

Smarter Planetを実現するための 統合機器管理アーキテクチャー

秋山 一人 西村 康孝 古市 実裕

A Design-Integrated Device Management Architecture for a Smarter Planet

Kazuhiro Akiyama, Yasutaka Nishimura and Sanehiro Furuichi

Smarter Planet の実現のため、多数かつ多種多様の機器からデータを収集し、適切に制御することのできる新しい統合管理アーキテクチャーを提案する。本提案ではゲートウェイと呼ぶ中間処理層を介して通信を行うことにより機器や管理製品の抽象化を実現している。これを多数のオフィス機器を抱えるお客様のパイロット・システムに適用し検証した結果、その有効性を実証できた。ここで提案したアーキテクチャーはソフトウェア製品としてリリースされ、お客様の実業務でも採用されることになった。

To realize the Smarter Planet initiative, this paper proposes a new integrated management architecture to collect data on efficiency from various types of intelligent devices in order to realize proper control of such devices. By communicating through an intermediary-layer gateway, the various devices and management products can be abstracted. The authors deployed this architecture in a pilot project on an actual client's site and verified its validity as well as its value for the client. Thus, this architecture has been released as a software product and deployed in the client's commercial environment.

Key Words & Phrases : デバイス管理, 機器管理, オフィス管理, CO₂ 管理, Smarter Planet, Smarter Office device management, office management, CO₂ management, Smarter Planet, Smarter office

1. はじめに

あらゆるものが互いに接続されて連携可能な状態になる” Interconnected” は、Smarter Planet における重要なコンセプトである。さまざまな情報が集約、統合されることによって新たな知見が得られ、よりスマートな意思決定が可能になる。近年、高度化、複雑化しているさまざまな機器から得られる情報を有効に活用し、適切に機器を制御することにより、新たな価値を提供することが期待されている。しかし、現状の機器は IT 管理システムと有機的に統合されていない。データ収集や機器制御は、機器ごとに局地的に行われており、統合管理のための課題は多い。

本論文では、多数かつ多種多様な機器が存在し、お客様から統合管理の要望が強いオフィス機器環境に焦点を当て、それらの機器を管理する上での課題と要件を明確にし、新たな要件に答える統合機器管理アーキテクチャーを提案する。

これにより、効果的な統合管理が可能となり環境負荷

やコスト削減、サービスの向上のような新たな価値を提供できるようになった。

2 章で機器統合管理の課題と要件を明確にし、3 章で統合機器管理アーキテクチャーの概要、4 章で株式会社リコー様と共同で行ったパイロット運用により、このアーキテクチャーを評価した結果および考察を述べる。

2. 機器統合管理の課題と要件

実際に統合管理の要望を受けた米国の大企業のオフィス環境を想定し、数万台に上る PC やプリンターなどのオフィス機器を統合管理する上での課題と要件について述べる。

2.1 機器の抽象化

機器管理においては、管理対象の数が膨大になるだけでなく、多種多様の機器を管理する必要がある。本論文のパイロット運用で管理対象とした PC やプリンターの場合、通常多数のメーカー、モデルが混在し、それぞれ使用している通信プロトコルやデータ形式が異なる。

また、各機器から送信されるデータは粒度が著しく異なる上、ネットワークに常時接続されていない機器も多く存

提出日:2010年9月6日 再提出日:2011年9月6日

在し、データの収集間隔を揃えることも困難である。

これらの機器を統合管理するためには、管理インターフェースを標準化することが最も知られた手法であるが [1]、既存の機器を含めすべての種類の機器で単一のインターフェースを持つことは現実的ではない。機器には SNMP [2] などの標準プロトコルが存在するが、プリンターなど特定の機器の種類に対するものであり、多種の機器の統合管理で単一のものとはならない。

Enterprise Service Bus (ESB) はサービスに対して抽象化階層を提供するアーキテクチャーであるが、大量で、しかも形式や粒度の異なる機器データを直接扱うのは困難である。

