

サーバサイド Java システムによる生産管理システムの構築

岡崎 徹*・井澤 正人**

要旨： 近年、PC の高機能化、低価格化およびインターネットの普及により、企業の業務に PC が不可欠なものとなった。さらに自社内のアプリケーションの開発が比較的容易になってきた。井澤は企業在職中、マイクロソフト社の Access を用いて自社向けの生産管理システムを構築した。利用台数が増えるにつれ、アプリケーションの入替作業などの保守に負担が増すようになった。この問題を解決するべくサーバサイド Java を用いてシステムを再構築し、両システムを比較考察する。

キーワード： サーバサイド Java システム、生産管理システム、クライアントサイドシステム、Ajax

Construction of a production control system by a server side system JAVA

Tooru Okazaki* and Masato Isawa**

Abstract: In late years it was it when a PC was indispensable for duties of a company by high function of a PC, price reduction and the spread of Internet. Besides, development of an application in own company became comparatively easy. Isawa built a production control system for own company with Access of Microsoft company during company holding the post. As use number increased, burdens came to increase for maintenance they replaced it, and to operate of an application. I rebuild a system with server side JAVA in order to solve this problem and I compare both systems and consider it.

Key words: server side JAVA system, production control system, client side system, Ajax

1. はじめに

井澤は企業在職中、マイクロソフト社の Access を用いて、自社向けの生産管理システムを構築した。開発は表 1 のとおりリスト作成から始め、一部門での初期運用を開始した。その後、工程票発行や購入伝票発行など徐々に機能を追加し、半年後の本運用ではアプリケーション数 7、部署数 6 となり、3 年経過後の最終運用では、アプリケーション数 11、部署数 7 を数えるまでとなった。

このシステムでは、図 1 のとおりファイルサーバ上にデータベースをおき、各クライアント PC に画面制御、業務ロジックなどを行う業務アプリケーションを配置す

る。そのため作成したアプリケーションは各部署で使用されている PC にそれぞれインストールしなければならず、その結果アプリケーションの修正や機能追加などを行うときは、その業務に関わる「すべて」のクライアント PC のアプリケーションを一斉に入れ替えなければならない(図 2)。PC の台数が限られていたときにはそれほど大きな問題ではなかったのだが、これだけ広範囲に使用されるようになると、プログラムの入替作業は業務の止まった休日や深夜に行わなければならない、管理者にとって大きな負担となってきた。

この問題を解決することを目的に産業情報専攻科の課題研究において、サーバサイド Java システムを用いた生産管理システムを構築した。さらに両システムの操作性

表 1 業務アプリケーションと関連部署
Table 1 Duties application and an associated post

		業務部署名						
		生産管理	資材係	外注係	検査係	総務	現場	技術科
アプリケーション	リスト作成	●	○	○			◎	○
	工程票発行	○		○				
	購入伝票発行	○	○					
	材料票発行	○	○	○				
	受入	○	○	○	○			
	進捗管理	○	○	○	○	○	◎	◎
	支払管理					○		
	原価見積集計	◎				◎		
	生産予定	◎	◎	◎	◎	◎	◎	◎
	発注実績検索	◎	◎	◎		◎		
	負荷時間集計	◎		◎				

●：初期運用 ○：本運用（初期運用から半年） ◎：最終運用（初期運用から 3 年）

* 山形県立産業技術短期大学校 (990-2473 山形市松栄 2 丁目 2 番 1 号)

* Yamagata College of Industry & Technology ,2-2-1 Matsuei,Yamagata,Yamagata,990-2473,Japan

