

IoT の柔軟な相互運用性を実現する ソフトウェアアーキテクチャに関する研究

M2019SE701 横山 史明

指導教員：沢田 篤史

1 はじめに

本研究では、IoT における柔軟性と相互運用性の確保をソフトウェア構造の問題と捉え、スマートホームにおいて相互運用性実現のための論理を組み合わせる段階で動的に追加できること（これを本研究では動的適応と呼ぶ）を可能にするソフトウェアアーキテクチャを提案する。提案するアーキテクチャにより、スマートホームにおける柔軟な相互運用を可能とするとともに、保守しやすい IoT アプリケーションの構築基盤を提供する。

2 IoT における相互運用性と柔軟性

2.1 IoT 機器の柔軟な相互運用

スマートホーム等で求められる IoT 機器の柔軟な相互運用について、簡単なアプリケーションを例に説明する。図 1 に洗濯完了通知アプリケーションの概念図を示す。

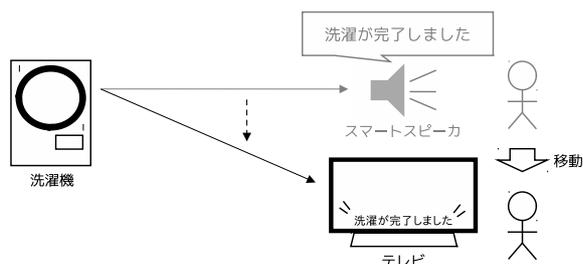


図 1: 洗濯完了通知アプリケーション

このアプリケーションは利用者に洗濯の完了を通知することを目的とし、通常時にはスマートスピーカーを介して洗濯完了メッセージを出力するように作られている。利用者は家庭内を移動することから、スマートスピーカーが適切な出力機器ではない場合もある。このアプリケーションが利用者の状況や嗜好に適応して効果的に目的を果たすために、例えば次のような対応が求められる。

- 洗濯が完了した時点での利用者の位置に応じて出力
- 利用者の就寝時間帯を考慮して出力

これを IoT 機器の連携を用いて実現するためには、次の各レベルの変更に対処可能であることが求められる。

- ネットワーク接続レベルの変更
センサによる利用者の位置の検出結果や時間帯に応じた、接続相手の IoT 機器の変更
- メッセージングプロトコルレベルの変更
相手の IoT 機器が受付可能なメッセージ通知プロトコルに応じた、メッセージングプロトコルの変更
- 意味レベルの変更
相手の IoT 機器の機能や利用者の嗜好に応じた、表現メディアやメッセージ内容の変更

2.2 IoT の相互運用性に関する課題

ネットワークを通じて機器同士を連携させるために様々な標準化が行われている [1, 2, 3]。さらに家電機器に特化した標準として、機器をホームネットワークによって連携させるための規格 [4] も策定されている。

IoT 機器導入のシナリオにおいて、各レベルに対して変更への柔軟な適応を実現するためには、既存の標準や製品群の単なる組み合わせでは難しい。いずれも、それぞれに閉じた範囲の IoT 機器同士が連携できることを第一義として定められているからである。標準をまたぐ IoT 機器の相互運用性を実現する機構が必要である。

2.3 利用状況に応じた柔軟なサービスの提供

情報家電の知的制御のためにオントロジーを用いる試みは古くから行われてきた [5]。前節で述べた標準 [1, 3] でも、ネットワーク接続レベル、メッセージングプロトコルレベルでの標準だけでなく、IoT に関連する概念とそれを表す語彙についての標準化を行っている [6, 7]。

意味レベルの相互運用性を確保するためのアプローチも提案されている。井垣らは、家電の連携を実現するネットワーク家電連携サービスを開発している [8]。サービス指向アーキテクチャに基づいて実現することで機器間の結合を疎にし、相互接続性や機器拡張性を向上させている。Desai らは、IoT アプリケーションの間で意味変換を行うセマンティックゲートウェイサービスのアーキテクチャを提案している [9]。他にも、意味レベルでの連携を可能とする IoT ゲートウェイのアーキテクチャが Datta らにより提案されている [10, 11]。これらゲートウェイサービスを特定領域へ応用した結果も報告されている [12]。

