スマートデバイスにおける Bluetooth の適正利用を考慮したソフトウェアアーキテクチャの設計

2015SE073 鈴木雄揮　2015SE039 河上勇介　 2015SE063 奥村康平

指導教員：野呂昌満

1　はじめに

近年，スマートデバイスの普及に伴い，ワイヤレスイヤホンやスピーカなどの Bluetooth 通信を用いた電子機器の利用が増加している．Bluetooth 機能は，一般に手動で起動させ，周辺の Bluetooth 対応機器を探索し，通信接続を終了するさいは，手動で機能を停止させている．

Bluetooth 機能が動作中の間，スマートデバイスは電力を多く消費し，セキュリティ面において脆弱となる．Bluetooth 機能は，周辺の Bluetooth 対応機器を探索するさい，使用者の認証を得ずにスマートデバイス情報を送受信している．この情報にはオペレーティングシステムや Bluetooth のバージョンなどの情報が含まれる [2]. Bluetooth 機能を自動で起動，停止させる方法は実現されておらず，通常のオペレーティングシステムでは使用者が起動，停止を行なわなければならない．

本研究の目的は，Bluetooth 機能の起動，停止を自動的に行なうためのソフトウェアアーキテクチャを提案し，その妥当性について考察することである．Bluetooth 機能を必要時のみ起動することでスマートデバイスの省電力化を図り，機能を自動的に停止させることで Bluetooth 機能の動作中に生じる情報漏えいしうる機会を最小化する．ソフトウェアアーキテクチャを提案することで，ソフトウェアを設計するさいに拡張性と保守性の確保する．

スマートデバイスは移動体であり外部環境は変化するので，その変化に応じて Bluetooth 機能を制御する必要がある．設計するソフトウェアの構造は，変化する状況に応じた振舞いが可能なコンテキスト指向により実現する．Bluetooth に関する記述は複数のコンポーネントを横断しているので，多相型によるコンポーネントの入れ替えではなく，構造の入れ替えにより実現する．本研究では，江坂らが提案する PBR(Policy-Based-Reconfiguration) パターンを適用する．これにより，制御の実行時に起こる環境の変化を検知し，構造や振舞いを自身で変更することが可能になる．設計したアーキテクチャに基づいた簡単なプログラムを作成し，考察として関連研究と比較を行ない，アーキテクチャの妥当性を確認する．

2　背景技術

2.1　PBR パターン [1]

江坂らは自己適応のためのアーキテクチャパターンとして PBR(Policy-Based-Reconfiguration) パターンを定義している．ポリシーと再構成の仕組みをそれぞれコンポーネント化し，動的再構成を実現する．PBR パターンの静的構造と動的振舞いを図 1，図 2 に示す．

