

江口 敦子  
EGUCHI Atsuko

海邊 裕  
KAIBE Hiroshi

宮澤 隆幸  
MIYAZAWA Takayuki

Java<sup>(注1)</sup>は、その適応力の高さから情報系システム分野で脚光を浴びているプラットフォームであるが、最近では組み込み機器あるいは制御システムへの適用でも注目されている。

当社は、情報系と制御系の両者を統合する分野のアプリケーションの作成に必要な非同期イベント駆動型の分散システム構築用フレームワークである ECJ<sub>TM</sub>(Event Centric for Java)を開発した。これにより情報系と制御系の高速かつシームレスな統合が実現でき、短期間でシステム開発が可能となる。ECJ<sub>TM</sub>の特長は軽量かつ高速なイベント配送であり、すべて Java で記述されている。

Java is a computer system platform suitable for business systems, and has recently become popular with its hardware-transparent properties. Java provides various application program interfaces (APIs), and is also being used for embedded and control systems.

Toshiba has developed ECJ<sub>TM</sub> (event centric for Java), which provides a mechanism for asynchronous event delivery to distributed computers. By combining this high-speed event delivery mechanism with conventional Java remote method invocation (RMI), it becomes easy to seamlessly integrate business and control systems.

## 1 まえがき

Sun Microsystems 社が開発し普及・標準化活動を展開している Java 環境は、そのプラットフォームの独立性による適用分野の幅広さから汎(はん)用計算機の標準環境としての成長が期待されている。現在主流の開発環境である JDK<sup>(注2)</sup>-1.1 では、情報系システムを構築するための API (Application Programming Interface) 群が数多く標準化されている。次期版の JDK-1.2 でさらに API 群が追加される予定である。これらの API 群をもつことで、Java は情報系のシステムを構築するための技術として成熟した環境になったものと考えられる。

一方、最近の動向として、Java の組み込みシステムや制御システムへの適用が注目を集めている。例えば、JavaSoft からは PersonalJava, EmbeddedJava などの組み込みシステム向け標準 API が提案されている。また、その他のベンダーも、組み込みシステムに関する Java 技術の発表を始めている。

組み込みシステムや制御システムを実現する場合に、イベント駆動型のシステム構築を提供する機能とネットワーク対応機能が要求される。そこで、当社はそれらの機能を提供するフレームワークである ECJ<sub>TM</sub>を開発した(図1)。

ここでは ECJ<sub>TM</sub>の概要と特長を述べる。

(注1) Java ならびにその他の Java を含む商標は、米国 Sun Microsystems 社の商標。

(注2) JDK は、米国およびその他の国における米国 Sun Microsystems 社の商標。

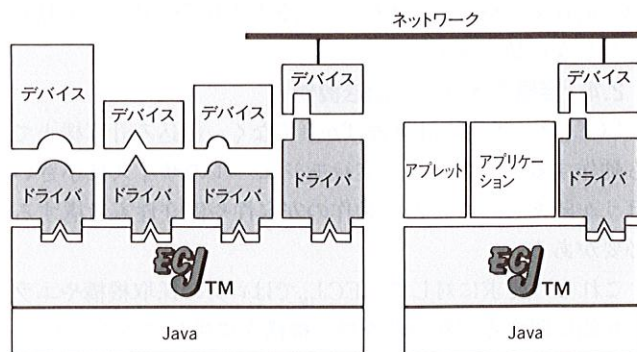


図1. ECJ<sub>TM</sub>の全体概要 ECJ<sub>TM</sub>は、Java 上に実装されたフレームワークである。それぞれのデバイスは対応するドライバを用意することで ECJ<sub>TM</sub>にプラグインできる。

Overview of ECJ<sub>TM</sub>

## 2 ECJ<sub>TM</sub>の概要

ECJ<sub>TM</sub>は、次の機能を提供するオブジェクト指向フレームワークである。

- (1) イベント駆動型のシステム構築が可能
- (2) Java による記述
- (3) 高速なネットワーク経由のイベント配送
- (4) 軽量のイベント配送機構
- (5) 既存システムの統合を容易にする機構

ここでは、これらの特長について述べる。

### 2.1 イベント駆動型のシステム構築

多くの制御システムは、非同期のイベント駆動型システ

ムとして実装される。ネットワークを介した分散アプリケーションに拡張する場合にも、イベント駆動型システムとしての構築が望まれる。しかし、標準 Java 環境として提供される分散オブジェクトフレームワーク RMI (Remote Method Invocation) は同期通信しかサポートしていない。