あらゆる機器を統合管理する目的のためには、述べてきたような機器の多様性を吸収する新しい抽象化の手法が必要である。

2.2 管理プロセスへの統合

IT 管理を効率よく行い、質の高いサービスを提供するためにさまざまな管理製品が提供されている。しかしこれらは従来の管理対象であるデータセンターやサーバーに対してのみ機能を提供し、機器に対しては直接のインターフェースを持たない。現在、お客様は機器に対してのみ独自の管理プロセスを行うことを余儀なくされている。近年 IT システムでは、CO₂ の排出管理や節電規制、情報漏えいなど法律で規制されるケースが存在する。機器管理においても IT システム管理と同様に、ITIL [3] のようなベスト・プラクティスに従った管理や、ISO14000 [4] のような確立された管理プロセスに従うことが求められている。そのためには、あらゆる機器が既存の管理製品から透過的に管理されることが必要である。

2.3 機器制御と自律化

現在提供されている機器管理製品の多くは、機器からさまざまな情報を収集、集計、可視化して問題を発見することができる。しかし、その問題が発見された機器に対して、効果的な改善策を施すことができないため、効果的なソリューションとはなっていない。「見える化」だけではなく、機器の適切な制御まで行えるアーキテクチャーが必要である。

また、管理対象の機器が多数であり、ネットワークに常時接続されていない機器があることから、サーバーから逐次制御を行うことは非現実的である。機器自身で自律的に制御を行う手法が強く求められる。

3. 新しい統合機器管理アーキテクチャー

ここでは 2 章で述べた問題を解決する統合機器管理アーキテクチャーを提案する。初めに設計方針および概要を説明し、その改善した個々の特徴について述べる。

3.1 アーキテクチャーの設計方針

可能な限り多くの種類の機器を管理対象とすること、最少の作業量で新しい機器に対応できること、さらに多数の機器、大量のデータに対処するため、以下のような方針で設計を行った。

- 標準に従った機器はそのまま管理対象となり、それ以外の機器は抽象化されて標準に従った機器と同様に管理される。
- 抽象化処理をプロトコル、データ形式、データ収集の間隔など、機器の多様性のカテゴリで階層化し、各階層でそれぞれの機器の種類における（事実上の）標準に対処することで、管理可能な機器の種類を最大化し、新しい機器に対する固有の実装を最小化する。
- 各階層で共通な処理は汎用化し、構成パラメーターや、XML などによる定義により柔軟に必要な処理ができるように実装する。
- 共通の処理はできるだけ下位（機器側）の階層で行い、上位に渡るデータ量および上位階層の処理量を減らす。

3.2 アーキテクチャーの概要

図 1 にこのアーキテクチャーの構成図を示す。

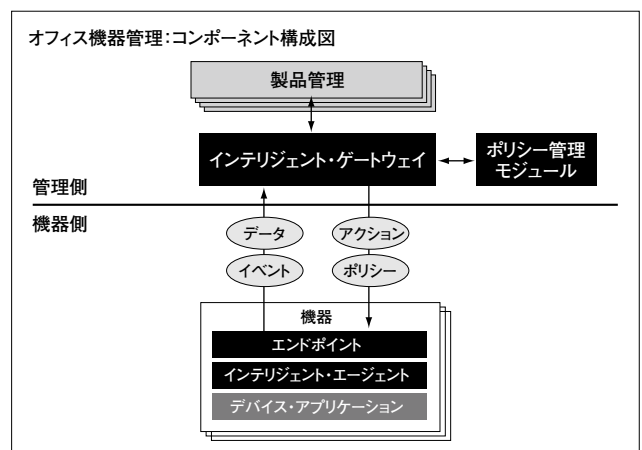


図1. アーキテクチャーの構成図

以下は主要なコンポーネントである。

- インテリジェント・ゲートウェイ
このアーキテクチャーの根幹となるモジュールであ

る。機器を抽象化する階層となり機器および管理製品と通信を行う。以下ゲートウェイと表記する。

- エンドポイント
機器側の実装される通信モジュールで、ゲートウェイと通信を行う。
- インテリジェント・エージェント
機器側の実装されるモジュールである。管理サーバー側で作成されたポリシーにより機器の制御を行う。以下エージェントと表記する。
- ポリシー管理モジュール
機器を自律的に制御するポリシーを管理するモジュールである。

3.3 アーキテクチャの特徴

3.3.1 標準インターフェースの採用

どの機器も、管理側に対して標準化された単一のインターフェースを持つことが望ましい。現在複数の標準化団体によって IT 管理インターフェースの標準化が進められており、このアーキテクチャーでは標準インターフェースおよびデータ形式として WS-Management (以下, WS-Man), CIM [5] の仕様を中核として採用している。

