

ロボット制御プログラム開発プラットフォーム “SotaPlatform” の開発

Development of "SotaPlatform" –Program Development Platform for Sota Robot

小林 稜, 神田 智子

Ryo Kobayashi and Tomoko Koda

大阪工業大学情報科学部

Department of Information Science and Technology, Osaka Institute of Technology

Abstract: 本研究では Vstone 社の普及型対話ロボット Sota の, 新規開発プラットフォーム "SotaPlatform"を開発した. 想定ユーザーは VstoneMagic で Sota の制御経験があり, Java アプリケーションの作成経験があるユーザーである. "SotaPlatform"は VstoneMagic にはない3つの機能がある. 第1に, OpenJTalk を用いた発話が可能であること, 第2に, Sota の3D モデルを表示しアニメーションさせることで, 実機の Sota を伴わずオフライン環境でのポージング作業が可能であること, 第3に, 3D モデルで指定したポーズや発話内容を xml 形式で保存し, 指定することで別の実行ファイルでも同様の発言及びポーズが再利用できることが可能であることである.

1. はじめに

近年, Pepper [1]やロボホン [2]といった対話型コミュニケーションロボットの導入が増加している [3]. これらのロボットが活躍する主なフィールドは, 対面業務を人に代わって行う場面である. 具体的な例として受付業務や面接業務をロボットにさせるケースが挙げられる [4,5]. このような対話ロボットの商用利用の増加には, グローバル社会への遷移に伴った受付の外国語対応や, 人件費削減を目的として多くの企業がロボットの導入意欲を高めていることが要因の一つとして考えられる. また, メーカーから SDK や制御プログラム開発用のプラットフォームが提供されているロボット製品の増加に伴い, エンドユーザーによって容易にロボットの制御プログラムの開発が可能になったことも対話ロボットの商用利用増加の要因として考えられる.

それに伴い, ロボットを制御する開発者がより必要とされると考えられることから, ロボットに実装する制御プログラムの開発効率の向上が求められる. 従って, ユーザーの開発効率を考慮した制御プログラムの開発プラットフォームが必要であると考えられる.

本研究では Vstone 社の普及型対話ロボット Sota を対象に, 新規開発プラットフォームを提案及び開発し, 既存の制御プログラムの作成手法よりも開発効率が向上することを期待する.

2. 既存のロボット制御プログラム と開発の問題点

Sota を制御するには, Java コーディングもしくは付属の制御プラットフォーム VstoneMagic [6]を用いる必要がある. コーディングで制御プログラムの開発を行う場合は, 外部プロセスや API を呼び出すことが可能である. しかしコーディングによる Sota の制御には, ロボット特有の制御ライブラリや基本的な Java 言語の理解が要求される為, ユーザーのノウハウが伴わない場合は基本的に VstoneMagic を使って制御する.

VstoneMagic では命令ブロックを順に配置し, ビルドすることで Sota の制御が可能となる. VstoneMagic の利点はユーザーのプログラミング能力を問わず容易に Sota を制御できる点である. また, VstoneMagic で作成したプログラムを直接修正し, 外部 IDE を使うことでデバックすることもできる.

その一方で, VstoneMagic には3つの問題点が存在していると考えられる. まず第1に, 発話命令を作成するには Vstone 社が提供する有料サービス Sota クラウドのライセンスを取得した上で, 音声合成機能の利用が前提となっていることである. その為ライセンスを取得していない場合は音声合成機能を利用できない. ライセンスを適用せずに同様の機能を実

装するには、自作もしくは外部で取得した API やソフトウェアを、作成したプログラムから呼び出すといった工夫が必要なため、Sota の仕様に関する知識や開発経験が少ないユーザーの開発難易度が上がる。

第 2 に、VstoneMagic には 3D モデルによるシミュレート機能が存在しない。したがって、意図した動作をロボットに行わせる場合は必ず実機を要する。実機が存在しない状態で Sota の動きを指定する場合、数値でポージングを指定する為、デバックするまでは Sota に設定した動きを確認できない。同じ環境下で並行して複数のユーザーがプログラムを作成するケースの現場では、複数の実機が必要となる為、開発コストの増加につながると考えられる。