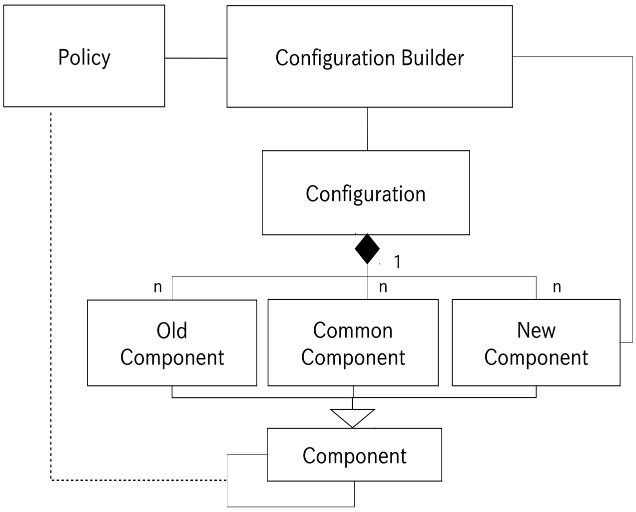

図 1　静的構造

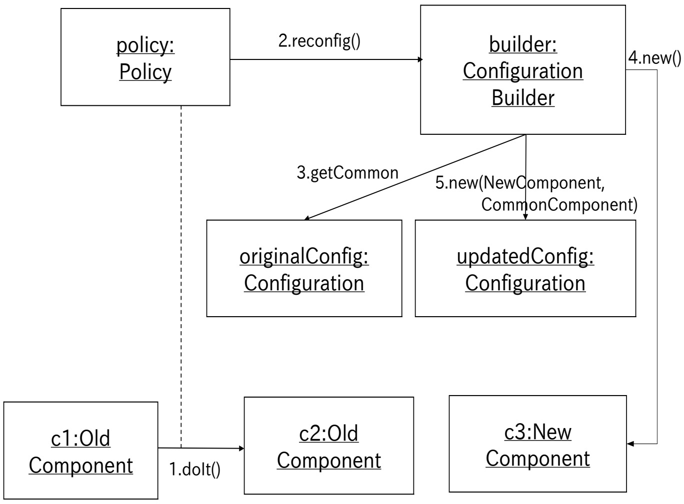

図 2　動的振舞い

- Policy：再構成の振舞いを定義する．Policy の振舞いとして，コンテキストの更新メッセージを横取りし Configuration Builder に再構成命令を送る．再構成する条件を満たすか評価し，Configuration の構成を

決定する．再構成の内容を独立して定義することで保守性を保証する．

- Configuration Builder：Component のインスタンス生成を行なう．Configuration Builder の振舞いとして，New Component を生成し，Updated Configuration を構築する．Policy に従って Configuration を再構成する．
- Original Configuration：再構成前の Configuration. Old Component と Common Component から構成される．
- Updated Configuration：再構成後の Configuration. Configuration Builder によって構築され，New Component と Common Component から構成される．

Component は次の３つからなる．

- Old Component：再構成前の Component.
- Common Component：再構成前と再構成後の共通の Component.
- New Component：再構成後の Componet.

Policy がコンポーネント間のメッセージを横取りし，Configuration Builder を起動する．Configuration Builder は Policy に従い New Component を生成することで Cofiguration を再構成する．

2.2 Bluetooth 通信 [5]

Bluetooth とはデジタル機器用の近距離無線通信規格の１つである．現在の Bluetooth は BLE（Bluetooth Low Energy）と呼ばれ，目的ごとに通信プロトコルやその使い方を定めたプロファイルが定義されている．Bluetooth で通信を行うさいは，用途に応じて必要な機能を提供できるプロファイルという規格を選択して通信を行っている．

2.3 GAP（Generic Access Profile）

デバイスの探索･接続の管理を行うプロファイルを GAP（Generic Access Profile）という．Bluetooth 通信を行うにはこのプロファイルが必要不可欠である．しかし，その性質上 Bluetooth 通信における脆弱性の原因となっている．

GAP は Broadcaste と Conection という２つの通信方法を定義している．すべての Bluetooth デバイスにはこれらの通信方法が実装されている．Broadcaste とは，データを発信するデバイス（Broadcaster）がデータを受信する不特定多数のデバイス（Observer）に対し，一方的にデータを送信するための通信方法である．Conection とは，通信のホストとなるデバイス（Central）と Central によって定められたタイミングでデータの送受信を行うデバイス（Peripheral）の間で，相互に送受信を行うための通信方法である．

3 アーキテクチャ設計

3.1 設計指針

我々は以下の２つを考慮し，コンテキスト指向アーキテクチャとして設計する．

- 省電力
- Bluetooth 通信における脆弱性の回避

スマートデバイスは限られた容量のバッテリで稼働時間を確保する必要がある．使用者のスマートデバイスの長時間利用を可能にするために，バッテリ残量が多い場合だけ Bluetooth 機能を自動的に起動させ，バッテリ残量が少ない場合は Bluetooth 機能を自動的に停止させることで省電力化を図る．

我々は使用者がスマートデバイス内のアプリケーションを動作させなければ Bluetooth 機能は不必要であると考えるので，Bluetooth 機能の起動および停止を行なうきっかけをアプリケーションの起動時と停止時とする．使用者の必要時のみ Bluetooth 機能を動作させるように Bluetooth 機能の起動と停止を制御する事で，情報漏えいしうる機会の最小化を図る．