このインターフェースを実装した機器であれば、ゲートウェイを介しての管理が可能となる。しかし実際にはこれから標準インターフェースを持つ機器はまだ少なく、ハードウェアのリソースなど機器の制限や以前の製品との互換性の問題により標準化することが困難な機器が存在する。次節でその対応を述べる。

3.3.2 階層化された抽象化处理

標準を採用していない、またはできない機器に対しては、ゲートウェイで機器の抽象化处理を行い、管理製品からはどの機器も同じインターフェースを持つようにする必要がある。図 2 にゲートウェイの階層化されたコンポーネント

構成図を示す。

以下に具体例を挙げながら各コンポーネントでの抽象化について述べる。例では機器からのデータ受信について述べているが、機器へのアクション送信は逆順に同様の処理が実行される。

1) Entry Point

機器と通信を行いプロトコルの抽象化を行う。WS-Man インターフェースだけではなく、既存の機器に使用されている標準プロトコル (HTTP, WSDM [6]) などに対応している。ここでは各プロトコルから取得したデータを次コンポーネントに渡す。標準プロトコルを実装できない機器に対しては、独自の Adapter を実装して対応する。例えば既に SNMP プロトコルに対応しているプリンターには、新たに SNMP Adapter を実装することになる。

2) Data Normalization

CIM 形式ではない CDM [7], MIB [8], CSV などのデータ形式をゲートウェイ標準のデータ形式である CIM に変換する。データの粒度が異なるもの (電力の単位など) は CIM の定義に従うようにこの時点で統一される。また、上位コンポーネントで不要な属性は、定義に従いここで削除される (フィルタリング機能)。データの欠如や異常もここで検出される。

3) Data Mapping

標準のデータ形式においても、個々の機器メーカーの実装によってデータの意味や要素名が異なる場合がある。例えば CIM でプリンターの印刷ジョブを記述する場合を考える。印刷枚数を示す属性は CIM クラスでは拡張属性となっているため、各メーカーで異なった要素名を使用した実装になり集計が困難になる。これらの違いを抽象化するため、各 CIM クラスに抽象化クラスを定義する。メーカーの実装クラスと、この抽象化されたクラスとの要素名マッピングを定義することにより、ここで要素名を変換し、統一することが可能になる。図 3 にマッピング定義の例を示す。

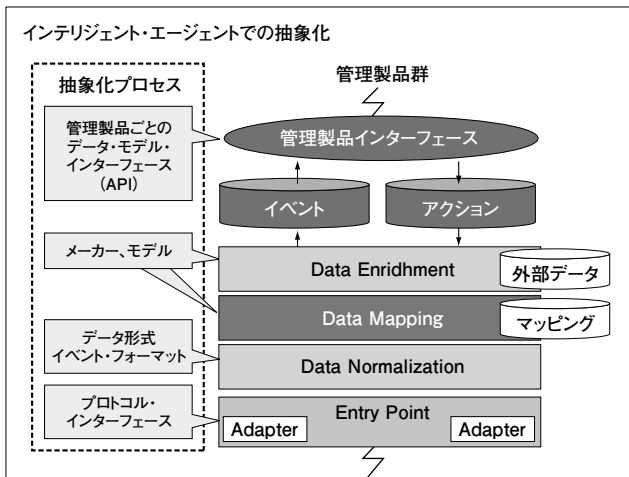


図2. コンポーネント構成図

```

<Definition>
<Vendor>
<Name> xxxx </Name>
<Class>
<AbstractName> CIM_PRINTJOB </AbstractName>
<Description> This is used for WS -Enumerating </Description>
<VendorClassName> Rxxx_PrintJob </VendorClassName>
<Attribute>
<AttrName> EVENT_ID </AttrName>
<VendorAttrName> JobID </VendorAttrName>
</Attribute>
<Attribute>
<AttrName> EVENT_TIME </AttrName>
<VendorAttrName> TimeSubmitted </VendorAttrName>
</Attribute>
<Attribute>
<AttrName> NUM_OF PAPERS </AttrName>
<VendorAttrName> UsedSheetsNumber </VendorAttrName>
</Attribute>

```

図3. Data Mappingの定義例

この例では、定義されたメーカーの機器から送られてくる印刷ジョブ情報では、要素名 <JobID> の内容が抽象化クラスにおける <EVENT_ID> の要素として使用される。

