N-027

# フレームワークでのメッセージ理解のための図式チュータリング

## Graphical Tutoring to Understand Message Passing in Java Framework

上野 敦子† 島川 博光†
Atsuko Ueno Hiromitsu Shimakawa

## 1. はじめに

Java 言語はフレームワークやメッセージパッシングなどのプログラミングに関する制約を多く設けている．すでに他の言語でプログラミングの基礎知識を身につけた学習者にとっては，この制約が Java を習得するための大きな難関となる．通常これらの制約は講義形式で学習者に教えられる．しかし学習者は教員の口頭説明を受けただけでは制約を十分に理解できず，その後のプログラミング作業につまずく．

そこで本研究では学習者のフレームワークやメッセージパッシングに対する理解支援を行うことを目的に，講義とそれに対応する演習の間に位置づけられる学習支援環境を提案する．本手法はシーケンス図を用いて Sender，Receiver，メッセージの記述順序，参照情報を図示する．これによりメッセージを記述するために必要な情報を学習者に明確に示す．またシステム内で教員の役割を果たすチュータが学習者の解答の正しさをコード依存関係と既知の情報から随時判定する．

## 2. Java プログラミング教育

### 2.1 Java プログラミングの学習困難性

Java 言語はプログラミングに関する制約が多い．学習者が理解に苦しむ Java の制約としてフレームワークとメッセージパッシングが存在する．

フレームワークは Java が持つ部品の概要とそれらの組み合わせ方法を合わせたものである[1]．Java はあらかじめ用意された部品を再利用してプログラミングを行う言語であり，プログラム作成のためには部品の知識が必須である．またメッセージパッシングとは図 1 に示すようなオブジェクト間のメッセージのやり取りをさす．メッセージを記述するためには Sender，Receiver，引数，返り値，おのおののオブジェクトから参照できる情報を意識してプログラムを作成すべきである．学習者はフレームワークとメッセージパッシングの双方を事前に理解していなければ Java プログラミングを行うことができない．

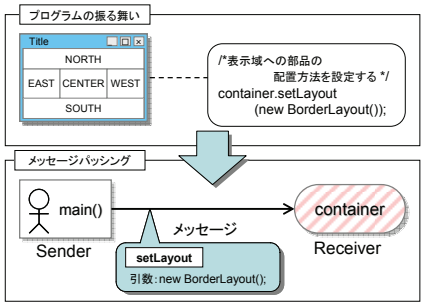


図 1 メッセージパッシング

† 立命館大学 大学院 理工学研究科

### 2.2 プログラミング教育の現状

大学における Java プログラミング教育では，フレームワークやオブジェクト指向の概念は講義形式の授業で教えられ，その後に Java プログラミング演習が行われることが多い．教員の口頭説明のみを受けた学習者は，Java プログラミングのおおまかなイメージを把握することはできても，プログラムを作成するさいに必要となるメッセージ記述の細かい知識を理解することができない．

## 3. メッセージパッシングの図式チュータリング

### 3.1 コード依存関係を用いたチュータリング

学習者のメッセージパッシングに対する理解促進を行う手法を提案する．本手法が提供する学習支援環境は講義とそれに対応するプログラミング演習の間に位置づけられ，講義で得られた知識を定着させる役割を持つ．

本手法は，シーケンス図を用いて Sender，Receiver，メッセージ記述順序，参照情報を図示する．これによりメッセージ記述に必要な情報を学習者に明確に示す．また教員の役割を果たすチュータが学習者の解答の正しさを以下に示すコード依存関係から随時判定する．

本研究では，Java プログラムを部品生成や結合などの機能を単位として分割し，それらをコードと呼ぶ．このコードの記述順序を表現するためにコード依存関係[2]を用いる．コード A を記述する前にコード B を記述しておかなければならないとき，A は B に依存するという．

Java のコード間に発生する依存関係は，その発生原因によって 2 種類に分類できる．1 点目はインスタンスの状態，2 点目はコードの記述順序を主とする依存関係である．

### 3.2 インスタンスの状態を主とする依存関係

Java プログラムに含まれるコードの多くはインスタンスの生成や設定，また設定が完了したインスタンスを利用して新たな処理を行う．このような場合にインスタンスの状態を主とする依存関係が発生する． 図 3 を例に挙げると，コード 1 は JButton インスタンスを生成しており，コード 3 ではそのインスタンスをメソッドの引数として利用している．よってコード 3 はコード 1 のインスタンス生成に依存していることが分かる．

