継承性を用いた多目的ロボット開発フレームワーク

YONENAGA, Shinsuke / 米長, 慎介

(出版者 / Publisher)
法政大学大学院情報科学研究科
(雑誌名 / Journal or Publication Title)
法政大学大学院紀要. 情報科学研究科編 / 法政大学大学院紀要. 情報科学研究科編
(巻 / Volume)
8
(開始ページ / Start Page)
223
(終了ページ / End Page)
228
(発行年 / Year)
2013-03
(URL)
https://doi.org/10.15002/00009882

継承性を用いた多目的ロボット開発フレームワーク

The Concept of Inheritance based Multi-purpose Robot Development Framework

米長　慎介
Shinsuke Yonenaga
法政大学大学院情報科学研究科情報科学専攻
E-mail: shinsuke.yonenaga.3q@stu.hosei.ac.jp

Abstract

Acceleration of demographic aging and demand of multi-purpose robot make the urgent need of the industry standardized middleware of robot technology, called RT-Middleware. Building of a multi- purpose robot requires to making a robot able to switch many its action behaviors. The existing built-in programming approaches cannot dissociate algorithms of action behaviors and other functions from the robot controller. The paper presents a robot development framework based on the concept of object-orientation to materialize the robot controller development. In this study, the proposed robot development framework enables a designed robot to have abstract behaviors with which an inherited robot of different behavior can be easily and efficiently built by stipulating inherited behaviors and overwriting methods from its parent. The advantage of introducing inheritance and polymorphism enabled approach is to increase reusability and enable dissociating algorithms of behaviors and intelligent functions. This framework builds a business model for multi-purpose robot. The allotment of different roles and reuse of available resources in a large-scale development advance the spread of multipurpose robots for supporting increasing number of the elderly who need support from robots.

1. 序論

近年我が国は不景気が続いている．内閣府が 2013 年 1 月に公表した景気ウォッチャー調査によると，円安や新政研の発足により景気動向指数は増加しているものの，依然として不景気を表す 50%以下を示しており，長期的にも 2007 年以降景気動向指数が 50%を超えた月は 20%にとどまっている[1]．一方で日本の少子高齢化も深刻な社会問題として取り上げられている．国立社会保障・人口問題研究所による「日本の将来推計人口」によると，人口分布を年齢別に見た場合，平成 72 年度の年少人口割合は 4.0 ポイントの減少，生産年齢人口割合は 12.9 ポイントの減少が見込まれる．これに対し老年人口割合は一貫して上昇し，16.9 ポイント増加すると推計されている[2]．人口と経済成長の間には直接的な因果関係はないものの，生産年齢人口低下による労働力不足は避けられず，日本の強みであった輸出産業は国外へ労働力を求めているのが現状である．

日本ロボット工業会の世界の産業用ロボット稼動台数調査によると，2011 年に世界で稼働する産業用ロボットは日本で 30 万台，世界で 115 万台にのぼる．世界的にみれば年々市場規模は拡大しつつあるものの，日本の産業用ロボットは 2010 年からの受注数伸び率は 2.7%であり，成長が著しいアジア諸国に比べて市場規模が伸び悩んでいることがわかる[3]．それらの背景を踏まえ，国内の企業では労働力としてのロボット，すなわちサービスロボットへの期待が高まりつつある．2004 年に産業技術総合開発機構はオープンなロボットテクノロジー技術の仕様書である「RT-MIDDLEWARE」を発表した．また，産業総合研究所が提唱する技術戦略マップの発表により，ロボットが実際の生活空間で活躍することを前提とした大規模な国家プロジェクトとして様々な取り組みが行われた [4]．2005 年に開催された愛知万博では，サービスロボットの実用化に向けた実証実験が行われ，実際のビジネスにおいても，清掃ロボットや食事支援ロボット，災害復旧作業を行う遠隔操作型ロボット等の導入が進んでいる．