4) Data Enrichment

機器から収集可能なデータは限られている。例えば印刷ジョブには機器のシリアル番号や印刷を行ったユーザー ID は含まれているが、保守に使用するためには機器がどの場所に設置されていて誰が管理者かという情報が必要であり、印刷枚数を集計して部門に課金するためには印刷を行ったユーザーがどの部門に含まれているか、という情報が必要である。

従来、これらのデータ統合は管理製品側で行われていたが、これをゲートウェイで実装することにより各管理製品での処理が軽減され、アプリケーション実装が容易になる。データ統合は定義に従い、収集したデータに外部データベースのデータを付加することにより行われる。図 4 に定義例を示す。

```

<Definition>
  <AbstractClass >
    <Name>CIM_PRINTJOB</Name>
    <TargetTable >
      <Name>USER</Name>
      <SourceColumn >USER_ID</ SourceColumn >
      <SourceDataType >STRING</ SourceDataType >
      <TargetAttr >USER_ID</ TargetAttr >
    </TargetTable >
    <Column>
      <Name>DEPT_CODE</Name>
      <Type>STRING</Type>
      <TargetAttribute>DEPT_CODE</TargetAttribute>
    </Column>
  </AbstractClass >
</Definition>

```

図4. データ統合の定義例

この例では、各印刷ジョブに含まれる USER_ID の値をキーとして外部 USER テーブル中にあるレコードを取得し、DEPT_CODE フィールドのデータ（組織名）が、印刷ジョブの DEPT_CODE フィールドに追加される。この処理により上位の管理製品で組織別の集計や解析が可能となる。

5) イベント、アクションのキャッシュ

処理されたデータは必要に応じてキャッシュに保存される。データ収集間隔が一様ではなく、上位側の処理でデータがそろふ必要がある場合には、データは蓄積され、収集が完了した時点で上位コンポーネントに通知される。

6) 管理製品インターフェース

機器のデータを取得し、制御を行う管理製品は被管理側に Soap や Java API、データベースやファイル経由などのインターフェースを提供しているが、製品ごとに異なっ

ているのが現状である。管理製品に対する個々のインターフェースを実装することで、機器はどの製品へもデータが送信でき、制御されることが可能になる。個々の管理製品が必要とするデータ変換や集計処理なども行われる。

3.3.3 ポリシーベースの機器制御

機器の制御を行うエージェントは、組み込み Autonomic Computing アーキテクチャー [9] に従ったモジュールである。上位管理製品によって定義、作成された制御ポリシーは、ゲートウェイにより機器に送信され機器を自律的に制御する。オフィス機器管理では、ユーザーの印刷操作に対する制御や PC、複合機に対する電源制御、構成変更などを行うことを想定している。

4. オフィス管理ソリューションによる検証

3章で述べたアーキテクチャーは、オフィス管理ソリューションとしてリコー様と共同で開発を行い、米国のお客様においてパイロット運用を行った。この章ではパイロットの概要とアーキテクチャーの検証結果について述べる。

