

# 分散ロボットシステムサービスにおける COP の適用検討

小野 建也<sup>1</sup> 菅谷 みどり<sup>1</sup>

近年多様化するロボット環境に対して、複数台のロボットによる分散したロボットサービスが考えられている。多様な環境を想定したプログラミングの手法の一つとしてコンテキスト指向プログラミングがある。また、ロボット開発のためのミドルウェアとして ROS がある。本研究では分散ロボットシステムにおける ROS を用いたコンテキスト指向プログラミングの分散システムへの適用の利点を検討する。

## 1. はじめに

近年ロボットの小型化、高性能化により、より身近なサービスや環境でロボットが利用されるようになってきている。ロボットを用いたサービスが多様化することにより、ロボットが動作する環境も、多種多様に広がってきている。また、こうした多様な環境の中で複数台のロボットを協調させるサービスも考えられる。

こうした広がりにおいて、ロボットは置かれている環境や状況に応じて振る舞いを変更し、最適な動作を選択することが期待される。置かれている環境や状況に適応するには、外界の情報を収集し、その変化により、振る舞いを変更する仕組みが必要である。コンテキスト指向プログラミング (Context-Oriented Programming)[1]は、こうした環境や状況(以下、コンテキスト)の変化に適応することを目的とした提案であり、近年、ロボットへの適用が検討されている[2]。一方、ロボットを分散化させた場合の効率的なデータ共有のアプローチとして ROS(Robot Operating System)を用いた方法が提案されている[3]。分散したロボットが、それぞれ自律、協調して動作するには、こうした効率的な分散化のためのミドルウェアを利用しつつ、動的に振る舞いを変更するコンテキスト指向プログラミングを取り込む等の方法が必要であると考えられるが、これらを用いた分散ロボットシステムの議論は十分になされておらず、統合した場合の有効性も示されていない。

そこで本研究は、分散ロボットシステムにおける課題について問題点を具体的な例を示して、これ

らの技術の適用可能性について検討することを目的とした。具体的には、まず、ROS を用いた場合の分散化の課題について述べ、その課題について、コンテキスト指向プログラミングの適用を行うことで解決する方法を検討するものとした。

## 2. 分散ロボットシステム

### 2.1 分散ロボットシステム例

分散ロボットシステムの例として以下のシステム「Order Anywhere」(図1)を提案する。

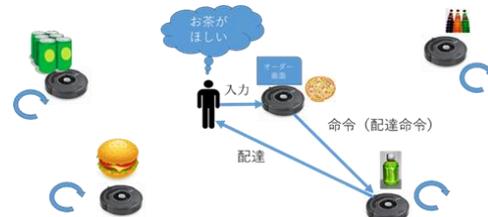


図1 分散ロボットシステムを用いた Order Anywhere サービス

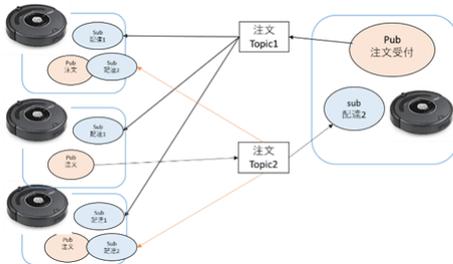
この分散ロボットを用いたサービスではテーマパークなどで巡回しているロボットがユーザ(客)から注文を受け付けており、注文がある場合、その情報が別のロボットに伝えられる。この際、別のロボットには商品の情報とユーザの座標情報が伝えられる。情報を受けたロボットは、現在商品を持っているか、届ける場所は何処かの判定を行い、最終的に商品を別のロボットがユーザに届けるというものである。

<sup>1</sup> 芝浦工業大学  
Shibaura Institute of Technology

## 2.2 課題

これに ROS を適用することによるデータ共有の効率化を考える。ROS はロボット開発のためのミドルウェアで、publisher/subscriber モデルのデータ共有方式を提供している。node 間（例のシステムの場合ロボット間）を topic 介してのデータ共有を行う。

一方、前述のシステムではクライアントに当たるロボットである node が増えると注文の情報である topic の数が増える。そうするとロボット 1 台の publisher に対して、すべてのロボットが個別に subscriber を持つ必要がある（図 2）。そうすると 1 台のロボットが多数の subscriber を持つことになり、クライアントであるロボットのシステムの複雑さ、負荷、が大きくなると考えられる。



## 3. 提案

### 3.1 COP

本提案では、ROSにおいて膨大な topic を制御する際に複雑化するプログラムの課題に対して、COP を用いた解決策を提示することを目的とする。COP ではコンテキストに応じた処理を、レイヤという単位でまとめ、レイヤをコンテキストに応じてアクティブ/ディアクティブさせることで動作を変える [2]。

### 3.2 COP によるトピック制御

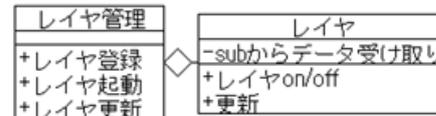
例においてコンテキストは各 topic から subscriber が取得したデータや、ユーザからの注文などがある。このコンテキストに応じたレイヤを判定して topic へのアクセスを制限する。例えば注文を受け付けているときは他のロボットがどんな注文を受けているかは必要な情報であるので注文受付のレイヤを実行しつつ、他のロボットが注文を受けたかを受信しなければならない。しかし、注文の配達中であれば注文受付のレイヤと配達の要請が

あったかを受信するレイヤは実行しない。

これにより M-to-M のモデルであった一つの publisher に対してすべてのロボットが subscriber を持つ必要がある問題をレイヤごとに subscriber へのアクセスを制御し、レイヤの数×(1-to-M)のモデルにすることで解決できると考える。

### 3.3 システム構成

レイヤごとに ROS の node を持ち、システム全体でレイヤを管理することで node の管理を行う。



システム起動時レイヤの登録を行い、コンテキストの更新に応じてレイヤの on/off を切り替える。

## 4. 考察

提案したシステムにおいてはレイヤを使うことで、コンテキストごとに扱う topic の数を制限することで M-to-M のモデルをレイヤの数×(1-to-M)のモデルとおくことができた。これにより過剰に topic を参照することを防ぎ負荷の軽減ができると考える。

## 5. 今後の課題

レイヤの動作を停止させることによる負荷軽減を示した。しかし、レイヤが非起動時に更新されたデータがレイヤ起動時にどう扱われるかを考える必要がある。例として配達の要請を受けるレイヤが非起動時に配達要請があった場合、レイヤが起動したときに配達が終わっているかどうかを知る必要がある。この場合には全体に配達を終了したことを伝えるシステムが必要になる。

### 参考文献

- [1] Rovert Hirschfeld, Pascal Costanza, Osccar Nierstarasz, "Context-Oriented Programming". Journal of Object Technology. 2008, 7(3), p.125-151
- [2] Harumi Watanabe, Midori Sugaya, Ikuta Tanigawa, Nobuhiko Ogura, Kenji Hisazumi  
A Study of Context-Oriented Programming for Applying to Robot Development  
COP'15 Proceedings of the 7th International Workshop on Context-Oriented Programming Article No. 4
- [3] Morgan Quigley, et al. "ROS: an open-source Robot Operating System". ICRA Workshop on Open Source Software. 2009.