

コンシューマ・システム論文

IoT機器に対するEXI利用時のスキーマ簡約によるROM必要量最適化

米澤 祐紀^{1,a)} 土井 裕介¹

受付日 2014年12月21日, 採録日 2015年5月21日

概要: IoT (Internet-of-Things) はさまざまな領域での応用が期待される。そのうちの1つに、スマートグリッドなどを含む標準仕様で構成されるシステムがある。このようなシステムでは、異なるベンダの異なる機器の統一的な制御に、多くの機器・機能を包含した標準が定義される。そのような標準仕様では、しばしば、XMLスキーマによるデータモデルやメッセージ形式を定義する。一方、標準で定義される多様なデータモデルを含むXMLスキーマは肥大化しがちである。このような肥大化したXMLスキーマを、IoTで用いられる組込み機器で処理するのは最適ではない。本研究では、計算機資源に制約のある組込み機器を、XMLを利用する標準仕様に統合するために、仕様で定義されたXMLスキーマを機器にあわせて最適化する方法 (XMLスキーマ簡約) を述べる。あわせて、本手法をEXIに適用した際のEXI文法のサイズ削減効果についても示す。

キーワード: IoT, XML, スキーマ, 組込み機器, EXI

IoT Device ROM Optimization for EXI Processing with XML Schema Contraction

YUKI YONEZAWA^{1,a)} YUSUKE DOI¹

Received: December 21, 2014, Accepted: May 21, 2015

Abstract: IoT (Internet-of-Things) is expected in various fields. One of particular applications is industrial systems such as smart grid. Standard specifications for interoperability are defined for integration of many kinds of devices from many vendors. W3C XML Schema is often used to define data models and message exchange format in those specifications. On the other hand, those schemas tend to become large because they contain many models and messages for many devices. It is not suitable for embedded devices to process the bloated XML schema. In this paper, we propose a method of XML schema contraction for integrating XML-based communication at constrained embedded devices. We also show that the proposed method reduces EXI grammar size.

Keywords: IoT, XML, schema, embedded system, EXI

1. はじめに

1.1 背景

Internet of Things (IoT:モノのインターネット) は、さまざまな領域で爆発的な普及が期待されている。IoTの先行例の1つに、スマートグリッドシステムを代表とする

各種産業向けシステムがある。そのようなシステムでは、異なるベンダの異なる機器の統合制御のために、各機器・機能に対応した標準が定義される。

データ処理においては、特に拡張性や比較容易性の観点から、データの型やあるデータとの関係を記述する、スキーマの存在が期待される。適切なスキーマが存在すれば、得られたデータが何であるか、2つのデータは比較可能なのか、等が自動的に判別可能である。また、あるスキーマから別のスキーマへの射影が定義できれば、個別の異なるスキーマによるデータを相互に比較・評価が可能になる。

¹ 株式会社東芝研究開発センター
Corporate R&D Center, TOSHIBA Corporation, Kawasaki,
Kanagawa 212-8582, Japan

^{a)} yuki.yonezawa@toshiba.co.jp

システム間でのデータ表現を共通化するために、ASN.1, JSON, W3C XML Schema といった標準的なデータ表現方法が存在する。特に W3C XML Schema は、名前空間の定義や他のスキーマの再利用などが可能であり、拡張性や再利用性に優れているため、複数のベンダ・多様な機器が混在するシステムにおけるデータ形式の定義において有力である。

W3C XML Schema を用いてスキーマ（以下、XML スキーマ）を定義する標準規格が存在する。たとえば、OpenADR2.0a [1], ZigBee SEP2.0 [2], IEC 61850-6 [3], HL7 (Health Level 7) [4] などがあげられる。これらの規格では、ビジネスドメインの専門家が UML ダイアグラムなどを用いて通信仕様を議論し、その結果から XML スキーマを定義し、そのスキーマによって規定される XML 文書を通信仕様上の交換メッセージ形式として用いる、という流れにより、結果的に XML スキーマが通信仕様の根幹となっている。これらの仕様は、一般に複数の機能の集合体であり、それぞれの機能に必要なメッセージ形式を定義するデータ型およびデータモデルが、XML スキーマによって定義される。

たとえば、OpenADR2.0a の通信仕様であるデマンドレスポンス機能*1 は、`oadrRequestEvent`・`oadrDistributeEvent`・`oadrCreatedEvent`・`oadrResponse` の 4 つのデータ型を用いる。また、ZigBee SEP2.0 の通信仕様の 1 つとして定義されているデマンドレスポンス機能*2 を構成する XML 文書には、`DemandResponseProgramList`・`EndControlList`・`EndDeviceControl`・`Response` の 4 つのデータ型を用いる。

1.2 課題

本研究で解決する課題は、計算機資源の潤沢なサーバと計算機資源の限られた多様なクライアント（組み込み機器）という非対称な環境において、通信仕様の定義に XML スキーマを用いることで生じる、組み込み機器における要求 ROM サイズ肥大化である。

XML は可読性と拡張性に優れている一方で、データ表現の冗長性の高さにより大きくなりがちであり、組み込み機器での取扱いは難しい。特に、データ表現の冗長性については、XML のバイナリ表現である EXI (Efficient XML Interchange) 標準により解決される [5]。EXI には、スキーマを利用しない符号化に対応する文法 (Built-In Grammar; 組み込み文法) とスキーマを利用した効率的な符号化に対応する文法 (Schema-Informed Grammar; スキーマ由来文法) とが存在する。Built-In Grammar を利用するには、動的に文法を生成する必要があるため*3、一般に RAM の制約が厳しい組み込み機器においてこの方式の利用は難し