4.1 パイロットシステムの概要

図 5 にシステムの構成図を示す。

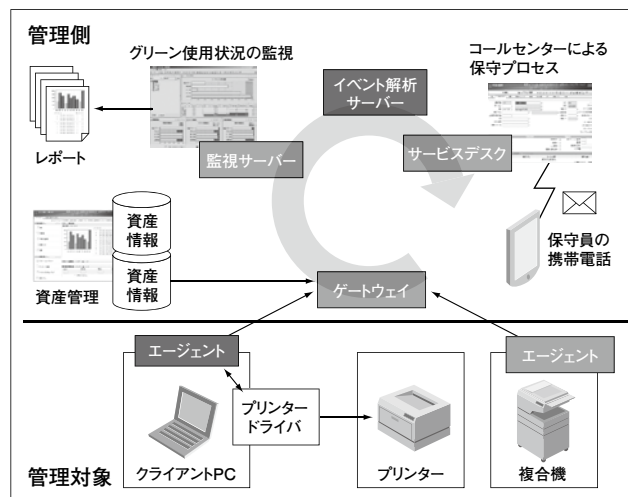


図5. ソリューション構成図

管理側はゲートウェイを除いてお客様が既に使用し、運用している管理製品である。

4.1.1 ソリューションのシナリオ

a) CO₂ の排出量改善

オフィス機器から印刷処理や電力消費によって排出される CO₂ を算出、可視化し、発見された問題点に対してユーザーに対する印刷制限や複合機の電源管理を行う。

b) 複合機の障害保守

複合機で発生する障害（用紙ジャム、保守員コール）を受信し、サービスデスクに通知して保守プロセスを実行する。

4.1.2 管理対象の機器

このパイロットでは以下の機器を対象とした。

- 複合機（リコー製）
リコー様のエージェントを導入した。
- クライアント PC（以下、PC と表記）
IBM アセットである制御モジュール [10] をベースとし、このアーキテクチャーに沿った拡張を行ったエージェントを導入した。
- プリンター
今回のパイロットではゲートウェイとプリンターが直接通信する実装を行わず、PC 上のプリンター・ドライバーから印刷情報を取得する制御モジュールの機能を使用した。

4.1.3 ソリューションの実現

a) CO₂ の排出量改善

図 6 に収集されたデータの表示画面を示す。印刷、電源情報を機器から収集し、消費された用紙やトナー、電力量を CO₂ 排出量に換算、組織やユーザー、機器別に集約して表示している。CO₂ 排出に関する片面印刷やカラー印刷などの印刷オプション、設定した目標に対する進捗などを表示することにより問題点および改善策が検出される。

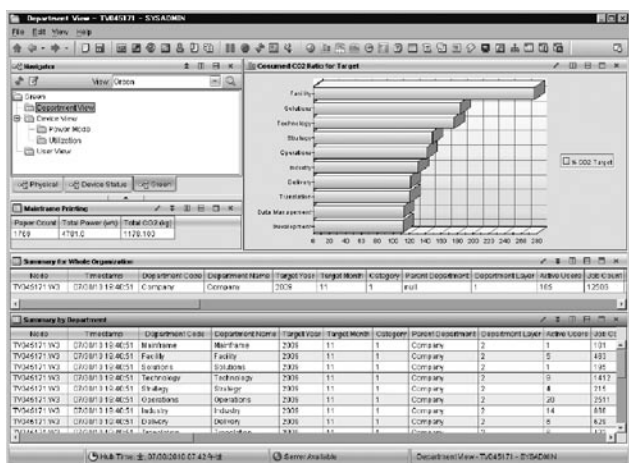


図6. 印刷状況の表示

b) 複合機の障害保守

複合機からの障害イベントに資産情報（位置、管理者情報）を付加してサービスデスクに転送する。自動化されたプロセスにより、保守に必要な情報を含んだメールが保守員の携帯電話に送信される。

4.2 検証結果および評価

提案した新しいアーキテクチャーの特徴について、それぞれ検証した結果及び評価を述べる。

4.2.1 標準インターフェースの採用

リコー製複合機とは標準インターフェースである WS-Man を介して通信を行い、データ形式は CIM に従った実装を行った。機器側の実装はリコー様で行い、インターフェース仕様の策定のみ共同で行った。

パイロットの結果、データ収集、機器制御とも要求された機能を満たすことが実証できた。標準に従った実装により統合機器管理が容易になることが確認できた。

4.2.2 階層化された抽象化处理

今回のパイロットでは複合機のほかに、PC 及びそれを介したプリンターを管理対象とした。これらは既存の HTTP プロトコルと CSV 形式のデータを使用しており、各コンポーネントで以下のような抽象化が行われた。印刷ジョブを例として述べる。

1) Entry Point

WS-Man インターフェースで通信する複合機に加え、PC とは HTTP プロトコルで通信した。ここでは複数のプロトコルが期待通り抽象化できることが検証された。

2) Data Normalization

ここで PC から受信した CSV 形式のデータを定義に従って CIM 形式に変換する（データ形式の抽象化）。今回のソリューションで使用しない属性はフィルタリングされる。必須属性の欠如や値の異常はここで検知、処理され、これらのデータは上位には渡されない。

3) Data Mapping

リコー社製複合機用および PC 用にそれぞれマッピング定義を用意する。機器データから定義に従い CIM の印刷ジョブ抽象化クラスの属性を抽出する。この時点ですべての印刷ジョブはメーカー固有の実装に依存しない単一のデータ構造を持つことになる。

