ZIPC の UML によるモデル化について

日本電気（株）NECエレクトロンデバイス システムＬＳＩ事業本部 システムＬＳＩ設計技術本部 設計システム部
川口 晃

1. はじめに

近年、マイコン組込みシステムの高機能、短納期化が進んでおり、その上に組み込まれるソフトウェアにも同様なことが言える。これに応じて、組込みソフトウェアの開発ツールも高機能、複雑化している。そこで、本稿では組込みシステム開発に最適なツール構築のため、ツール自身をモデル化し要求仕様の定式化を試みた例を紹介する。対象ツールとして状態遷移設計ツール ZIPC を、モデル化の記法として UML(Unified Modeling Language)を使用した。

2. ZIPC のモデル化

ZIPC のモデル化は、次の 2 段階について行なった。まず、ツール自身の内部の振る舞いをモデル化すること。これは、ZIPC を実現する立場で、ZIPC というツールが内部でどのような処理をどのようなデータ構造を用いて行なっているのかを、明確に定式化する目的で行なった。次に、その内部のモデルを参照しながら、ZIPC ユーザがツールを使用する際、ユーザと ZIPC がどのように振る舞うのかをモデル化した。

2.1 ZIPC の内部モデル

■ 外部インタフェースと内部の振る舞いを規定

最初に、アクタとユースケースを用いて ZIPC の外部インタフェースと内部の振る舞いを規定した。アクタとは、ヤコブソンの OOSE(参考文献)でシステム外部にあってシステムに刺激を与えるものと定義されている。また、アクタの刺激を受けてシステムが示す振る舞いをユースケースという単位で扱う。

まず ZIPC から見て、アクタは何か、またそのアクタからの刺激を受け、ZIPC が行なう振る舞い、すなわちユースケースは何かを定式化

◆アクタと状態遷移表◆

アクタと状態遷移表の関係

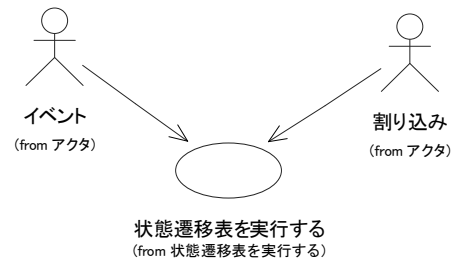

図2-1　アクタの定義

した。図 2-1 に、UML で表記したアクタとユースケースの定義を示す。

UML では、人の形をしたオブジェクトがアクタを示し、ユースケースは、楕円形で表わされる。ここでは、ZIPC の振る舞いを［状態遷移表を実行する］というユースケースで代表させ、そのユースケースに関係するアクタとして、［イベント］と［割込み］という 2 種類のアクタを定義している。これまで、このような情報は［暗黙の了解］、［規定の事実］、［個人の認識］など、あいまいに扱われる傾向があった。このような情報を、モデルとして可視化、定式化することで、モデルがどのような立場、視点で作成されているかを明確にすることができる。また、ツール開発に携わる人々（仕様決定者、開発者など）の開発対象の理解度の向上、開発者間で明示的な情報の共有を図ることができる。

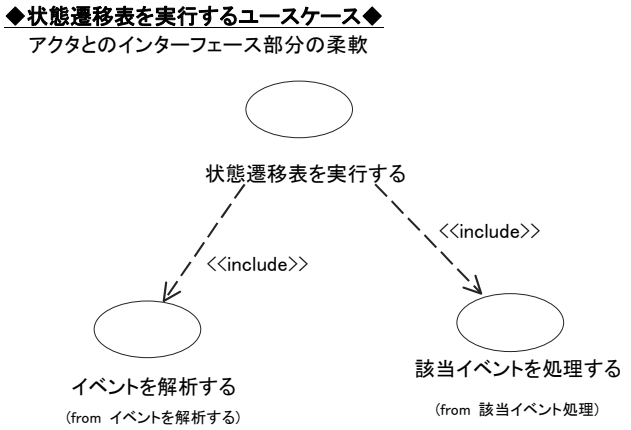

図2-2 ［ 状態遷移表を実行する ］ユースケースの分析

■ ZIPC の振る舞いの分析

次に、ZIPC の振る舞いを代表する［状態遷移表を実行する］というユースケースが、どのような振る舞いの集まりなのかを規定するため、ユースケース図を作成した(図 2-2)。

