

コンテキスト協調を考慮したIoTシステムのための ソフトウェアアーキテクチャの実用性の評価と考察

— スマートコンビニを題材として —

M2017SE009 村瀬賢

指導教員：野呂昌満

1 はじめに

IoT(Internet of Things) システムは、移動体としての組込みシステムとサービスが協調し、振舞いは外部環境や利用者の要求の変化に対応するものとして実現される。このことから、組込みシステムおよびサービスを、外部環境や利用者の要求(コンテキスト)に応じて再構成するものとして設計・実現する試みが盛んに行われている [1].

組込みシステムとサービスの協調の論理は、互いのコンテキストの影響を受けて変化することから、複雑になりやすい。江坂らは、コンテキスト協調を考慮した IoT システムのためのアーキテクチャ [5] を提案している。江坂らは、複数のコンテキストと、それらが互いの影響を受けて変化する振舞いとの関係をコンテキスト協調として定義した。このアーキテクチャは、コンテキストおよび横断的コンサーンとなる非機能特性をモジュール化して分離している。コンテキストに応じた振舞いのための構造を動的再構成により実現している。コンテキスト協調のための構造を自己反映的な動的再構成の仕組みとして定義し、コンテキスト協調論理のための記述を、分割統治的に整理可能としている。

本研究の目的は、江坂らのアーキテクチャに基づいてコンテキスト協調を行なうアプリケーションを設計・試作し、この結果から有用性について考察することである。アーキテクチャの実現としてアプリケーションフレームワークの設計・実装を行ない、この成果物を用いてスマートコンビニアプリケーションの試作を行なう。アプリケーションフレームワークにおいて、動的再構成のためのポリシーはホットスポットの1つとなる。このホットスポットを充足するためのコンポーネント部品の標準化が可能になれば、コンポーネント部品の組合せとして記述されたポリシーに基づく動的再構成が可能になる。

2 背景技術

2.1 コンテキスト協調

IoT システムでは、物理的な場所に計算場が定義され、計算場において協調が行なわれる。Epsilon モデル [3] では、計算場をコンテキストとして取り扱い、このコンテキストに応じたオブジェクト間の協調を定義する。すなわち、組込みシステムとサービスのコンテキストと振舞いの関係は、計算場コンテキストに応じて動的に再構成される。江坂らは、このコンテキスト間の協調をコンテキスト協調として定義している。

コンテキスト協調の実現のために、コンテキスト記述には複数のコンテキストの組合せに応じた振舞いが定義される。協調に関係するコンテキストが複雑になれば、

コンテキストの組合せは爆発的に増大する。関連する振舞い定義の数は、最大ですべての協調対象の依存するコンテキストの組合せの数にのぼる。もし、コンテキスト依存の振舞い毎に多相型を定義する場合、その数はコンテキストの組合せの数が必要となる。このことから、コンテキスト記述の分割を考える必要がある。

江坂らは、コンテキスト記述の複雑さを解消する方法として、階層的に記述する方法を用いている。鶴林ら [4] は、コンテキストと振舞いの関係を変化させるコンテキストをメタコンテキストとし、コンテキスト階層を定義することで、このコンテキスト記述を整理している。IoT システムにおいて構成要素のコンテキストと振舞いの関係は計算場コンテキストに依存して決定する。コンテキスト階層を定義した場合、計算場に関するコンテキストをメタコンテキストと捉えることができる。このコンテキストの階層は多段に拡張することができ、このことから、コンテキスト記述を分割統治的に整理することができる。これは手続き指向実現において、場合分けを複数の関数で記述することに相当する。

以上をまとめると、IoT システムにおいて、計算場コンテキストをメタコンテキストとして定義することで、コンテキスト協調に関する記述は階層的に記述され、分割統治的に整理することができる。結果としてコンテキスト協調論理の簡便な記述が得られる。本研究では、江坂らの簡便なコンテキスト記述を可能とするために設計された IoT システムのためのソフトウェアアーキテクチャに基づき、アプリケーションを試作し、このアーキテクチャの実現可能性について考察する。

2.2 PBR パターン

自己適応計算は、自己表現に基づいて再構成することにより振舞いを制御する。アスペクト指向は静的再構成、コンテキスト指向はコンテキストに応じた動的再構成を行なう自己適応計算である。PBR パターンは、コンテキスト指向およびアスペクト指向を統一的に取り扱う。