PBR パターンを適用し，Bluetooth 機能の自動的な起動，停止に関する処理を独立して定義することで，Bluetooth 機能の制御の変更が容易になる. これにより，Bluetooth 機能の保守性を確保する．

Bluetooth 機能の起動および停止を行なうきっかけは異なり，Bluetooth に関する記述は複数のコンポーネントに横断する．我々は，起動と停止が求められる状況を整理し，それぞれをコンテキストとして定義する．多相型ではなく Bluetooth 機器を含む構成と Bluetooth 機器を含まない構成の２つを入れ替えることで複数のコンポーネントにおける Bluetooth に関する記述をまとめて変更する．これにより，Bluetooth 機能の起動，停止を行なう．

3.2 Bluetooth 制御のためのコンテキスト指向アーキテクチャ

本アーキテクチャではアプリケーション起動時のコンテキストをバッテリ残量とする．バッテリ残量に応じた Bluetooth 機能の起動，停止が可能となる．また，アプリケーション停止時のコンテキストを動作中のアプリケーション一覧とする．使用者の意思に応じた Bluetooth 機能の停止を可能とする．

バッテリ残量に応じた Bluetooth 機能の起動，停止に関する振舞いと，動作中のアプリケーション一覧に応じた Bluetooth 機能の起動，停止に関する振舞いを Bluetooth 制御ポリシーとして定義する．

アプリケーションを起動させるさい，バッテリ残量を評価し，バッテリ残量が基準値を上回っている場合は，スマートデバイスコンポーネントを再構成し，Bluetooth 機能を起動させる．バッテリ残量が基準値を下回っている場合は，Bluetooth 機能を停止させ，スマートデバイスコン

ポーネントを再構成する．

アプリケーションを停止させるさい，動作中のアプリケーション数を評価し，その数が 0 となった場合，Bluetooth を停止させ，スマートデバイスコンポーネントを再構成する．コンテキストとポリシーの定義に基づき，PBR パターンを用いて設計を行なった．図 3 に本研究の静的構造を示す．

- Bluetooth 非対応スマートデバイス：Bluetooth センサを持たない状態のスマートデバイス
- Bluetooth 対応スマートデバイス：Bluetooth センサも持った状態のスマートデバイス
- Input Device：使用者が操作するデバイス
- アプリケーション：Music アプリケーションなど
- バッテリ：スマートデバイスの電池残量
- Bluetooth センサ：デジタル機器間で Bluetooth 通信を行なうためのセンサ
- Bluetoot アクチュエータ：Bluetooth 通信を行なうデジタル機器
- コンテキスト：アプリケーション起動時はバッテリ残量，停止時は動作中のアプリケーション一覧
- Bluetooth 機能アクティベータ：Bluetooth 非対応デバイスと Bluetooth 対応デバイスを入れ替え，Bluetooth センサにメッセージを送る．
- Bluetooth 制御ポリシー：コンテキストに基づいて Bluetooth の起動と停止の制御に関する振舞いを定義する．使用者の操作によるメッセージを横取りし，Bluetooth 機能アクティベータに再構成命令を送る．

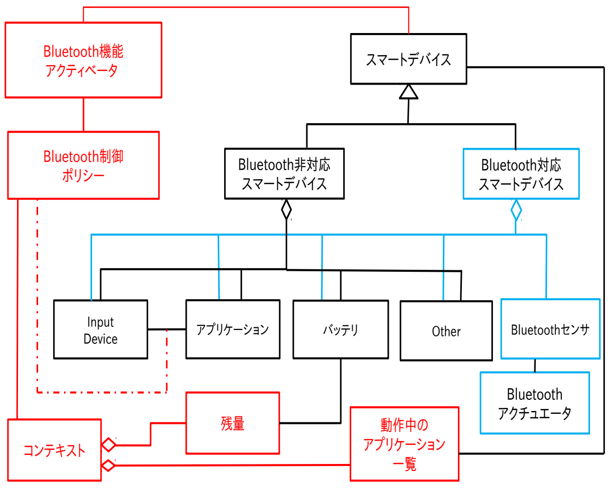

図 3　静的構造

使用者がアプリケーションを起動させるさいの振舞いとして，そのメッセージを Bluetooth 制御ポリシーが横取りする．Bluetooth 制御ポリシーはバッテリ残量に応じて Bluetooth 機能アクティベータに命令を送る．Bluetooth 機能アクティベータが Bluetooth 非対応スマートデバイスと Bluetooth 対応スマートデバイスを入れ替え，Bluetooth センサにメッセージを送る．これにより，Bluetooth は動的に起動，停止する．図 4 にアプリケーション起動時の動的振舞いを示す．

