

コンテキスト指向に基づくスマート家電制御システム

2018SE038 前川友輔 2018SE069 大野靖弘

指導教員：沢田篤史

1 はじめに

スマート家電が普及し、日常生活で使用する機器がインターネットに接続されるようになった。現在のスマート家電では様々なメーカーが独自の仕様や規格に基づいて機器を開発しており、異なるメーカーの機器の間で様々なデータを共有することができないという問題が存在する。課題を解決するために、異なる規格にまたがってデータを相互に利用し、利用者や環境への適応が可能なスマート家電制御システムを実現する必要がある。

利用者に向けた家電制御の実現には利用者や環境情報に合わせて動的な変更を行う機能が必要である。多様な状況に応じた動的な切り替えは複雑になり開発は容易なものではない。また、スマートホームの実現を考えると複数のスマート家電を同時に制御することも並行して行わなければならない。その開発は容易ではないといえる。

本研究では室内環境に柔軟に適応できるスマート家電制御システムのアプリケーション構築を目的とする。アプリケーション構築にあたって、我々は身近な空調機器を題材とした。室内環境への適応を実現するために気温・湿度などの情報に合わせた柔軟な振り舞いを行う。このようなシステムを繰り返し開発するためには、共通のソフトウェア基盤が必要となる。本研究では、その基盤としてコンテキスト指向に基づくソフトウェアアーキテクチャの定義をする。アーキテクチャの定義には江坂ら [1] が提案した PBR パターンを引用する。

本研究を通じて、ソフトウェアアーキテクチャの定義を行い、データを相互に利用し、利用者や環境への適応が可能なスマート家電制御システムを実現するための見通しをつけることができた。また、アーキテクチャの設計とプロトタイプシステムの実装を通じて、PBR パターンの有用性について確認することができた。

2 スマート家電における環境適応に関する課題

2.1 スマートホーム

一般社団法人電子情報技術産業協会 [2] はスマートホームについて、様々な利用者の生活様式や要望に合ったサービスの提供が可能になった状態の家であると定義している。日常で使用する家電や住まいに設置されたセンサなどとインターネットを結びつける事で、利用者の要望に合ったサービスを実現する。

2018 年度のスマートホーム国内市場調査 [3] では、2018 年時点では 3 兆 936 億円という規模である。2025 年には 3 割近くの拡大が見られ、市場規模は 4 兆 240 億円まで伸びると予測されている。エアコン、照明、ロボット掃除機

などの身近な家電については 2025 年に向けて市場が拡大していくという予測が立てられており、今後は現在よりも多くの利用者がスマート家電を導入していくことが考えられる。

2.2 家電制御にあたっての課題

前節で説明したスマートホームの実現にあたって、現状では、スマート家電利用者が利便性を完全に享受することは難しい。なぜなら、様々なメーカーが独自の仕様や規格に基づいて機器を開発しており、異なるメーカーの機器の間で様々なデータを共有することができないからである。スマート家電の例として、エアコンを挙げる。エアコンでは、スマートフォンの専用アプリを用いることで稼働状況や温度の設定の確認ができたり、過去の稼働状況を管理し、学習して快適な運転を行うことができる。現状では、各製品がセンサからそれぞれが取得した情報を基に自身の制御をしており、他の家電製品にそのデータを使用することは難しい。スマートホームやスマート家電のこのような課題を解決するために、異なる家電の規格でも利用者個々の生活様式や要望に合ったサービスを可能にした家電制御システムの実現をする必要がある。

家電制御システムの実現にあたって問題となるのが、利用者や環境情報に合わせて動的な変更を行う機能が必要ということである。状況に応じた動的な切り替えを実現するには、複雑な切り替え条件を実行中に変更する仕組みが必要となり、その開発は容易ではない。また、スマートホームの実現を考えると、複数の家電製品を同時に制御することも並行して行わなければならない。開発は容易ではないといえる。