アプリケーションが柔軟に接続先や振舞いを変更する際に、利用者の状況や嗜好に応じた変更や、接続先の IoT 機器に応じた変更に関する論理は、アプリケーション本来の論理とは異なる関心事に基づく。保守性が高く、柔軟で相互運用可能な IoT アプリケーションを開発するためには、これらの要素技術の統合を系統的に支援するための共通基盤が必要である。

利用者の状況や嗜好をコンテキストと捉えた場合に、コンテキストによりホームネットワークの制御を行う研究もおこなわれている [13, 14, 15]。これらの手法を利用すると、コンテキストにより IoT アプリケーションの振舞いの制御を変更可能である。しかしこれらはコンテキストに応じて接続先の IoT 機器を利用するための手順を記述した論理を動的に切替えることを目的にしている。柔軟な相互運用を実現するためには、アプリケーションに対して相互運用性実現の論理を動的に追加し、動的適応を行うことができる必要がある。

て通常は「スマートスピーカ」が指定されているが、利用者が居間にいる時は「TV」を通知先に変更するものとする。そして、変更先の「TV」はRESTメッセージを受け付け、完了通知を音声出力する機能はなく、画面に文字を出力する機能のみが搭載されているものとする。

図には説明のためにメッセージ通知を矢印で併記しており、番号がメッセージの順番を表す。洗濯完了通知アプリケーションは「洗濯機」へ完了通知依頼メッセージを送付することで開始され、動作中は「人感センサ」を用いて利用者の居場所を検知しているものとする。

動的再構成は、利用者の移動を検知した「人感センサ」が起こす「コンテキスト」の変更（メッセージ1）をきっかけに行われる。「IoT 機器再構成ポリシー」がセンサからのメッセージを横取りし、変更後のコンテキスト（利用者位置＝「居間」）に応じて、「洗濯機」が完了メッセージを通知すべきIoT機器を「スマートスピーカ」から「TV」に切り替えるよう、「アダプタファクトリ」に再構成の指示を行う（メッセージ2）。

「アダプタファクトリ」は、通知先の切り替えに必要なアダプタ「洗濯機 2tv」を生成し「TV」に関連付けるとともに、「洗濯機」の終了通知先をこのアダプタに切り替える（メッセージ3）。アダプタには、意味、メッセージングプロトコル、ネットワーク接続のレベルでの切り替えに必要な論理を生成（メッセージ4）することで「洗濯機」からの完了通知を「TV」で出力できるようにする。その後、「洗濯機」が、完了通知を通知先に送信（メッセージ5）すると、「洗濯機 2tv」を構成する変換器がそれぞれのレベルでの変換を行い（メッセージ6）、通知先の「TV」に通知する（メッセージ7）。

通信相手のIPアドレスを変更する方法や、メッセージ形式とそれに伴う通信手順を変更する方法など、以上の振舞いを実現する技術には様々な選択肢がある。このアーキテクチャでは、動的適応のためのアダプタおよび変換器を、特定の実装技術には依存しない変換論理のためのモジュールとして定義している。

4 考察

4.1 アーキテクチャに基づく実装例

提案するアーキテクチャに基づいて、異種IoT機器間の柔軟な相互運用が可能となることを示すために、簡単なメッセージ通知アプリケーションの実装を行った。

実装したプロトタイプアプリケーションのハードウェアおよびネットワーク構成を図5に示す。

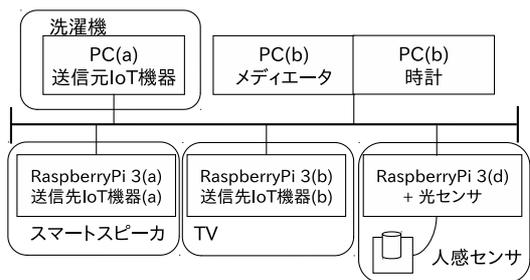


図 5: メッセージ通知アプリケーションの構成

これまでに説明してきたアプリケーション（図1）を想定した構成となっているが、実際の家電機器の代わりにRaspberry Pi 3とノートPCを用いている。これらの機器は、すべて同一ネットワークセグメント上に接続した。