PBR パターンの静的構造と動的振舞いを図 1(a), (b) に示す。PBR パターンは、コンテキストの変化を含むポリシー (Policy), ポリシーに応じて変化する再構成後のオブジェクト群を代表するアスペクトオブジェクト (AspectObject), 再構成の仕組みとして、アスペクトオブジェクト (AspectObject) のインスタンス生成を行なうファクトリ (Factory) から構成される。オブジェクト (Object) 間のメッセージを横取りし、ポリシー (Policy) がその記述に従ってファクトリ (Factory) を起動し、ファクトリ (Factory) はアスペクトオブジェクト (AspectObject) を

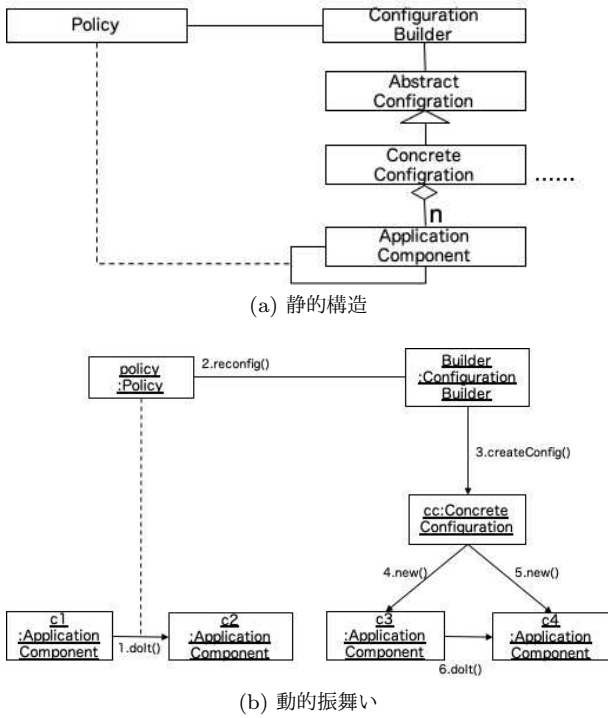


図 1: PBR パターン

生成する。ここでは、オブジェクト間のメッセージの横取りを、関連クラスによって表現している。

3 アーキテクチャ

PBR パターンおよび、このパターンを用いた江坂らのアーキテクチャについて説明する。このアーキテクチャは、PBR パターンを用いて IoT システムに特徴的なコンサーンを統一的に取り扱い可能としている。

3.1 IoT システムの横断的コンサーン

江坂らのアーキテクチャは、IoT システムにおける横断的コンサーンを分離している。IoT システムは、組込みシステムとサービスが連携して実現される。[6] で識別された、組込みシステムに特徴的なコンサーンとして、次の 4 つを分離している。

1. 並行性コンサーン
2. コンテキストコンサーン
3. 実時間性コンサーン
4. 耐故障性コンサーン

さらに IoT システムに特徴的なコンサーンとして次の 2 つを識別し、分離している。

1. 位置透過性コンサーン
2. メタコンテキストコンサーン

図 2 にこれらのコンサーン間の関係の概要を示す。IoT システムにおいて、組込みシステムは、その機能の一部をサービスとして実現する。組込みシステムには、サービスにアクセスするために位置透過性に関連するメッセージ通信が実現される。このメッセージ通信は位置透過性コンサーンとして、組込みシステムとサービス間に横断

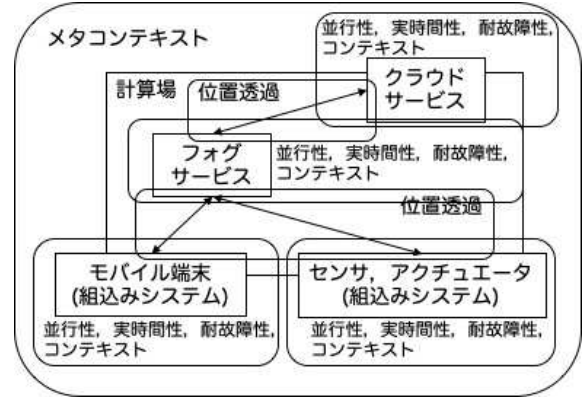


図 2: コンサーン間の関係の概略

する。組込みシステムおよびサービスは、それぞれ並行に動作し、コンテキストアウェアで実現され、対象によっては耐故障処理および実時間処理が実現される。このことから、組込みシステムおよびサービスそれぞれには、コンテキスト、並行性、実時間性、耐故障性コンサーンが横断する。コンテキスト協調は計算場における協調を実現することから、コンテキスト協調に関するメタコンテキストコンサーンは特定の計算場に存在する組込みシステムとサービス全体に横断する。