3 利用者に応じた室内環境適応のアーキテクチャの設計

本研究の目的は室内環境に柔軟に適応できるスマート家電制御システムのアプリケーション構築である。複数の空調機器間でセンサの情報を共有し、制御するシステムを開発する。システムの開発にあたって、我々は基盤となるソフトウェアアーキテクチャの定義をした。基盤となるアーキテクチャには、江坂ら [1] が提案した PBR パターンを適用する。PBR パターンを適用し作成したアーキテクチャでは、利用者が起動ボタンを押した際と室内温度が変更された際をきっかけに、室内環境におけるコンテキストの変化に応じて動的に利用する機器の再構成を定義する。

3.1 PBR パターン

PBR (Policy-Based Reconfiguration) パターンは江坂らの提案している静的および動的に再構成を行う自己適応

のためのアーキテクチャパターンである。自己適応とは自身の実行状態や動作環境を監視することで機能的振る舞いを実行時に変更することを示す。

3.2 動的な変更のためのソフトウェアアーキテクチャ

本研究で提案するアーキテクチャは、室内環境に柔軟に適応させるために動的な振る舞いをさせる必要がある。この要求を実現するには定義するアーキテクチャの設計に前節で提示した PBR パターンを用いるのが良いと考えた。

PBR パターンを用いた、利用者に応じた室内環境適応のためのアーキテクチャの基本構造を図 1、動的振る舞いを図 2 に示す。

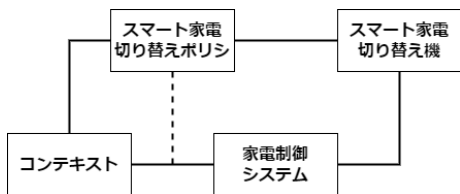


図 1 室内環境適応のためのアーキテクチャ

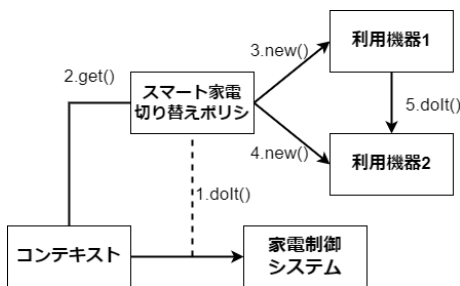


図 2 室内環境適応のためのアーキテクチャの動的振る舞い

家電制御システムでは、気温のコンテキストの変更もしくは利用者からのボタン入力を検知した時、利用機器や設定の再構成を行うメッセージ通信を行う。スマート家電切り替えポリシーが家電制御システムとそれぞれのセンサとのメッセージ通信を横取りし、コンテキストの情報を更新する。更新されたコンテキストをもとにスマート家電切り替えポリシーが利用機器や設定の変更を行う。そしてスマート家電切り替え機が利用機器や設定の再構成を行う。

ここで定義した基本構造を環境適応システムに適用した場合、気温やボタンによって変化したコンテキストの値を元にスマート家電切り替えポリシーが利用機器や設定を再構成することで、利用者の状況に合わせて利用する機器を柔軟に変更することが出来る。

3.3 利用者に応じた室内環境適応のためのシステムの設計

本節では室内環境適応のためのシステムを前節の基本構造を参考にし、ソフトウェアアーキテクチャを設計する。

本研究で提案する室内環境適応システムの静的構造を図 3 に示す。

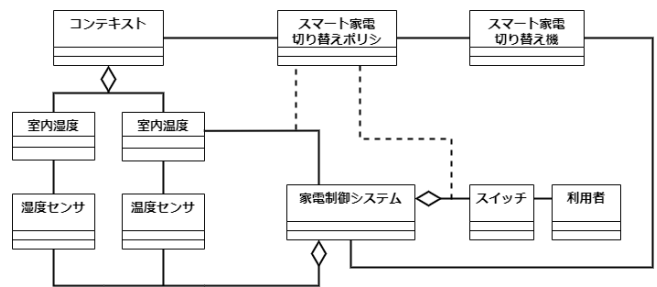


図 3 PBR パターンを適用した静的構造

各コンポーネントについて以下に説明する。

- コンテキスト
- スマート家電切り替えポリシー
- スマート家電切り替え機
- スイッチ
- 家電制御システム

コンテキストは室内環境適応システムのポリシーの判断に必要な情報を保持する。本研究では環境適応に向けたスマート家電システムの設計であるので、気温、湿度をコンテキストとして記述し変更の際に利用する。

