いたので5分ほど遅れます。」	
13 R	「返信はありますか」	
14 U	「はい。木戸出先生の部屋でお待ちしています。」	
15 R	「分かりました。お伝えします」と言って返信内容を管理用サーバに送信する	
16 R	特殊行動「回転+観察」をm回繰り返す	<イベント>
17 R	人間を発見する	
18 R	「見つけました。行きましょう。追跡します」	
19 R	人間追跡の開始	
20 U	人間の位置に対して適切な距離と方向を維持しながらついていく	
21 U	ポスターの位置まで戻り、数秒ポスターを眺める	
22 R	しばらくクラッキングを続ける	
23 U	ポスターを理解したので、追跡を終えるためタッチセンサに触れる	<イベント>
24 R	「追跡をやめますか」	
25 U	「はい」	
26 R	「追跡を終了します」	
27 R	ゴールに向かってトポロジカル移動する	<イベント>
28 R	ゴールに到着	
29 R	停止し「到着しました」と言う	

図 13 FA2 と EA4 の調停例 (テキスト)
Fig. 13 Mediated output (Text).

番号は要求が送信されたタイミングを表し、図 13 の行番号に対応している。複数の Command が記入された項は、4.3 で述べたようにアプリケーションが Command 間の実行順序を木構造で表現して送信していることを示す。Event 項は、アプリケーションがイベントに対する行動としてフラグを立てたことを示す。もしロボットが IU という排他単位をもたない場合、7~9 行目で FA がユーザとコミュニケーションをとろうとする間、ロボットは案内を続行し、ユーザから離れていってしまう。図 13 の 7 行目と 11 行目では、4.3 のルールに従い、PRIMA はイベント駆動の出力を CS の維持よりも優先した。このルールがない場合、

(注1): ビデオから分かりにくい通信関係のイベントは、ビデオに字幕を付けることで発生時を被験者に示した。



図 14 FA2 と EA4 の調停例 (Video)
Fig. 14 Mediated output of FA2 and EA4 (Video).

7 行目でロボットは人間に頭を触れられても反応せず案内を続けてしまう。また、CS 維持のルールがなく EA と FA を均等に出力するアルゴリズムであった場合、対話や人間発見処理の間にロボットは案内を割り込ませようとする。15 行目では、EA4 が CC を切断する動作 (表 5 : No.14) を要求したので、PRIMA は 4.2 のルールに従い FA2 (表 6 : No.8) を優先した。ここで CC 維持のルールがない場合、メッセージ応答の後 CS 維持ルールにより案内が再開してしまい、ロボットは人物追跡が続行できない。

被験者は以上のビデオを見ながら、用紙の調停例に対し被験者の主観で実行順が不自然な個所を記述する。個々の行は以下 (2), (3) を除いて AB を示しており、今回の実験では被験者には行と行の入換え操作のみ許した。ロボットの出力を変化させる要因となるユーザ発話以外のロボット入力をイベントと定義し、図 11, 図 12 の右側に示している。また、FA に由来する行には色を付けて表示している。今回の実験では、被験

表 5 EA4 から PRIMA への要求
Table 5 Requests from EA4.

No.	Root Command(arg), Leaf Command1(arg), ... ;	Event
0	Q&A("どちらに御用ですか?");	
2	Speak("こちらです。参りましょう"); Topological Move(EA DEF POSITION KIDODE ROOM); Stop_Moving();	
10	Speak("メッセージを受け取りました"); Speak("木戸出です。会議が長引いたので5分ほど遅れます。"); Q&A("遅延はありますか?");	○
14	Speak("分かりました。お伝えします"); Topological Move(EA DEF POSITION KIDODE ROOM); Stop_Moving();	
28	Speak("到着しました");	○

表 6 FA2 から PRIMA への要求
Table 6 Requests from FA2.