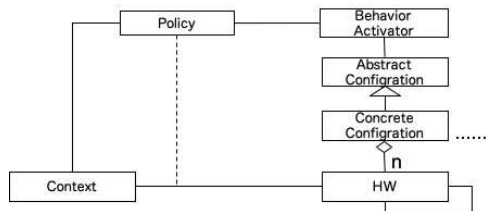
以下、江坂らの提案した IoT システムのアーキテクチャを説明する。アスペクト指向により非機能特性に関する横断的コンサーンをモジュール化し、コンテキスト指向によりコンテキスト、メタコンテキストに関するコンサーンをモジュール化する。PBR パターンを適用することにより、これらを統一的に扱う構造として設計している。ここでは、コンテキスト、メタコンテキストコンサーンについて議論したいので、並行性、実時間性、耐故障性コンサーンについては省略する。

3.2 コンテキスト

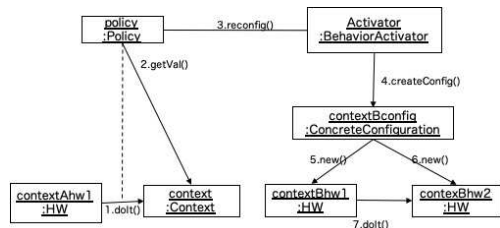
コンテキストに関連する記述を分離した構造を定義するために、PBR パターンを適用している。コンテキスト指向プログラミング言語にあるように、コンテキストとこれに応じた振舞い、振舞い活性化手続きを分離し、独立に変更できるように設計する。静的構造と動的振舞いを図 3(a), (b) に示す。PBR パターンを適用し、ポリシーをコンテキスト (Context)、ファクトリを振舞い活性化手続き (BehaviorActivator) とした。Context は、HW 間のメッセージ通信を横取りし、その状態を変化させ、BehaviorActivator を用いて状態に応じた ContextHW を活性化し、ContextHW にメッセージを送る。

3.3 メタコンテキスト

メタコンテキストに関する構造については、PBR パターンを自己反動的に適用し、コンテキスト協調に関連する記述を分離している。これにより、メタコンテキストとコンテキストに関する記述は、独立して変更できる。自己反動的に適用することで、メタレベルとベースレベルを同じ構造で定義したことから、理解も容易である。静的構造と動的振舞いを図 4(a), (b) に示す。Object 間のメッ

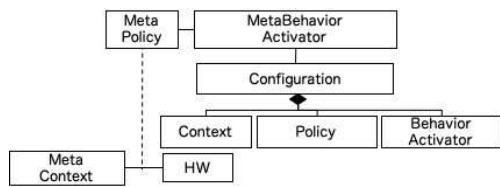


(a) 静的構造

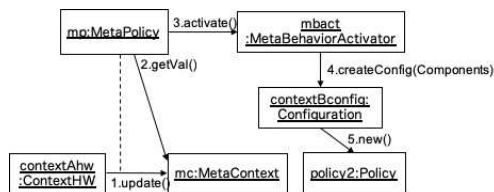


(b) 動的振舞い

図 3: コンテキストウェアハードウェア



(a) 静的構造



(b) 動的振舞い

図 4: コンテキスト協調

セージを横取りし、メタコンテキスト (*MetaContext*) の状態を変化させる。 *BehaviorActivator* は、 *MetaContext* の状態が変化したさいにコンテキストアスペクトの *BehaviorActivator* に定義されるコンテキストと振舞いの組を動的に再構成する。

4 事例検証および考察

本研究では、スマートコンビニを事例とし、江坂らのアーキテクチャに基づいてアプリケーションを試作することにより、この有用性について考察する。

仮想店舗システムは仮想籠としてのスマートデバイスと仮想棚サービスが協調する(図5)。仮想籠は、様々な店舗を移動し、入店した店舗の仮想棚から店舗商品一覧または、全商品一覧を取得し、表示する。店舗内にはディスプレイが設置され、店舗の在庫数に応じて仮想棚から商品一覧を取得し表示する。特定の店舗の在庫数の変化に伴い、特定の地区の顧客群の嗜好が特定され、顧客群の嗜好に応じて商品一覧の表示順は変化する。すなわち、仮想籠は位置コンテキスト、仮想棚は在庫コンテキスト