い。したがって、スキーマを利用した符号化方式が組み込み機器においては適している。

スキーマ由来文法において利用される EXI 文法は、およそ XML スキーマのサイズに比例する。前述の XML を利用する標準規格は、しばしばその複雑なビジネス要求を充足するため、多様な機器機能をすべて包含した XML スキーマを生成する。結果的に、標準規格に準拠した EXI 利用製品を作る場合、機器側に搭載する必要がある EXI 文法のサイズが大きくなる、という課題がある。たとえば、ZigBee SEP2.0 における EXI 文法のサイズは、ある実装においては 38 kB 程度であり、これはサーモスタットのようない一部の組み込み機器には過大であるといえる。

本論文では、EXI 文法に割り当て可能な ROM 容量として、組み込み機器の計算機資源の容量から、10 kB 以下を目標とする。組み込み機器には、RFC7228 [7] に記載されている Class 1 機器を想定する。Class 1 機器は、CoAP (Constrained Application Protocol) のような計算機資源に制約のあるノード用に設計されたプロトコルの利用を想定されている。たとえば Castellani らは、組み込み機器における Web サービスにおいて、CoAP と EXI を組み合わせたアーキテクチャを検討し、評価している [8]。

Class 1 機器で利用可能な計算機資源は、RAM は 10 KiB、ROM は 100 KiB 以下が目安となる。そのため EXI 文法を組み込み場合、100 KiB 程度に対して、妥当な比率に抑えなくてはならない。そこで、本論文では、全体の 10% に相当する 10 kB 以下とした。

一方、個別の機器に着目した場合、その機器に必要なデータモデルはごく一部である場合が多い。ただし、実際には XML スキーマのデータ構造（抽象化）は機器にあわせたものとは限らず、機器に必要な部分を残して不要部分を排除する。

本研究では、個別の機器が利用する部分にあわせて、XML スキーマの抽象構造を最適化する処理を、XML スキーマ簡約と呼ぶ。特に本論文では、XML スキーマのモデルの簡約化手法の提案とその効果を評価する。

2. 関連研究

IoT を組み込み機器で実現するには、高効率なメッセージ送信が望まれる。たとえば、Priyantha 他 [9] らは、IEEE802.15.4 で通信を行う組み込み機器 (MSP430 48 kB ROM) 上に、REST によるウェブサービスを実装した。彼らは、メッセージサイズを小さくするために、短縮タグを用いた XML 文書とその XML 文書を処理できる専用のパーサを用いた。圧縮した XML 文書を用いずに、短縮タグを用いる手法に関しては、文献 [10] で議論されている。この手法は、XML タグの書き換えをとらなうため、標準に準拠した環境においては適用できない。

Lerche et al. [11] も文献 [9] と同様に TelosB (MSP430,

*1 文献 [1] 8.1.1 を参照。

*2 文献 [2] 16.7 を参照。

*3 文献 [6] 8.4 Built-in XML Grammars に記述されるように、XML の構造を示す文法が XML 文書のエンコード・デコードにともない動的に生成される。

48 KB ROM, 10 kB に RAM) の環境で, SOAP ベースのセンサを実装している. 彼らの手法は, 組込み機器で XML 文書を処理できるが, 送信時間が顕著に増加すること示した. よって, SOAP のような XML を利用する Web サービスには, よりメッセージサイズの小さい通信が必要であるとされた.

Kyusakov et al. [12] は, 組込み機器向けの EXI エンコーダ・デコーダ: EXIP を作成し, ルネサス社製 M16C/65 プロセッサ上で gSOAP を用いた SOAP に基づく通信を実現している. EXIP はストリーム型のメッセージ処理 API を持っており, 省メモリ処理が可能な構造になっている. また, デコードに用いる XML スキーマは動的に生成することも, 静的なオブジェクトとして持つこともできる.

EXIP は組込み機器に適した構造を持っているが, 先に述べたように EXI の性質により, デコードには EXI 文法が必要になる. スキーマが大きくなった場合, EXI 文法は組込み機器に搭載するには適切でない大きさになる場合がある. これに対し, Caputo et al. [13] は, センサに用途を限定し, デコーダを削除しエンコーダのみに改造した EXIP を Contiki に搭載している. これを STM32W を搭載したマイコンに搭載している.

EIGEN [14] は, 組込み機器をターゲットとした EXI 処理エンジンである. EXIP と異なり, EIGEN は組込み機器が利用するデータを構造体の形で限定し, コードの自動生成により機器ごとに最適化された EXI 処理エンジンを生成する. ただし, EIGEN においては, 受信する EXI ストリームについてはそのままデコードできる必要があることから, デコーダに搭載する EXI 文法は XML スキーマに由来するものであり, XML スキーマのサイズに比例したサイズの ROM が必要になるという課題が残っている.

3. 機器仕様にあわせた XML スキーマ簡約

本節では, XML スキーマで宣言・定義する膨大なデータモデルやその抽象定義から, 個別の機器に合わせたデータモデルのみを残す, XML スキーマ簡約化手法を提案する (図 1). ここで, 仕様などにより定められたサイズの大きい XML スキーマを元スキーマ, 機器ごとに最適化されたスキーマを機器簡約化スキーマと呼ぶ.

本提案手法は, 機器で利用するデータを記述した XML 文書例を用い, 一定の手続きにより元スキーマを簡約化した, 機器簡約化スキーマを生成する. 生成する機器簡約化スキーマは, 元スキーマの複雑さを継承しない. よって, 機器簡約化スキーマは, XML 文書の妥当性検証, EXI ストリームのデコードなどを, 最小限の ROM で動作させる.