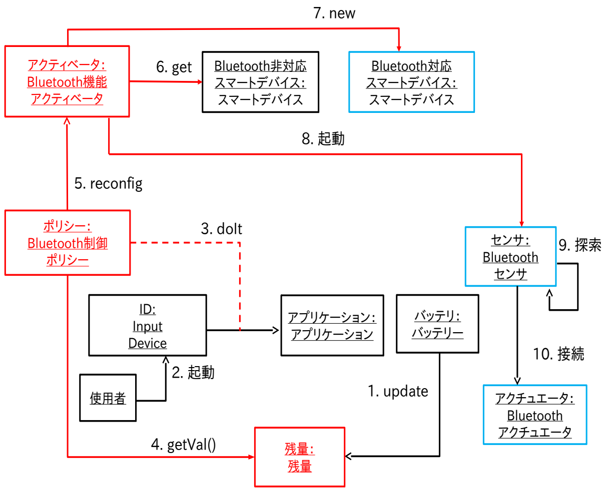

図 4　アプリケーション起動時の動的振舞い

使用者がアプリケーションを停止させるさいの振舞いとして，そのメッセージを Bluetooth 制御ポリシーが横取りする．Bluetooth 制御ポリシーは動作中のアプリケーション数が 0 になったさいに，Bluetooth 機能アクティベータに命令を送る．Bluetooth 機能アクティベータが Bluetooth 対応スマートデバイスと Bluetooth 非対応スマートデバイスを入れ替え，Bluetooth センサに停止メッセージを送る．これにより，Bluetooth は動的に停止する．図 5 にアプリケーション停止時の動的振舞いを示す．

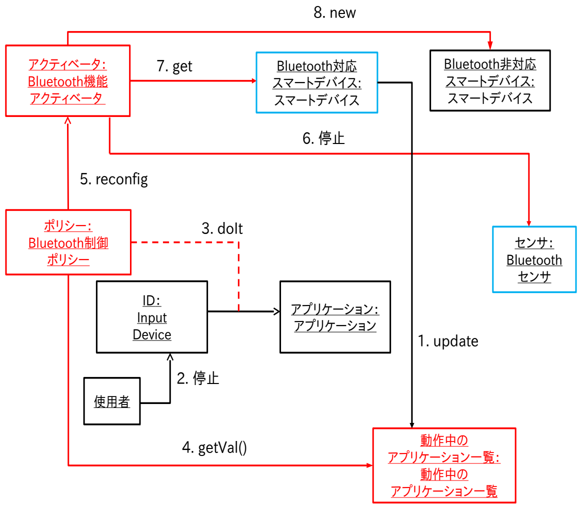

図 5　アプリケーション停止時の動的振舞い

4 考察

武田ら [3]，村尾ら [4] の研究は，センサの切り替えという点で本研究と共通しているので，これらの研究と比較する．

4.1 関連研究

武田らは，GPS と加速度センサから得られる情報を元にスマートデバイスが GPS 測位を自動的に行なう省電力手法を提案している．この研究は使用者の状態によって GPS 測位を行なうか判断しており，使用者の状態を知るために加速度センサを用いている．加速度センサから得られる情報によって GPS センサを動的に制御している．

村尾らは，使用者の状態ごとに 5 つのセンサの組合わせを変更するコンテキストアウェアシステムを提案している．センサの組合わせによって必要なセンサのみを切り替える必要がある．センサの切り替えには村尾らの開発した CLAD というセンサ管理デバイスを用いる．