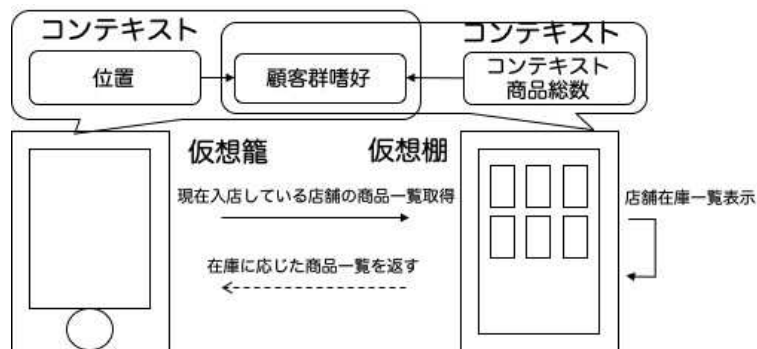


図 5: 仮想店舗システム

に応じて振舞い、特定の地区の顧客群の嗜好は、計算場コンテキストとなる。顧客群の嗜好コンテキストに応じて、位置コンテキストおよび在庫コンテキストに応じた振舞いは変化する。

コンテキスト記述を簡便にするために、階層的なコンテキストの構造を持つアーキテクチャが定義されている。このアーキテクチャに基づくことで、コンテキストに関連する記述は分割統治的に整理して記述される。このアーキテクチャに基づく仮想店舗システムの詳細構造を図6(a), (b) に示す。仮想籠は位置コンテキスト、仮想棚は在庫コンテキストに応じて *BehaviorActivator* によって再構成する。顧客群の嗜好コンテキストに応じてそれぞれの *BehaviorActivator* が再構成され、仮想籠と仮想棚のコンテキストと振舞いの関係が変化する。コンテキストに応じた個々の振舞いの再構成と、メタコンテキストに応じた協調の再構成をそれぞれ分離して定義することができた。さらに、性別によって顧客群の嗜好に応じた構成方法を変更するさいには、性別コンテキストに応じた再構成を上位に定義することで対応できる。

2.1章で述べたように、コンテキストを階層的に定義しない場合、コンテキスト協調に関する記述は switch-case 文一文で全てのコンテキストの組合せを記述することに相当し、複雑で理解しづらいものとなる。例えば、仮想棚は3種類の商品の在庫の有無それぞれの組合せに応じた8種類の商品一覧の表示方法を定義している場合、ここに嗜好も関連して表示方法を変更しようとするれば、その種類の数が商品有無の組合せに乘算される。商品の種類が多くなれば、その組合せは爆発的に増大する。

[7, 2] は、IoTシステムにおける個々の構成要素の動的再構成のためのアーキテクチャを提案している。孫ら [7] の提案するアーキテクチャは、ポリシーに基づくコンポーネントの再配置による動的適応を行なう。Keeney ら [2] の提案するアーキテクチャは、ポリシーに基づいてメタモデルを選択し、ベースレベルのコンポーネントに反映する。一方、江坂らのアーキテクチャは、これらのアーキテクチャを基礎とし、さらにコンテキスト協調を考慮した動的再構成のためのアーキテクチャを提案した。[7, 2] はいずれもコンテキスト協調を行なうさいに、その記述の複雑さは実現に依存する。鶴林ら [4] のコンテキスト階層を参考にし、コンテキスト記述を階層的な構造から定義されている。この階層構造は PBR パターンを自己反映

