

携帯電話における設定システムのアスペクト指向実現

2001MT024 平位 明日香 2001MT106 辻 香織
指導教員 野呂 昌満

1 はじめに

組込みソフトウェアは製品の開発サイクルが短いので、変更に対応できる構造にする必要がある。一方で、オブジェクト指向開発では、一般に、変更における柔軟性が向上すると言われている。しかし、オブジェクト指向開発では、解決できないコンサーン横断問題（横断的に関連する処理）[4] が確認されている。オブジェクト指向を組込みソフトウェアに適用しても、同様の問題があると予想される。コンサーン横断問題は、POP 技術により解決が試みられている。

我々は、組込みソフトウェアのひとつである携帯電話ソフトウェアをオブジェクト指向開発し、横断的処理の存在を確認した。オブジェクト指向開発では、コンサーン横断問題が残るので、変更における柔軟性が低下する。これらの横断的処理をモジュール化するためには、一般的な POP 技術のひとつであるアスペクト指向の適用を考えた。

本研究では、アスペクト指向技術が組込みソフトウェアの柔軟性に寄与するかを考察する。組込みソフトウェアにおけるオブジェクト指向によるコンサーン横断問題を、アスペクト指向により解決することを目的とする。

携帯電話ソフトウェアは複数のハードウェアを並行に処理するので、典型的な組込みソフトウェアだと考えられる。上記の目的を達成するために、携帯電話ソフトウェアの設定システムを例にとり、開発をおこなう。

研究方法として、オブジェクト指向で実現した設定システムの構造を観察し、横断的な処理をアスペクト指向により分離する。ソフトウェアの追加、変更を行った例を用いて、柔軟性が向上したことを示す。

平位は主に設定に関する状態遷移機構を、辻は主にデータベースシステムを担当した。

2 オブジェクト指向開発

携帯電話における設定システムをオブジェクト指向により実現し、横断的に関連した処理を確認した。設計、実現する際に、Java でオブジェクト指向実現した。実現した設定システムを図 1 に示す。

2.1 設定メニューリスト

オブジェクト指向により実現した携帯電話の設定メニューリストについて述べる。本研究では、設定メニューリストを設定画面の集まりと定義する。設定メニューリストは、構造が階層構造である。メニューをたどる処理

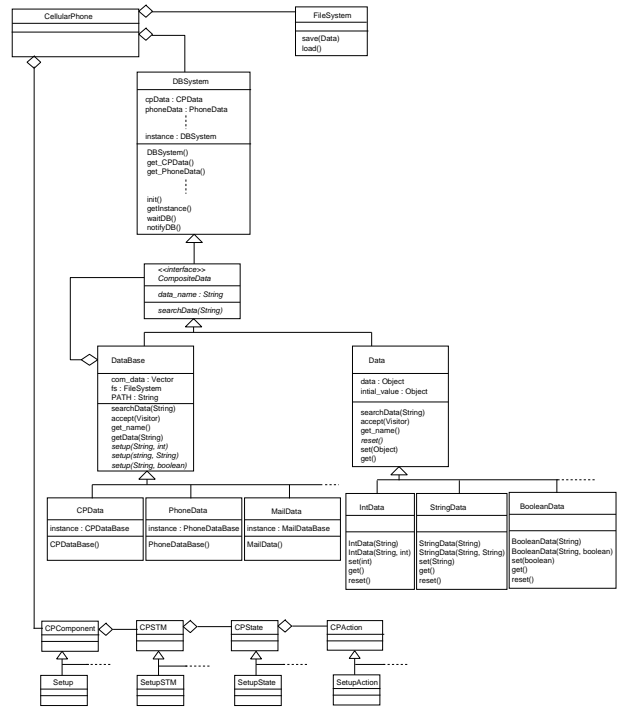


図 1 設定システムのクラス図

が状態の遷移と考えることができるので、状態遷移機構で表現できる。設定メニューリストは、State パターン [2] を用いて実現した。単一の状態遷移機構で設定システムを実現した場合、構造が複雑になる。そこで、個々のサブシステム毎に状態遷移機構をもたせる構造にした。作成した設定システムの状態遷移図を図 2, 3 に示す。decideState へのアクションで、データベースに値を設定する。これ以外のアクションはメニューの遷移である。

