

に多大な時間・能力が必要となる。この作業はソフトウェア開発には直接的な関係はなく、無駄な工程である。

プロセスを定義することにより、属人化を防ぐことができる。プロダクトが作成者に依存することがなくなるので、生産性・品質を一定に保つことができる。また、無駄な工程を省くことができ、生産性を向上させることができる。

さらに、富士電機の自動販売機制御ソフトウェア開発にドメインを限定したプロセス定義をすることで、自動販売機制御ソフトウェアの特徴から起きる開発プロセスでの問題点を考慮することができる。富士電機で問題となっている現状のプロセスを改善した新しいプロセスを定義することは、生産性・品質を向上することにつながる。

4 プロセスの記述方法

我々はプロセスを記述する方法として、UML を用いることにした。その理由は以下の3つが挙げられる。

- 図で表現する
 - 図で表現することでプロセスを記述し理解しやすい
- 様々な側面からの記述ができる
 - 複数の図を使うことでプロセスを1方向からではなく、静的な側面や動的な側面など角度を変えて表現できる
- 標準的である
 - 数ある表記法の中で、一般に広く使用されており標準的である

UML の4つの図を使って現状プロセスを記述することにした。以下の他にもユースケース図、オブジェクト図、コラボレーション図、コンポーネント図、配置図がある。しかし、開発プロセスを記述する上では不必要であると判断した。

我々は、現状プロセスを記述する上で能動的なオブジェクトと受動的なオブジェクトの2種類に分けた。

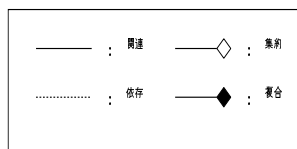
- 開発者
 - 能動的なオブジェクトであり、電制(電子制御部)や三重設計などである。
- 中間生成物
 - 受動的なオブジェクトであり、アイテム一覧や開発方針書などである

能動的なオブジェクトは、アクティビティ図で記述し、受動的なオブジェクトは、状態チャート図で記述することにした。

ここでは、UML の表記方法を簡単に説明する。

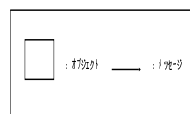
クラス図

クラス図は、システム内に存在するクラス同士の静的な構造を記述する。クラスとは、システム内に存在する「もの(オブジェクト)」を抽象化したものである。以下の図は、主な表記法である。



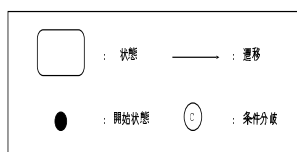
シーケンス図

シーケンス図は、相互作用を時系列に並べる。特に、相互作用に関係しているオブジェクトをライフラインによって示し、オブジェクトが交換するメッセージを時系列に並べる。ライフラインとは、オブジェクトが生成している状態を意味する線である。横軸にオブジェクトを複数並べ時間は上から下へと進んでいき、その間にどのオブジェクトがどのオブジェクトに対してメッセージを送るのかを記述する。以下の図は、主な表記法である。



ステートチャート図

ステートチャート図は、オブジェクトの内部の状態の変化を表す。状態は時間や外部的なイベントにより変化する。以下の図は、主な表記法である。



アクティビティ図

アクティビティ図では、オブジェクトのアクティビティ(活動)の流れを表す図であり、システムの動的な側面を記述する。つまり、一連の作業を完結するために必要な個々のアクションとその結果をとらえる。また、アクティビティ図は外部のイベントによらない自動的な状態遷移だけを記述した特殊な状態図ともいえる。以下の図は、主な表記法である。



以上のUMLをプロセスに当てはめて、プロセスの記述に用いることにした。

5 現状プロセスの記述

富士電機の現在の開発方法を理解し、現在の問題点を見つけるために、UMLの4つの図を使って、現状プロセスを記述した。

以下は、現在の開発プロセスをUMLで表記した図と説明である。(一部省略)

5.1 クラス図

開発者と中間生成物との関係は、関連がある。中間生成物同士の関係は、集約や依存の関係がある。図2の中で、電制とアイテム一覧との関係は、makeであり、電制がアイテム一覧を作ることの意味している。また三重設計とアイテム一覧の関係は、三重設計がアイテム一覧を使うという意味である。また、アイテム一覧と開発方針書との関係は、依存を表している。開発方針書の変更にもない仕様書の変更も必要であることを意味している。

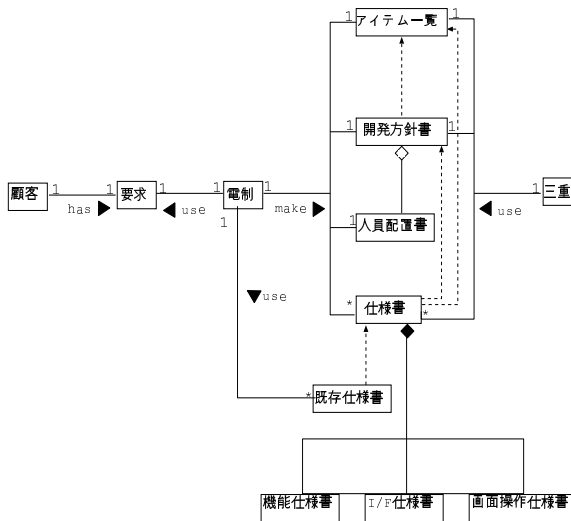


図2: クラス図

5.2 シーケンス図

シーケンス図はメッセージの実行順序やオブジェクト間の協調関係を表現することで、プロセス全体の流れを表している。

図3では、まず顧客から電制に要求というメッセージが送られる。電制からアイテム一覧作成というメッセージが送られ、アイテム一覧というオブジェクトが生成される。生成されたオブジェクト(アイテム一覧)は仕様書というオブジェクトを作成する際に参照される。仕様書の作成が終了すると電制から三重設計に作業が移る。