4) Data Enrichment

資産情報、ユーザー情報を付加する。複合機からの警告イベントには機器の設置されている店、フロア、位置、管理者などが付加され、印刷ジョブ情報には印刷者が所属する部門情報が付加される。機器データに対しお客様既存の IT 情報を有機的に統合できた。

5) イベント、アクションのキャッシュ

PC は常時接続ではなく、接続された PC からの Polling をトリガーとしてデータ収集、ポリシー配布が行われる。管理製品での表示に必要なデータがそろうまでデータはキャッシュされる。ポリシー配布のアクションも PC が接続されるまでキャッシュで待機する。

6) 管理製品インターフェース

今回使用した IBM Tivoli Monitoring (以下、ITM) が提供する Java クラス・ライブラリーを使用する。CIM データを Java のオブジェクトに変換し、送信メソッドをコールする。

これらの抽象化処理により、ITM 上では複合機からの印刷ジョブも PC を介したプリンターの印刷ジョブも同様に扱うことで、集計やレポート、解析が統合的に行えることを実証できた。

お客様は今まで、これらの目的で多数の機器管理製品を使用していたが、これが ITM のみで行えることを示した。

またこれらの機器は、データの可視化だけではなく、警告状態の検知、判別、サービスデスク製品への通知など、通常の IT 機器と同じ管理プロセスへ組み込むことが可能になった。

4.2.3 ポリシーベースの機器制御

データ収集及び解析の結果、期待されない印刷オプションの使用や、必要がない時間帯に複合機がフルパワー状態で稼働していることが発見された。ここに採用された改善策を示す。

- 1) 環境負荷やコストの低い両面印刷や白黒印刷を推奨するため、業務で必要がない部門やユーザーに対して印刷制限を行う。
- 2) それ以外の印刷に対しても、意図しない片面やカラー印刷の防止として、警告表示を行って確認させる。
- 3) 大量の印刷を行う場合には、個人や部門の複合機ではなく、印刷センターなどにある大規模な複合機にしか出力させない。
- 4) 非就業時間帯にフルパワーで稼働している複合機をスタンバイ状態とする。

これらの制御ポリシーを作成、配布し、期待通りに制御できることを確認した。パイロットでは制御による改善効果までは検証できなかったものの、制御時に管理側と通信を行わず、自律的に制御できることがお客様からは非常に高い評価を得、これが採用へのキーとなった。

4.2.4 課題と要件に対する評価のまとめ

• 機器の抽象化 (2.1 節)

数十台を対象とするパイロットでは多数の管理対象に対する要件の実証はできなかったが、フィルタリングや異常検知の機能により管理製品側に渡るデータ量およびその処理作業が大幅に低減され、数万台対象の実運用に耐えうる実装であると評価された。

ネットワーク非常時接続である PC に対しても、管理製品側では何の考慮をすることなく定期的なデータ取

集が期待通り行えた。

複合機との標準に従った実装によるアプローチが正しいこと、それ以外の場合でも機器の抽象化により異なった種類の機器を統合管理できることを実証し、オフィス機器に対しても、サーバーやデータセンターと同様の管理ができることを示せた。

• 管理プロセスへの統合 (2.2 節)

このパイロットでは、監視やレポート、前述した障害保守のシナリオなど既存の IT 機器と同じ運用を行った。機器は管理製品インターフェースを通じてこれら管理製品の機能からアクセスされる。お客様は今までの IT 機器とまったく同じように機器を扱えるため、お客様自身が管理プロセスを定義、作成できた。これにより機器が容易に管理プロセスに統合されることが検証された。

• 機器制御と自律化 (2.3 節)

問題点の発見に加えて、改善策として機器に柔軟な制御を行うことが検証できた。ポリシーの作成、配布、適用および運用が多数の機器に対してお客様の期待通りに行えた。適切にプロセス管理製品と統合することにより、継続的な改善プロセスの運用や制御の自動化が期待できる。

4.3 お客様における価値

今回のパイロットの結果、このソリューションによる以下のような改善効果が導出された。

- 単一製品による統合管理が可能になり、従来の多数の管理製品の運用作業、コスト低減、さらに要員育成、配置のコスト削減
- 不必要な印刷操作、印刷オプションの制限による環境負荷、印刷コストの改善
- 複合機の電源制御による環境負荷、電力コストの改善
- 機器の障害管理自動化によるサービスデスクの保守コスト、サービス応答時間の改善