4.2 本研究の位置付け

我々が提案したアーキテクチャを用いれば，ポリシーの変更が容易なので上述の関連研究の方式を実現することが可能である．我々はコンテキストの変化に応じてセンサを起動，停止させる振舞いをポリシーに定義した．加速度センサの変化に応じて GPS センサを自動的に起動，停止させる振舞いをポリシーに定義することで武田らの方式を実現できる．したがって，我々が提案したアーキテクチャは，センサを自動的に起動，停止させる制御システムのアーキテクチャの一般形として位置付けられる．また, 村尾らは CLAD によってセンサを起動，停止させているのに対し，本研究ではセンサの起動，停止処理をソフトウェア内に独立して定義した．これにより，機能を開発，運用するためのコストが安価となり，また，用いるセンサなどの変更が容易であるという利点が得られる．

我々の提案したアーキテクチャに基づくことで，プログラムの構成が簡便となる．アーキテクチャを設計せずに実装を行なう場合は，プログラムが複雑化し変更が難しくなる可能性があると考える．

4.3 本研究の応用

使用者は，通信する Bluetooth 対応機器と利用するアプリケーションによって，Bluetooth 機能が必要かどうかが決まる．例として，音声を出力しないメモアプリケーションを利用するさいは，使用者の周辺にあるワイヤレスイヤホンと通信接続する必要はない．

本研究では，Bluetooth 機能を制御するさい，バッテリ残量と動作中のアプリケーション一覧をコンテキストとした．応用として，そのような物理的なデバイスではなく，使用者の利用状況などの抽象的な状態をコンテキストとすることで様々なハードウェアに対応できる．そのコンテキストに基づいた判断を行なうポリシーを設計することで，より使用者の意思に応じた機能の制御が可能になると考える．

5 おわりに

本研究では，Bluetooth 機能を利用するさいのスマートデバイスの省電力化とセキュリティ面の脆弱性の回避を課題とし，Bluetooth 機能の自動制御のためのソフトウェアアーキテクチャを設計した．省電力化の方法として，使用者が Bluetooth 機能を必要としない場合に機能を停止させる．Bluetooth の脆弱性の回避に関して，使用者が Bluetooth 通信を用いて周辺機器と接続する場合のみ起動させることで，スマートデバイス情報が漏えいしうる機会を最小化した．本研究では，Bluetooth 機能の動的制御においてポリシーの変更が容易なアーキテクチャを定義した．これにより，上述の関連研究の方式を実現できるので，本研究で提案したアーキテクチャの妥当性が確認できる．今後の課題として，Bluetooth 通信を行なうさいの信号強度にも対応する必要があると考える．Bluetooth の電波信号は電子レンジなど，他の機器が発する電波によって干渉を受ける場合がある．そのさいは，安定した通信接続が行われていないにも関わらず，スマートデバイスの電力を消費してしまう．Bluetooth 通信の信号強度が弱い場合は，Bluetooth 機能を停止させることで，さらに省電力化できると考える．また，本研究では非機能要求として，PBR パターンを適用することで拡張性と保守性を考慮したアーキテクチャを設計した．Bluetooth 通信を行なっている最中でもバッテリ残量の低下によってに通信を切断することがあり，Bluetooth 機能に求められる非機能特性を考慮し，再構成する方法を洗練する必要があると考える．

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(2018), pp. 3-15.

[2] 折尾彰吾, 上田浩, 上原哲太郎, 津田侑: ワイヤレスデバイスのもたらすロケーションプライバシー問題に関する一考察, コンピュータセキュリティシンポジウム 2012 論文集, Vol. 2012, No. 3(2012), pp. 262-269.

[3] 武田恭典, 安積卓也, 西尾信彦: 端末ローカル情報のみで実現する GPS センシング省電力機構, マルチメディア, 分散, 協調とモバイル (DICOMO2011) シンポジウム, 2010, pp. 1467-1475.

[4] 村尾和哉, 寺田努, 竹川佳成, 西尾章治郎: ウェアラブルコンピューティングのための消費電力を考慮したコンテキストアウェアネスシステムの構築, 情報処理学会論文誌, Vol. 50, No. 5(2009), pp. 1456-1466.

[5] Bluetooth SIG: 従来型プロファイル, https://www.bluetooth.com/ja-jp/specications/proles-overview, 2018.