現在多くの企業がサービスにおけるロボット分野を IT 分野と同様に競争力を支える産業戦略の一つとして注目しており，様々なビジネスモデルが提唱され，それらは「ロボットインテグレーション」として議論と研究が進んでいる．将来は様々なサービスにおけるニーズや抱えている問題を分析し，いくつかの既存技術を組合せ，ユーザの要求に合わせたシステムを作りあげるロボットテクノロジーの「ソリューションビジネス」が活躍し，少子高齢化問題の解決や，社会をよりよくする産業として強化・発展することが望まれる．そのためのビジネスモデルや開発を促進するフレームワークが必要である．サービスロボットは環境の変わりやすい日常生活で自立して活動するため高度な知能と行動のバリエーションが必要である．本研究ではそれらを効率的に開発できる手法と，しかるべき教育を受けた「普通の技術者」がロボット開発を容易に行えるよう，各開発分野を独立化，再利用性を高めるための開発手法を提案し，それをサポートするフレームワークを構築する．

2. 関連技術

2.1 OpenRTM-aist

OpenRTM-aist は，独立行政法人産業技術総合研究所・知能システム研究部門・タスクインテリジェンス研究グループが開発・配布を行っている RT-MIDDLEWARE 実装の

Supervisor: Runhe Huang, Professor

一つである．OpenRTM-aist では，ロボットシステムを作る際に，RT コンポーネントと呼ばれるパーツごとにプログラムを作成し，ブロックのように RT コンポーネントをつなぎ合わせることでシステムを構築する．ロボットシステムの各モジュールを外部に依存しない形で実装することにより，独立性や再利用性の高いモジュールを容易に作成可能である OpenRTM-aist は，ネットワーク透過性，OS 非依存性，プログラミング言語非依存性を重視して 分散オブジェクトミドルウエアである CORBA を用いて実装されている．RT-Middleware を利用したコンポーネント指向によるロボット開発により，以前より明確にロボット開発の役割を分担し，技術を再利用することが可能になった．

2.2 OpenHRP

今日のロボット研究開発では，異なる専門知識を有する開発者による大規模な開発が行われている．それらの大規模開発にはロボットのための統合開発環境が不可欠である．また，開発途中で実験を行った場合，ロボットが予期せぬ動作を行い，ロボット自身や動作環境が破損する危険性がある．そのため，そのような実験はまずシミュレータで試験を行い，安全性を確認後，実機にして実験するのが望ましい．OpenRTM-aist と連携し動作する OpenHRP は，ロボットの動作及び作業シナリオの正当性，妥当性を容易に検証するために提供されるロボット開発のためのソフトウェアプラットフォームである．ロボットのコントローラや，指定した環境下での動作を動力学レベルで確認することができる[6]．OpenHRP でシミュレーションを行う場合，まずロボットモデルや環境モデルを作成し，コントローラに対応する RT コンポーネントを実装し，ロボットモデルの各関節にデータを与える必要がある．このツールを利用することによって，効率的にロボットの開発を行うことができるが，RT コンポーネントベースのコントローラ開発はロボットの各動作に注目したサービスレベルの設計には向いておらず，利便性や再利用性の面で問題がある．

2.3 Choreonoid

Choreonoid は産業総合研究所の中岡慎一郎が中心となり開発している，オープンソースのロボットの動作・振り付け開発ソフトウェアである[7]．Choreonoid では，OpenHRP などで利用されているロボット３Ｄモデルを視覚的に操作し，時間単位ごとにモデルのパーツごとにポーズを指定，以前のポーズから現在のポーズまでの起動をスムーズに生成，つなぎ合わせることによってロボットの多様で自然な動作を作成することが可能である．この機能は２足歩行型ロボットにも対応し脚を用いたステップ等を含む動作についてもロボットが転倒せずに実行可能な動作となるよう，システムが自動で身体バランスを補正するモジュールを標準で提供している．これにより，ヒューマノイドロボットの脚を含む全身動作についても，CG アニメーション開発と同様の感覚で作成することが可能である．生成されたロボットの動作は時間単位ごとの関節角度で構成されており，そのデータを利用することで容易にロボットの動作を各関節の角度ベースで作成・利用することができる．本研究ではロボットの動作作成に Choreonoid を利用し，それらを適当な形式に変換する．

