

Ext JS による技術センター業務依頼システムの構築

○西岡祐介, 甲斐崇浩, 相川勝, 森圭史朗

宮崎大学工学部教育研究支援技術センター

概要

宮崎大学工学部教育研究支援技術センターでは、教員等の業務依頼に対し技術支援を行っている。これまで、業務依頼の申請方法は紙ベースで行われており、大変労力が費やされていた。そこで、この工程をシステム化にすることで、これまでの作業を簡素化でき、依頼者が申請しやすい環境を構築することができた。本発表では、このシステムで SPA（シングルページアプリケーション）を実現するために導入した技術である Ext JS について報告する。

1. はじめに

宮崎大学工学部教育研究支援技術センター（以下、技術センター）では、教員等からの業務依頼に対し技術支援を行っている。これまで技術センターに業務を依頼する場合、依頼者は紙ベースによる業務依頼書を作成し、技術センターに提出する必要があった。また、技術センター側も依頼を受けた業務に対して、担当者を調整する際、他の業務との兼ね合いを調べるために他の申請書類を探す等の手間がかかり、依頼者・技術センターともに時間と労力が費やされていた。そこで、これらの問題を解消するため、業務依頼の方法を Web アプリケーションによるシステムで運用することを目指し、技術センター内でシステム開発グループが立ち上がった。この開発は、新たな技術習得の一環として行われ、開発メンバー間で技術共有を行いながら協力してシステム開発が進められた。開発では、システム設計計画から始まり、2015 年にクライアントサイドに Bootstrap を利用した HTML、サーバサイドに Rails アプリケーションと PostgreSQL サーバで構成されたシステムを構築し、運用を開始した。しかし、クライアントサイドで用いた HTML では、今後、新たな機能を実装する際、拡張性・操作性に対する要求を満たせない部分があったため、要求を満たす機能を持つ SPA を用いたシステム開発を目指すこととした。そこで、SPA を実現する JavaScript フレームワークの一つである Ext JS を用いて、業務依頼システムの機能を再構築したので、今回は、その内容について報告する。

2. Ext JS とは

Ext JS とは、Web アプリケーション開発用の JavaScript フレームワークの一つである。Ext JS の特徴として、豊富な UI ウィジェット、クロスブラウザの互換性、コンポートモデル拡張性の高さなどがある。

また、SPA により、単一ページによる Web アプリケーションを構築でき、高い操作性を実現可能である。アプリケーションの構造は、MVC（Model-View-Controller）、MVVM（Model-View-ViewModel）アーキテクチャ（図 1）を選択でき、目的にあった構成で開発することができる。

開発では、Ext JS アプリケーション単独で進めることができるため、クライアントサイド、サーバサイドが独立して開発を進めることができる点も利点の一つである。

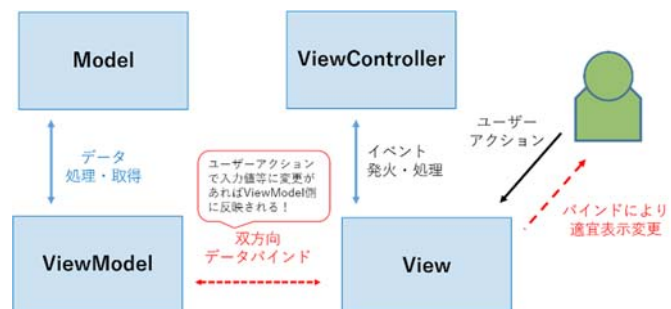


図 1 MVVM アーキテクチャの概略図

3. システム設計及び開発

業務依頼システムは、2015 年に HTML と Rails アプリケーションを用いた構成により構築されており、今回は、本システムを Ext JS アプリケーションと Rails アプリケーションを用いた構成に再構築することを目的とした。再構築後のシステム構成を図 2 に示す。クライアントサイドは、3 つの Ext JS アプリケーションで構成され、それぞれ以下のような機能

を持つ。今回は、各 Ext JS アプリケーション毎に開発担当者を分け、独立して開発した後、統合して1つのシステムとして構築した。

- 依頼者 (Customer)

業務依頼申請、業務終了報告書作成・確認

- 技術職員 (Tech)

業務依頼確認、業務完了報告書作成・確認

- 管理者 (Admin)

