

# インタラクティブシステムの設計に関する考察

16SE070 佐藤 奏帆

指導教員 野呂昌満

## 1 はじめに

近年、多くのインタラクティブシステムが実現されている。インタラクティブシステムではユーザが対話形式でシステムを操作する。インタラクティブシステムの例として Web アプリケーションが挙げられる。

本研究では、手続き指向、オブジェクト指向に基づく設計、および、MVC アーキテクチャスタイルを適用したインタラクティブシステムのアーキテクチャについて比較し、考察する。例題として、アラート Web アプリケーションを用いる。手続き指向、オブジェクト指向に基づいて設計し、さらにオブジェクト指向設計に対して MVC アーキテクチャを適用して設計する。このアプリケーションに考えられる仕様変更への対応のしやすさについて比較する。

## 2 アラート Web アプリケーション

本研究では、アラート Web アプリケーションを例題として設計を行なう。このアプリケーションでは、Web ブラウザ上で図 1 に示すような画面が表示されることを想定している。画面上部にカウントダウンされる現在時間が表示されている。中部には、計測を開始するためのボタンと、停止するためのボタンが配置されている。下部には、設定時間を 5 分と 3 分で切り替えるスイッチが表示され、このスイッチを押下すると、上部の現在時間が切り替わる。

## 3 アーキテクチャ設計

手続き指向に基づいて設計したものを図 2 に示す。画面表示手続きから、設定時間変更、時間計測手続きが実行される。時間計測は、1 秒カウント手続きを設定時間カウントダウンされるまで繰り返し実行する。1



図 1: アプリ画面

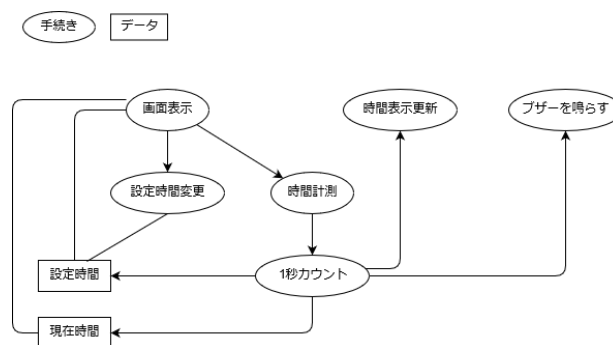


図 2: 手続き指向設計

秒カウントでは、1 秒カウントダウンした時間を時間表示更新手続きによって表示させる。設定時間カウントされた場合はブザーを鳴らす手続きによって時間切れを通知する。

オブジェクト指向に基づいて設計したものを図 3 に示す。アラートシステムは、画面、ブザー、現在時間、タイマーを持つ。画面は図 1 に示したように各種ボタン、スイッチ、時間表示パネルを持つ。タイマーは設定時間を持ち、時間切れになるまで、現在時間の更新を行なう。

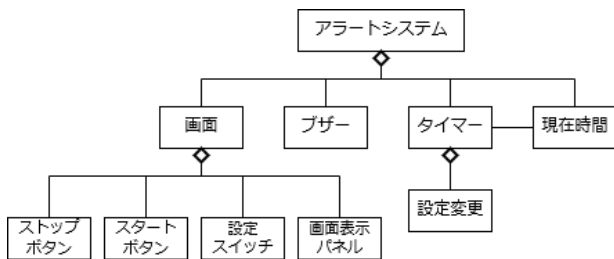


図 3: オブジェクト指向設計

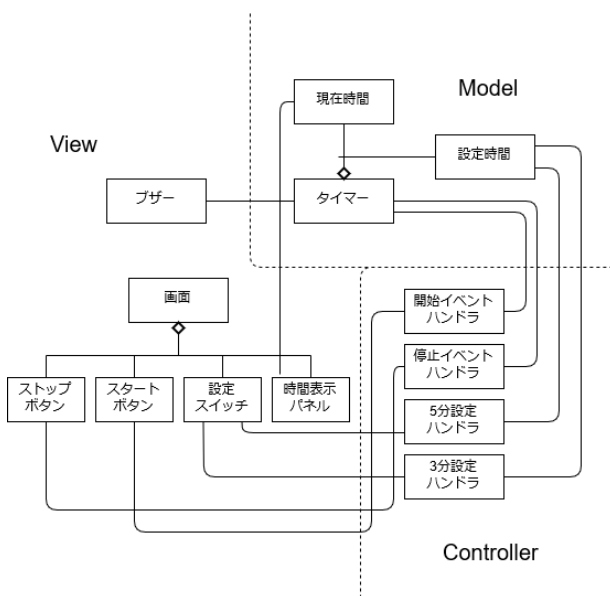


図 4: MVC アーキテクチャスタイルを適用した設計

図 3 に対して MVC アーキテクチャスタイルを適用して設計したものを図 4 に示す。ブザーや画面は出力の役割を持つ View に分類される。ブザーと画面表示パネルはそれぞれ Model のタイマーと現在時間のリスナとして登録される。Model は状態が変化したらリスナにこれを通知する。Model は動的に追加されたリスナに通知を行ない、具体的にどのオブジェクトがリスナか識別しない。各種ボタンとスイッチが押下されたイベントを Controller としてのイベントハンドラは検知し、適切なモデルに通知する。このように View と Controller は Model から独立して定義され、これらの変更は Model に影響を与えない。

## 4 考察

例題のアプリケーションに対して、どのような変更が行われるか考え、その変更のしやすさに関して優劣を表 1 にまとめた。設定時間の変更に関しては、手続き指向では、設定時間に関連する手続きすべてに関して修正が必要となる。オブジェクト指向では、設定時間オブジェクトの多相型を定義し、画面からこの設定時間を設定するメッセージを書き換える必要がある。MVC では、設定時間オブジェクトと設定スイッチの多相型を定義し、これらに関連付けるイベントハンドラを定義する。画面自体にどの Model にメッセージを送るかといった修正は必要ない。見た目の変更に関しては、手続き指向においては、複数の表示手続きに影響が及ぶことが考えられる。オブジェクト指向はこの例においては画面オブジェクトの多相型によって、表示方法毎に独立して定義することが可能となる。MVC においては、View に独立して記述可能となる。通知方法の変更に関しては、手続き指向とオブジェクト指向は、通知方法に関連するコンポーネントを定義するだけでなく、これを実行するための起動側についても変更が必要となる。MVC においては、View はリスナとして実現されることから、通知のきっかけを起こす Model に対しての影響はない。

表 1: 比較結果

	手続き	オブジェクト	MVC
見た目の変更	×		
設定時間変更	×		
通知方法の変更	×	×	

## 5 おわりに

本研究では、アラート Web アプリケーションを例題として、手続き指向、オブジェクト指向、MVC アーキテクチャによる設計を比較した。インタラクティブシステムにおいて MVC アーキテクチャスタイルを適用することが有用であることがわかった。本研究の事例は単純なものであり、比較した内容に大きな差がなかった。今後、より複雑な例について実装まで行なった上で、比較する必要があると考えた。