提案手法は, XML スキーマの型定義の特徴を用いた簡約化手法である. 文献 [15] で示されているように, XML スキーマにおいて多様性を効率的に取り扱うために, オプション化と派生型の定義が用いられる. たとえば, 複数種

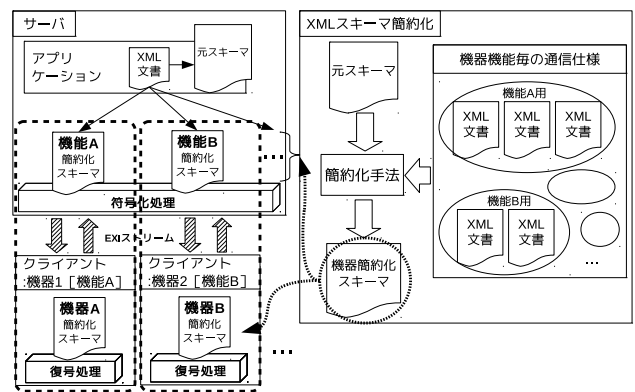


図 1 提案する簡約化 XML スキーマを機器ごとに適用したシステム構成

Fig. 1 System structure: The proposed XML schema contraction method is applied for each device specification.

類のセンサのデータモデルを定義する場合, 各センサに共通の項目と, 機器種別ごとに固有の項目が存在する. 共通の項目には機器 ID などがあり, 異なる項目には各種センサが取得できる値などが考えられる. またセンサのメーカーや型番などの違いにより, 取得できる値や利用可能なパラメータにばらつきが生じる可能性がある. XML スキーマは, このようなパラメータの違いをオプション項目として扱うことができる. また, 型派生は, 派生型を利用したプログラムの利便性・再利用性の向上などが利点としてあり, 本研究が対象とする ZigBee SEP2.0 などでもデータ型の定義に広く利用されている.

提案手法では, 機器で利用するデータを記述した XML 文書を雛形として, XML スキーマに定義・宣言されている型定義とオプション要素の必要性を判断する. 組込み機器の多くは, 特定の用途・機能に向けて設計されている. よって, その用途・機能に無関係な型定義は不要であることから, 多くの型定義やオプション要素が不要である場合がある.

XML スキーマに定義・宣言されているオプション要素は, 次の 2 つである. 1 つ目は, Global Element の定義であり, XML 文書のルート要素を定義する. ルート要素は多くの場合メッセージ種別に対応し, 機器は多くの場合そのサブセットのみを利用する. 2 つ目は, 型定義と型を構成する Element・Attribute の定義である. 型定義は特定の目的に対して行われ, 仕様によっては個別の機器にとって不要な定義を多く含む. また, Element および Attribute の定義のうちオプションなものについて, そのオプションの指定により動作に影響を与えないもの, 無視されるものについては削除可能である.

本論文が対象とする XML スキーマの簡約化処理は, 次の 2 つである. 1 つ目の簡約化処理は, オプションの簡約化である. XML スキーマでは, 記述してもしなくてもよい項目 (オプション項目) を定義できる. 一方, 個別の機

器が利用する項目はある程度決まっている。したがって、機器が利用範囲へのオプション項目の絞り込みは、XML スキーマ簡約として有効な手法である。また、EXI 固有の問題として、オプションの項目が連続すると、EXI 文法の状態遷移数が爆発的に増大する、という問題がある*4。その項目がない場合は機器にとってエラーとなるような、その機器にとっての必須項目があれば、オプション項目の必須項目への置き換えで、さらに文法を簡約化できる。

2つ目の簡約化処理は、派生型の簡約化である。オブジェクト指向設計において、多様な機器・機能を効率的に管理するために、中間的な抽象化モデルを定義し、個別の機器は種別ごと、機器ごとの派生型として定義する。このように管理されたモデルの処理には、対象とするモデルとその上位に位置するモデルとを処理しなくてはならないため、複雑さが増大する。中間の抽象モデルを簡約化し、個別のデータモデルを具体化することで、XML スキーマを単純化できる。

3.1 提案手法を用いたシステム構成

本節では、提案手法により生成できる機器簡約化スキーマを用いた、クライアントサーバモデルのシステム構成とその通信手順を述べる。

図1に示すように、システムは、組込み機器における複数種類のクライアントと、計算機資源の豊富なサーバで構成する。クライアントは、標準などで定義される本来の通信仕様に対応するスキーマ（元スキーマ）に含まれる多くのデータモデルから、自身が利用する機能を利用するのに十分なデータモデルを含んだ、機器簡約化スキーマあるいはこれに対応する EXI 文法と、EXI ストリームから値を取得する復号処理モジュールを持つ。サーバは、アプリケーションモジュール・元スキーマ・各々の機器の種別に対応する機器簡約化スキーマ・符号化処理モジュールの4つを備える。

次に、サーバからクライアントへの EXI ストリーム送信について説明する。サーバのアプリケーションが、元スキーマをもとに作成した XML 文書のあるクライアントに送信する。XML 文書を受け取った符号化処理モジュールは、送信先であるクライアントに対応する機器簡約化スキーマを選択し、受け取った XML 文書を機器簡約化スキーマを用いて符号化し、そのスキーマに対応する EXI ストリームを生成する。この EXI ストリームがクライアントに送信される。クライアントは、受信した EXI ストリームを、機器簡約化スキーマを用いて復号し、アプリケーションにその結果を渡す。

本方式の特長は、以上に述べたように、サーバ側のアプリケーションで用いる XML 文書、およびそれにとまう

アプリケーションロジックを変更せずに、クライアント側が利用する仕様に比例する程度に、EXI 文法を軽量化することである。本方式で提案する XML スキーマ簡約化手法は、特定の機器で利用する機能に対応する XML 文書の集合をもとに、これらを受信するのに十分な機器簡約化スキーマを生成する。XML 文書は変更されないため、サーバ側アプリケーションが利用する XML 文書は元スキーマに沿ったアプリケーションロジックによるものを継続して利用できる。一方で、EXI 符号化に用いるスキーマは、サーバ側アプリケーションが用いるすべての機能ではなく、クライアント側が用いる機能に絞り込んだものとなる。よって、図1に示すように、サーバ側アプリケーションでは元スキーマ、組込みの機器との通信には機器簡約化スキーマをそれぞれ利用でき、組込み機器側の EXI 文法を大幅に軽量化できる。