** テン・ソフトウェア株式会社 (992-0011 米沢市中田町 926 番地の 1)

** TEN SOFTWARE CO.,LTD. ,926-1 Nakata,Yonezawa,992-0011,Japan

を比較考察する。

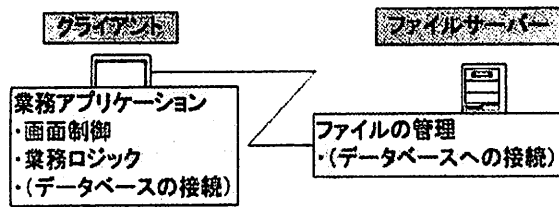


図1 Accessでのシステム構成
Fig. 1 System constitution in Access

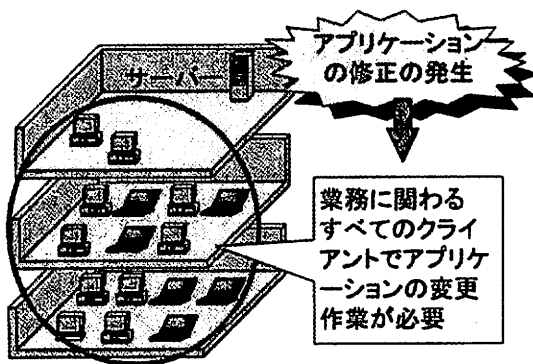


図2 クライアントサイドシステムでのアプリケーションの修正例

Fig. 2 An application revision example by a client side system

2. 生産管理システムの構成

2.1 サーバサイド Java システム

サーバサイド Java システムでは、図3のとおりデータベースはもちろんのこと、クライアントサイドシステムではクライアント PC にあった画面制御、業務ロジックなどの業務アプリケーションもサーバ上に配置し、クライアント PC には、HTML の送受信を行う Web ブラウザのみを配置する。

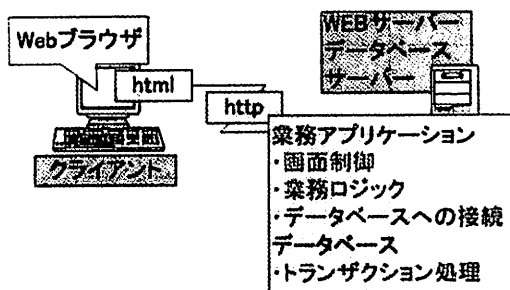


図3 サーバサイド Java でのシステム構成
Fig. 3 System constitution by server side JAVA

その結果、クライアントサイドシステムで問題となっていた業務アプリケーションはサーバ上にも存在することになるため、業務アプリケーションの入替作業はサーバ上のみ行えば良くなり、管理者の負担が大きく軽減される。