2.4 RTC-Handle

RTC Handle は OpenRTM-aist で不足しているロボットシステム構築の支援を行う RT コンポーネントである．各制御を RT コンポーネントとして開発し，ミドルウェア上で動的にそれらの接続を切り替えることにより，サービスレベルの意味を持った RT コンポーネントによるプログラミングを可能にする．しかし，多目的ロボットのような様々な動作に意味をもち，それらを多様に変化する環境下で自在に操る場合，よりロボットの行動の意味に重点をおいたプログラミングが必要であり，実行環境などミドルウェアからプログラミングにおける制約を受けるべきではない．また RT コンポーネントとして提供されているため，利用する開発者によって様々な使い方や応用が可能である反面，「ロボット動作を制御するコントローラを開発する」といった明確な目的がある場合には，よりプログラミング手法や設計アプローチを提供し，標準化するフレームワークとしてのあり方が望ましい．

3. 継承性を用いたロボット開発手法

本研究の対象は様々な動作に対応し，実行することができる，汎用的で多目的なロボットである．具体的に言えば，多くの行動目的を持ち，その行動目的が変更されるたび，パーツや関節をハード的に変更する必要が無いロボットである．このような汎用的なロボットは多くの行動目的に対応できればできるほど，またその一つ一つの行動が効率的であればあるほど，優れたロボットであると言える．しかし，各ロボット開発者が一からモデルを作成し，そのモデルのための動作パターンを一から作成し，それらを組み合わせて毎回行動目的を作成していては，上記のような評価基準の下では，優れたロボットが開発しにくい状況となってしまう．汎用的なロボットが様々な動作に対応するということは，それだけ汎用的，一般的な関節構成となることが必然である．本研究では上記の特徴に着目し，一般的な関節構造を共通化し，それらに継承の概念を導入することで独立性・再利用性を高める開発手法を提案する．

3.1 オブジェクト指向

オブジェクト指向とは，あるデータ群とそれを使用するための基本的な操作・手続きをひとまとめにし，その合併体を一つのモジュールとしてみなした「オブジェクト」の相互作用としてシステムの振る舞いをとらえる考え方である．オブジェクト指向という考え方ができた背景として計算機の性能向上により，従来に比べ大規模で複雑なソフトウェアが書かれるようになってきたことが第一に挙げられる．そこでソフトウェアの再利用，部品化といったようなことを意識した仕組みの開発や，ソフトウェア開発工程の体系化などが行われるようになり，現在のロボットシステム開発においても開発工程の体系化が議論されている．また，多目的ロボットはモデリング技術，機械工学に関わるプログラミング，人工知能プログラミングなどの開発が

入り混じり，非常に開発作業や構成が複雑化しており，今後ロボットシステムがソリューションとして提供された際，同じ問題に陥る可能性もある．複雑さを克服する手段として，オブジェクト指向では密接に関連するデータと手続きはひとまとまりのオブジェクトとして定義し，かつ一度オブジェクトとして定義したものは，それがどのように実現されているかを知らずともその固体の機能を使うことができるという性質を利用することで複雑さを解決する．これらは抽象化の性質により成り立っている．また，オブジェクト指向によって我々人間のもつ現実世界に対する知識や行動，物体などの情報を，よい形で効率的にプログラミング内に表現することも可能となる．オブジェクト指向ではそれらの情報を我々が世界を知覚・認識する際に無意識に使っているのと同様な区分け方にしたがって固体や概念を区分け分割し，それらをまとまりとして表現する[8].

3.2 継承に関わる開発アプローチの追加

本研究では継承性を用いてロボット開発効率を高めるため，従来の手法に加えて，骨組モデル，骨組みロボットを継承した実装モデル，実装モデルへの動作パターンの登録，ライブラリを用いたロボットコントローラの開発というアプローチを追加している．これらのアプローチを利用することで，ロボットモデルの開発，モデルの動作の開発，モデルを制御するコントローラの開発それぞれを独立化させることが可能になり，専門分野の開発に集中することが可能になる他，再利用性も高めることができる．なお，これらのアプローチは従来の手法に情報を付加する方法で行われるため，本ツールのアプローチで開発したモデルは従来の手法でも利用することが可能である．