なお、提案手法は、各々の機器が利用する機能に対応するデータモデルの集合をもとに機器簡約化スキーマを生成する。機器簡約化スキーマの量は、機器の機種数、あるいは最大限に見ても機器のファームウェアのバージョン数に比例する量となることを見込まれ、IoT 機器のスケールに比して十分に小さく、現時点のサーバが持つストレージ・処理能力を考えれば十分に対応可能なオーバヘッドである。

4. XML スキーマ簡約の設計・実装

XML スキーマ簡約の設計・実装の方針として、XML スキーマにおけるオプションと派生型の簡約化方法を説明する。

本研究では、機器は少数のパターンの XML 文書のみを受信し、これ以外のものは処理できないと仮定する。また、各機器が受信可能な XML 文書のパターンは、少数の具体的な XML 文書により表現できるものとする。

4.1 オプションの簡約化

本節では、対象とする XML 文書から、元スキーマの不要なオプション項目を削除することにより、XML 文書の妥当性を損うことなくスキーマを簡約する手法を説明する。

XML スキーマにおけるオプション項目は、ルート要素になりうる Global Element とその Element を構成する要素が対象である。Global Element のうち、ある XML 文書で利用されていないものは、オプション項目として解釈できる。Element を構成する要素では、指定されている属性からオプション項目と分かる。具体的には、minOccurs 属性が 0、use 属性が optional の場合である。

本論文では、次に示す手順で、オプション項目を簡約化する。

- (1) XML 文書が利用する要素・属性の一覧を作成
- (2) 手順(1)で作成した一覧に対応する元スキーマの要素・属性を記録

*4 項目数 n に対して遷移数は $(n-1)!$ となる。

```
<xs:element name="note" type="Note" />
<xs:complexType name="Note">
  <xs:sequence>
    <xs:element name="title" type="xs:string" />
    <xs:element name="body" type="xs:string" minOccurs="0" />
    <xs:element name="author" type="xs:string" />
  </xs:sequence>
</xs:complexType>
```

図 2 オプション要素を含んだ XML スキーマの例

Fig. 2 Example of an XML schema with optional elements.

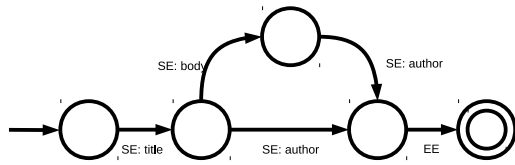


図 3 図 2 の EXI 文法に対応するオートマトン

Fig. 3 Automaton corresponding to the EXI grammar shown in Fig. 2.

```
<note>
  <title>こんにちは</title>
  <author>Tarou Toshiba</author>
</note>
```

図 4 図 2 の XML スキーマに対し妥当な XML 文書

Fig. 4 An example of valid XML document for XML schema shown in Fig. 2.

(3) 記録した要素・属性だけを含んだ機器簡約化スキーマを出力

オプション簡約化による EXI 文法への影響について、図 2 に示す XML スキーマと図 4 に示す XML 文書を用いて説明する。

図 2 に示した XML スキーマは、Note 型を構成する name="body" の要素がオプション項目である。この XML スキーマから生成できる EXI 文法は、図 3 に提示するようなオートマトンを生成する。なお、図中の遷移に与えられたラベル (SE:title など) は EXI の内部表現に対応する ([6] Table 4-1)。

手順 (2) では、図 3 で示した各状態のうち、利用する状態の記録を意味する (図中では各状態にマーク)。図 4 の XML 文書を用いる場合、図 5 に示すように状態を記録する。これにより、EXI 文法の中から、図 4 の XML 文書に必須な項目を明らかにできる。

手順 (3) では、図 6 に示す EXI 文法のオートマトンを生成を意味する。これにより、図 6 は、図 4 を受け入れる限度内で最適化される。

4.2 派生型の簡約化

XML スキーマでは、抽象化モデルに基づく型から、継承による具体的な型に派生させることができる。図 7 の例では、C に基づく型の具体的な型として D と E が派生している。また、C は B から、B は A からそれぞれ派生した型である。

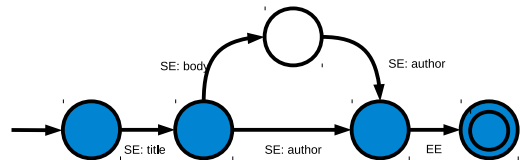


図 5 オプション簡約化手順 (2): 図 4 の XML 文書を用いて、必須な要素・属性を記録したオートマトン

Fig. 5 Step 2 of optional elements contraction: It records essential elements and attributes, which are used in the XML document instance shown in Fig. 2.

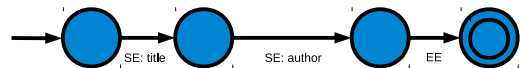


図 6 オプション簡約化手順 (3): オプション簡約化による機器簡約化スキーマの EXI 文法に対応するオートマトン

Fig. 6 Step 3 of optional elements contraction: It only contains essential elements.