スマート家電切り替えポリシーとはコンテキストに応じて利用機器や設定の決定を行っている。本研究でポリシーは気温や湿度による室内環境の情報と不快指数 [4] を用いて記述する。不快指数は湿度が高いときに体感温度は高くなり、湿度が低いときに体感温度が低くなる。このような関係から気温と湿度から求められる蒸し暑さの指数で数字が大きいほど蒸し暑く不快であると定義されている。体に感じる蒸し暑さは気温や湿度以外に風速の条件によって左右される。本研究では室内環境による適応であるので、風の影響が無いことから不快指数が室内の環境適用に利用できる。表 1 は不快指数の判断をもとに感じる体感の違いを示す。

表 1 不快指数による体感の違い

不快指数の値	体感
55～	寒い
55～60	肌寒い
60～65	何も感じない
65～70	快い
70～75	暑くない
75～80	やや暑い
80～85	暑くて汗が出る
85～	暑くてたまらない

スマート家電切り替え機とは、スマート家電切り替えポリシーからのメッセージをもとに利用機器や設定を変更す

る。空調機器の変更を行う事で室内環境に適応した機器を利用することができる。

スイッチは利用者が利用する際に押されるインターフェースとなる。スイッチが押されることで家電制御システムへ押されたという通知が行われその通信をもとに動的な変更を実現する。

家電制御システムでは、室内環境に応じて利用機器の変更を行うシステムである。コンテキストの情報を用いて変更された利用機器や設定を用いて室内環境適応を実現する。

本研究において動的に変更するきっかけは、ユーザが起動ボタンを押した際と室内の温度が変更された時とする。家電制御システムから送られる室内温度更新のメッセージ、もしくは利用者からのボタンが入力されたメッセージをスマート家電切り替えポリシーが横取りする。スマート家電切り替えポリシーがコンテキストの更新を行い、その更新した情報を用いてどのような処理を実行するかを決定する。スマート家電切り替え機ではポリシーによって決定された使用する家電をもとに利用機器や設定を変化させる。このようにしてコンテキストに応じた室内環境適応を実現している。利用者がボタンを押すことで動的な変更を行う振る舞いを図4、気温が変更されたことで動的な変更を行う振る舞いを図5に示す。

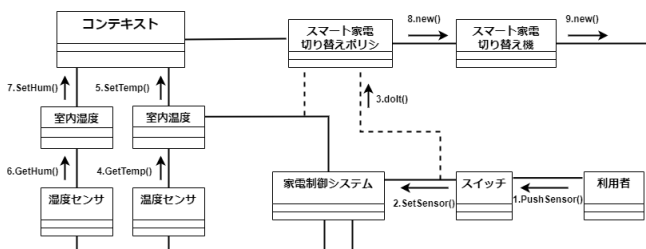


図4 利用者がボタンを押すことで変更を行う振る舞い

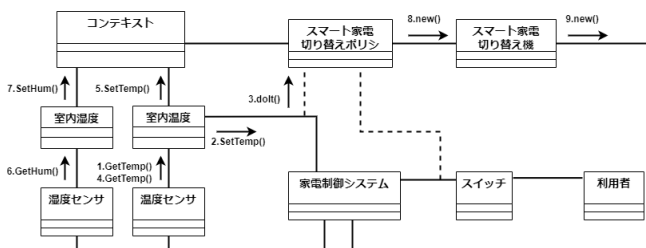


図5 室内温度の変化で変更を行う振る舞い

図4では利用者からのボタン入力のメッセージの際に、家電制御システムとボタンの通信をきっかけに、スマート家電切り替えポリシーが家電制御システムとボタンとのメッセージ通信を横取りすることを表している。家電制御システムはスイッチが入力(1.PushSensor())されたかを2.SetSensor()でセンサを呼び出して確認する。そしてボタンが押されたというメッセージが家電制御システムに送られる。このメッセージ通信をスマート家電切り替えポ