業務依頼確認・承認、業務担当者・時間調整

Ext JS アプリケーションで構成されているクライアントサイドと Rails アプリケーションと PostgreSQL サーバで構成されているサーバサイド間のデータの送受信は、JSON データを用いて行われており、本システムでは、Ext Direct と呼ばれる機能を利用することで JSON データを一括送受信し、サーバの負荷を減らすことも行っている。

今回、構築した Ext JS アプリケーションは、データバインディング機能を活用するため、MVVM アーキテクチャを採用した。このデータバインディング機能により、View で値を入力した際、入力値が ViewModel に反映される。この値を別の View で表示する際に

ViewModel から値を取得することにより、DB を介することなく入力値を表示することができる。この機能を用いることにより、入力フォームと確認画面での値の受け渡しの様子を図3のように表現することができた。また、デザインに関しては、Ext JS アプリケーション全体のデザインを一括管理できる「theme」を適用し、容易にデザインを変更することができる。Ext JS では、既存の theme の他にも新しく theme を作成することもできるため、今回は、デフォルトの theme に対し、メインとなる色、コンポーネントの余白の調整を行った新しい theme を作成し、それを本システムに採用した。

4. システム運用

開発終了後、開発メンバーによるテスト・デバッグを経て、2017年10月30日に運用を開始した。運用後は、一部ブラウザの互換性による不具合が生じたが、その他は無事、運用できている。

5. おわりに

今回、業務依頼システムの再構築を Ext JS アプリケーションと Rails アプリケーションを用いた構成で行った。今回は、再構築ということで前回構築時のシステム設計の基本的な資料を活用することができ、時間短縮に繋がると想定していたが、Ext JS の技術習得に時間がかかり、目標の納期までに完成することができなかった。原因としては、今回、使用した Ext JS のバージョンが Ext JS6 であり、技術情報が少なかつた点が挙げられる。Ext JS は商用版と GPLv3 ライセンス版があり、利用するだけなら GPLv3 ライセンス版で十分だったが、技術情報等の収集・取得には商用版ライセンスでないといけないものもあった。ただ、情報が少なかつた中でも開発グループで協力して、新たな技術を習得できたことにより、今後のシステム開発にも十分活用できる。本技術を用いた今後の展望として、カレンダー、グラフ等の UI を利用した Web アプリケーション開発が考えられる。そのために、今後も技術の研鑽を進めたいと思う。

参考文献

1. Loiane Groner. Mastering Ext JS Second Edition, Shroff Publishers & Distributors Pvt Ltd, 2014

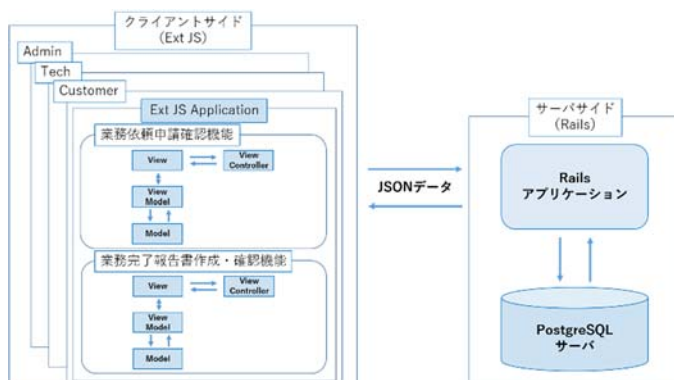


図2 再構築後のシステム構成

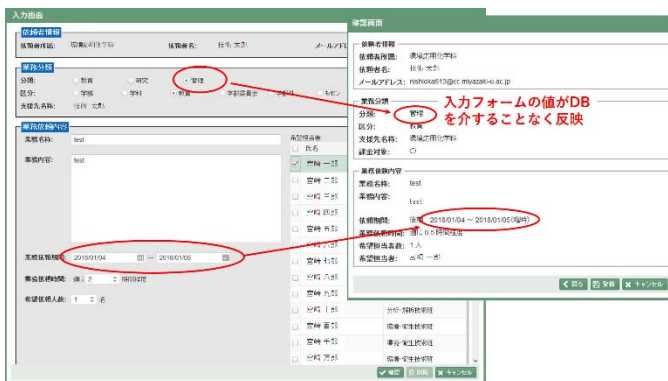


図3 データバインディング機能