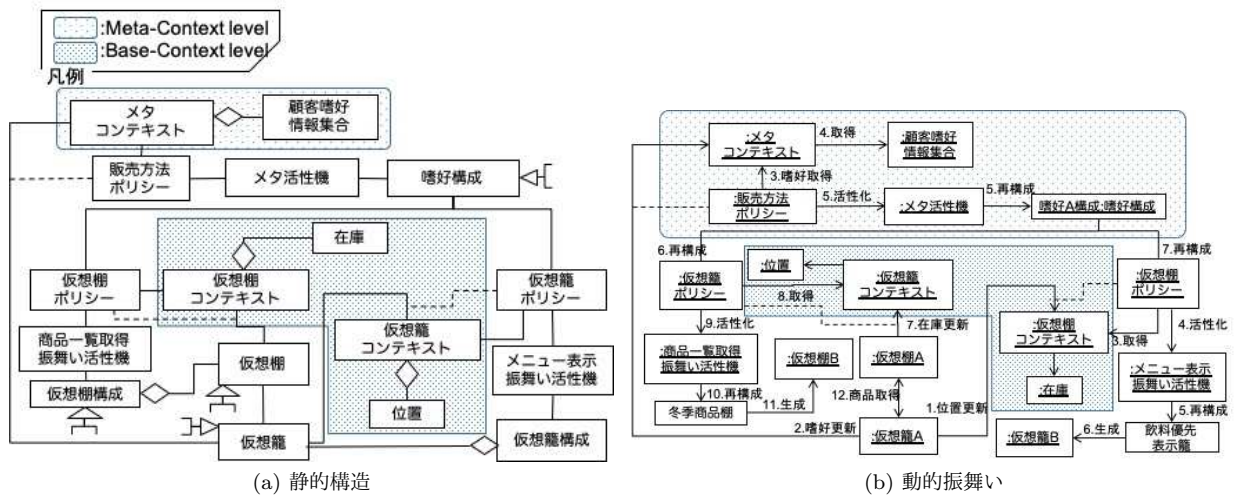


図 6: メタコンテキストを分離した仮想店舗システムの詳細構造

的に適用することで定義している。同じ構造のパターンからコンテキスト階層を定義したことにより、コンテキスト協調に関する記述は簡便で理解しやすいものとなる。

5 今後の課題

江坂らのアーキテクチャに基づき、スマートコンビニアプリケーションの設計を行なった。Merezeanu および孫らのアーキテクチャに基づいて設計を行なった場合には、コンテキスト協調について、構造ではなく手続きの問題として、コンテキストに応じた振る舞いの変更が取り扱われる。江坂らのアーキテクチャに基づくことにより、コンテキストおよびメタコンテキストそれぞれを自己反映的に適用した共通構造によって取り扱うことが可能となることを確認した。振る舞いを分割統治的に整理し、それぞれの役割を持ったコンポーネントとして定義できることを確認した。今後、コンテキスト協調コンサーンを分離した場合としない場合でプログラムコードを記述し、複雑さが軽減されることを確認する。

本研究では、簡便なコンテキスト記述を可能とする IoT システムのためのソフトウェアアーキテクチャに基づき、スマートコンビニアプリケーションの設計を行なった。アーキテクチャは、複数のコンテキストと振る舞いの組合せに関する記述は膨大で複雑になり易いことから、簡便なコンテキスト記述を可能とするために自己反映的な動的再構成の仕組みを持つ構造として定義されている。このアーキテクチャに基づく設計では、コンテキスト記述は分割統治的に整理され、簡便になることを確認した。今後の課題として、アーキテクチャに基づく実現としてのアプリケーションフレームワークの定義および、生成系の実現可能性について考察する。

参考文献

[1] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos: "Context aware computing for the internet of things: A survey",

IEEE communications surveys & tutorials, Vol.16 No.1, pp.414-454 (2014).

[2] John Keeney and Vinny Cahill: "Chisel: A policy-driven, context-aware, dynamic adaptation framework", In Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on, pp. 3-14. IEEE (2003).

[3] Ubayashi Naoyasu and Tamai Tetsuo: "Modeling collaborations among objects that change their roles dynamically and its modularization mechanism", Systems and Computers in Japan, Vol.33 No.5, pp.51-63 (2002).

[4] 鷓林尚靖, 玉井哲雄: "オブジェクト間の協調動作を表現する自己反映並行計算モデル", 情報処理学会研究報告ソフトウェア工学 (SE), Vol.1996 No.112 (1996-SE-112), pp.25-32 (1996).

[5] 江坂篤侍, 野呂昌満, 沢田篤史: "コンテキスト協調を考慮した iot システムのためのソフトウェアアーキテクチャの設計 (採録決定)", ソフトウェア工学の基礎, Vol.25, 2018.

[6] 江坂篤侍, 野呂昌満, 沢田篤史, 繁田雅信, 谷口弘一: "コンテキストウェアネスを考慮した組込みシステムのためのアスペクト指向アーキテクチャの設計", ソフトウェア工学の基礎, Vol.24, pp.3-12 (2017).

[7] 孫静涛, 佐藤一郎: "IoT 環境における動的適応能力を備えた再構成可能なアーキテクチャ", マルチメディア, 分散協調とモバイルシンポジウム 2016 論文集, Vol.2016, pp.1116-1121 (2016).