リシが横取り(3.doIt())することで再構成のきっかけとなる。気温センサの値を取得するために4.GetTemp()によって室内温度を取得する。そしてスマート家電切り替えポリシーでコンテキストの値を取得するために5.SetTemp()によって温度を更新している。次に、湿度センサの値を取得するために6.GetHum()によって室内温度を取得する。そしてスマート家電切り替えポリシーでコンテキストの値を取得するために7.SetHum()によって湿度を更新している。その後、スマート家電切り替えポリシーはスマート家電切り替え機を生成し、メッセージ通信を行うことで利用機器や設定の再構成が行われる。そして再構成された利用機器や設定をもとに家電制御システムの更新を8.new()が示している。また、切り替えられた利用機器や設定の情報を家電制御システムの更新を行う(9.new())ことで実現している。図5では、気温センサの気温変化のメッセージの際に、コンテキストの室内温度と気温センサの通信をきっかけにスマート家電切り替えポリシーがメッセージ通信を横取りすることを表している。家電制御システムは気温が変化したかを1.GetTemp()で気温センサを呼び出して確認する。気温センサを呼び出した際に、前回動的な変更が行われた気温と比較する。そして気温が変化していた場合、変更されたというメッセージが家電制御システムに送られる。このメッセージ通信をスマート家電切り替えポリシーが横取りすることで再構成のきっかけとなる。その後の処理は利用者がボタンを押すことで動的な変更が起こった際の振る舞いと同様のものが行われる。

4 利用者に応じた室内環境適応のためのシステムの実装

4.1 室内環境適応システムプロトタイプの実装

3章で提案するアプリケーションアーキテクチャに基づいて、室内環境適応システムプロトタイプをJavaで実装した。コンテキストのクラスでは利用機器や設定の判断に必要な気温と湿度に関する記述を行った。スマート家電切り替えポリシーのクラスではコンテキストの情報を用いて利用機器や設定の決定を行った。

室内環境適応システムでは室内温度に対応して動的に変更ができていないことを確認する必要がある。しかし本研究では室内環境適応システムにセンサを接続しセンサの値によって変更することが出来なかった。そのためセンサの代わりとしてキーボード入力を行う事でセンサの値の変更を行い動的な変更ができていないことを確認した。

4.2 実行結果

実装した家電制御システムプロトタイプの実行結果の例として、利用者がボタンを押した際に検知して行う実行の流れを図6に示す。

図6ではボタン入力で1を取得したとみなした場合の実行結果を示している。ボタン入力のメッセージ通信をスマート家電切り替えポリシーが横取りしコンテキストの

```
Problems @ Javadoc Declaration Console
Main [12] [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (2022/01/13 22:31:33)
ボタンを入力する場合は1を押してください
1
ボタンが入力されました
気温を入力してください
35
入力された気温は「35.0」です
湿度を入力してください
80
入力された湿度は「80.0」です
室内温度: 35.0度 室内湿度: 80.0%
エアコン: 27度 扇風機: 中風 で起動
```

図6 家電制御システム 実行結果 1

値を参照する。現在の室内温度と室内湿度を取得するためにコンテキストクラスで実現した GetTemperature() と GetHumidity() を呼び出す。本研究ではセンサを実装できていないので、GetTemperature() と GetHumidity() ではキーボード入力により気温と湿度を取得する。キーボード入力より「気温が 35 度, 湿度が 80 %」と入力し取得したとみなす。取得した気温と湿度を用いて不快指数を計算すると「90.9」となるので、実行結果では「気温が 35 度, 湿度が 80 %」, 「エアコンの設定温度を 27 度で設定し, 扇風機を中風で起動する」に変更が行われ表示されている。また, 1 以外が入力された際は, ボタン入力が行われなかったと判断を行い実行されない。