ここでは、ノートPC(a)を洗濯機に見立て、ソケット通信を用いて独自形式のメッセージ通信を行うクライアントとした。また、Raspberry Pi 3(a)をスマートスピーカに見立て、洗濯機からのソケット通信を待ち受けるサーバとした。Raspberry Pi 3(b)はTVに見立て、HTTP通信でメッセージを受け取るサーバとした。Raspberry Pi 3(c)には、光センサ（Grove Light Sensor）を接続して人感センサを模擬している。PC(b)をメディエータとしているが、これは、動的適応を仲介するために実装上導入した要素である。また、IoT機器として時計（PC(b)を便宜上共有）も接続し、時間をコンテキスト情報として用いることができるようにしている。

これらの機器構成を用い、前章で提案したアーキテクチャに基づいて次のシナリオを実装し、動作確認をした。

- メッセージ送信先の変更（ネットワーク接続レベルの適応）：洗濯機（PC(a)）のメッセージ送信先を、TV（Raspberry Pi 3(b)）に変更
- メッセージ形式の変更（ネットワーク接続およびメッセージングプロトコルレベルの適応）：洗濯機（PC(a)）のメッセージの送信先をTV（Raspberry Pi 3(b)）に変更し、HTTPアダプタによりプロトコルを変換
- 機能とIoT機器構成の変更（三つのレベルの適応）：人感センサの検知結果と時間帯に応じ、IoT機器構成ポリシーによって、メッセージ送信先をTV（IoT機器（b)）に変更するとともに、アダプタの動的生成により、メッセージングプロトコルをHTTPに、通知機能を音によるビーブ機能に変更

プロトタイプアプリケーションにおいてこれらのシナリオを実現するために、前章で提案したアーキテクチャを図6に示すように具体化した。この設計では、IPアドレスやメッセージングプロトコル、機能の動的変換の役割をメディエータに担わせる。

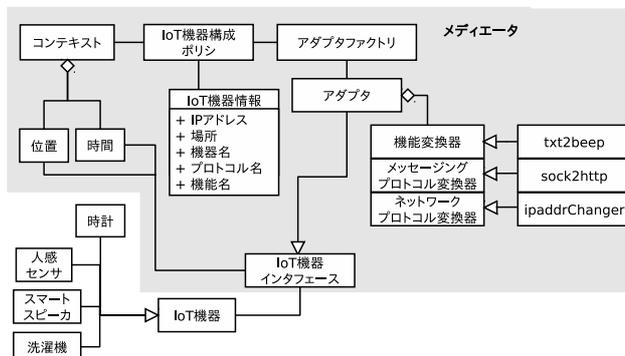


図 6: メッセージ通知アプリケーションの詳細構成

この図にあるように、IoT機器を相互に連携させるための論理をアダプタの切り替えにより、そして利用者の状況や嗜好に適用させるための論理をコンテキストの切り替えにより、それぞれ実現できることが確認できた。

4.2 アーキテクチャ設計の妥当性

本研究で実現した動的適応は、Reflection パターン [19] や Rainbow フレームワーク [20] や、その他の一般的なフレームワークを利用して実現することが可能であると考えられる。それらに対し、本研究のアーキテクチャでは、スマートホームでの利用を想定していることから、過度の複雑性を回避し、性能や保守性への悪影響を防ぐ設計としている。すなわち本研究は、自己反映ないしは自己適応計算を一般的なモデルとして表現することを目的とするこれらのアーキテクチャを、コンテキストに応じた適応による再構成に特化して詳細化したものであると位置づけることができる。

5 おわりに

本研究では、スマートホームなどで用いられる IoT アプリケーションにおける柔軟性と相互運用性の確保を可能とするソフトウェアアーキテクチャを提案した。アーキテクチャの設計にあたり、IoT 機器を相互に連携させるための論理と、利用者の状況や嗜好に適用させるための論理とを明確に分離させながら、IoT 機器間の多層にわたる動的適応のための構造を定義した。これにより、スマートホームにおける柔軟な相互運用を可能にし、保守しやすい IoT アプリケーションの構築基盤を提供する。

提案するアーキテクチャに基づいて、IoT 機器間の動的適応が可能であることを示すために、機器間の簡単なメッセージ通知アプリケーションを実装した。この実装を通じ、利用者の状況に応じてメッセージ送信先の IoT 機器を変更し、その機器に合わせた形態のメッセージを通知できることを確認した。また、利用者の状況に適応する論理と、異なるプロトコルを持つ機器へ適応する論理とが分離できることも確認した。