5. 統合機器管理アーキテクチャーの有効性と今後への期待

このアーキテクチャーの有効性と今後への期待について述べる。

5.1 アーキテクチャーの有効性

前章で述べたように、考案したアーキテクチャーが目的とする効果的な統合機器管理を行えることが検証できた。

今回のパイロットを通して、多数かつ多種多様な機器情報が資産や保守、人事などのIT情報と有機的に統合されることがお客様に有用な知見をもたらし、さらに機器が制御できることから、効果的な改善策がお客様によって導き出されることが実証できた。今回採用された改善策(制御ポリシー)はすべてお客様の発案によるものである。

以下はパイロットの結果、お客様から受けた今後に向けての要望である。

5.1.1 異なる機器間での連動制御

データを統合的に扱うだけではなく、異なった機器が連動して動作することによりさらにスマートな管理が可能になる。プロトタイプとして実装した以下のPCと複合機の連携シナリオでは、さらなる環境負荷やコストの軽減が期待できる。

- ある部門のPCが全てログオフした場合、その部門の複合機が即時にスタンバイ状態になる

ほかの機器、電灯や空調などの機器と連動すればさらに価値のあるソリューションが提供できる。

5.1.2 ソリューションの拡張

今回はCO₂排出量改善と機器の障害管理のシナリオを実装したが、機器からデータを収集し制御を行う仕組みの上では、以下のようなさまざまなソリューションが提供可能である。

- 機器の消耗品発注管理、在庫管理
- 予防保守、保守員の作業管理
- セキュリティー、コンプライアンス順守

5.2 今後のアーキテクチャーに対する期待

5.2.1 Smarter Planetへの統合

オフィス機器に限らず、すべての機器を統合することがこのアーキテクチャーの目的である。現在ビル管理ソリューションとの統合が進められており、工場や家庭、都市まで管理できる統合機器管理アーキテクチャーとするためには、さらに多様な機器を使用した検証が必要である。

5.2.2 標準化へのドライバー

標準化されたインターフェースが実装されていれば、より容易に統合管理が可能になることは実証された。各メーカーは標準化に興味を持っているが、そのメリットを模索している。

このアーキテクチャーを実装したソリューションの価値を機器メーカーや管理製品開発者に示すことにより、特に新規に開発される機器に対して標準インターフェースの採用を推進することができる。さらに環境負荷の指標となるデータ属性など十分に策定が進んでいない標準化の促進も期待する。

5.2.3 機器での自律管理

ポリシーを送信して制御を行うアプローチは最終的な管理対象が膨大な数になる今回のケースでは非常に有効であった。扱うべきデータが大量になる場合には、さらにインテリジェントな機器上での集約や解析、異常検知などの処理が重要になる。エージェント側に組み込みACアーキテクチャーの自律管理や階層化解析の仕組みを適用、拡張し、よりスマートで効果的な機器管理を可能にすることが期待される。

6. おわりに

パイロット運用により、オフィス機器に対する統合管理ソリューションの有効性、導入効果が実証された。このアーキテクチャーを実装したオフィス機器管理製品としてIBM Tivoli Monitoring Agent for Multi-function Devices がリリースされ、このお客様で採用された。

オフィス機器以外に対しても、幾つかのパイロット適用を行いこのアーキテクチャーの検証を行っているが、以下のような課題も見つかっている。

1) IT 標準とインダストリー標準との差異

このアーキテクチャーはIT管理の標準(DTMF/OASYS)をベースとしているが、例えばセンサーを管理対象としたエネルギー管理ソリューションにおいてはIEC [11] や SEP [12] の対応が必須となる。今後各インダストリーにこのアーキテクチャーを適用するためには、それぞれ要求される標準に適切に対応する必要がある。

2) Data Normalization の課題

対象機器によっては単純に正規化できない形式でデータを送信する場合がある。データ形式がその都度変わるケースや特殊なエラー処理が必要なケースなどが存在する。この対応として各コンポーネントの前後にプリ・プロセス、ポスト・プロセスを置くようなアーキテクチャー拡張を行っている。