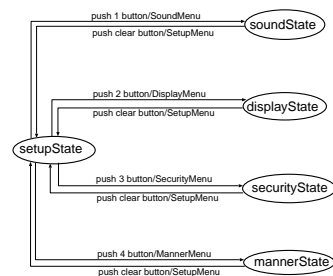


図 2 設定に関する状態遷移図

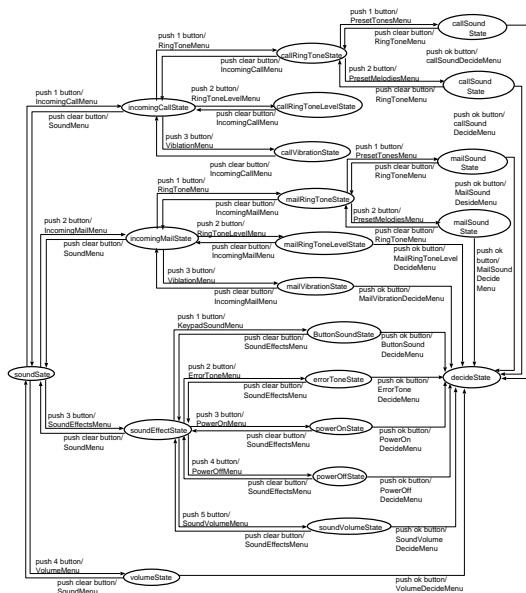


図 3 音関連の設定に関する状態遷移図

ことを目的とした設計である．待ち受け画面を設定する処理を図 4 に示す．

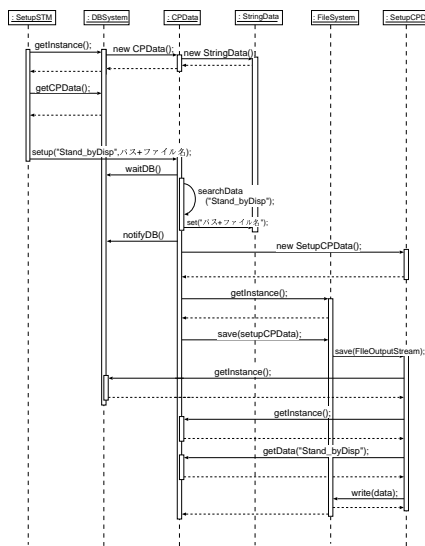


図 4 待ち受け画面設定に関する処理のシーケンス図

2.2 データベースシステム

オブジェクト指向により実現したデータベースシステムについて述べる．アプリケーションからデータ構造を垣根なくアクセスできるという理由により，オブジェクト指向データベース（以下，OODB）[3]としてデータベースシステムを実現した．本研究では，設定，およびリセット，読み出しに関する処理について実現した．実現する際に，以下のパターンを用いた．

- Composite パターン
データを木構造で管理
- Singleton パターン
データベースのインスタンスが複数存在すると，データの整合性がとれなくなるので，インスタンスを 1 つだけつくることを保証
- Visitor パターン
木構造で実現したデータベースを操作するために使用

設定用サブシステムは階層構造をもつ．保守性の向上を目指し，この階層構造とデータベースの構造を対応一致させた．

2.3 設定システム

設定システムは，設定メニューリストを実現した状態遷移機械とデータベースシステムによって構成される．設定の処理は，状態遷移機械のアクションからデータベースシステムにアクセスする．データ永続性の実現は，データベースシステム内にファイルシステムを保持し，ファイルシステムが補助記憶処理を行う構造とした．これは，アプリケーションインタフェースとデータ永続性処理を分離することで，システムの保守性向上させる

3 アスペクト指向開発

複数のオブジェクトに横断的に関連する処理を，アスペクト指向により分離する．オブジェクト指向で実現した設定システムから横断的コンサーンを抽出する．設定システムにおける横断的コンサーンを 3.1 節に示す．

3.1 コンサーンの抽出

オブジェクト指向で実現した設定システムから 4 つの横断的コンサーンを抽出した．

- カスタマイズコンサーン
- 排他制御コンサーン
- インスタンス生成コンサーン
- データ永続性コンサーン

カスタマイズコンサーン

データを設定する処理は，設定システムがもつ状態遷移機械と横断的に関連している（図 5 参照）．データを設定する処理では，データベースシステムにある設定項目を更新しなければいけない．設定項目を設定，およびリセットする処理が，設定システムの状態遷移機械と横断的に関連している．