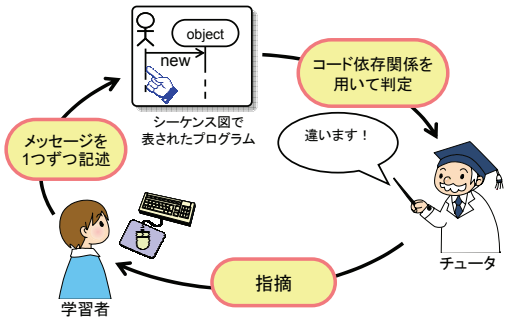


図 2 提案手法の概要

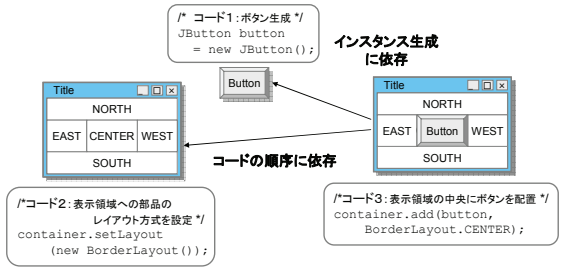


図 3　コード依存関係の発生理由

### 3.3　順序を主とする依存関係

Java フレームワークでは，メソッド実行に順序が前提として存在する場合がある．このような場合，学習内容や学習者のレベルによっては，依存関係の発生理由を厳密に教えるよりもコードの記述順序の説明に重点を置くべきである．図 3 中のコード 3 から 2 への依存関係を例に挙げると，コード 2 は Container インスタンスにレイアウトの設定をしており，コード 3 ではその Container インスタンスに対して部品配置を依頼している．この依存関係を説明するさいには，「部品を配置する前にレイアウト方式を設定するべき」のような，記述順序を基にした説明が一般的に望まれる．よってコード 3 はコード 2 に対して順序を主とした依存関係を持つ．

### 3.4　オブジェクトが保持する情報

Java 言語の各オブジェクトは他のオブジェクトへの参照やプリミティブ値を保存するための変数を保持している．変数値を参照できる範囲はその変数の宣言場所によって異なる．インスタンス変数であればそのインスタンス内，メソッド内で宣言されたローカル変数であればそのメソッド内でのみ変数を利用できる．本手法では変数保存時にどの領域内で保存するかを学習者に問い，現在参照できる情報を領域別に一覧表示する．これにより学習者は変数の参照領域の違いを正しく認識することができる．

### 3.5　依存関係と既知の情報からの判定

本研究では学習に必要な Java の部品群をあらかじめ用意しており，学習者は部品やそれに含まれるメッセージを選択することで学習を進める．部品には代表的な変数とコンストラクタ，メソッドが含まれている．コンストラクタとメソッドにはそれぞれの引数に渡すべき実数とそのメソッドが依存するコードが設定されている．

学習者によってコードが記述されると，コードの前提条件に設定されているコードがすでに書かれているかを調べる．またコード内で利用しているインスタンスが Sender の領域内から参照可能であるかを調べる．これにより学習者の記述の正しさを判定する．

## 4.　チュータリングシステム

提案手法を実現するチュータリングシステムを Macromedia Flash[3]を用いて実装する．プログラム作成のために必要となるクラスやメソッド構成はあらかじめ用意されており，学習者はメソッド内の処理部分を記述することによって学習を進める．プログラムの記述はマウスクリックやドラッグで行う．図 4 にメッセージを記述する場合の動作例を示す．まず学習者は Sender と Receiver を選択する．すると Receiver が持つメソッドの一覧が表示されるので，その中から目的のメソッドを選択する．返却値を持つメソッドの場合は，返却値の保存場所を指定する必要がある．学習者がメッセージを記述したとき，それが正しくない記述であればチュータから指摘が与えられ，正しい記述であれば次の記述に進むことができる．

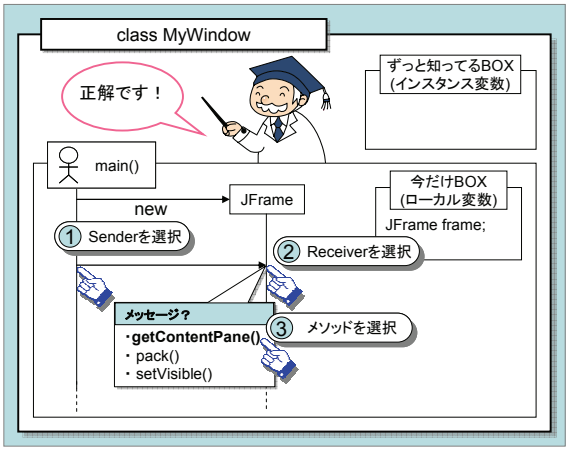


図 4　チュータリングシステム

## 5.　既存研究との比較

本手法と関連研究[4]，[5]とを表 1 にて比較する．本手法はメッセージパッシングのチュータリングを可能としている．またプログラミング作業中に学習者に指摘を与えることができる．しかし，プログラムの自動判定のためにあらかじめ Java の部品群を用意しておく必要がある．

表 1　既存研究との比較

| | ITPs[4] | Jmaster[5] | 本手法 |
|---|---|---|---|
| 目的 | プログラミングの学習支援 | プログラミングの学習支援 | メッセージの理解支援 |
| 対象言語 | C，Pascal など | Java | Java |
| 作業中の指摘 | △ | × | ○ |
| 再利用性 | × | ○ | × |

## 6.　おわりに

本研究では学習者のメッセージパッシングに対する理解支援を目的に，シーケンス図を用いたチュータリングシステムを提案した．今後はより効果的な指導のタイミングや指導内容を考察するために，実験・検証を行う予定である．

## 参考文献

[1] R. E. Johnson, 中村宏明, 中山裕子, 吉田和樹 著, パターンとフレームワーク, 共立出版 (1999)
[2] 上野敦子, 島川博光, コード依存関係を用いた Java フレームワークの チュータリング, 電子情報通信学会/情報処理学会 情報科学技術レターズ, vol.4, pp. 331-332 (2005)
[3] http://www.macromedia.com/jp/software/flash/
[4] N. Pillay, Developing Intelligent Programming Tutors for Novice Programmers, ACM SIGCSE Bulletin, pp.117-121(2003)
[5] 高岡詠子，斉藤将志，谷口景介，Java プログラミング教育支援システム Jmaster の開発と教育効果，第 2 回情報科学技術フォーラム(FIT2003)，pp.381-382 (2003)