

# Fogに基づくIoTアプリケーションのためのコンテキスト指向アーキテクチャの設計

M2015SE003 川畑 賢史

指導教員：野呂 昌満

## 1 はじめに

近年、ICTの発展に伴い、組込み機器をネットワークに接続し、相互通信を行なうInternet of Things(以下、IoT)[3]が普及してきた。この組込みシステムは移動体として実現され、サービスを連携させることで機能を実現する。クラウド[7]上のサービスと連携することによる実時間性能の低下の問題を解決するためにフォグ[2]が提案されている。様々なサービスと連携することでアプリケーションドメインは拡大し、高機能化してきている。複数の非機能特性を考慮しながら実行時に変化するサービスや移動体に応じた振舞いを実現するには動的再構成の方法を考へることが重要である。Truongらは、クラウド上のサービスに応じたフォグの構成を動的に変更するアーキテクチャ[4]を提案している。

移動体、フォグ、クラウドは相互に影響を受けて再構成されると我々は考へる。この場合、Truongらの提案するアーキテクチャは移動体に関する視点が欠落している。移動体にとって重視する非機能特性に応じてフォグの構成の変更が必要となる。実行時にアクセス可能なフォグやクラウド上のサービスは変化するのて、移動体は非機能特性を考慮しながら、状況に応じたサービスを1つ選択する必要がある。

本研究では、移動体として実現される組込みソフトウェアおよびフォグを動的に再構成するアーキテクチャを提案する。非機能要求を充足したIoTアプリケーションの開発支援を実現する。

移動体およびフォグの動的再構成に関する問題としてアーキテクチャを設計する。動的再構成に関する要求を整理し、アーキテクチャの設計指針を定義する。移動体およびフォグをコンテキスト指向設計し、コンテキスト間の協調によってそれぞれの再構成を可能とする。Clementsらが定義するアーキテクチャスタイル[5]を参考にアーキテクチャを設計する。考へでは、動的再構成に関する要求に対するアーキテクチャの妥当性とTruongらの提案するアーキテクチャとの比較によるアーキテクチャの新規性および利点を確認する。

## 2 背景技術と関連研究

### 2.1 背景技術

#### 2.1.1 IoT

IoTとは、センサやアクチュエータ等の組込みデバイスをネットワークに接続し、相互通信を行なう仕組みや概念のことである。IoTを用いることで、必要な情報をリアルタイムで取得することが可能となる。ユビキタス計算やパーベイシブ計算と同じ概念である。IoTの参照アーキテクチャ[2]は次の4層からなる(図1)。

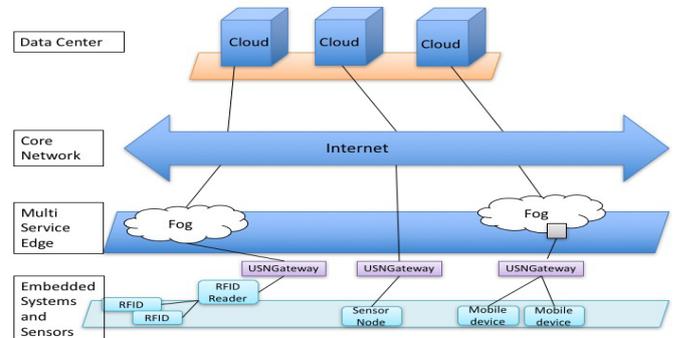


図1 IoTの参照アーキテクチャ

#### 1. Data Center, Cloud Layer

データセンタが置かれ、このサーバ上にサービスを配置する。

#### 2. Core Network Layer

IPネットワークなど、複数のサブネットワーク間でデータを送信、交換する。

#### 3. Multi Service Edge Layer

組込みシステムやセンサなどをコアネットワークにアクセス可能とするために、この層に配置されるゲートウェイによってコアネットワークとの橋渡しを行なう。フォグはこの層に対応する。

#### 4. Embedded Systems and Sensors Layer

組込みシステム、センサ、アクチュエータなどのデバイスが存在する。これらは、一般にコアネットワークにアクセスする機能を持たない。

### 2.1.2 フォグ

フォグとはクラウドより、地理的に分散し、デバイス・ユーザの近距離に配置することで、情報の提供にかかる遅延を短くすることを目的としている。類似の概念としてエッジがある。クラウドに対して、IoTを制御する上での遅延などの問題を解決するためにクラウドを補完する概念である。フォグ/エッジはクラウドにとってかわるものではなく、フォグとクラウドを連携することで新たなサービスを提供することを目的としている。

### 2.2 関連研究

#### 2.2.1 移動体に関する研究

欧州電気通信標準化機構により、モバイルエッジコンピューティング(以下、MEC)の標準化が進められている[1]。[1]では、MECの参照アーキテクチャとフレームワークが提案されている。図2は、MECの参照アーキテ