No.	Root Command(arg), Leaf Command1(arg), ... ;	Event
6	Q&A("何か御用ですか?");	○
8	ActiveSensingHuman("どちらにいらつしやいますか?");	
17	Speak("見つけました。行きましょう"); Tracking Human();	○
23	Stop_Moving(); Q&A("追跡を終了しますか?");	○
25	Speak("追跡を終了します");	

者の順序の入換えに対し以下の制限を設けた。

(1) が付与された行は複数の AB を含むため例外的に行中に割り込み可能

(2) [が付与された区域は単一の AB より構成されるため割り込み不可能

(3) イベント発生タイミング変更の禁止

5.2 結果

今回の実験では 12 の被験者から延べ 120 試行の回答を得た。集計の結果 120 試行中 76% で、行動順変更は求められず PRIMA の出力に肯定的な結果を得た。しかし大きく分けて 2 種類の場面で、被験者による順序変更が多く指摘された。

一つは CS の終端判定に関する部分である。否定的回答であった 24% の試行のうち 56% が CS の終端判定に関係していた。4.3 で述べたように PRIMA は、親ノードの実行終了までに AB が要求されなければ、CS を切断する。そのため、入力待ち等でアプリケーションに制御が返り PRIMA の WM が空となる場合に、連続性が保証されない。例題では、EA3 でアプリケーションが図書検索の結果を待つ部分と、FA5 でアプリケーションがビデオサーバに問い合わせる部分で、他アプリケーションに割り込まれている。ユーザはロボットが話の途中で急に話題を変えたと感じてしまい、否定的な結果となった。これに対して、アプリケーションが CS の切断を防ぐ目的で実行する予定の行動を前もって PRIMA に予約した場合、PRIMA は AB の実行完了後即座にその子ノードを実行しようとするため、イベント発生までロボットを静止させるといった特殊な行動を要求する必要がある。これは開発

者に負担となるため、CS の明示的・暗黙的な連続性要求を簡潔に記述する方法が必要とされる。

二つ目はコミュニケーションを開始する場面である。PRIMA は CC の維持についてのみ責任をもち、接続の責任はプログラマが負う。そのため図 11 の 1 行目のように、アプリケーションが対面動作を省いた対話を要求した場合、3.3 で定義した CC の接続前に対話が開始されてしまう。25%の被験者が (19 行目) 人間との対面行動を完了してから (9 行目) 発話を行うべきとの回答を行った。

5.3 考 察

今回行ったアンケートでは、回答の際に設けた制限以外でロボットの挙動で不自然な点を、被験者に自由に記述させた。回答をまとめ、前節の結果と合わせ、本システムに今後必要と思われる機能を以下に挙げる。

- (1) CS の明示的・暗示的要求
- (2) ロボット状態の復帰
- (3) AB 間への割り込み

(1) 理想的には CS の範囲が、アプリケーションにより明示的に定義されている状態が望ましい。現在明示的な記述方法として、Command に CS 要求という新たな属性値を加え、その値が真の場合、PRIMA は次に要求される Command まで CS 維持を保證する、といった仕組みを検討中である。始端と終端の両方を記述する方法に比べれば簡単であると期待しているが、「必須ではないが、連続実行されれば望ましい部分」をプログラマが理解し記述することは困難である。そこで、本研究では上記の値を定義済みのポーリング機能をもった Command を検討している。例えば「Q&A」は、ロボットが発話してからユーザの回答を待ち、更に回答に対するアプリケーションの反応があるまで、CS を維持する。将来的には、こういった入力待ちを含むコミュニケーション行動の類型化が必要となる。

(2) ロボットは人間とコミュニケーションをする際に様々な状態を満たす必要があるが、EA と FA が共存する環境では、アプリケーションが望んだ状態の維持が他アプリケーションの出力により阻害される。本研究は IU 及び CS の保証により、実行中のアプリケーション内における状態維持を目指した。また、中断中のアプリケーションも必要とする状態として CC に着目し、その維持を試みた。以上の試みによりロボットにある程度の一貫性を与えることに成功したが、これらは状態遷移を抑止する手法であり応答性を損な

う原因となっている。また応答性を極度に失うことを避けるため状態遷移の完全な抑制は行っていない。

そのため本研究では行動抑制は最小限にとどめ、ロボットの状態復帰による問題解決を目指す。割り込み前の状態への復帰を保證することにより、連続性に関する制約が減少し応答性の向上が期待できる。対話の流れなど時系列を含む状態の復帰は困難が予想されるため、ターゲットとするのは現在のセンサ情報で確定する状態である。このとき問題となるのは、復元すべき状態と復元すべきでない状態の区別である。例えばロボットの位置は、もとの位置にまで戻った方が良いのか、戻る必要がないのか区別する必要がある。将来的には、プログラマの明示的宣言なしに、AB 木の親ノードを探索することでアプリケーションが求める状態を推定する。