3.3 ロボットモデル

様々な形や機能を持つ多目的ロボットにおいて，そのロボット群(例えば人間型など)が必ず持つ共通の要素を開発のベースとして再利用することにより，ロボットモデルを効率よく開発することができる．共通化できる要素には関節構造やロボット識別番号などの具体的な情報と，実際の振る舞いは異なるが，本質的な意味合いにおいて同義となる，抽象的な情報が存在する．それらのひとかたまりの共通化されたロボットの情報を骨組みモデルとする．骨組みモデルは具体的な情報と抽象的な情報を複数持つ．これらの情報はモデルの継承やコントローラの開発から参照される情報であり，この情報を参照し合うことにより，各作業を独立化することが可能となる．骨組モデルにはそのモデルが行うであろう行動の名称が複数登録されている．図1は人間型ロボットの骨組みモデルがもつ付加情報の一例である．具体的な情報に当たるのがモデル名や作者などの基本情報とモデルの関節構造に関わる情報(従来のロボットモデル)である．それに対し，人間型ロボットが行うであろう「歩く」や「座る」などの動作名はロボットにより振舞いは異なるが，本質的な意味は共通である抽象的な情報である．これらは後述の抽象動作パターンであり，実際にどのように関節を動かすかなどの情報は含まれていない．

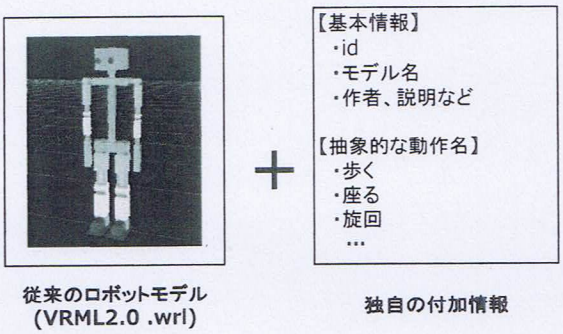

図1　ロボットモデルの表現.

3.4 モデル間の継承と実装

骨組みモデルを指定することで共通部分を再利用し，そのモデルをベースに開発する方法をロボットモデルの継承と定義する．また骨組みモデルを継承したモデルを実装モデルとする．図2はモデル同士の継承で引き継がれる情報を示している．実装モデルは骨組みモデルの持つ最低限の関節構造とセンサをすべて保有する他，骨組みモデルの持つ抽象的な動作も持つことになるが，各実装モデルは関節構造や外装を拡張，抽象的な動作パターンを具体化することでそれぞれ差別化を図る．動作パターンの具体化は継承した骨組が保有する抽象動作パターンに Choreonoid で作成した動作パターンファイルの情報を管理する付加情報ファイルの id を実際の振る舞いに登録することができ，また実装モデルオリジナルの行動も追加することが可能である．

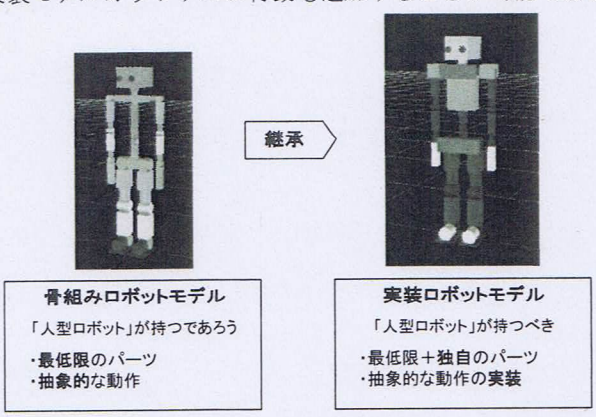

図2　実装モデルが引き継ぐ要素.

