

M-001

XML ルール圧縮による監視コンポーネント向け軽量ルールエンジン A Light-Weight Rule Engine for Video Security Device based on XML Rule Compression

阿倍 博信† 中島 宏一 峯村 治実†
Hironobu Abe Koichi Nakashima Harumi Minemura

1. はじめに

近年の防犯意識の高まりにより、監視カメラを用いた映像監視システムが普及しつつある。特に、映像符号化技術、IP ネットワーク技術の進歩により、高精細映像のデジタル IP 伝送を可能としたネットワーク映像監視システムが注目されている[1]。映像監視業務では、監視の見落としなどによる重大事故を発生させないための機能強化が求められる一方、誤報の抑止による業務効率化、コスト削減などのニーズが高くなっている。

上記要求に対するアプローチとして、近年のセンサ技術や映像処理技術の発展により、映像監視システムにおいても、各種センサからの出力や監視カメラで撮影した映像データの映像処理結果を元に様々なイベントを発生させ、その結果を処理することによるシステムの効率化が可能となりつつある。例えば、映像処理結果と関連するセンサの出力を組み合わせてることによりイベントの信頼性の向上を図ることができる[2]。

本論文では、カメラやレコーダなどの組み込みシステムを前提とする監視コンポーネントに搭載し、イベント処理を直接実現する軽量ルールエンジンの開発、及び評価について述べる。

2. 技術課題

一般的な映像監視システムでは、各種センサの他、組み込みシステムを前提とするカメラ、レコーダ、ビューワなどの装置(以下、監視コンポーネント)と制御 PC から構成される。制御 PC にて監視カメラや各種センサで発生したイベントを処理し、その結果を元に、各コンポーネントの制御(エンコード制御、蓄積制御、表示制御など)を行う。この場合、監視カメラやセンサなどから発生した全てのイベントを制御 PC で処理することになるため、システムコストが高くなるとともにイベントが集中したときのシステムのレスポンス低下が課題であった。

3. 軽量ルールエンジンの開発

2.で設定した技術課題に対して、カメラやレコーダなどの監視コンポーネントに搭載し、制御 PC を使用せずコンポーネント内部でのイベント処理を実現する軽量ルールエンジンを開発した。

3.1 システムの概要

図 1 に本ルールエンジンを適用した映像監視システムのシステム構成について示す。

本システムの特長について以下に示す。

- ・ルールを XML で記述しているため、市販またはオープンソースのツール(XML エディタや XML パーサ)を利用

でき、開発コストを削減できる。

- ・監視コンポーネントはリソース(CPU、メモリ)が限られているため、XML エディタによるルールの作成や XML パーサによるルールの解析は管理用 PC で行う。
- ・ルールの実行部分(ルールエンジン)は監視コンポーネントで動作し、汎用部とデバイス依存部から構成される。

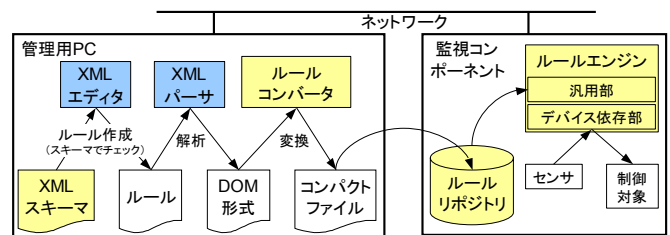


図 1 システム構成

3.2 ルール処理フロー

本システムでは、XML エディタや XML パーサなどの XML 処理系は、管理用 PC に搭載し、XML エディタを用いた動作ルールの編集や XML パーサによるルールの解析は管理用 PC にて行い、その解析結果を各監視コンポーネントに配布する。また、動作ルールの監視コンポーネントへの格納形式は、DOM などの標準形式を用いる方法と、独自形式を用いる方法が考えられる。一般的に動作リソースが限られている監視コンポーネントでは、DOM 形式でルールを格納すると、監視コンポーネントのコストアップにつながりかねない。今回は、管理用 PC にてルールを XML パーサで解析して得られた DOM 形式のデータを、動作ルールに特化した独自のコンパクト形式に変換し、監視コンポーネントのルールエンジンにはコンパクト形式で格納する。

3.3 ルールコンバータ

管理用 PC で動作し、XML 形式で記述された動作ルールを読み込み、XML パーサによるスキーマチェックを行い DOM 形式にて内部メモリに展開後、コンパクト形式へ変換・出力する機能を持つルールコンバータを Windows アプリケーションとして実装した。

3.4 ルールエンジン

ルールエンジンは、将来的な監視カメラへの搭載を想定し、一般的な監視カメラと同等の CPU 性能(Arm 200MHz)を持つ組み込み Linux 環境[3]上に実装した。

監視コンポーネントは組込み機器のため、メモリなどのリソース量が限られているため、ルールエンジンは軽量でなければならない。このため、目標メモリサイズを通常の XML パーサを使用する場合の 25%削減に設定した。また、監視コンポーネント上のルールエンジンで動作ルールを解釈し、外部や内部で発生したイベントに対する制御をリアルタイムに行う必要がある。このため、イベントの発生から制御開始までの目標応答性能を PC にてイベント処理を実行する場合と同等の性能に設定した。

†三菱電機(株), Mitsubishi Electric Corporation

今回実装した軽量ルールエンジンの機能は以下の通りとなる。

- ① イベント (条件) の発生をチェックする。
- ② 発生した条件を満たすルールをルールリポジトリから検索し、対応するアクションを見つける。
- ③ アクションを実行する。

