

シンセティックメディアにおけるソフトウェア部品の開発*

5M-2

平野亮太† 田中 譲†

北海道大学工学部電気工学科‡

1 はじめに

ソフトウェアの生産性と品質の向上に対する有効な手段としてソフトウェアの部品化と再利用性が考えられる。これらが実現された環境においてユーザは必要な部品を開発し、それらを既存の部品とともに組み立てることができる。本稿ではこのような環境が実現されているシンセティックメディアシステム (IntelligentPad[2]) においてその構成部品であるパッドの開発を行なうためそのモデル化を行なった。

2 IntelligentPad

IntelligentPad ではすべての部品をパッドと呼ばれる紙のイメージを持ったメディアオブジェクトとして統一し部品化を行なっている。また、ユーザの直接操作により自由にレイアウトできるとともに、貼り合わせることができパッドの機能を合成することができる。

各パッドの内部状態の値の一部は外部からアクセス (set/get) することができる。アクセス可能な要素は可観測要素、あるいはスロットと呼ばれる。パッドの貼り合わせはこれらのスロットの結合によって行なわれ、スロットを通してデータのやりとりが行なわれる。これらの内部機構はパッドが持つスロットとスロットアクセスに関するメソッドを提供することによって定義される。例えばスロット value に対して setvalue, getvalue というメソッドを必要に応じて加える。

本稿で行なう部品の開発はパッドの内部機構を定義することに重点を置いて話しを進める。

3 System Configuration

IntelligentPad ではすべてパッドによって構成されている。このため問題領域に存在するオブジェクトは1対1でパッドとして実現されたり、複数のオブジェクトを1つのパッドにグループ化して実現される。これらパッド化の基準、指針となるのは既存する部品の利用

である。ボタン、スイッチ、テキスト入力などのユーザとの Interface となる基本入出力パッドが多数用意されているため、これらのパッドの利用がアプリケーションの全体像を決める際に大きく影響する。そのため既存部品の利用を考えた全体像を考える必要がある。

さらに IntelligentPad ではパッドの貼り合わせ (スロット結合) によってパッド間のデータ通信が行なわれる。このためパッド化しようとするオブジェクトの通信に制約が与えられる。このためスロット結合によるデータの流れを考慮しなければならない。

そこでシステムの静的モデルの仕様としてオブジェクト図を取り上げ、それらのオブジェクトをパッド化したパッドの構造図を作成することを考える。図1にオブジェクト図の例を示す。

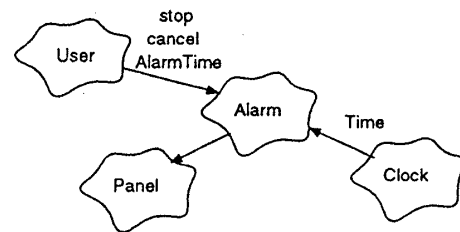


図1: AlarmClock オブジェクト図

User は Alarm に cancel, stop, AlarmTime などの値を入力する。これらはボタン、数値入力などの既存のパッドを利用できる。また時間入力には時、分、秒があるためこれらをまとめることができる。さらに Clock を Alarm 内部に取り込みグループ化することができる。このようにして概念上のオブジェクトをパッド化し、そのパッドが通信に必要な要素をスロットとする。

その結果得られたパッドの構造図を図2に示す。AlarmClock, TimeInput は新しく開発しなければならないパッドを意味し、アイコン間を結合するラベルはそのパッドに必要なスロットを意味し、各アイコン間を結合している矢印は、ラベルのスロット値をアクセスすることを意味する。つまり source 側が子パッドで destination 側が親パッドとなり、その向きからパッド間の貼り合わせ構造が木構造として得られる。

*Development of Software Components in Synthetic Media System

†Ryota HIRANO

‡Yuzuru TANAKA

§Department of Electrical Engineering, Faculty of Engineering, Hokkaido University.