3.5 動作パターン

骨組モデルは，その骨組モデルが行うであろう行動の名称を複数登録することが可能である．この名称のみ登録された，行うであろう行動を抽象動作パターンとする．また，骨組みモデルを継承した実装モデルで実際に抽象動作パターンの振る舞いを具体化する場合，モデルの関節角度や角速度が必要となるが，それらの情報を動作パターンと呼び，モデルとは別のファイルとして記述する．この実装動作パターンは骨組モデルの抽象動作パターンや実装モデルのオリジナルの行動の実際の振る舞いを定義したものであり，実装モデルは骨組から継承した抽象動作パターンや

オリジナルの行動に対応する動作パターンを登録することで自身の振る舞いを決定することが可能である．本アプローチでは，制御を行うコントローラ内でロボットの関節や値の制御を記述することは独立化を妨げると考え，コントローラ内では動作パターンを指定することでモデルに動作を行わせる手法をとる．これにより，コントローラは探索，推論といった本来重要となるアルゴリズムに集中することが可能となる．本研究では動作パターンは前述の Choreonoid に対象となる骨組みモデルを読み込み，視覚的に生成することを前提としている．動作パターン開発の流れを図 3 に示す．図が示すように，同じ抽象動作パターンでも様々な実装が行われる．実装モデルの開発者はその中からどの動作パターンを登録するべきか，多目的ロボットのニーズに応じて適切なものを選択する必要がある．

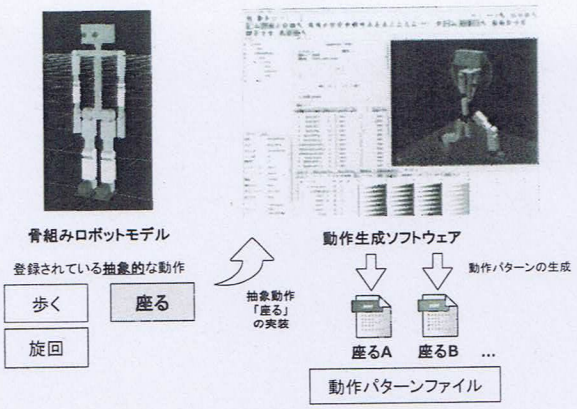

図 3　動作パターンの定義.

3.6 多様性の実現

骨組みモデルから継承した抽象動作パターンに Choreonoid で開発した動作パターンを登録する流れを図 4 に示す．複数の新たなモデルが共通の骨組モデルを継承し作成された場合，それらは骨組みモデルで定義された共通の関節，センサ構造を最低限保有していることが保証されており，骨組みモデルで定義された関節構造，センサを利用した骨組みモデルのための動作パターンを同じように利用することが可能である．よって骨組みモデルを継承した実装モデルは登録する動作パターンの実装によってロボットの振る舞いを切り替えることが可能であり，またコントローラ側では抽象動作パターンを指定し開発しているため，どのような動作パターンが登録されていようとコントローラの実装に影響を与えず，コントローラ開発分野と動作パターン開発分野を完全に独立することが可能である．これらはオブジェクト指向における多様性と同じ原理であり，同じ骨組モデルを継承した実装モデル同士で同じ動作パターンを再利用可能であるため，生産性を高めることができる．コントローラからモデルの抽象動作を呼び出す場合，その動作が実装されているかはわからない．それらは後述の動作パターンの動的取得を利用することにより解決することが可能である．

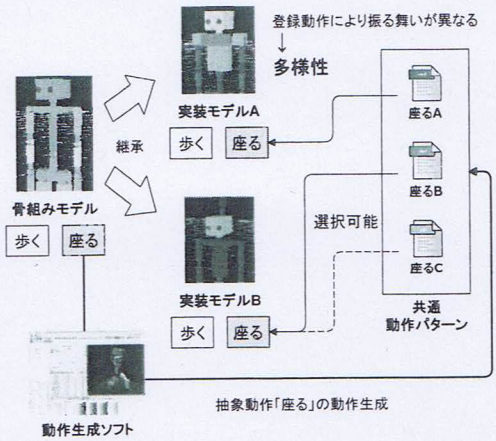

図 4　モデルの継承と多様性.