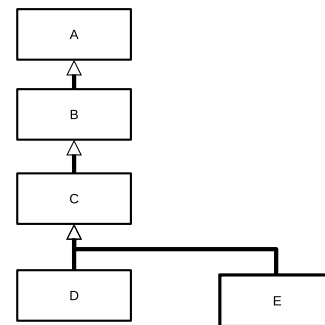


図 7 派生型簡約化の手順 (1): D と E は C から、C は B から、B は A から、それぞれ派生した型であることを示している

Fig. 7 Step 1 of derivative type contraction: D and E are derived from C. C is derived from B. B is derived from A.

派生型の簡約化は、ある具体的な型の元となっている抽象的な型の要素と属性を、1つの型に簡約する。元スキーマが、図 7 に示す派生関係で、かつ、XML 文書で D だけを利用する場合、A から D を 1つの型に簡約化する。

一方、図 7 に示す派生関係で、XML 文書で D と E を利用する場合は、図 8 に示すように、A, B, C の要素を含む抽象的な型 X と、X から派生した D と E、との 3つの型を作成する。仮に、抽象的な型 X を作成せず、A から D と A から E をそれぞれ 1つの型とした場合、簡約化した 2つの型は独立した型となり、オブジェクト指向における構造化を壊してしまう。

以上をふまえて、派生型の簡約化に対し、次の 3つの手順で実現した。

- (1) XML スキーマの xs:extension から、派生関係を調査
- (2) 手順 (1) の調査結果を用いて、最も抽象化されている派生元から分岐点を検知
- (3) 分岐点に至るまでの型を、1つの型に変換した機器簡約化スキーマを出力

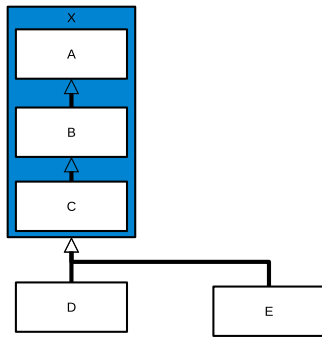


図 8 派生型簡約化の手順 (2): A, B, C を 1 つの型 X とし, X が D と E を派生する

Fig. 8 Step 2 of derivative type contraction: A, B and C are summarized in a class X. D and E can be derived from class X instead of class C.

次に、派生型簡約化による EXI 文法への影響を説明する。EXI 文法は、XML スキーマに定義されている各型に対応する EXI 文法を作成する。つまり、図 7 に示すような派生関係の場合では、5 つの EXI 文法が必要である。

一方、図 8 に示すように派生型関係を簡約化することで、A, B, C を 1 つの型で表現できる (図 8)。これを EXI 文法にした場合、5 つ必要だった EXI 文法を 2 つ削減でき、ROM サイズ削減につながると予測する。

5. 評価

今までに述べた設計をもとに構築した XML スキーマの簡約化エンジンを用いて、提案手法を評価する。まず、既存の XML スキーマに対する適用可能性・有効性について、オプション・派生型のそれぞれを評価し、その後、XML スキーマを小さくする方法として一般的に考えられる従来手法と比較する。

5.1 評価に用いる XML スキーマと XML 文書例

図 1 で示したように、本システムは、サーバから組込み機器への EXI ストリーム送信を想定する。評価に用いる標準仕様には、組込み機器での利用が想定されている、OpenADR2.0a と ZigBee SEP2.0 を用いる。

OpenADR2.0a と ZigBee SEP2.0 は、想定されるユースケースの規模が異なる。OpenADR2.0a が想定しているユースケースは、デマンドレスポンス機能だけである。ZigBee SEP2.0 のユースケースは、デマンドレスポンス機能以外に、複数の機能を提供する。そのため表 1 と表 2 に示すように、定義・宣言されている型や要素数が異なる。

ここでは、2 種類の XML スキーマを用いることにより、個々のスキーマの特徴から本手法の有効性がどのように変化するかを示す。XML スキーマの定義方法は本報告で述べたとおりである一方、その利用は、想定するユースケースやその設計に依存する。そこでユースケースの規模が異なる 2 種類のスキーマを用いることで、本提案手法の有効

表 1 OpenADR2.0a におけるオプション簡約化による定義・宣言数の比較: 機器簡約化 XML スキーマは、oadrResponse・oadrDistributeEvent の XML 文書を受信可能

Table 1 Difference the number of declaration or definition between the original XML schema and an contracted XML schema on OpenADR2.0a. The contracted XML schema can accept oadrResponse and oadrDistributeEvent message encoded as an XML document.

項目	元スキーマ	機器簡約化スキーマ	
		オプション簡約	派生型簡約
Element	48	33	33
ComplexType	13	11	10
SimpleType	10	7	7

表 2 ZigBee SEP2.0 におけるオプション簡約・派生型簡約による定義・宣言数の変化。機器簡約化スキーマは、DemandResponseProgramList・EndControlList・EndDeviceControl の XML 文書を受信可能

Table 2 Difference the number of declaration or definition between the original XML schema and a contracted XML schema in ZigBee SEP2.0. Contracted XML schema accepts DemandResponseProgramList, EndControlList and EndDeviceControl messages encoded as an XML document.