参考文献

- [1] *oneM2M: Standards for M2M and the Internet of Things*, available from [\(https://www.onem2m.org/\)](https://www.onem2m.org/)(accessed 2021-01-15).
- [2] Bassi, A., Bauer, M., et al.(eds.): *Enabling Things to Talk — Designing IoT Solutions with the IoT Architectural Reference Model*, Springer, 2013.
- [3] W3C: *Web of Things at W3C*, available from [\(https://www.w3.org/WoT/\)](https://www.w3.org/WoT/)(accessed 2021-01-15).
- [4] エコーネットコンソーシアム: ECHONET Lite 規格の特徴と概要, 入手先 [\(https://echonet.jp/about/features/\)](https://echonet.jp/about/features/) (参照 2021-01-15) .
- [5] 山田 知秀, 飯島 正, 他: オントロジーを利用した情報家電エージェント協調アーキテクチャ, 第 19 回人工知能学会全国大会, 1B2-03, 2005.
- [6] *Ontologies Used for oneM2M*, available from [\(https://www.onem2m.org/technical/onem2m-ontologies/\)](https://www.onem2m.org/technical/onem2m-ontologies/)(accessed 2021-01-15).
- [7] W3C: *Semantic Sensor Network Ontology*, available from [\(https://www.w3.org/TR/vocab-ssn/\)](https://www.w3.org/TR/vocab-ssn/)(accessed 2021-01-15).
- [8] 井垣 宏, 中村 匡秀, 他: サービス指向アーキテクチャを用いたネットワーク家電連携サービスの開発, 情報処理学会論文誌, Vol. 46, No. 2 (2005), pp. 314–326.
- [9] Desai, P., Sheth, A. P., et al.: Semantic Gateway as a Service Architecture for IoT Interoperability, *Proc. IEEE MS 2015*, pp. 313–319, 2015.
- [10] Datta, S. K., Bonnet, C, et al.: An IoT Gateway Centric Architecture to Provide Novel M2M Services, *Proc. IEEE WF-IoT 2014*, pp. 514–519, 2014.
- [11] Datta, S. K. and Bonnet, C: Smart M2M Gateway Based Architecture for M2M Device and Endpoint Management, *Proc. IEEE iThings, GreenCom and CPSCoM 2014*, pp. 61–68, 2014.
- [12] Jabbar, S, Ullah, F., et al.: Semantic Interoperability in Heterogeneous IoT Infrastructure for Healthcare, *Wireless Communications and Mobile Computing*, Vol. 2017, Article ID 9731806 (2017).
- [13] 小林 英嗣, 依田 育生: “ホームネットワークにおけるコンテキストとユーザ操作履歴を用いたサービス制御方式の提案”, 情報処理学会論文誌, Vol. 47, No. 2 (2006), pp. 507-520.
- [14] Bucchiarone, A., Marconi, A., et al.: “A Context-Aware Framework for Dynamic Composition of Process Fragments in the Internet of Services”, *Journal of Internet Services and Applications*, Vol. 8, No. 6, 23 pages (2017).
- [15] Krishna, A., Le Pallec, M., et al.: “IoT Composer: Composition and Deployment of IoT Applications”, *Proc. IEEE/ACM ICSE-Companion 2019*, pp. 19-22 (2019).
- [16] Bass, L., Clements, P., et al.: *Software Architecture in Practice*, Third Edition, Addison-Wesley, 2012.
- [17] Gamma, E., Helm, R., et al.: *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1994.
- [18] 江坂 篤待, 野呂 昌満, 他: インタラクティブシステムのための共通アーキテクチャの設計, コンピュータソフトウェア, Vol. 35, No. 4 (2018), pp. 3–15.
- [19] Buschmann, F., Meunier, R., et al.: *Pattern-Oriented Software Architecture, Volume 1: A System of Patterns*, Wiley Publishing (1996).
- [20] Garlan, D., Cheng, S.-W., et al.: Rainbow: Architecture-Based Self-Adaptation with Reusable Infrastructure, *Computer*, Vol. 53, No. 10, pp. 46–54, IEEE Computer Society (2004).