3) Energy Management System (EMS) の管理

実際のソリューションにおける管理対象は個々の機器だけではなく、BEMS (Building) や HEMS (Home), FEMS (Factory) などのEMSであることが多い。EMSはそれぞれ自体閉じた管理システムであり、データ収集や機器制御の機能を持つ。大部分のEMSは単体で配下の機器を管理する目的で作成されているため、外部インターフェースが限られている上、独自のプロトコルやデータ形式を使用している。これらのEMSをどのようにこのアーキテクチャーの“標準化”されたインターフェースで統合管理すればよいか、実プロジェクトでのさらなる検証が必要である。

4) ゲートウェイの階層化 (クラウド対応)

実際のソリューションでは、管理製品をクラウド上に置いてサービスを提供したいという要望が非常に強い。エネルギー管理など複数ビルや都市全体の管理ではクラウド対応は必須になる。個々の機器がクラウド上の管理側と通信することは現実的ではなく、階層構造の機器管理が要求される。ゲートウェイの階層化に対しては、単純なゲートウェイ間通信のインターフェースだけではなく、セキュリティーやプライバシー、データのフィルタリングや段階的なデータ統合などの要件が存在する。

参考文献

- [1] An architectural blueprint for autonomic computing: <http://www.eecs.harvard.edu/~chaki/bib/papers/autonomic.pdf>
- [2] Simple Network Management Protocol (RFC 1157) : <http://www.ietf.org/rfc/rfc1157.txt>
- [3] Van Bon, J. IT Service Management: An Introduction. IT Service Management Forum. Van Haren Publishing, UK, 2002.
- [4] ISO14001: <http://www.env.go.jp/policy/j-hiroba/04-iso14001.html>
- [5] DTMF Standard: WS-Management, CIM: <http://www.dmtf.org/standards>
- [6] OASIS Standard;WSDM: <http://www.oasis-open.org/specs/index.php#wsdm1.1>
- [7] IBM Tivoli Common Data Model: <http://www.redbooks.ibm.com/abstracts/redp4389.html>
- [8] Management Information Base (RFC 1213) : <http://www.faqs.org/rfcs/rfc1213.html>
- [9] 秋山一人他: 組み込み Autonomic Computing アーキテクチャーの考案, 2008年7月 IBM ProVISION No.58: http://www.ibm.com/ibm/jp/provision/no58/ibm_paper3.html
- [10] 古市実裕, 工藤道治: GUI ベースのコンピュータに適したアクセス制御ポリシー管理方法の提案, 2008年9月情報処理学会: <http://fw8.bookpark.ne.jp/cm/ipsj/search.asp?flag=6&keyword=IPSJ-JNL4909013&mode=PRT>
- [11] International Electrotechnical Commission: <http://www.iec.ch/>
- [12] Smart Energy Profile: <http://www.zigbee.org/Standards/ZigBeeSmartEnergy/Overview.aspx>



日本アイ・ビー・エム株式会社
大和ソフトウェア開発研究所
Tivoli 開発
アドバイザー・ソフトウェア開発エンジニア

秋山 一人 Kazuhito Akiyama

[プロフィール]

1990年、日本 IBM 入社。システム管理、デバイス管理を行うミドルウェア製品の開発を経て2006年よりソフトウェア開発研究所にてオートノミック・コンピューティング関連製品の開発およびソリューション技術支援に従事。現在は Smarter Planet ソリューションのアーキテクトを担当。IBM Master Inventor.
akiy@jp.ibm.com



日本アイ・ビー・エム株式会社
大和ソフトウェア開発研究所
Tivoli 先進技術 & Ind. Sol.
ソフトウェア開発エンジニア

西村 康孝 Yasutaka Nishimura

[プロフィール]

2006年、日本 IBM 入社。以来、大和ソフトウェア開発研究所にてシステム管理製品のソリューション開発に従事。以来、アセットマネジメント製品、サービスマネジメント製品の開発、品質検証を経て、現在 Tivoli 先進技術 & Ind. Sol. にて Smarter Building 製品の開発を担当。
nishi2go@jp.ibm.com



日本アイ・ビー・エム株式会社
大和ソフトウェア開発研究所
アドバイザー・ソフトウェア・エンジニア

古市 実裕 Saneshiro Furuichi

[プロフィール]

1996年、日本 IBM 入社。東京基礎研究所にて、モバイルコンピューティング、システムソフトウェア、情報セキュリティーなどの研究に従事。2007年12月よりソフトウェア開発研究所にて、セキュリティーや Smarter Planet 関連の製品開発に従事。情報処理学会会員。IBM Master Inventor.