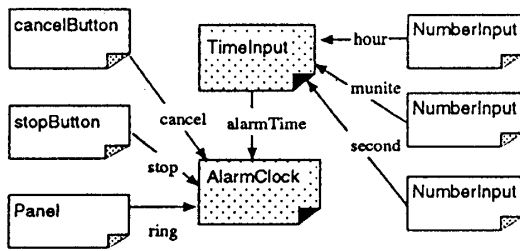


図 2: AlarmClockPad 構造図

4 Pad のモデル化

次に新規に開発するパッドを仕様化する。まずパッドの振舞いを明らかにしパッドの機能を明確にしなければならない。そのために動的モデルを構築する。

まずパッドの取り得る状態を考える。これらの状態を遷移させるイベントはスロットアクセスのみであるので取り得る状態を考えれば良い。またその結果得られる状態を構造化、可視化して記述するために D.Harel の Statecharts[1] を用い、状態を遷移させる事象をスロットアクセスに対応させるため次のような規則を与えた。ラベル付きの遷移のラベル名をスロット名 (pulse, cancel, stop, alarmTime) とし、その遷移はラベル名のスロットがアクセスされたときに遷移を開始する。また状態 ringing 内の timeofday, [finish] はその状態でしかアクセスされない属性値を意味し必要に応じスロットとして定義される。図 3 にこの結果を示す。

また状態図から各状態を識別する状態変数を定義することができる。例えば図 3 で状態 alarmON alarmOFF の対になった 2 つの状態は状態変数 ON(true/false) を用いて識別することができる。このように状態変数を導入することによってスロットアクセスが到着したときのパッドの内部状態を識別することができる。

ところで状態を遷移させる事象はスロットアクセスに対応させているため、その遷移規則を与えることによってメソッドのシナリオを与えることができる。このとき遷移規則を記述に関しては

- 現在の状態に関する制約
- 実際に行なわれる動作
- 実際に行なわれる動作後の遷移

を記述すればよい。これらは状態変数を導入したことによって容易に行なえる。例えば、図 3 の例で update という状態は Statecharts の AND-component で区切られた active にアクセスされる状態であるが、一定間隔のスロットアクセス (pulse) を受け現在の時刻 timeofday

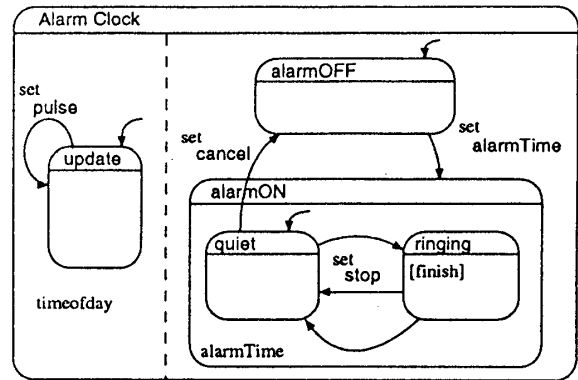


図 3: Statecharts for AlarmClock

を更新する。もう一方の状態が alarmON であればその substate である quiet ,ringing においてその動作を記述する。

5 おわりに

本稿では IntelligentPad におけるパッドの開発についてモデル化をおこなった。

IntelligentPad で部品開発を行なう特徴としては、オブジェクトの静的構造、オブジェクトの通信はすべてパッドの貼り合わせ構造の中で行なわなければならない、そのためオブジェクトのグループ化、パッド化は貼り合わせ構造を考慮し行なう必要がある、しかしパッド間の通信がスロットアクセスにより統一されているためパッドの動的振舞いの記述が独立して行なうことができる、などが挙げられる。

今後の課題としては、表記法としてオブジェクト図、パッドの構造図、Statecharts などを用いているが、各ダイアグラム間での関係がなされていない。そこでこれらをラウンドトリップに行なうことによって完成までのプロトタイプを示しながら開発を進めるといったような支援を行なうことが考えられる。

参考文献

- [1] David Harel. On visual formalisms. *Communications of the ACM*, Vol. 31, No. 5, pp. 514-531, 1990.
- [2] 長崎祥, 田中譲. シンセティック・メディアシステム:intelligentpad. *コンピュータソフトウェア*, Vol. 11, No. 1, pp. 36-48, 1994.