ユースケース図は、ユースケースの間にどのような関係があるかを示す図である。

この図では、ユースケース間の Include 関係が示されている。これは、ユースケース［状態遷移表を実行する］が、ユースケース［イベントを解析する］、［該当イベントを処理する］を含んでいることを示している。［イベントを解析する］、［該当イベントを処理する］の 2 種類のユースケースがモデル化されたのは、［イベント］アクタとのインタフェースを、［イベントを解析する］ユースケース内部に局所化し、［イベント］アクタと ZIPC とのインタフェースの柔軟性を、より少ないコストで確保するためである。

■ ZIPC の動的な振る舞いを規定

このように、ZIPC が内部でどのような振る

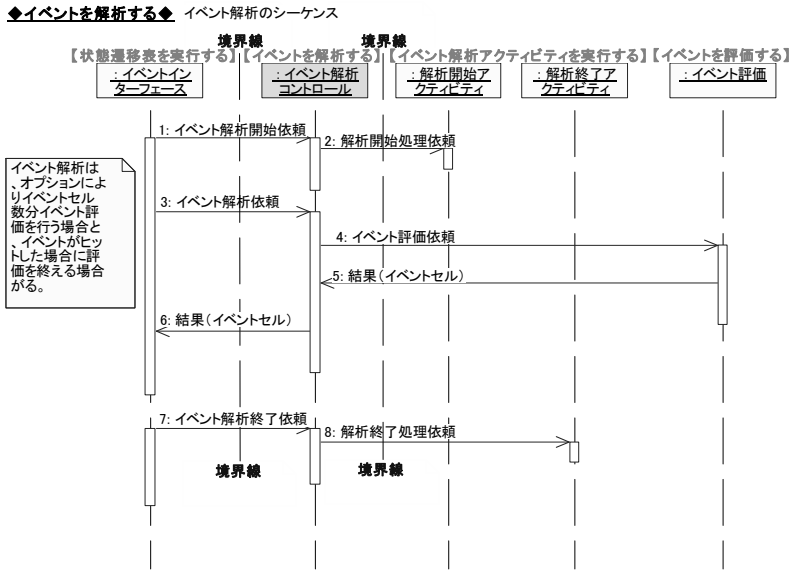

図2-3 イベントを解析する メッセージシーケンスチャート

舞いをするのかをユースケース単位でモデル化していく。この過程で、ユースケースの動的な振る舞いを規定することが必要になる。このため、メッセージシーケンスチャートを記述した。これは、ユースケースの振る舞いを、関係するクラス間のメッセージのやりとりで記述した図で、ユースケースの動的な振る舞いを詳細に表現できる。

■ ZIPC 内部のデータ構造のモデル化

次に、ZIPC 内部に存在するデータ構造をクラス図を用いてモデル化した。図 2-4 で示すのは、状態遷移表を管理するために必要なデータ構造を規定したクラス図である。クラスは四角形で表わされており、クラス間を結ぶ、一端がひし形になっている直線は集約を表わしている。集約は、part-of の関係を示す。このようにして、ZIPC 内部に存在するデータ構造を、クラス図として明示的に規定した。

2.2 内部モデルの効果

このように、アクタ、ユースケース、クラス、さらに、それらの間の関係を示す図（ユースケース図、メッセージシーケンスチャート、クラス図など）を用いて、ZIPC の内部的な振る舞いをモデル上で規定した。これらのモデルは、次の用途に用いることができる。

□設計の効率化

ユースケース間の関係を精密に分析し、再利用できるもの、拡張して使用できるものを抽出したり、クラス構造を見直し、より効率的なデータ構造を構築するなど、モデルを用いてツールの効率的な実現を行なうことができる。

□情報共有

理解しやすいモデルの形でツールの振る舞いを規定しているので、ツールの仕様決定者、開発者間で、齟齬の少ない情報共有を図ることができる。

□仕様書の作成

モデルは、ZIPC の内部構造を網羅的に規定している。モデルを構成するユースケースや、クラスなどの詳細な記述を追加することで、ZIPC の内部仕様書として使用することができる。

2.3 ユーザ側から見た ZIPC のモデル化

前節では、ZIPC を実現する側の立場で、内部モデルの作成を行った。ここでは、ZIPC を使用するユーザの立場で、ユーザと ZIPC の振る舞いのモデル化を行う。ユーザと ZIPC の振る舞いをモデル化するには、ユーザが状態遷移設計するために行なうべき振る舞いを、ZIPC の内部モデルを参照しながら、モデル化してい