アクションを実行中でも別のイベントが発生しうするため、これらの3つ機能をそれぞれ別のスレッドで実行する。また、ルール検索処理はデバイス非依存であるが、条件チェックとアクション実行はデバイス非依存の汎用部とデバイス依存部分が混在する。上記検討により、軽量ルールエンジンのモジュール構成を図2の通りとする。

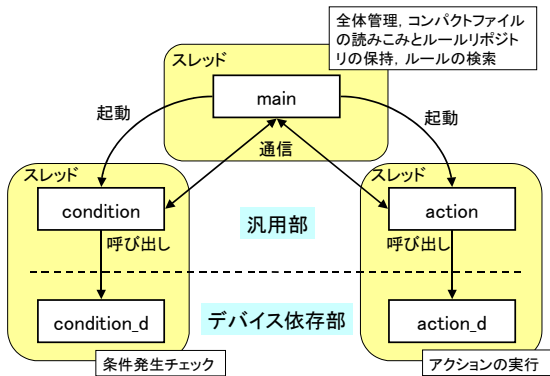


図2 軽量ルールエンジンのモジュール構成

4. 評価

実装した軽量ルールエンジンの処理性能とメモリ使用量について下記の通り評価を行った。

4.1 評価用ルールセット

評価に用いるルールは、サイズの異なる以下2つのルールセットを用いた。ルールセットはXML形式にて記述し、ルールコンバータを用いてコンパクト形式に圧縮して使用する。

①ルールセット1

サイズ：XML：207バイト，圧縮後：102バイト
内容：ボタンがONになるとカメラに短いコマンドを送るルール1つのみのシンプルなルールセット

②ルールセット2

サイズ：XML：2093バイト，圧縮後：920バイト
内容：実運用を想定し、4つのOR条件のANDを条件とするルール、4つのAND条件のORを条件とするルール、5つのAND条件のTHENを条件とするルールの3つのルールから構成される複雑なルールセット

4.2 処理性能の評価

4.1の評価用ルールセットを用いて、組み込みLinux環境(Arm 200MHz)上に実装した軽量ルールエンジンの処理性能評価を行った。比較用に、PC(Pentium4 3.8GHz)上にインストールしたLinux上にて軽量ルールエンジンを動作させて性能評価を行った。その結果を表1、表2に示す。

表1 処理性能評価結果(ルールセット1)

| | 初期化処理 (msec) | イベント処理 (msec) | |
|----------|--------------|---------------|-------|
| | | イベント1 | イベント2 |
| Pentium4 | 8.9 | 2.2 | 0.1 |
| Arm | 16.4 | 1.3 | 0.2 |

表2 処理性能評価結果(ルールセット2)

| | 初期化処理 (msec) | イベント処理 (msec) | | | |
|----------|--------------|---------------|-------|-------|-------|
| | | イベント1 | イベント2 | イベント3 | イベント4 |
| Pentium4 | 10.1 | 1.5 | 1.4 | 1.6 | 1.8 |
| Arm | 31.9 | 1.4 | 1.5 | 1.5 | 1.5 |

表1、表2の結果を見ても分かる通り、組み込み環境の場合、コンパクト形式のファイルの読み込みなどをはじめとする初期化処理に処理時間がPC環境の倍程度かかっているが、一旦メモリ内にルールを展開してしまえば実時間処理に必要なイベント処理はPC環境とほぼ同程度の性能が出ていることが分かった。

4.3 メモリ使用量の評価

4.1の評価用ルールセットを用いて、組み込みLinux環境(Arm 200MHz)上に実装した軽量ルールエンジンのメモリ使用量評価を行った。比較用に、同じ環境に通常のXMLパーサであるlibxml2を用いてXML形式で記述されたルールを読み込む評価プログラムを作成して評価を行った。その結果を表3に示す。

表3 メモリ使用量の評価結果

| | ルールエンジン (Arm版) | libxml2 (Arm版) |
|---------|----------------|----------------|
| ルールセット1 | 160KB | 208KB |
| ルールセット2 | 156KB | 232KB |

表3の結果を見ても分かる通り、軽量ルールエンジンを使用することにより、通常のXMLパーサを使用する場合と比較してメモリ使用量の削減が可能であることが分かった。特に、XML形式で記述されたルールセットが長いルールセット2の場合は特にデータ圧縮によるメモリ削減効果について確認できた。

5. おわりに

本論文にて、制御PCではなくカメラやレコーダなどの監視コンポーネントにて、動作ルールに基づき自律的にイベントを処理することによりレスポンス向上を目的とした軽量ルールエンジンを開発した。

組み込みLinux環境(Arm 200MHz)上に軽量ルールエンジンを実装し、その処理性能及びメモリ使用量について評価を行った。評価実験の結果、組み込みLinux向けに開発した軽量ルールエンジンは、XML形式で記述されたルールを独自形式に圧縮することにより、組み込み向けCPUでも軽量に動作することが確認でき、目標性能(メモリ使用量、応答性能)を達成できたため、実時間処理が必要な組み込み機器にも広く適用が可能であることを検証できた。今後は、本ルールエンジンの実用化に向けた評価を行っていく予定である。

参考文献

- [1] 矢野経済研究所：2008～09年版 ビジュアル・コミュニケーションシステム市場 Vol.1 ネットワークカメラ編 (2007)。
- [2] 山田陽一，山口政巳，本玉靖和：ユビキタスセンサネットワークにおける映像監視システム，沖テクニカルレビュー，Vol. 72 No. 4 (2005)。
- [3] アットマークテクノ Armadillo-9：http://www.atmark-techno.com/products/armadillo/a9