項目	元スキーマ	機器簡約化スキーマ	
		オプション簡約	派生型簡約
Element	312	3	3
ComplexType	312	23	17
SimpleType	25	11	11

性を評価する。

OpenADR2.0a のデマンドレスポンス機能*5 を構成する XML 文書のうち、組込み機器が受信する文書は、oadrDistributeEvent と oadrResponse の 2 つである。

ZigBee SEP2.0 のデマンドレスポンス機能*6 を構成する XML 文書のうち、組込み機器が受信する文書は、DemandResponseProgramList・EndControlList・EndDeviceControl の 3 つである。これらメッセージ関する XML スキーマの UML クラス図を図 9 に示す。

5.2 オプション・派生型簡約の効果

本論文が提案する簡約化手法であるオプション簡約と派生型簡約は、複数の機能を持つ XML スキーマに対して有効であることが分かった。具体的には、ZigBee SEP2.0 では、デマンドレスポンス機能以外に多くの機能を備えている XML スキーマにおいて、デマンドレスポンス機能を切り出した際に有効に作用することを確認した。一方で、すでに機能が特定されている XML スキーマに対しては、あまり効果的でないことが分かった。具体的には、OpenADR

*5 文献 [1] 8.1.1 より。

*6 文献 [2] 16.7 より。

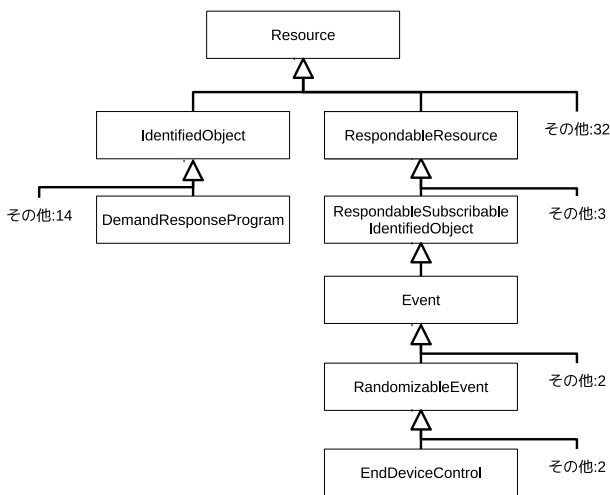


図 9 ZigBee SEP2.0：デマンドレスポンス機能に関する元スキーマの UML クラス図

Fig. 9 UML class diagram for classes required for demand response of ZigBee SEP2.0.

表 3 元スキーマとデマンドレスポンス機能を対象にした機器簡約化スキーマから生成できる EXI 文法のオブジェクトサイズ：OpenADR2.0a では oadrResponse・oadrDistributeEvent, ZigBee SEP2.0 では DemandResponseProgramList・EndControlList・EndDeviceControl の XML 文書が受信できる

Table 3 Object size of EXI grammar: Contracted XML schemas accept XML documents for oadrResponse and oadrDistributeEvent in OpenADR 2.0a, and, DemandResponseProgramList, EndControlList, and EndDeviceControl in ZigBee SEP2.0, respectively.

	元スキーマ (バイト)	オプション簡約 (バイト)	派生型簡約 (バイト)
OpenADR 2.0a	2,759	2,015	2,013
ZigBee SEP2.0	37,817	2,711	1,887

2.0a のように、デマンドレスポンス機能しか持たない XML スキーマである。

表 1, 表 2, 表 3 は、提案手法であるオプション簡約と派生型簡約に対する削減結果である。各手法による削減効果について詳細に述べる。表 3 に示す EXI 文法のオブジェクトサイズは、EIGEN [14] を用いた。

派生型簡約の評価では、オプション簡約化後の XML スキーマを用いる。これは、提案した派生型の簡約化手法が、1 対 1 の派生関係を想定していることに起因する。一般的な型の派生関係では、1 つの型から複数の型に派生するため、想定する派生関係は稀である。一方、オプション簡約化後の XML スキーマにおいては、派生する型が限られるため、想定する派生関係が成り立ちやすい。

5.2.1 オプション簡約の効果

オプション簡約が、OpenADR2.0a と ZigBee SEP2.0 の両方に対し、EXI 文法を小さくできることを確認した。特

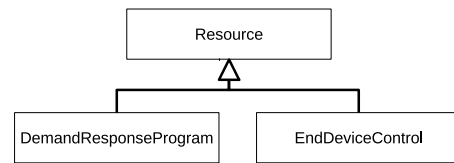


図 10 ZigBee SEP2.0 のデマンドレスポンス機能を対象とした簡約化手法適用後の UML クラス図

Fig. 10 UML class diagram for classes after applying proposed method on demand response in ZigBee SEP2.0.

に、複数の機能を持つ ZigBee SEP2.0 では、EXI 文法を 37,817 バイトから 2,711 バイトまで、約 93%削減できた。

OpenADR2.0a は、デマンドレスポンス機能に限定された XML スキーマである。本提案手法のオプション簡約は、ROM サイズ削減の効果はある一方で、OpenADR2.0a において、その削減効果は相対的に少ない。これは、元スキーマがすでにデマンドレスポンス機能に必要な項目のみで構成されており、オプションとして扱える要素が少ないためである。表 1 に示すように、定義・宣言数は削減されており、また表 3 に示すように、EXI 文法のオブジェクトサイズは、元スキーマから約 27%削減できた。

ZigBee SEP2.0 は、デマンドレスポンス機能以外にも、多くの機能を提供している。そのため、利用されない型とオプション項目が削減されることとなり、オプション簡約が有効に作用した。ZigBee SEP2.0 に対して、本提案手法のオプション簡約により、表 2 に示すように、定義・宣言数を全体の 94%削減できた。また表 3 に示すように、EXI 文法のオブジェクトサイズは、元スキーマから 93%削減できた。

5.2.2 派生型簡約の効果

派生型簡約化では、ZigBee SEP2.0 に対してのみ、EXI 文法のオブジェクトサイズを削減できることを確認した。OpenADR2.0a では、想定した派生関係が 1 カ所しか存在しなかった。そのため、EXI 文法のオブジェクトサイズは、おおむねオプション簡約時と同じである。

ZigBee SEP2.0 では、EXI 文法のオブジェクトサイズを、オプション簡約時に対して約 30%、元スキーマに対して、約 95%まで削減できた。派生型簡約により定義されるデータ型の UML クラス図を図 10 に示す。まずオプション簡約化により、図 9 中の「その他」部分と、型を構成するオプション項目のうち不要なものを削除した。そして派生型簡約により、図 10 に示すように、派生型関係を簡約化した。