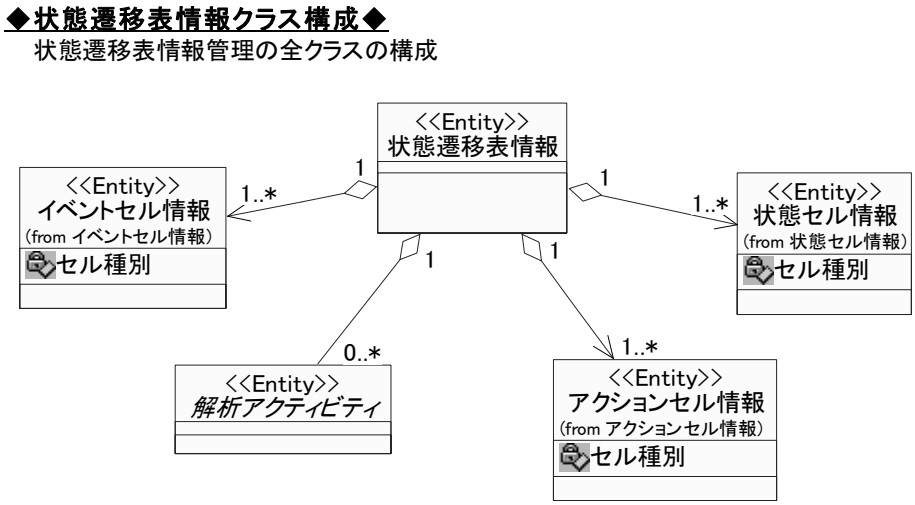

図2-4 データ構造をクラス図で表現

く。

■ ユーザと ZIPC の振る舞いを規定

ZIPC ユーザをアクタとした時の、ユーザと ZIPC の振る舞いをユースケースを用いてモデル化した(図 2-5)。2.2 で示した ZIPC を実現する側の視点と比較すると、モデルがまったく異なっている。どのような立場で、何を表現したいかを明確にすることが、有益なモデルを作成するために必要であることがわかる。