3.7 動作パターンの動的取得

本研究の開発アプローチではロボットの制御部分と振る舞いの記述が分離しており，かつ実装されるべき対象が明確に定義されているため，ロボットの振る舞いは開発過程に限らず，動的に切り替え，もしくは取得することが可能である．動作パターンの動的な取得はネットワーク上に構築された動作パターンリポジトリによって実現される．ロボットモデルは動作パターンが必要になった段階でリポジトリにアクセスし，必要な情報を取得する．本アプローチではモデル継承で得られた抽象動作パターンをすべて実装する必要がないが，その場合に実装しなかった抽象動作パターンがコントローラから呼び出された場合に，ロボットが行動すべき内容を自動で取得する目的も含まれている．

4. 実装と実験

上記のような拡張された骨組み，抽象インターフェイス，動作パターンパッケージ及び必要な動作パターンの動的取得を現在の OpenRTM-aist 環境で実現させるためには，それらを機能的，視覚的にサポートするための開発環境及び実行環境が必要である．そもそも OpenRTM-aist や OpenHRP は複数のシステムの集合であり，それらのシステムは CORBA を利用して通信を行い，ひとつの大規模なシステムとして稼働している．本研究で開発する開発環境，実行環境も，CORBA を利用して各システムの通信に介入し，本来のシステムに変更を加えることなく，定義したアプローチのサポートを実現する．

4.1 継承性を管理する開発フレームワーク

アプローチ拡張フレームワークは従来の OpenHRP を利用した開発手法に本研究のアプローチを追加するためのフレームワークである．本フレームワークを利用した開発の流れを図 5 に示す．OpenHRP や OpenRTM-aist など多くの RT-Middleware に関わるツールは Java などの統合開発環境で知られる Eclipse のプラグイン形式で提供されており，それぞれのツールは，ワークスペースやネームサーバを利用したコンポーネント間通信を利用して連携を取り合って

いる．本ツールは，それらの開発手法から余計な手間をかけることなく利用可能にするため，Eclipse のプラグインとして，従来のツールに機能を追加する形で提供する．これにより，従来のアプローチとの切り替えが容易になる他，バージョンアップなどで従来のツールに変更が加えられたとしても，問題なく本ツールを利用することが可能である．OpenHRP の持つモデル構造との連携は，Java のリフレクション技術を用いて操作した．

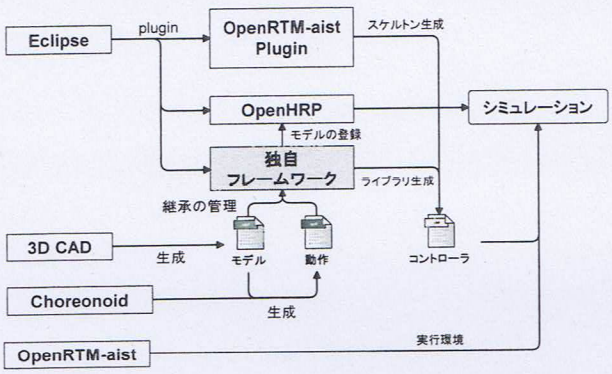

図 5　提案手法の開発の流れと使用技術との関連図．

4.1.1　付加情報の作成

各モデルと動作パターン，制御を行うコントローラに付加する情報を視覚的に記述することを可能にする．それらの付加情報は xml 形式でそれぞれのデータに必要な情報が記述された付加情報ファイルとして定義されており，テキストエディタで直接で情報を編集することも可能である．付加情報ファイルは対象となるデータと同じ階層に拡張子の異なる同名のファイルとして保存される．

4.1.2　付継承関係の管理

OpenHRP 上で管理されているモデルの付加情報に指定された対象となるデータを適切な形で取得する．継承関係のモデルデータ，各関節の角度や角速度の値，コンポーネントなどの，付加情報ファイルの対象となるデータを開発環境内で利用できるよう読み込む．それらは状況に応じて動的に提供される．

4.1.3　モデル専用ライブラリの自動生成

自身が開発したモデル，及びモデルに対応する付加情報を元に，制御を行うコントローラで利用できるライブラリや，シミュレーション上のモデルとコントローラを繋ぐ「コントローラブリッジ」と呼ばれる機能を起動するためのバッチファイルなどを生成する．そのモデル専用のライブラリを利用して，制御を行うコントローラの開発者は関節などの情報を気にすること無くロボットモデルをオブジェクトに見立て，ロボット間の継承関係や動作の実行など，完全なオブジェクト指向としてプログラミングを行うことが可能である．