5.3 従来手法と提案手法との比較評価

元スキーマを縮小する一般的な手法に、元スキーマから XML 文書で利用するモデルのみを用いて、XML スキーマを再構築する手法がある（以下、従来手法）。従来手法適用時には、定義されているモデルの内部構造は変更しない。一方、提案手法は、XML 文書の妥当性検証に必要な文法

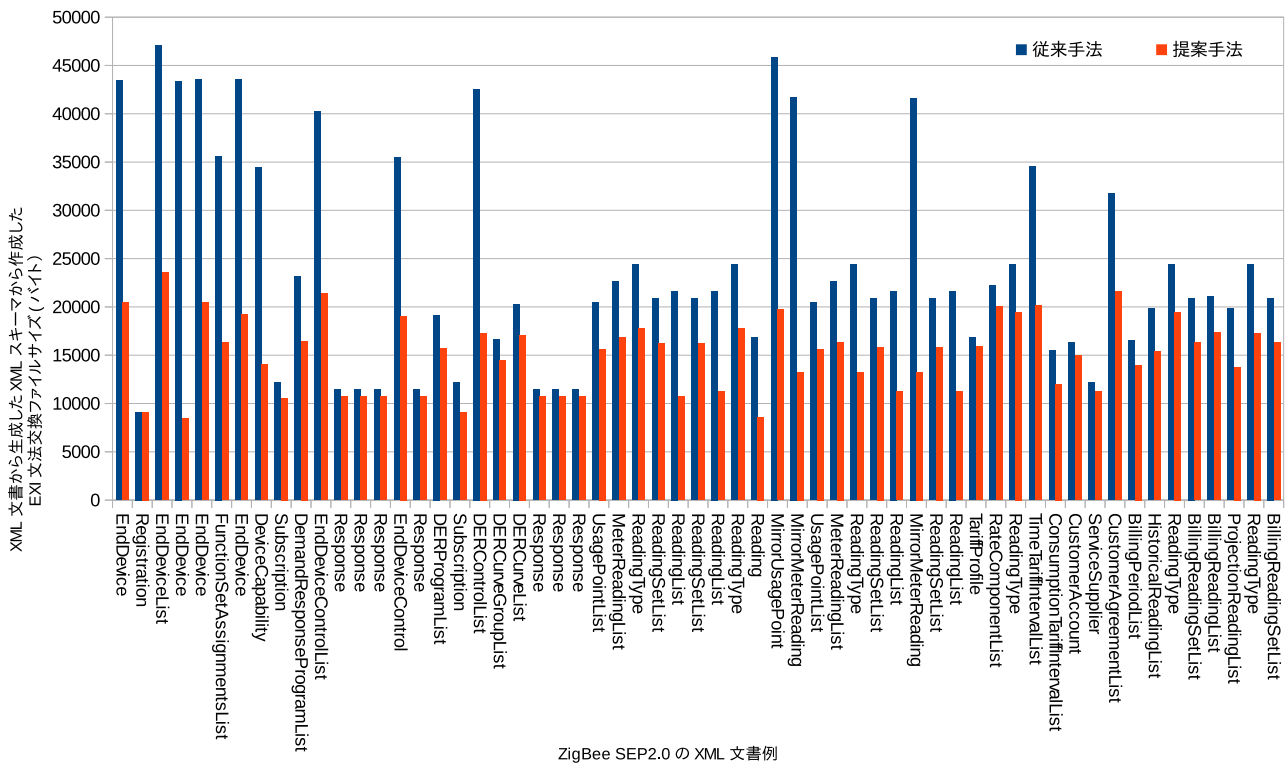


図 11 ZigBee SEP2.0 付属の XML 文書に対して、既存手法と提案手法適用後の EXI 文法交換ファイルのファイルサイズ

Fig. 11 File size of EXIGram in existing method vs proposed method. XML documents are extracted from examples of ZigBee SEP 2.0 specification.

を、元スキーマを用いて簡約化する。

従来手法と提案手法との比較には、5.2 節で利用した ZigBee SEP2.0, XML 文書には、標準ドキュメントに付属する 60 種類の XML 文書を用いる。

ここでの評価には、それぞれの手法を通して作成される XML スキーマから生成できる EXI 文法交換ファイル [16] のファイルサイズを用いる。EXI 文法交換ファイルは、XML スキーマの宣言・定義内容を EXI 文法に変換した XML 文書である。今回は、EXI エンコーダ・デコーダのオープンソースライブラリである OpenEXI*7を用いて、EXI 文法交換ファイルを生成する。

各 XML 文書に対して、従来手法と提案手法とをそれぞれ用いて生成した XML スキーマから、生成した EXI 文法交換ファイルのファイルサイズを、図 11 に示す。図 11 に示すように、提案手法は、従来手法と同一かそれ以下のファイルサイズになっている。ファイルサイズが同一であるのは、対象とする XML 文書が、宣言・定義されているオプション要素をすべて利用し、かつ、派生型を利用していない場合である。一方、この条件に入らない場合は、従来手法よりも小さい EXI 文法交換ファイルを生成できている。

6. おわりに

本研究では、標準仕様などにより定義される、さまざまな機器仕様を包含する XML スキーマを、個別の組込み機器が利用する部分のみを残して機器別に最適化する、XML スキーマ簡約化手法を提案した。

多様な機器を包含するモデルの定義において、オプション定義による多様性の吸収と、中間的な抽象化型の定義と派生による多様性の定義が、モデルの複雑化の要因であると仮定し、オプション簡約化と型派生簡約化のそれぞれを検討した。オプション簡約化は、ある XML 文書から、利用しないことが確定しているオプション項目をスキーマから削除する、という簡約化を行う。派生型簡約化では、オプション簡約化によって単純化されたモデルの派生関係の、派生先と派生元とを 1 つにまとめる簡約化を行う。

