

# Java 部品管理システムを利用した ソフトウェア開発環境の実例\*

7C-5

植木 紀昌      青木 寛      野村 竜太郎      木村 耕  
電気通信大学 情報工学科

## 1 はじめに

コンポーネントウェア [1] の進展によって、ソフトウェア部品 (以下部品) の再利用に関わる技術が新たな局面を迎えた。とくに、広くネットワーク上に散在する部品を収集、組み立てるアプリケーション開発は、今後の新たな工法として注目される [2]。

われわれは、上記の工法の実験基盤として Java 対応の部品管理システム (Parts Management System for Java:以下 PMS/J) を構築し、これをベースとしたソフトウェア統合開発環境 (Jide;Java-IDE) を実現した。

## 2 Jide の設計方針

Jide はネットワークを介して Java 部品を収集し、これを組み立てるソフトウェア開発環境を提供する (図 1)。このような環境では、ややもするとネットワーク上の部品分類方式の不具合やアクセス方式の不統合による部品検索のオーバーヘッドを起し、そして部品の継承や実装の際に生じる煩雑なコード編集、またインスタンス生成やメソッドの利用に必要な部品情報の閲覧等による作業能率の低下を生じる。

Jide では、以上のような弊害に対処するために、次に述べる二つの機能を実装した。

### 2.1 ネットワーク上の部品管理

Jide では、部品検索システムとして PMS/J を利用している。PMS/J は、ネットワーク上に散在する部品管理サーバ (PMS/J サーバ) と情報端末としてのクライアントからなる、Java により実装された部品検索システムである。再利用可能な部品をサーバ上のリポジトリに蓄積し、不特

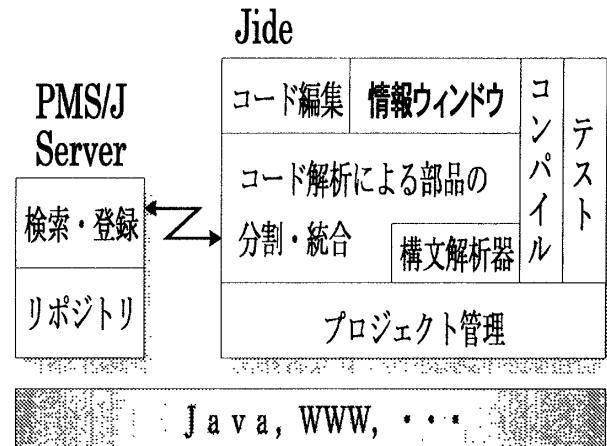


図 1: システム構成

定のユーザはその部品を検索、取得することができる。

Jide では、PMS/J と接続することで再利用可能な部品の検索・登録を統合的に管理している。また、サーバのうちの一つをプライマリサーバとして位置づけ、開発ソフトウェアの保存場所としてサーバのリポジトリを利用している。

### 2.2 情報ウィンドウ

部品情報を動的に閲覧し、ソースコード編集等を効率的に行うためのシステムとして、情報ウィンドウを考案した。情報ウィンドウはシグネチャ情報を階層的に整理して表示する。シグネチャ情報には、次のようなものが含まれる。

- クラス関係：クラス名，継承するクラス名，インタフェース名
- メソッド関係：メソッド名，戻り値型，アクセス制御，その他の接頭辞，引数，発生する例外
- 変数関係：変数名，型，アクセス情報，その他の接頭辞，初期値
- 各種ドキュメント

\*A prototyping of software development environment on Parts Management System for Java by Norimasa Ueki, Hiroshi Aoki, Ryutaro Nomura and Koh Kimura, The University of Electro-Communications

### 3 実験・評価

現状で Jide はアルファ版が完成している。これを使って実験を行い、情報ウィンドウによるシグネチャ情報閲覧の効率を主に操作性の面から測定し、一般のテキストエディタ環境と比較した。

#### 3.1 方法

開発途中のソースコードにおいてメソッドを書くことを想定し、被験者にソースコード中で記述すべき識別子に対するシグネチャ情報を表示してもらう。

このときシグネチャ情報を格納しているリソースによる場合分け (表 1) を行い、表示させるまでの操作にかかる時間を計測する。

表 1: シグネチャ情報の格納場所による分類

環境	分類	情報格納場所/表示方法
一般環境	A	同一コードファイル/ エディタをスクロールさせる。
	B	異なるコードファイル/ エディタに別ファイルを を読み込む。または別の ウィンドウに読み込む
	C	API ドキュメント/ WWW ブラウザで表示
Jide	D	情報ウィンドウ/ 常設ウィンドウで表示

#### 3.2 結果

3人の被験者に数回ずつメソッドを作る動作を行ってもらった。その過程で情報閲覧操作にかかる平均時間は次の通り (表 2) に得られた。

表 2: 情報閲覧操作の平均所要時間

分類	ファイルの閲覧回数	時間 (秒)
A	-	7.3
B <sub>1</sub>	1 回目	22.6
B <sub>2</sub>	2 回目以降	6.4
C <sub>1</sub>	1 回目	17.5
C <sub>2</sub>	2 回目以降	6.6
D	-	2.8

分類 B および C については、情報の格納されるファイルに対して一回目の閲覧はファイルを開くといった作業が付加されるため、二回目以降と

大きく異なる。そのため、二回目以降は閲覧操作時間を別途算出した。分類 A および D に関しては、常に画面上に表示されているため、二回目以降に違いは見られなかった。

#### 3.3 考察

あるソースコードにおいて、分類 A, B, C にあたる識別子の割合は、およそ 1:3:4 であった。さらに分類 B<sub>1</sub> と B<sub>2</sub> の比は、ほぼ 1:1、分類 C<sub>1</sub> と C<sub>2</sub> の比は 3:5 であった。これらの値と実験の結果を組み合わせると、そのソースコードにおいて情報閲覧操作にかかる時間は、Jide の方が一般の環境の四分の一以下で済むことがわかる。

この実験では、実装者のソフトウェア部品に対する理解度や閲覧に至る状況、環境による誤差、労力といった条件付けがなされていないために、シグネチャ情報閲覧に関する評価は完全ではない。また、今回の実験において情報ウィンドウはシステム API に含まれるクラスの情報は無視して評価を行っているため、表示している部品の数も比較的少ない。その分、目的の情報にたどり着くまでの時間は、通常より短かったと推測される。

しかし、情報ウィンドウのようにソースコードとは別に情報を整理閲覧できる機構を備えることで、情報閲覧操作に対するオーバーヘッド減少に効果のあることが認められた。

### 4 おわりに

本論文では情報ウィンドウによる作業能率の向上に関する実験を行った。現在、ネットワーク上の部品管理システムとリンクによる効果について実験を進めている。

Jide は、PMS/J と平行して開発が行われている。今後はバイナリ部品を含めたさまざまな形態のデータのサポートや情報ウィンドウの改良などを予定している。

#### 参考文献

- [1] 青山幹雄: コンポーネントウェア: 部品組立て型ソフトウェア開発技術, 情報処理, Vol.37, No.1, pp.71-79(1996).
- [2] 野呂昌満: ネットワークソフトウェアの新しい工法, 情報処理学会研究報告, SE110-13-3, pp.93-96(1996).