基本構造の構築に PBR パターンを用いることで, 室内環境適応システムでコンテキストクラスとスマート家電切り替えポリシクラスを切り離して記述できるようになった。今後センサの追加や利用する空調機器の変更を行う場合, コンテキストやポリシによる利用機器と設定の判断の追加や変更が容易になり, 保守性が高められた。

5 考察

本研究では身近な空調機器を題材とし, 基本構造を具現化した環境適応システムのためのアプリケーションアーキテクチャに基づいて, 室内気温と室内湿度をもとに利用機器と設定の動的な切り替えを可能にした。PBR パターンを適応することでコンテキスト, スマート家電切り替えポリシ, スマート家電切り替え機などを分離して記述することができる。コンテキストとポリシを分けて記述することで, 環境適応を実現する際の変更の影響が最小限となり保守性が優れている。PBR パターンを適用しなかった場合, 同一のクラスにコンテキスト, ポリシ, 切り替え機を記述することになりクラス内の記述量が増加する。様々な状況に対応するシステムではクラスの記述量が増加すると機能の追加や変更が困難になる。このことから PBR パターンを適用して基本構造を構築したことは妥当だったと考える。

6 おわりに

スマート家電が普及し, 日常生活で使用する機器がインターネットに接続されるようになった。スマート家電の登場により, 利用者は個々の生活様式や要望に合ったサー

ビスを受けることが可能となり, 今よりも便利な暮らしを実現することができるようになった。現状のスマート家電の問題として, 様々なメーカーが独自の仕様や規格に基づいて機器を開発しており, 異なるメーカーの機器の間で様々なデータを共有することができないということがある。スマートホームやスマート家電のこのような課題を解決するために, 異なる規格にまたがってデータを相互に利用し, 利用者や環境への適応が可能なスマート家電制御システムを実現する必要がある。

本研究では室内環境に柔軟に適応できるスマート家電制御システムのアプリケーション構築を目的とする。アプリケーション構築にあたって, 我々は身近な空調機器を題材とした。システムの実現には利用者や環境情報に合わせて動的な変更を行う機能が必要である。多様な状況に応じた動的な切り替えは複雑になり開発は容易なものではない。さらに, スマートホームの実現を考えると複数の異なる規格に基づくスマート家電を同時に制御することも並行して行わなければならない, その開発は容易ではないといえる。このようなシステムを開発するためには, 共通のソフトウェア基盤が必要となる。本研究では, その基盤として江坂ら [1] が提案した PBR パターンを用いた。

本研究を通じて, スマート家電を同一時期の同一メーカーの製品に統一するなど, 同一の規格に基づく製品を揃えなければスマート家電利用者が利便性を完全に享受することができないという課題に対して, ソフトウェアアーキテクチャの定義によって, スマート家電制御システムを実現するための見通しをつけることができた。また, また, PBR パターンと他のデザインパターンとで家電制御システムを実装したときの比較や考察も行い, PBR パターンの有用性についても確認することができた。

参考文献

- [1] 江坂篤侍, 野呂昌満, 沢田篤史, “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, pp. 3-15, 2018.
- [2] 一般社団法人 電子情報技術産業協会 スマートホーム部会 将来ビジョン検討会, “JEITA スマートホーム 将来ビジョン”, https://home.jeita.or.jp/smarthome/deliverables/pdf/society5.0_smarthome_vision.pdf, 2020. (Accessed 2022.1.10)
- [3] 株式会社富士キメラ総研 Fuji Keizai Groupe, “『スマートホーム市場総調査 2018』まとまる (2018/10/26 発表 第 18096 号)”, <https://www.fcr.co.jp/pr/18096.htm>, 2018. (Accessed 2022.1.10)
- [4] 国土交通省 気象庁, “湿度・気圧・日照時間について”, <https://www.jma.go.jp/jma/kishou/known/faq/faq5.html>, 2018. (Accessed 2022.1.10)