

Java 高信頼化ミドルウェア「MISAKI」の開発*

4U-06

豊岡 明 神余 浩†

荒井兼秀††

三菱電機(株) 産業システム研究所‡

Mitsubishi Electric Research Laboratory

1 はじめに

産業用途向けデバイスにおいては、ソフトウェアで障害が発生した場合でも異常状態のまま停止せず、再起動等による機能の回復や自己診断等を行うことが求められる。また、障害の要因を特定し改修を行う手掛かりとなる障害情報を、収集・保存する必要がある。

この要求に対応するため、従来の産業用途向けデバイスで動作する Java アプリケーションでは、想定しうる全ての障害への対処手続きをアプリケーション内に作り込む手法が取られてきた。¹しかし、Java の特徴でもある、ソフトウェアコンポーネントの動的なダウンロードと実行が行われる環境においては、複数コンポーネントの組み合わせ、デバイス間の非互換性等に起因する障害が発生しうる。このような環境では全ての障害を前もって想定することは難しく、上記の手法では解決できない。

この問題を解決するため、Java アプリケーションで発生した全ての障害を検出し、障害対応処理を行うことを可能とするミドルウェア「MISAKI」を開発している。MISAKI を使用することにより、従来の手法では網羅できずに見逃されていた障害の発生を検出することが可能となる。また、検出した障害の情報をログ情報として記録すると共に、障害対応処理を行うことが可能となる。

2 機能

MISAKI の機能を以下に示す。

* Development of a high-reliable Java middleware: MISAKI

† Akira Toyooka, Hiro Kanamaru

†† Kanehide Arai

‡ Industrial Electronics & Systems Laboratory, Mitsubishi Electric Corporation

(1) 障害検出

MISAKI の障害検出機能は、Java アプリケーションの実行過程において発生し、アプリケーション内でキャッチされなかった例外とエラーを一括してキャッチし、障害と認識する。また、キャッチした例外とエラーの情報を基に障害対応処理を実行するとともに、MISAKI のログ機能を用いて障害情報をログ出力する。

なお、MISAKI を使用して Java アプリケーション内で発生した例外・エラーを検出する場合、Java アプリケーション側に MISAKI 上で動作させるための特別なインタフェース等は必要ない。MISAKI は、単体で動作することを想定して作成された通常の Java アプリケーションに適用できる。

(2) ログ

MISAKI の障害検出機能が検出した障害情報と、Java アプリケーションが生成したログ情報をログ出力先に出力する。ログ出力先は、MISAKI の設定に基づき決定される。具体的には、ローカルのファイル、標準出力、ネットワーク上のサーバ、デバイスのベンダが追加するログ出力モジュールが挙げられる。

Java アプリケーションが生成した情報を MISAKI のログ機能を用いてログ出力するには、Java アプリケーション側に MISAKI のログ機能を呼び出すためのコードが必要である。

(3) 障害対応処理

障害検出機能が検出した障害情報から行うべき障害対応処理を選択・実行する。障害情報と障害対応処理の対応付けはデバイスのベンダが行う。具体的な障害対応処理としては、デバイスの再起動、アプリケーションの実行停止、ソフトウェアコンポーネ

ントの削除等がある。

3 構成

MISAKI を適用したシステムの全体構成を図 1 に示す。MISAKI は Java アプリケーションと共に Java 実行環境上で動作する。

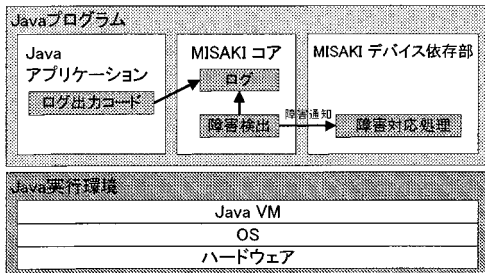


図 1 MISAKI 適用システムの全体構成

MISAKI は MISAKI コアと MISAKI デバイス依存部から構成される。MISAKI コアは障害検出機能とログ機能を実現する。MISAKI コアは Java の機能のみを使用して実現されており、適用デバイスには依存しない。MISAKI デバイス依存部は適用デバイスに依存する機能を実現する部位であり、デバイスの開発者がデバイス毎に開発する必要がある。

MISAKI コア内のログ部は、Java アプリケーションおよび MISAKI コア内の障害検出部からのログ情報を受け取り、ログ情報の文字列への加工とログ出力先への出力を行う。

MISAKI コア内部の障害検出部は、Java 実行環境の機能を用いて Java アプリケーション内で発生した例外およびエラーの検出を行う。また、検出した例外およびエラーの情報をログ部にログ出力すると共に、同情報を MISAKI システム依存部内の障害対応処理部に通知し、障害対応処理の実行を実現する。

MISAKI システム依存部内の障害対応処理部は、MISAKI コア内の障害検出部から障害情報を受け取り、必要に応じてデバイスに特有の障害対応処理を行う。

4 模擬デバイスへの適用

本ミドルウェアの有効性を検証するために、PC 上に構築した模擬デバイス環境内の Java VM 上で MISAKI を動作させ、Java アプリケーションで障害が発生した際の動作を確認した。

確認では、メモリリークの障害を有する Java アプリケーションを、MISAKI あり・なしそれぞれの環境で動作させた。MISAKI の有無以外のハードウェアおよびソフトウェアの条件は同一とした。

(1) MISAKI なしの環境

Java アプリケーションを実行し一定時間が経過すると、Java アプリケーションのスレッドは OutOfMemoryError により終了した。模擬デバイスはハングアップ状態に陥った。

(2) MISAKI ありの環境

Java アプリケーションを実行し一定時間が経過すると、Java アプリケーションのスレッドは OutOfMemoryError により終了した。その後 MISAKI がこれを検出し、障害の発生をログに出力するとともに Java VM および Java アプリケーションの再起動を行った。模擬デバイスは、一旦ハングアップ状態に陥ったものの、Java VM と Java アプリケーションの再起動とともに再び正常状態に復帰した。

5 おわりに

Java アプリケーションでの障害の発生を検出し、障害対応処理と障害情報のログ出力を実現するミドルウェア「MISAKI」を開発した。また MISAKI を PC 上に構築した模擬デバイス環境で動作させ、その有効性を確認した。

参考文献

- 1) Sun microsystems: JAVA EMBEDDED SERVER SOFTWARE,
<http://www.sun.com/software/embeddedserver/>