ECJ<sub>TM</sub>は、この標準 Java 環境を補完するフレームワークとして位置付けられる。すなわち、非同期のイベント配送機能を提供するフレームワークである。

## 2.2 Java による記述

ECJ<sub>TM</sub>はすべて Java 言語で書かれている。さらに、プラットフォームに依存するライブラリを使用せず実装されている。このため、Java 環境のもつプラットフォーム独立性を損なうことなく、JDK-1.1 環境であればどのようなプラットフォームでも動作する。

## 2.3 高速なネットワーク経由のイベント配送

高速な応答性が要求される分野でも利用できるようにするため、ECJ<sub>TM</sub>の設計と実装の段階で、メモリ管理やスレッドのコンテキスト切換えなどの比較的時間を要する処理の最適化を図った。

これにより ECJ<sub>TM</sub>のイベント配送速度は、ネットワークを経由した場合でも数ミリ秒以下となった。この値は実行環境に依存するが、C 言語などで実装した場合と比べても見おとしめない値である。

## 2.4 軽量のイベント配送機構

ECJ<sub>TM</sub>は、汎用の計算機ばかりでなく、組み込み計算機上でも動作する。このため、実行モジュールのサイズは小さいほうが望ましい。また、動作の安定性や保守性も考慮する必要がある。

これらの要求に対して、ECJ<sub>TM</sub>ではログの採取機構やエラー状態に関する例外処理を扱う機構を提供している。また、通信機構や制御機器に応じたドライバを、それぞれプラグイン可能な構造にしている。これにより、対象とするシステムに応じた必要最小限の構成を構築することができる。

## 2.5 既存システムの統合を容易にする機構

ECJ<sub>TM</sub>ではイベントの統合モデルをサポートしている。このモデルは、既存の制御システムなどで独自に規定されたイベント群を抽象化し、これらをオブジェクト指向の継承関係として体系化したものである。

継承関係に基づいてイベントを任意の型に変換することが可能となり、既存のシステムで用いられる複数の異種イベントを容易に統合することができる。また、ネットワークなどへのアクセスを提供するドライバ層も抽象化して扱っているため、通信層のプロトコル変更なども柔軟に実現できる。

## 2.6 ECJ<sub>TM</sub>の適用分野

以上の特長を備えた ECJ<sub>TM</sub>は次の分野での利用が考えられる。

- (1) 遠隔監視制御システム イン트라ネットなどを經由した、プラントなどの遠隔監視あるいは遠隔制御システム
- (2) 家庭内情報システム 家庭内の情報機器などを遠隔地から、例えば公衆回線経由でアクセスするシステム
- (3) オフィス環境の簡易管理サービス オフィス内の計算機やプリンタなどの状態を管理ホストから監視して障害などの検出を行うサービス
- (4) エンターテインメント分野 インターネット上でのゲームやマルチメディア情報の配送をするサービス
- (5) 車載計算機への適用 家庭内情報システムと車載計算機システムとのインターネットを經由した統合に利用

このほか、教育向けのシステムや情報キオスクなど、Java を使用するさまざまなシステムで利用可能と考える。

## 3 ECJ<sub>TM</sub>の機能構成

ここでは、ECJ<sub>TM</sub>の構成要素について述べる。

ECJ<sub>TM</sub>は次の要素から成っている。

- (1) ECJ<sub>TM</sub>イベント
- (2) ローカルマネージャ
- (3) ハンドラ
- (4) ドライバ
- (5) トランスポート

この構成を図 2 に示す。以下にそれぞれについて説明する。

### 3.1 ECJ<sub>TM</sub>イベント

ECJ<sub>TM</sub>が扱うイベントである。これはオブジェクト指向の抽象化を行ったイベントであり、イベント ID とデータから構成される。

イベント ID は 64 ビットの空間をもつ。インターネット上のさまざまな資源をユニークに識別する必要があるため、IP アドレスなどの既存の識別子を内部に含められる構成としている。データは、イベント ID ごとに任意の利用方法が許されており、Java Object Serialization を利用することでインスタンスの情報を保持することも可能である。これにより、永続オブジェクトなどの扱いも可能となっている。

### 3.2 ローカルマネージャ

ECJ<sub>TM</sub>の管理部である。内部にイベントディスパッチャとイベントテーブルをもつ。イベントテーブルはイベント ID とイベントのハンドラの対応表である。この対応表に基づき、イベントディスパッチャは各ドライバやトランスポートからの ECJ<sub>TM</sub>イベントを適切なハンドラへ引き渡す。