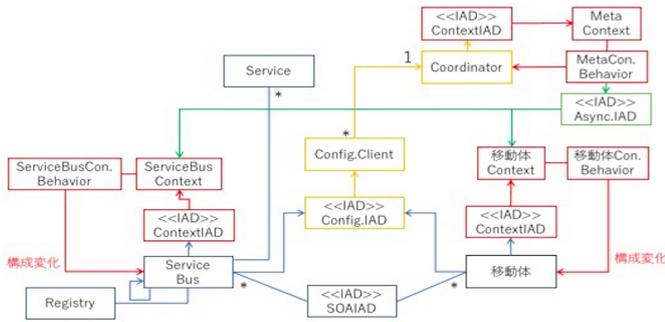


図 4 Module View に基づき記述したアーキテクチャ

断的関心事として特定する。我々は横断的関心事として次の4つの関心事を特定し、それぞれアスペクトとして分離した。表1は、横断的関心事とアスペクトの関係を表す

表 1 横断的関心事とアスペクトの関係

横断的関心事	アスペクト
Configuration Concern	Configuration Aspect
SOA Concern	SOA Aspect
Context Concern	Context Aspect
Communication Concern	Asynchronous Aspect

表したものである。Configuration Concern を Configuration Aspect として抽出。SOA Concern を SOA Aspect として抽出。Context Concern を Context Aspect として抽出。Communication Concern を Asynchronous Aspect として抽出。図4の黄線のモジュールは Configuration Aspect、青線のモジュールは SOA Aspect、赤線のモジュールは Context Aspect、緑線のモジュールは Asynchronous Aspect を表す。

### 3.4 C&C View

C&C View には、Module View で記述したアスペクトを織り込んだ実行時のコンポーネントの関係を記述する。図5は本研究で提案するアーキテクチャを C&C View に基づき記述したものである。

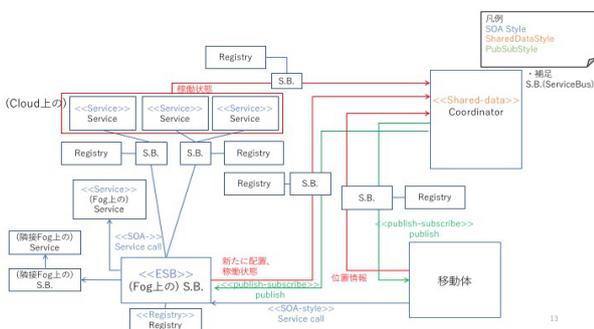


図 5 C&C View に基づき記述したアーキテクチャ

C&C View に定義されている次の3つのスタイルを用いて記述し、実行時のコンポーネントの振舞いに対する役割について説明する。

- Shared Data Style
- SOA Style
- Publish Subscribe Style

移動体、Fog、Cloud Service は Coordinator に対して稼働状態や位置情報を送信する。Coordinator の持つメタコンテキストは移動体、Fog、Cloud Service の共有データと定義できるので、Coordinator と移動体、Fog、Cloud Service に Shared Data Style を適用する。Fog および Cloud の Service は実行環境が異なる場合でも相互作用を可能とすることを目的としている。Fog および Cloud の Service、Registry、Service Bus は SOA Style を適用する。Coordinator から移動体および Fog にコンテキスト変更のメッセージを非同期に送る。Coordinator と移動体、Fog、Cloud Service に Publish-Subscribe Style を適用する。移動体は Service Bus を介して Service を利用するか、Coordinator に位置情報を送信する。Fog は Service Bus を介して Service を利用するか、Coordinator に稼働状態を送信する。Coordinator は Service Bus を介して移動体と Service Bus を再構成させる。

### 3.5 Allocation View

Service の配置および Coordinator の配置にはいくつかのパターンが存在し、それぞれのパターンについて説明する。

図6は本研究で提案するアーキテクチャを Allocation View に基づき記述したものである。Service は、アプリケーションによってはクラウドとフォグに配置されることが考えられる。クラウド上の Service は Data Center 層に、フォグおよび隣接するフォグ上のサービスは Multi Service Edge 層に記述した。Coordinator は、アプリケーションによってはクラウド、フォグ、移動体に配置し、分散・連携することが考えられる。Coordinator の配置は7通り存在する。図6では、Data Center 層、Multi Service Edge 層、Embedded Systems and Sensors 層それぞれに記述した。

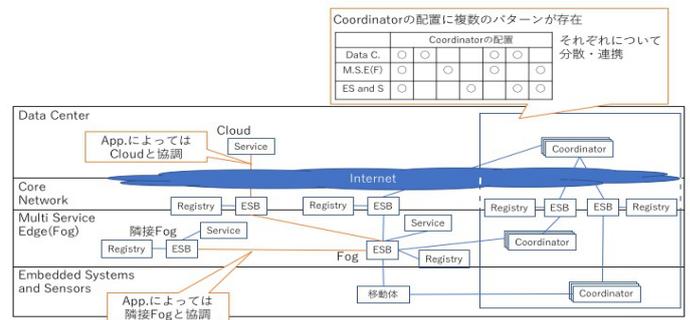


図 6 Allocation View に基づき記述したアーキテクチャ

## 4 考察

本章では、提案したアーキテクチャの有用性と、関連研究と比較し本研究で提案するアーキテクチャの新規性および利点を考察する。

#### 4.1 提案したアーキテクチャの有用性の考察

提案するアーキテクチャに基づかずに安直に実現した場合、次の問題があると考えた。

- コンテキストの協調に関する視点の欠落
- 非機能要求を充足したアプリケーションの実現が困難

移動体に対して、実行時に変化するフォグおよびクラウド上のサービスに応じた処理を定義しなければならない。フォグに対して、クラウド上のサービスおよび移動体の状況を考慮した処理を定義しなければならない。コンテキストの協調に関する視点が欠落していることから、これを実現した場合、移動体とフォグに対して協調に関する記述が散在することとなる。