(3) 本研究は 3.1 で述べたように、IU の連続性に関する制約を重視し、IU 内では割り込まないとした。しかし 4.1 にあるとおり、IU の実装である AB の大きさは開発者の記述に依存し、応答性を大きく損なう可能性がある。アンケートにおいても、ロボット発話や音声認識の最中に割り込みたいという意見が多く見られた。この問題に対し、AB 間での割り込みを許し、割り込まれた場合は AB を最初からやり直すことで解決を目指す。その場合、IU を先頭から再開することでユーザに対して不自然な印象を与えずに復帰できる、という仮定が必要になる。しかし多くのプログラマは、AB 間では割り込まれないことを前提に開発を行っているため、中途半端な中断でユーザを惑わせる可能性があり、単純には解決できない。

6. む す び

本研究は、独立に開発された複数のアプリケーションをパーソナルロボットが同時に実行することで、環境に応じた情報サービスの親和的な実行を目指した。本研究ではこれら二つの行動を、アプリケーション開発時の独立性を維持したまま、しかもロボットの振舞いに不自然さをもたらさずに統合するため、アプリケーション出力を調停するミドルウェアを用いる。この調停の基準として、人間-ロボットコミュニケーションにおける連続性と応答性に着目した。本研究では、Information Unit, Communication Channel, Communication Stream の 3 レベルのコミュニケーションモデルからなる調停戦略により、連続性と応答性のトレードオフの解決を目指した。戦略実現に際し

ては、アプリケーションプログラムの負担の軽減を目指して、ミドルウェアを設計した。最後に、本システムの出力の様子を撮影したビデオを被験者に見せることにより、調停結果の評価を行った。本システムの出力結果はおおむね肯定的な評価を受けたが、調停戦略に改善点も見受けられた。今後、これらの改善を行う場合プログラムの労力とのトレードオフをとっていくことが重要となる。