4.2　コントローラ制御の実現

フレームワークが生成するライブラリを使用したコントローラ開発の実現方法を図 6 に示す．従来の OpenHRP におけるロボットコントローラ開発では，シミュレーション上のモデルの各関節と制御を行うコントローラはコントローラブリッジと呼ばれるコンポーネントを介して通信を行なっていたが，このコントローラブリッジと接続するコントローラは 1 MHz から 2 MHz で繰り返し実行されており，高度なアルゴリズムを記述するには向いていない．よって本ツールでは，コントローラブリッジと制御を行うコンポーネントの間に中間のコンポーネント(動作制御モジュール)を作成し，コントローラの開発を容易にしている．自動生成ライブラリはアプローチ拡張フレームワークがモデルの情報を元に自動生成される Java ライブラリである．ロボットの基本的な情報の他，継承したモデルや抽象動作パターンに 1 対 1 で対応するメソッドを保持している．これにより，本アプローチでは通常のプログラムと同じ要領でコントローラの開発を行うことが可能である．動作パターンの制御方法には PD 制御を用いる．PD 制御はモーターの制御などに用いられる手法で，関節角度の変化に比例したトルク(P 制御)と，角速度の変化に比例した微調整(D 制御)を行う手法である．PD 制御は P 制御に比べ，関節に発生するいきすぎ(振動)を抑えることが可能である．

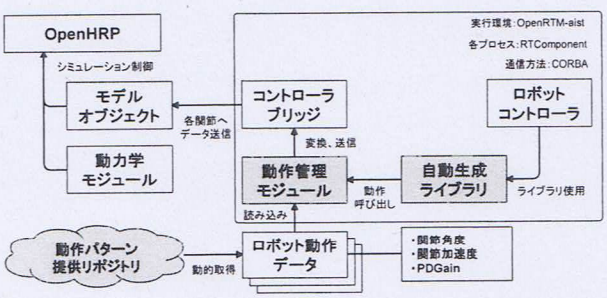

図 6　モジュールによるコントローラ制御．

4.3　動作パターン提供リポジトリ

動作パターン提供リポジトリはコントローラがライブラリを通して動作パターンを呼び出し，動作管理モジュールが該当するファイルを見つけられなかった際に呼び出されるネットワーク上のシステムである．リポジトリは Google 社の Google App Engine 上で，同社が提供している標準のフレームワークを利用し開発を行った．

4.4　フレームワークを利用した開発

図 7 にアプローチ拡張フレームワークのスクリーンショットを示す．図のように，ロボットモデルの登録，継承，付加情報の作成，追加，それらを利用したコントローラ専用ライブラリの生成などを OpenHRP 上で視覚的に行うことができる．その際モデルの開発者は動作パターンの中身や実装するコントローラを意識する必要はない．生成されたモデル専用ライブラリを用いてプログラミングを行った

ソースコードを図 8 で示す．通常のプログラミングと同様に，オブジェクトが歩くなどの動作メソッドを呼び出すというオブジェクト指向プログラミングを実現しており，機構学の知識がない開発者でも容易に開発が可能である．ロボットは複数のセンサを持つことが可能であるが，現在は加速度の取得のみ可能でそれ以外の入力機能やイベント駆動のメソッドは存在しない．

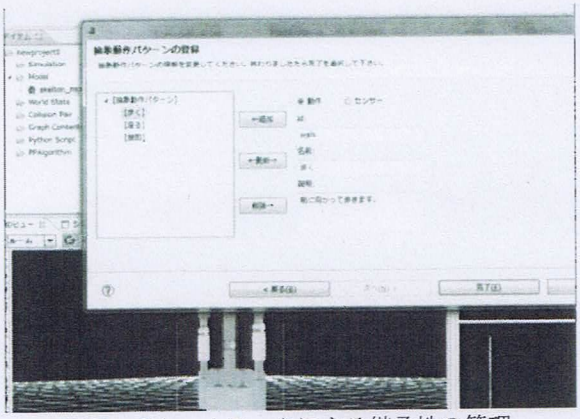

図 7　フレームワークによる継承性の管理.

```
import jp.opensdt.model.MySkeltonRobot;


public class ImplementTest {
    public static void main(String[] args) {
        MyImplementRobot iRobot = new MyImplementRobot(null);
        MySkeltonRobot   sRobot = iRobot; // inheritance

        sRobot.dowalk(); // do walk(implement)
        for(int i=0;i<3;i++){
            sRobot.doturn(); // turn 90*3(abstract)
        }
        sRobot.dosit_down(); // do sit down(abstract)

        sRobot.getAbsTransform();
        sRobot.getActors();
```

図 8　ライブラリによるコントローラ記述.