第 3 に、ポーズ単位での保存ができない為、ポーズの再利用ができず開発効率が下がる点である。

3. ロボット制御プログラム開発プラットフォームの概要

提案する新規開発プラットフォーム"SotaPlatform"では前章の 3 点の課題を解決する。想定ユーザーは VstoneMagic で Sota の制御経験があり、Java アプリケーションの作成経験があるユーザーとする。

第 1 の解決策として、発話機能に関しては日本語テキストに対応しているオープンソースの音声合成ソフトウェア OpenJTalk [7] を Sota にインストールしておき、外部プロセスとしてシェルスクリプトを呼び出すメソッドを、生成コードに記述することで解決する。第 2 の解決策として、SotaPlatform では Sota の 3D モデルを表示し、モデルをアニメーションさせることで、実機の Sota を伴わずオフライン環境でのポージング作業を可能にする。最後に、第 3 の解決策として、3D モデルで指定したポーズや発話内容を xml 形式で保存し、指定することで別の実行ファイルでも同様の発言及びポーズが再利用できることを可能にする。

SotaPlatform の開発環境は、ソフトウェアは eclipse, Unity2017 1.1.f1(64bit), Blender2.79, プログラミング環境は C#, Java, Python 2.7 である。

4. SotaPlatform の機能

4.1 SotaPlatform のファイル構成

SotaPlatform では、発話とポーズ、一連の動作を作成して xml 形式のファイルに保存が可能で、それぞれのファイルには、出力コードの一部や発話内容、又はパラメータが記述されている。作成した xml フ

ァイルは、図 1 のコード作成画面にある右側のドロップダウン群から実行させたい発話や動作ファイルを指定することで、降順で設定しているテンプレートがソースコードに記述される。各発話及びポーズ、一連のファイルの構成は図 2 のようになっている。発話と動作を指定するファイルは act ファイルと名付け、ポーズを ps ファイルと名付けた。

4.2 発話機能

プラットフォーム内の入力フォームに発話内容と発話名を記述することで、発話内容が xml 形式の act ファイルとして保存される。作成した命令は図 3 のドロップダウンで選択することができる。

発話機能を実装するにあたり Sota 内に OpenJTalk をインストールした。実行フローは図 4 の通りになっており、発話内容を引数に発話メソッドから OpenJTalk を介して音声合成を行うスクリプトを外部プロセスとして実行することで合成結果を wave ファイルとして出力する。このスクリプトでは音声合成で使用する音声データと辞書など必要なオプションを指定している。また、音声の生成と同時に wave ファイルを再生することで発話を行う。



図 1 SotaPlatform コード作成画面

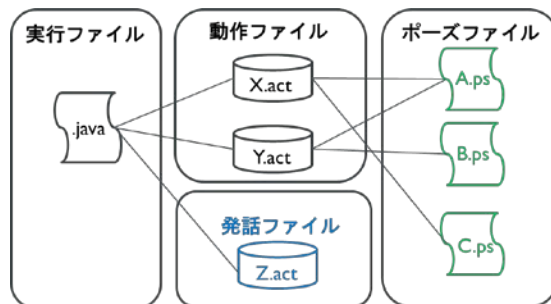


図 2 SotaPlatform のファイル構成



図 3 発話作成画面



図 4 発話機能実行フロー

4.3 動作およびポーズ作成機能

ポーズ作成は、ポーズ/動作の作成画面に切り替え、図 5 の画面右側のスライダーを動かすことで、画面背部に描画されている Sota の 3D モデル (Blender で作成) の関節調整を行う。次にユーザーが定めた 3D モデルのポーズを維持した状態で保存名を付けることで、各関節レベルの情報を含んだ xml 形式の ps ファイルとして保存することができる。

次に動作の作成について示す。本プラットフォームで示す“動作の作成”とは、ポーズを複数指定することで、1つのメソッドに一連のポーズを保存する事を表す (図 6)。図 5 の画面下部では作成済みのポーズを左から実行順に指定することで act ファイルとして動作を作成することができる。このとき遷移するポーズ間の速度をミリ秒単位で指定することが可能となっている。また各ブロックを押下するとドロップダウンで指定しているポーズが 3D モデルにアニメートされる為、ポーズの内容を確認できる。

4.4 コードのプレビュー機能

4.1 でも述べたように、コードの作成画面右側のドロップダウンで指定した動作や発言を反映したコードを出力する前にプレビューとして確認できる (図 7)。発話や動作は main メソッドで記述され、どの記述が指定した発話動作であるかコメントから確認で