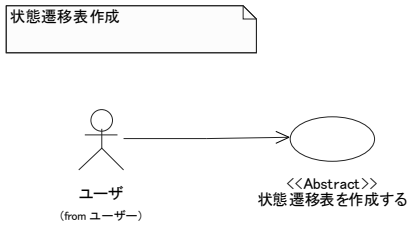

図2-5　ユーザとZIPCの振る舞いを規定

■ ユーザと ZIPC の振る舞いを分析

上述のユーザと ZIPC の振る舞いを、ユースケース図を用いて分析した。図 2-6 に一例を示す。

■ ユーザと ZIPC の動的な振る舞いを規定

次に、ユースケースのうち、動的な振る舞いを明示する必要があるものについてメッセージシーケンスチャートを作成した。図 2-7 に一例を示す。

■ ユーザが定めるべきデータ構造の規定

ZIPC を使用するにあたって、ユーザが定めるべきデータ構造をクラス図を用いて規定した。図 2-8 に一例を示す。

2.4 ユーザの視点に立ったモデル化の効果

□ツールとして不足している機能の洗い出し

ユーザ側の視点で、状態遷移表作成のプロセスをモデル化することで、ツールがサポートするべき機能の洗い出しが容易になる。

□ユーザインタフェースの改善点の洗い出し

ツールを使用するユーザのために、ツール側が、どのようなインタフェースを提供するべきかを議論するためのたたき台として使用できる。

□ユーザマニュアル

ユーザ側の視点で、状態遷移表作成のプロセスをモデル化しており、ユーザマニュアルとして使用できる。

まとめ

以上、簡単に UML を用いた ZIPC のモデル化について述べた。この試みを行なった感想として、次のようなことが言える。

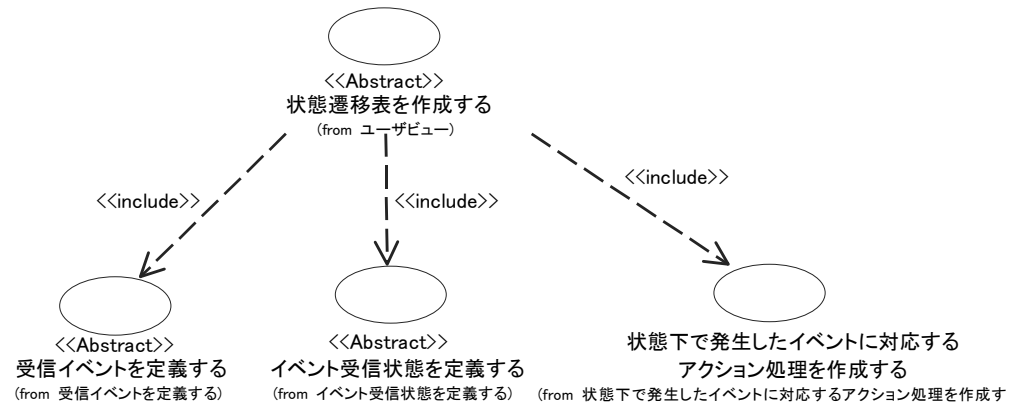

図 2-6 ユーザと ZIPC の振る舞いを分析

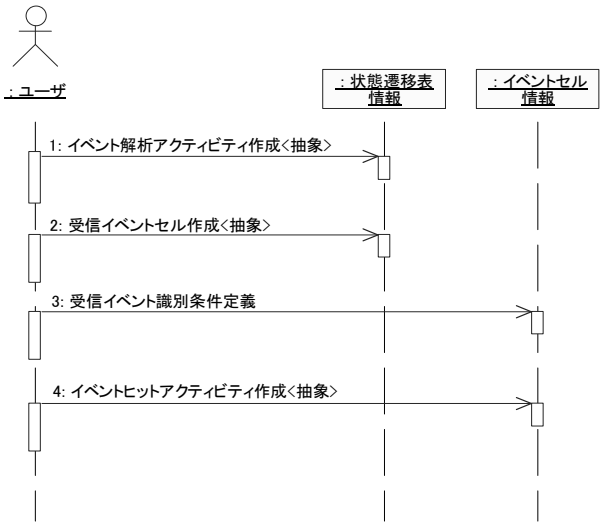

図 2-7 ユーザと ZIPC の動的な振る舞いを規定

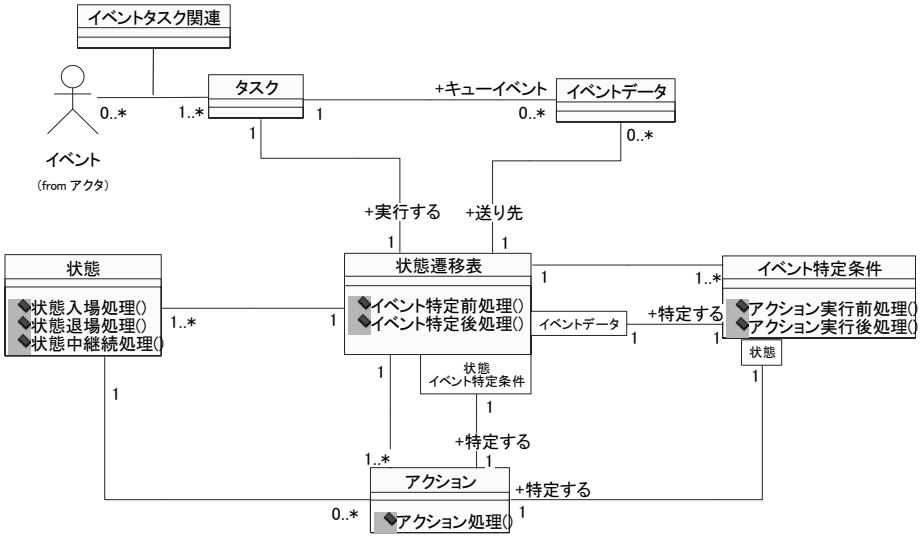

図 2-8　ユーザが定めるデータ構造の規定

□ツールの機能は非常に複雑、多岐にわたっており、全てを正確に理解し、把握することが難しい。

□UML により、ツールの機能やデータ構造、ユーザインタフェースを規定、可視化した内部モデル、ユーザ側のモデルを作成、活用すれば、過不足のない機能の作り込み、見通しの良いツール設計、使いやすいユーザインタフェースの設計が容易になる。

□このようなモデルは、機能仕様書、ユーザマニュアルなど、ドキュメントとしての価値も大きい。

最後に、今回の UML モデルの作成者である CATS の萩原氏と、このプロジェクトの機会を与えていただいた渡辺氏に感謝する。

参考文献

Jacobson,I.et al.:Object-Oriented Software Engneering, Addison-Wesley(1992)

（かわぐち あきら）