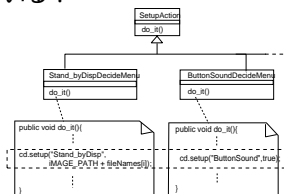


図 5 設定に関する処理

排他制御コンサーン

排他制御に関する処理は，データベースが保持したい値を読み書きする処理と横断的に関連している（図6参照）．設定項目を設定，およびリセット，読み込む際の排他制御に関する処理が，データベースシステムと横断的に関連している．本来データベースでは，排他制御に関する処理は必然的に備わるが，分離することにより排他制御関連モジュールの再利用性が向上すると考えた．

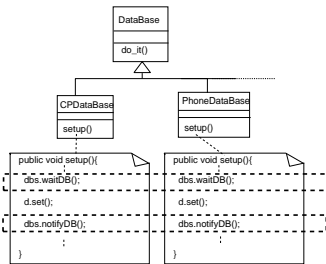


図6 排他制御に関する処理

インスタンス生成コンサーン

インスタンス生成に関する処理は，設定システムがもつ状態遷移機械と横断的に関連している（図7参照）．設定項目を設定，およびリセット，読み込む場合のインスタンス生成をする処理が，設定システムの状態遷移機械と横断的に関連している．インスタンス生成に関する処理が散在しているので，コンサーンとして抽出した．

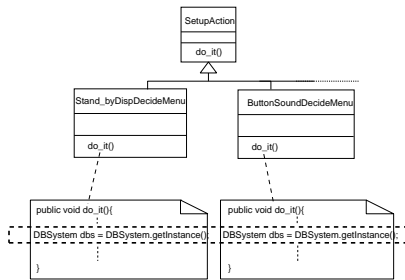


図7 インスタンス生成に関する処理

データ永続性コンサーン

データ永続性に関する処理は，データベースシステムと横断的に関連している（図8参照）．データ永続性に関する処理では，設定された項目が携帯電話の電源を切っても消えずに残るようファイルに保存する．設定項目の設定，およびリセット後のデータを保存する処理が，データベースシステムと横断的に関連している．設定項目を更新した後にファイルシステムに保存する処理が散在しているので，コンサーンとして抽出した．

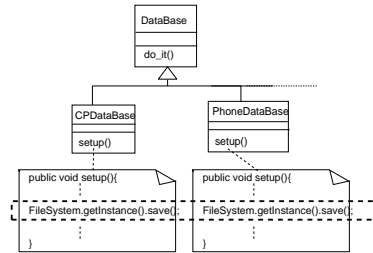


図8 データ永続性に関する処理

3.2 アスペクト指向による設計と実現

抽出したコンサーンにより，設定処理に関するアスペクトを規定し，ソフトウェアを設計，実現する．アスペクト指向により分離したデータベースアスペクト，および設定メニューリストアスペクト，データ永続性アスペクト，排他制御アスペクトを図9に示す．

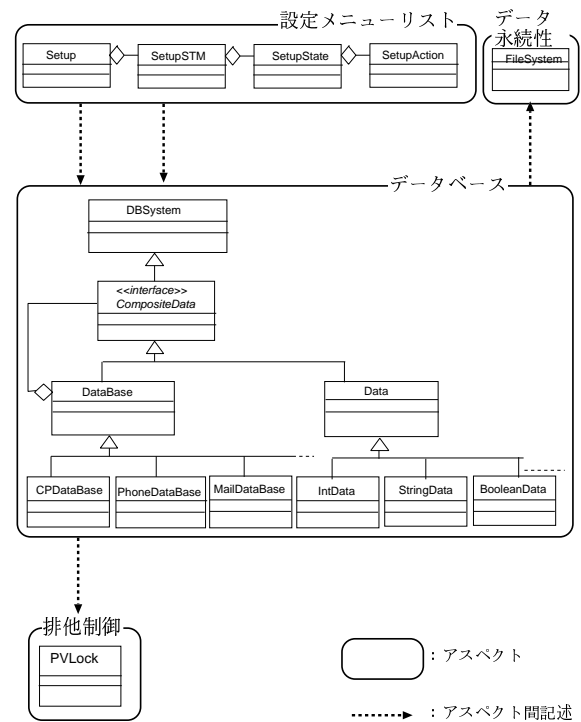


図9 アスペクト指向により設計した設定システムのクラス図

4 考察

本研究で設計した設定システムを実現し，柔軟性について考察する．本節では，オブジェクト指向，およびアスペクト指向により実現した設定システム各々を比較し，柔軟性について議論する．