### 3.3 ハンドラ

ローカルマネージャによって ECJ<sub>TM</sub>イベントの ID に応じて起動される処理手続きである。このハンドラは抽象クラ

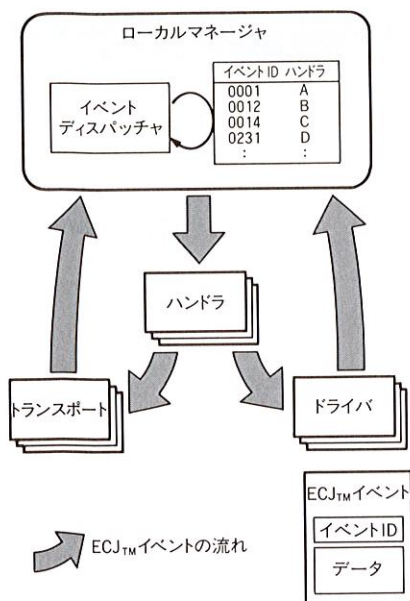


図2. ECJ<sub>TM</sub>の機能構成 ECJ<sub>TM</sub>は、イベント、ローカルマネージャ、ハンドラ、ドライバ、トランスポートから構成される。  
Functional architecture of ECJ<sub>TM</sub>

スとして定義されており、それぞれのECJ<sub>TM</sub>イベント特有の処理ロジックを継承して記述する。

### 3.4 ドライバ

既存の制御システムなどで独自に規定されたイベント群をECJ<sub>TM</sub>イベントに変換する、あるいはその逆変換を行うドライバである。ドライバは抽象クラスとして定義されており、この実装は制御システムのベンダーあるいはドライバのサードパーティベンダーが供給することになる。

また、ECJ<sub>TM</sub>イベントのイベントIDの空間は、これらのベンダーごとに利用可能な領域が予約されているため、複数の異種のシステムを統合する場合にもイベント空間の衝突による混乱を避けることが容易である。

さらに、ECJ<sub>TM</sub>の実行系は動的に実行時にドライバを追加あるいは削除することを許している。このため、ユーザーによるドライバの選択や変更が可能となり、システムの動的な統合あるいは変更が容易になる。

### 3.5 トランスポート

ECJ<sub>TM</sub>イベントをネットワーク経由で配送する機構を提供する。すなわち、異なる計算機上で実行されるECJ<sub>TM</sub>の実行系間のイベント通信路として機能する。

このトランスポートは、前述のドライバの一種として定義される。このため、ドライバと同様にユーザーは通信層のプロトコル変更などを容易に行える。

## 4 ECJ<sub>TM</sub>の動作例

ここでは、ECJ<sub>TM</sub>の簡単な適用例として共有ホワイトボー

ドアプリケーションを考える。

共有ホワイトボードアプリケーションとは、ネットワークにまたがる複数の計算機上で、同時に複数のユーザーによる描画が可能な電子黒板のようなものである。ECJ<sub>TM</sub>のIPマルチキャストのサポートと、マウスの操作などによる大量のイベント対応能力を示す例題である。

データの流れ、すなわちイベントの流れは以下に示すものとなる。この例題では、ポインティングデバイスの位置の変化がイベントである。つまり、ある計算機上でユーザーがマウスを操作して絵を描くと、カーソルの位置変化が発生し、この変化がイベントとして他の計算機上に伝わり、すべての計算機上で同じ絵の再生が可能になる。図3にこのアプリケーションのイベントの流れを示す。

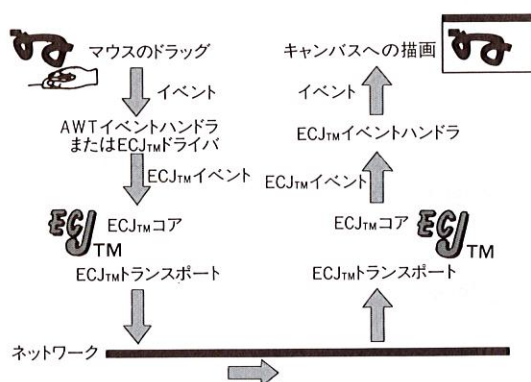


図3. ホワイトボードアプリケーションにおけるイベントの流れ イベントの流れは、マウスの操作から始まり、ECJ<sub>TM</sub>を経由し、キャンバスへの描画で終わる。