それぞれについて、組込み機器での利用が想定されている標準仕様である、OpenADR2.0a と ZigBee SEP2.0 の XML スキーマを用いた。元スキーマの EXI 文法サイズと提案手法を用いてデマンドレスポンス機能に特化させた簡約化スキーマの EXI 文法サイズとを比較した。

機器簡約化スキーマによる妥当性が検証できることを確認したうえで、著者らによる EXI デコーダのオブジェクトサイズとオープンソースによる EXI 文法交換ファイルとで比較した。

*7 <http://openexi.sourceforge.net/>

OpenADR2.0a では、オプション簡約化により必要な EXI 文法のオブジェクトサイズを約 2.7KB から約 2.0KB へと約 27%程度削減できた。ZigBee SEP2.0 では、オプション簡約化と派生型簡約化により EXI 文法のオブジェクトサイズを約 37KB から約 1.8KB へと約 95%程度削減できた。また従来手法との比較において、提案手法が従来手法とほぼ同等かそれ以下の EXI 文法交換ファイルを生成できることを確認した。これにより、本報告が目標とした RFC7228 における Class1 機器に対し、SEP2.0 のような大型仕様に対応する EXI 文法の組込みが可能になり、標準に基づいて相互動作する機器の生産が容易になる。

以上から、XML 簡約により、個別の組込み機器に合わせた最適化と、XML スキーマ・XML 文書による仕様に厳格なデータ処理とを両立できる。これにより IoT 機器による XML スキーマの標準利用に向けた障壁を 1 つ減らすことができたといえる。

参考文献

- [1] OpenADR Alliance: OpenADR 2.0 Profile Specification A Profile (2011).
- [2] ZigBee Alliance: Smart Energy Profile 2 Application Protocol Standard (2013).
- [3] IEC 61850-6: Communication networks and systems for power utility automation Part 6: Configuration description language for communication in electrical substations related to IEDs (2009).
- [4] Health Level Seven: HL7 Version3, (online), available from <http://www.hl7.org/> (accessed 2015-05-07).
- [5] Bournez, C.: Efficient XML Interchange Evaluation, W3C Working Draft (online), available from <http://www.w3.org/TR/exi-evaluation/> (accessed 2015-05-07).
- [6] Schneider, J. and Kamiya, T.: Efficient XML Interchange (EXI) Format, W3C Recommendation (online), available from <http://www.w3.org/TR/exi/> (accessed 2015-05-07).
- [7] Bormann, C., Ersue, M. and Keranen, A.: RFC7228: Terminology for Constrained-Node Networks, Internet Engineering Task Force (online), available from <https://tools.ietf.org/html/rfc7228> (accessed 2015-05-07).
- [8] Castellani, A.P., Gheda, M., Bui, N., Rossi, M. and Zorzi, M.: Web Services for the Internet of Things through CoAP and EXI, *Communications Workshops (ICC), 2011 IEEE International Conference*, pp.1-6 (2011).
- [9] Priyantha, N., Kansal, A., Goraczko, M. and Zhao, F.: Tiny web services: design and implementation of interoperable and evolvable sensor networks, *Proc. 6th ACM conference on Embedded network sensor systems*, pp.253-266, ACM (2008).
- [10] Augeri, C.J., Bulutoglu, D.A., Mullins, B.E., Baldwin, R.O. and Iii, L. C.B.: An Analysis of XML Compression Efficiency, *Proc. ACM Workshop on Experimental Computer Science* (2007).
- [11] Lerche, C., Laum, N., Moritz, G., Zeeb, E., Golasowski, F. and Timmermann, D.: Implementing Powerful Web Services for Highly Resource-Constrained Devices, *2011*

- IEEE International Conference on Pervasive Computing and Communication Workshops*, pp.524-527 (2011).
- [12] Kyusakov, R., Eliasson, J. and Delsing, J.: Efficient Structured Data Processing for Web Service Enabled Shop Floor Devices, *Proc. IEEE International Symposium on Industrial Electronics (ISIE) 2011*, pp.1716-1721 (2011).
- [13] Caputo, D., Mainetti, L., Patrono, L. and Vilei, A.: Implementation of the EXI schema on wireless sensor nodes using Contiki, *Proc. 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pp.770-774 (2012).
- [14] Doi, Y., Sato, Y., Ishiyama, M., Ohba, Y. and Teramoto, K.: XML-Less EXI with Code Generation for Integration of Embedded Devices in Web Based Systems, *Proc. 2012 International Conference of the Internet of Things*, pp.80-87 (2012).
- [15] Walmsley, P.: *Definitive XML Schema*, Prentice Hall PTR (2001).
- [16] Kamiya, T.: EXIGram (EXI Grammar) Interchange — Format Specification (online), available from <https://www.dropbox.com/s/etpuchaddplq2s/EXIGram.pdf> (accessed 2015-05-07).



米澤 祐紀 (正会員)

2011年3月慶應義塾大学大学院政策・メディア研究科修士課程修了。2011年4月より株式会社東芝研究開発センターでスマートグリッド、組込み向けXML処理系の研究開発に従事。



土井 裕介 (正会員)

2000年3月慶應義塾大学大学院政策・メディア研究科修士課程修了。2011年3月東京大学大学院情報理工学系研究科博士後期課程単位取得退学。2011年9月博士(情報理工学, 東京大学)取得。2000年4月より株式会社東芝研究開発センターで分散システム, ホームネットワーク, 電子タグ, 組み込み向けXML処理系等の研究開発に従事。電子情報通信学会, IEEE, ACM各会員。