4.1 データベースシステムの変更における柔軟性の考察

オブジェクト指向、およびアスペクト指向により実現した設定システムの構造を比較し、データベースシステムの変更における柔軟性について考察する。設定システムは、設定メニューリストの状態遷移機械とデータベースシステムにより構成される。オブジェクト指向開発では、設定、インスタンス生成に関する処理が散在しており、データベースシステムを分離することが困難である(図10参照)。一方、アスペクト指向開発では、散在する処理が局所化できており、変更が容易である(図11参照)。アスペクト指向により、横断的な処理をモジュール分割したことで、データベースシステムの変更における柔軟性が向上した。

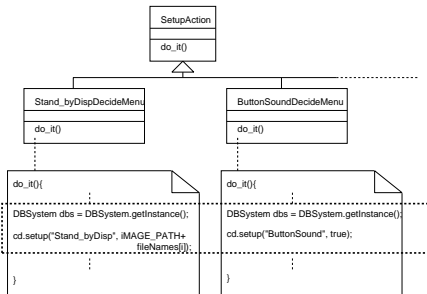


図10 オブジェクト指向コードにおけるデータベースアクセスコード

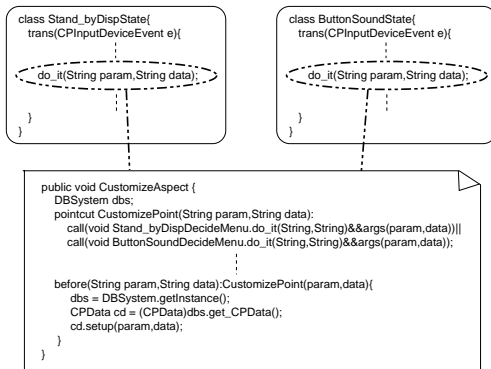


図11 カスタマイズコンサーンに関するアスペクト間記述

4.2 排他制御に関する処理における再利用性の考察

ファイルシステムでデータを保存する際に排他制御を行う例をもとに、排他制御に関する処理の再利用性について考察する。データベース内の情報更新時には、排他制御を行う必要がある。ファイルシステムで保存および呼び出す際にも排他制御を行いたい場合、オブジェクト指向では、データベースの情報更新時と同様の処理を記述する必要がある。アスペクト指向により実現すると、排

他制御に関する処理をモジュール分割できるので、保存および呼び出すメソッドをジョインポイントと指定するだけでよい(図12参照)。排他制御に関する処理をモジュールとして分離したことで、ファイルシステムで排他制御を行う際に再利用することができた。

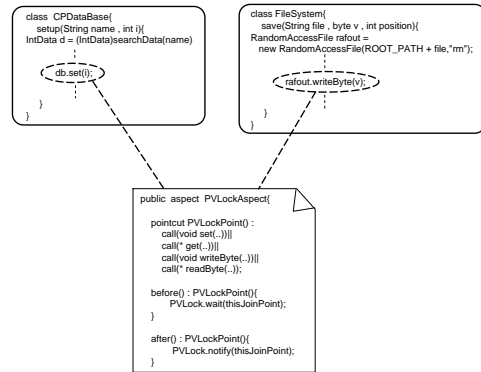


図12 ファイルシステムに排他制御の処理を追加する際のアスペクト間記述

5 おわりに

本研究では、アスペクト指向により携帯電話における設定システムを設計、実現した。ソフトウェアの追加、変更における柔軟性をオブジェクト指向、およびアスペクト指向により実現した設定システムを比較し考察した。アスペクト指向によりコンサーンを抽出し分離することで、設定システムの柔軟性が向上することを確認した。本研究で実現した携帯電話制御ソフトウェアの構造を他の組込みソフトウェアに適用し、有用性について検証することを今後の課題とする。

謝辞

本研究を進めるにあたり、二年間御指導いただいた野呂昌満教授、有益な助言をいただいた熊崎敦司先生、大学院の後藤修平さん、石見知也さん、小久保佳将さん、八木晴信さんに深く感謝いたします。

参考文献

- [1] AspectJ <http://eclipse.org.aspectj/>
- [2] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: Design Patterns Elements of Reusable Object-Oriented Software, Addison-Wesley, p. 395(1995).
- [3] Robert J.Muller: Database Design for Smarties:Using Uml for Data Modeling, (1999).
- [4] COMMUNICATIONS of the ACM, Vol.44, No.10, p.168 (2001)