Event flow of whiteboard application

AWTとはJavaが提供するGUI (Graphical User Interface) ライブラリフレームワークであり、ユーザーによるマウスの操作などはこのAWTを経由してシステムに配送される。このアプリケーションでは、マウスのボタン押下げによるイベントの発生を逐次ECJ<sub>TM</sub>のイベントに変換する。変換は、ECJ<sub>TM</sub>イベントのデータとしてマウスの位置情報を格納することで実現される。

このECJ<sub>TM</sub>のイベントをトランスポートに引き渡すことで、トランスポートはネットワーク上にまたがる他の計算機上のECJ<sub>TM</sub>に通信メッセージを送信する。これにより、他の計算機上へECJ<sub>TM</sub>イベントを転送できる。

ECJ<sub>TM</sub>イベントを受け取った計算機は、イベントに含まれるマウスの位置情報を取り出し、AWTへの描画命令に変換する。これにより、ネットワークにまたがる共有ホワイトボードアプリケーションが構築可能となる。

このアプリケーションを、LAN上のPentium<sup>(注3)</sup> 200 MHz相当のPC/AT互換機で動作させたところ、ネットワークの

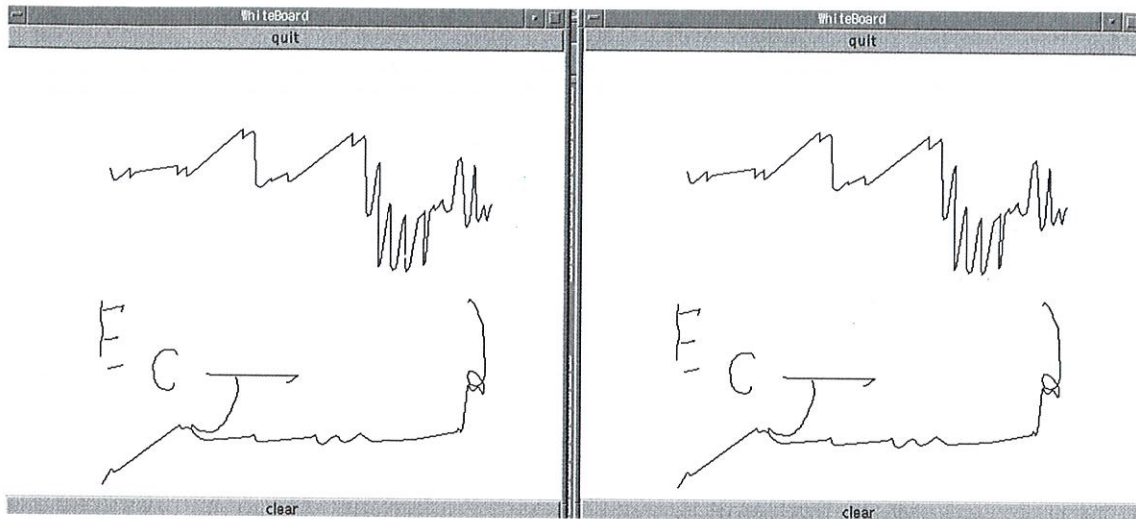


図4. ホワイトボードの実行画面 両方のホワイトボード上に同様の図形が描画されている。  
Snapshot of whiteboard application

存在を意識させない速度で描画が可能となった。この画面イメージを図4に示す。

興味のある方は利用いただきたい。WWWページのURLは、<http://www2.toshiba.co.jp/ecj>である。

## 5 あとがき

ECJ<sub>TM</sub>の概要と特長を述べた。

ECJ<sub>TM</sub>はJavaSpaceあるいはInfoBusなどの分散フレームワークで、非同期メッセージを扱うための拡張としても利用可能である。今後、JavaSpacesにおけるRMIの代替通信手段として、あるいはInfoBusをネットワーク拡張する手段としての実現方法を考える。

また、現在、ECJ<sub>TM</sub>の試用版を東芝WWWサーバにて公開している。ここで取り上げた共有ホワイトボードアプリケーションに関してもソースコード付きで配布している。

(注3) Pentiumは、インテル社の商標。



江口 敦子 EGUCHI Atsuko

府中工場 電算機ソフトウェア部。  
分散オブジェクト指向ミドルウェアの開発に従事。  
Fuchu Works



海邊 裕 KAIBE Hiroshi

研究開発センター S&S 研究所。  
分散システム構築技術に関する研究に従事。  
Systems & Software Research Labs.



宮澤 隆幸 MIYAZAWA Takayuki

研究開発センター S&S 研究所。  
分散システム構築技術に関する研究に従事。情報処理学会  
会員。  
Systems & Software Research Labs.