文 献

- [1] <http://www.sony.co.jp/SonyInfo/QRIO/>
- [2] 吉見 卓, “快適生活支援ホームロボット - ロボット情報家電コンセプトモデル aprialpha の開発,” 日本機械学会誌, vol.106, no.1021, pp.909-911, 2003.
- [3] 藤田善弘, “NEC におけるパーソナルロボットの開発,” 日本ロボット学会誌, vol.20, no.7, pp.16-19, 2002.
- [4] 石黒 浩, 日浦亮太, “コミュニケーション支援ロボットビジネス,” 日本ロボット学会誌, vol.20, no.7, pp.12-15, 2002.
- [5] 景山浩二, 石田健蔵, “エンタテインメントロボットビジネス,” 日本ロボット学会誌, vol.20, no.7, pp.8-11, 2002.
- [6] J. Buhmann, W. Burgard, A. Cremers, T.H.D. Fox, F. Schneider, J. Strikos, and S. Thrun, “The mobile robot rhino,” AI Magazin, vol.16, no.1, pp.31-38, 1995.
- [7] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, “MINERVA: A second-generation museum tour-guide robot,” Proc. 1999 IEEE International Conference on Robotics and Automation (ICRA1999), pp.1999-2005, 1999.
- [8] A. Kobayashi, I. Kume, A. Ueno, Y. Kono, and M. Kidode, “A robot programming model for mediating between familiarity-oriented behaviors and environment-oriented behaviors,” Proc. 7th of the World Multi-Conference on Systemics, Cybernetics and Informatics (SCI2003), pp.295-302, 2003.
- [9] A. Kobayashi, A. Ueno, I. Kume, Y. Kono, and M. Kidode, “Robot middleware architecture mediating familiarity-oriented and environment-oriented behaviors,” Proc. 5th IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA2003), pp.544-551, 2003.
- [10] P.J. Prodanov, A. Drygajlo, G. Ramel, M. Meisser, and R. Siegwart, “Voice enabled interface for interactive tour-guide robots,” Proc. 2002 IEEE Int. Conference on Intelligent Robots and Systems (IROS), pp.1332-1337, 2002.
- [11] R. Bischoff and V. Graefe, “Dependable multimodal communication and interaction with robotic assistants,” Proc. 11th IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN2002), 2002.
- [12] N. Roy, J. Pineau, and S. Thrun, “Spoken dialog management for robots,” Proc. 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000), P00-1013, 2000.
- [13] 岡本昌之, “キャッシュを用いた漸次的 pdfa 学習と対話型エージェントへの適用,” 信学論 (D-I), vol.J86-D-I, no.8, pp.524-531, Aug. 2003.
- [14] 小林哲則, “会話ロボットの実現に向けて,” 信学技報, 2003HCS-1, 2003.
- [15] H. Mizoguchi, T. Sato, K. Takagi, M. Nakao, and Y.H. Hatamura, “Realization of expressive mobile robot,” Proc. 1997 IEEE International Conference on Robotics and Automation (ICRA1997), pp.581-586, 1997.
- [16] 小野哲雄, 今井倫太, 石黒 浩, 奈良良平, “身体表現を用いた人とロボットの共創対話,” 情処学論, vol.42, no.6, pp.1348-1358, 2001.
- [17] Y. Matsusaka, T. Tojo, and T. Kobayashi, “Conversation robot participating in group conversation,” IE-ICE Trans. Inf. & Syst., vol.E86-D, no.1, pp.26-36, Jan. 2003.
- [18] Sun Microsystems, “Java(tm) 2 platform standard edition 5.0 api specification.” <http://java.sun.com/j2se/1.5.0/docs/api/>
- [19] OMG, “CORBA.” <http://www.corba.org>
(平成 16 年 12 月 10 日受付, 17 年 4 月 9 日再受付)



小林 亮博 (正員)

1999 京大・工学卒。2005 奈良先端科学技術大学院大学博士課程単位取得退学。同年、NICT 独立行政法人情報通信研究機構専攻研究員、現在に至る。ヒューマンロボットコミュニケーションの研究に従事。情報処理学会、人工知能学会、ロボット学会各

会員。



久米 出

1991 京大・理学卒。1997 東工大理工学研究科情報科学博士課程単位取得退学。現在奈良先端科学技術大学院大学助手。ソフトウェア保守・発展に興味をもつ。情報処理学会、日本ソフトウェア科学会、ACM、IEEE Computer Society 各会員。



上野 敦志

1991 東大・工・航空卒．1996 同大大学院航空宇宙工学専攻博士課程単位取得退学．同年，奈良先端科学技術大学院大学情報科学研究科助手．2004 大阪市立大学工学部情報工学科講師，現在に至る．博士（工学）．ロボットの学習，自律システムの研究に従事．

人工知能学会会員．



河野 恭之（正員）

1989 阪大・基礎工・情報工卒．1994 同大大学院基礎工学研究科博士後期課程了．同年（株）東芝入社．同社関西研究所研究主務などを経て，2000 奈良先端科学技術大学院大学情報科学研究科助教授．マルチモーダル理解，音声対話，ウェアラブルインタフェースなどヒューマン・マシンインタラクションの研究に従事．情報処理学会，人工知能学会，日本認知科学会，IEEE-CS，ACM 各会員．博士（工学）．



木戸出正継（正員：フェロー）

昭 45 京大大学院工学研究科修士課程了．同年東京芝浦電気（現，東芝）総合研究所入社．同社総合企画部，関西研究所，東芝アメリカ社を経て，平 12 奈良先端科学技術大学院大学情報科学研究科教授．京大工博．パターン認識，ロボットビジョン，ヒューマンインタフェース，ウェアラブルコンピュータに関する研究に従事．情報処理学会フェロー，IAPR（国際パターン認識協会）フェロー，本会業績賞，高柳記念奨励賞，などを受賞．情報処理学会関西支部長，本会理事，MVA 国際ワークショップ組織委員長，IEEE ウェアラブルコンピュータ国際シンポジウム実行委員長などを歴任．