5. まとめ

継承性，多様性を用いた本アプローチとフレームワークを利用することで，各分野の独立性，再利用性が高まり，各開発者が得意な分野に集中し，研究，開発効率の向上が図ることが可能である．例えば CAD でモデルを作るのが得意な人はロボットモデルのみ，動力学が得意な人は実装動作パターンのみ，プログラムが得意な人は制御を行うコントローラのみ開発し，他の分野は第三者が作成したものを利用すれば，より多くの人が RT-Middleware に関わり効率よく開発が可能であり，それでこそ RT-Middleware のメリットを活かすことができる．また，各分野を完全に独立化することによって新たなビジネスや活動を生む可能性があり，骨組の継承関係，動作パターンの概念によって再利用性が向上することで，同じ骨組を継承したロボットの生産ラインをある程度共通化でき，複数のモデルに対して自社や研究室の実装動作パターンや制御コントローラを提供するなど，縦のラインを越えた様々な連携ができるようになり，ロボット市場の活性化が期待できる．

6. 今後の課題

本研究では独立性，再利用性を高めるアプローチを視覚的に補助するフレームワークを開発したが，ロボットモデル継承の定義，動作パターンの記述方法と制御方法，コントローラ側から呼び出す場合のパラーメータ化，ロボットのセンサに対応するイベントメソッド，モデル間の協調や各関節における動作実行の非同期化，実行環境の整備など，実用化には仕様を固めなければならない部分は多い．また，各分野には優れたツールや RT-Middleware と同様のミドルウェア仕様が今後も登場していくと考えられるため，柔軟な対応が必要不可欠である．

参考文献

[1] 内閣府政策統括官室経済財政分析担当. “景気ウォッチャー調査”. 景気ウォッチャー調査. (オンライン), 入手先 <http://www5.cao.go.jp/keizai3/watcher/watcher_menu.html >, (参照 2013-01-30)

[2] 国立社会保障・人口問題研究所人口動向研究部. “日本の将来推計人口”. 国立社会保障・人口問題研究所. (オンライン), 入手先 <http://www.ipss.go.jp/syoushika/tohkei/newest04/point.pdf>, (参照 2013-01-30)

[3] 日本機械工業連合会, 日本ロボット工業会. “２１世紀におけるロボット社会創造のための技術戦略調査報告書”. 一般社団法人 日本ロボット工業会. (オンライン), 入手先<http://www.jara.jp/publication/dl/rt.pdf>, (参照 2013-01-30)

[4] ロボット技術戦略マップ検討会. “技術戦略マップ 2010(ロボット分野)”. NEDO:独立行政法人 新エネルギー・産業技術総合開発機構. (オンライン), 入手先< http://www.nedo.go.jp/content/100109957.pdf>, (参照 2013-01-30)

[5] 日本ロボット工業会. “世界の産業用ロボット稼動台数”. 一般社団法人 日本ロボット工業会. (オンライン), 入手先<http://www.jara.jp/data/dl/kado.pdf>, (参照 2013-01-30)

[6] 比留川博久, 金広文男, 中岡慎一郎, 末廣尚士, 神徳[]徹雄, 安藤慶昭, 中村仁彦, 山根克, 齋藤元, 川角祐一郎, " 分散コンポーネント型ロボットシミュレータ OpenHRP3", 第 25 回日本ロボット学会学術講演会, September 2007 .

[7] 中岡，三浦，森澤，金広，金子，梶田，横井: “ヒューマノイドロボットのコンテンツ技術化に向けて - クリエイターによる多様な表現の創出が可能な二足歩行ヒューマノイドロボットの実現 -”, Synthesiology, vol.4, no.2, pp.80-91, 2011.

[8] 米澤明憲, “オブジェクト指向計算の現状と展望”, Information Processing Society of Japan, vol 29, pp.290-294, Apr 1988.