きる。尚、出力されるコードにインポートさせたいライブラリやメソッドがある場合は、バイナリの、“lib.ini”と“post.ini”ファイルに各ライブラリとメソッドを記述することで、テンプレートメソッドとして作成されるコードに追記される。

4.5 SotaPlatform の想定利用例

Sota に簡単な動作と短い発話を複数回行わせたい場合に本プラットフォームを利用する。また出力後のコードは eclipse など制御用ライブラリのパスが通っている外部 IDE を用いてビルド及び Ant 実行で Sota の実機に転送する。問題がなければ同一ネットワーク内の Sota に SSH 接続を行い、コマンドラインから転送したプログラムを実行する。



図 5 ポーズ/動作作成画面

```
// (動作)両腕交互
public static void ryouude_pose(CSotaMotion m){
    //ポーズファイル名：右腕あげる
    ExcuteMotion(m,-1200,-275,300,275,500);

    //ポーズファイル名：左腕あげる
    ExcuteMotion(m,-300,-275,1200,275,500);
}

```

図 6 動作メソッドの出力部分



図 7 コードプレビュー画面

5. 提案開発プラットフォームの評価

価

本プラットフォームによる開発生産性の向上効果評価テストを行った。評価テストを行うにあたり、前述の想定ユーザーとして、Java アプリケーションの作成経験がある評価者4人に対し、事前に Sota クラウドを利用せずに VstoneMagic を用いるケースと、コーディングのみで Sota を制御させるケースの両方を与えた。いずれのケースも発話とポーズを1回以上含む実行ファイル作成を条件とした。

その結果、4人全員が両ケースとも30分以上の作成時間を要した。次に、各評価者に対して、SotaPlatform を使用し、前述のケースで作成したものと同様の条件で実行ファイルを作成してもらった結果、所要時間は4人とも15分前後であった。これらことから、提案プラットフォームによる開発効率の向上が認められたと考える。SotaPlatform の使用感に対するヒアリングにおいて、全員から 3D モデルによるポージング機能が作業効率の向上に貢献していたとして高い評価を得た。

6. おわりに

本研究では Vstone 社の普及型対話ロボット Sota の、新規開発プラットフォーム SotaPlatform を開発した。想定ユーザーは VstoneMagic で Sota の制御経験があり、Java アプリケーションの作成経験があるユーザーである。SotaPlatform は VstoneMagic にはない3つの機能がある。第1に、OpenJTalk を用いた発話させることができること、第2に、Sota の 3D モデルを表示しアニメーションさせることで、実機の Sota を伴わずオフライン環境でのポージング作業が可能になること。第3に、3D モデルで指定したポーズや発話内容を xml 形式で保存し、指定することで別の実行ファイルでも同様の発言及びポーズが再利用できることを可能になることである。

今後の課題として、プログラミングや IDE の利用に自信がないユーザー層にも SotaPlatform を利用できるようにするには、VstoneMagic と同様に、特定の IDE を使わず、プログラムのビルド及び実行を自動化させる必要があると考える。

謝辞

本研究の一部は、科研費「基盤(C)26330236, 17K00287」の交付を受けて実施した。

参考文献

- [1] 株式会社 SoftBank ロボット Pepper 製品情報
<https://www.softbank.jp/robot/consumer/products/>
- [2] シャープ株式会社 ロボホン <https://robohon.com/>
- [3] 株式会社矢野経済研究所コミュニケーションロボット市場に関する調査を実施
<http://www.yano.co.jp/press/press.php/001668> (2019.1.31 閲覧)
- [4] ロボスタ すべての病院やクリニックの待合室に Pepper
<https://robotstart.info/2017/06/30/pepper-biz-medival.html> (2019.1.31 閲覧)
- [5] 採用面接でロボットが面接官を務める企業で劇的效果 (株式会社リツアン STC)
http://biz-journal.jp/2017/04/post_18773.html (2019.1.31 閲覧)
- [6] ヴイストン株式会社 Sota 取扱説明書 VstoneMagic を使ってみる
<http://www.vstone.co.jp/sotamannual/index.php?VstoneMagic%E3%82%92%E4%BD%BF%E3%81%A3%E3%81%A6%E3%81%BF%E3%82%8B> (2019.1.31 閲覧)
- [7] OpenJTalk
<http://open-jtalk.sourceforge.net/> (2019.1.31 閲覧)