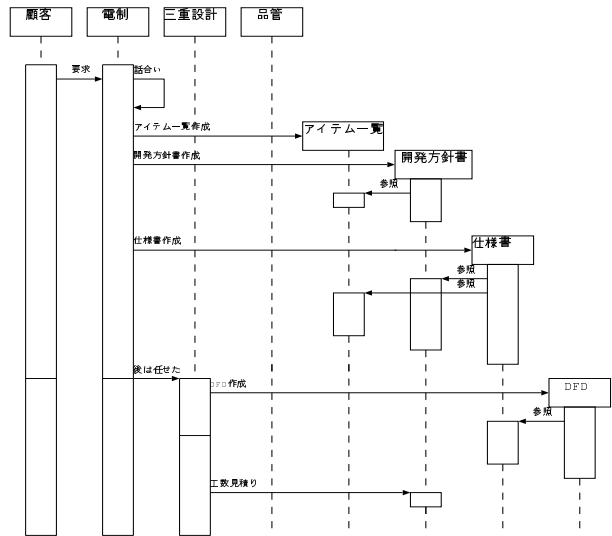


図3: シーケンス図

5.3 ステートチャート図

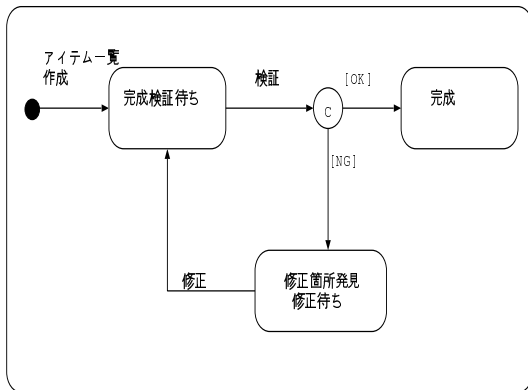
ステートチャート図では、中間生成物の振舞いを表している。つまり、オブジェクトを中間生成物として、一つのオブジェクトに視点を置き、オブジェクトの内部の状態を記述している。

図4では、アイテム一覧というオブジェクトに対し記述している。アイテム一覧作成というイベントで検証待ち状態になり検証の結果、アイテム一覧完成状態となる。アイテム一覧の他にも中間生成物の状態は、全てステートチャート図で記述する

5.4 アクティビティ図

アクティビティ図では、開発者の振舞いを表している。つまり、開発者が中間生成物を作る上でのアクションを記述している。

図5は、電制がアイテム一覧を作成する際のアクティビティ図である。要求というオブジェクトによりアイテム考案というアクションをして、アイテム一覧記述というアク



アイテム一覧

図 4: ステートチャート図

ションをする。結果、アイテム一覧というオブジェクトが生成される。他の中間生成物も同様に、開発者の側面から記述することができる。

6 今後の課題

これまでに、現状のプロセスを記述した。しかし、一般的なソフトウェア開発プロセスのレベルなので、問題点を見つけることができていない。細かく現状のプロセスの記述をすることで問題点を見つける。その後、ARTICを取り入れたプロセスを明らかにすることで現状とARTICとの比較・検討を行ない、プロセスを定義する。

参考文献

- [1] 大森 章司, 竹田 麻奈美, 寺田 享平: 自動販売機制御ソフトウェアの構成法に関する研究, 南山大学経営学部情報管理学科卒業論文, 1998
- [2] 富士電機株式会社: ARTiC(Abstraction for Real Time Control)
- [3] 富士電機株式会社: ARTiC リアルタイム制御における抽象化の手引き
- [4] 富士電機株式会社: ARTIC による分析・設計・実装方法 初版, 1998
- [5] Hans-Eriksson, Magnus Penker: UML ガイドブック, 株式会社トッパン, 1998

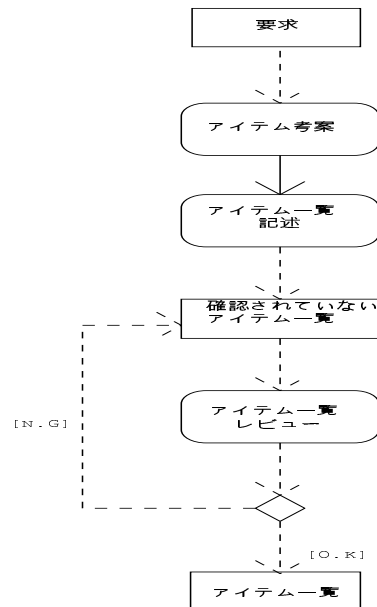


図 5: アクティビティ図