移動体およびフォグについて、アクセス可能なサービスを定期的に検索する必要がある。この定期的な検索はアプリケーションの実時間性能の低下の要因となる。定期的な検索の間隔を長くした場合、タイミングによっては移動体とフォグとクラウドの実行時の変化に対応できず、非機能要求を充足した最適なアクセス対象の選択を実現することが困難となる。

提案するアーキテクチャは、構成要素間の関係の変化に動的に対応可能とし、それぞれの構成要素を独立して定義している。安直に実現した場合と比較して、提案するアーキテクチャに基づいて実現することにより、実時間性能および信頼性の向上すると考えた。

提案するアーキテクチャは、メタコンテキストを持つ Coordinator によるベースコンテキストの協調を実現している。コンテキストの協調により、動的な構成要素間の関係の変更を実現することで、非機能特性を考慮した最適なアクセス対象の選択を可能としている。また、移動体とフォグはそれぞれ独立して定義され、協調に関する記述は Coordinator としてモジュール化している。

非機能特性を考慮して柔軟に Coordinator の配置を決定することが可能である。Coordinator の配置のパターンは、前章の図 6 に示した。例えば、実時間性能を重視する場合には、より移動体の近くに Coordinator を配置することが考えられる。信頼性を重視する場合には、Coordinator を分散配置することが考えられる。

#### 4.2 車車間アドホックネットワークに関する研究との比較

Truong らによって提案されている FSDN VANET[4] と本研究で提案したアーキテクチャの比較を行ない、本研究の新規性および利点を考察する。FSDN VANET は車車間アドホックネットワークのための実時間性能とモビリティを考慮したアーキテクチャである。

FSDN VANET は、実行時に変化するクラウド上のサービスに応じて、フォグの構成を動的に変更する。この構成の変更は、クラウド上に配置された SDN Controller によって実現される。実行時に変化するフォグとクラウド上のサービスに応じて移動体を再構成すべき場合がある。この場合、FSDN VANET は移動体に関する視点が欠落している。移動体の再構成を実現するためには、移動体はフォグを経由して再構成情報を取得する必要がある、実

時間性能の低下の要因となる。本研究で提案するアーキテクチャは、移動体とフォグの動的な再構成を実現する。Coordinator により移動体とフォグのコンテキストを変化させ再構成を行なう。移動体の再構成が必要な場合には、移動体はフォグを経由した再構成情報の取得を必要としないことから、FSDN VANET と比較して、実時間性能が良いと考えられる。

実時間性能の向上が求められる場合、クラウド上に SDN を配置することを前提としていることから、困難である。本研究で提案するアーキテクチャは、移動体とフォグの再構成を実現する Coordinator を独立して定義している。前節で説明したとおり、非機能要求に応じて Coordinator の配置を柔軟に決定することが可能となっている。

## 5 おわりに

本研究は、非機能要求を充足した IoT アプリケーションの開発を目的とし、コンテキストに応じて Fog および組込みソフトウェアを動的に再構成するアーキテクチャを提案した。結果として、提案するアーキテクチャに基づいて実現することで、コンテキスト間の協調によって移動体と Fog を再構成することが可能となった。

## 参考文献

- [1] European Telecommunications Standards Institute, “Mobile Edge Computing (MEC); Framework and Reference Architecture,” *Technical Report*, 2016.
- [2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog Computing and Its Role in the Internet of Things Characterization of Fog Computing,” *in the first edition of the MCC workshop on Mobile cloud computing*, pp. 13-15, 2012.
- [3] K. Ashton, “That ‘Internet of Things’ Thing,” *RFID Journal*, 2009.
- [4] N. B. Truong, G. M. Lee, Y. G. Doudane, “Software Defined Networking-based Vehicular Adhoc Network with Fog Computing,” *Integrated Network Management*, pp. 11-15, 2015.
- [5] P. Clements, F. Bachmann, L. Bass, D. Garkan, J. Ivers, R. Little, R. Nord, and J. Stafford, *Documenting Software Architectures Views and Beyond Second Edition*, Addison Wesley, 2010.
- [6] P. Oreizy, M. M. Gorlick, R. N. Taylor, D. Heimbigner, G. Johnson, N. Medvidovic, A. Quilici, D. S. Rosenblum, A. L. Wolf, “An Architecture-Based Approach to Self-Adaptive Software,” *IEEE Intelligent Systems*, vol. 14, pp. 54-62, 1999.
- [7] P. Mell, T. Grance, “The NIST definition of cloud computing,” *Technical Report*, 2011