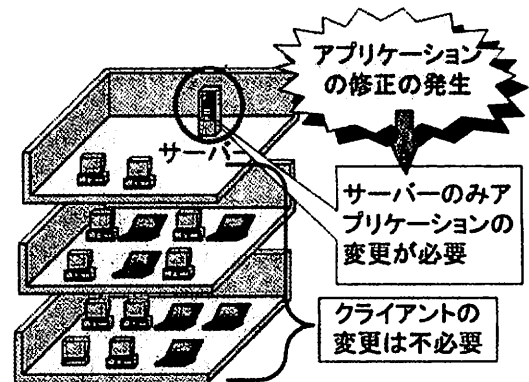


図4 サーバサイドシステムでのアプリケーションの修正例

Fig. 4 A revision example of an application by a server side system

2.2 生産管理の流れ

生産管理の流れは図5の通りである。各処理における主な機能は以下のとおりである。

(0) ログイン処理

ログインしたユーザの権限（部署）をもとにアプリケーションを切り替える。

(1) 製造番号登録

製番テーブルに製造番号、製品名、製作台数などの製造の関わるデータを登録する。

(2) ユニット登録

ユニットテーブルにユニット名、ユニット記号、ユニット数などのユニットに関するデータを登録する。

(3) 部品登録

部品テーブルに図面番号、品名、数量、仕様、規格など部品に関するデータを登録する。

(4) 伝票の入力

伝票テーブルに工程記号、工程名、発注先、発注数など伝票に関するデータを登録する。

(5) 伝票の発行

(4)で入力された伝票を発行する。

(6) 伝票の受入

発行された伝票を受入する。

(7) 進捗管理

伝票の発注/受入/納期遅れ状態などを確認する。

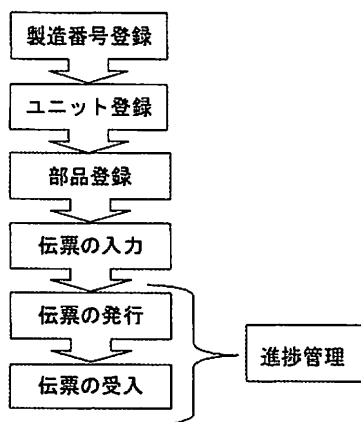


図5 生産管理の流れ

Fig. 5 A flow of production control

詳細なシステム画面構成は、本論末尾の図7の「システム構成」のとおりである。

3. 結果

3.1 保守管理性について

サーバサイド Java をもちいることにより、当初の目的どおり業務アプリケーションの入替作業などが発生しても負担の大きくならない生産管理システムを構築することができた。また Java で生成した Web ページは、標準 HTML のみで構成したため現在使われている Web ブラウザであれば問題なく動作した。そのためクライアント PC の性能、機種さらには OS の種類、ブラウザの種類などをほとんど気にする必要がなくなり、クライアント側の管理が大変楽になった。

3.2 操作性について

クライアントサイドシステムでは、クライアント側で画面の制御や業務ロジックの処理を行うため、「製品番号から製品名を表示させたり、単価と数量から金額を求める」などといったデータの加工や細かい画面制御がシームレスにできる。

一方、クライアント側で HTML の送受信しか行わないサーバサイドシステムでは、先ほどの製品名を求める処理も、金額を求める処理も、その都度サーバに処理を依頼しなければならず、そのたびに「待ち時間」が発生してしまう。

その結果、ユーザにとって使い勝手の悪いものとなってしまった。

4. 操作性向上の改善案

4.1 Ajax^{(1),(2)}

Ajax とは、"Asynchronous JavaScript + XML" の略で、今年初め Adaptive Path の Jesse James Garrett 氏によって名づけられた技術である。

Ajax では、クライアントサイド側にユーザとサーバとの仲介者 -Ajax エンジン- を導入する。このエンジンが必要ときにサーバと通信を行い、必要な HTML を書き換える。この通信は送信ボタンを押すなどのユーザの意志とは非同期にバックグラウンドで行われるためユーザを待たせることなく画面を書き換えることができる。

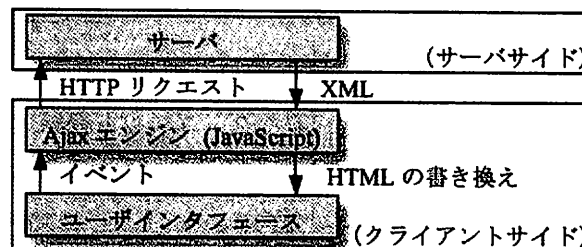


図6 Ajax の概念図

Fig. 6 A conception diagram of Ajax

たとえば、ある入力フォームの入力を終え、次のフォームへカーソルが移動したときに、その入力値の値をバックグラウンドでサーバへ送り、エラーチェックや製品名などの結果を表示させたり、金額を求めるなどの計算も行うことができるのではないかと考えられる。そうすればサーバサイド Java システムで問題となった使い勝手が大きく向上するのではないかと考えられる。

5. おわりに

当初の目的の通り、管理者にとって保守性の高い生産管理システムを構築することができた。しかしその一方、入力データのチェックや金額などの計算を行うときにも、サーバへ送受信が発生し、そのたびに作業が中断してしまい、シームレスな作業ができないものとなってしまった。この問題点を解決する手段として、Ajax 技術の利用が考えられる。過去にクライアントで使用するブラウザの種類、バージョンの違いによる JavaScript の互換性が問題になった時期があった。しかしその後 JavaScript の標準化作業が行われ、現在は互換性は大きく向上したといわれている。事実 Ajax を使った「google サジェスト⁽³⁾」や「goo ラボ エリア情報検索実験⁽⁴⁾」では、動作環境として Internet Explorer 6 以降となっているが、MacOS X 上の Firefox 1.0.6 でも問題なく動作した。

この論文は、平成 15 年度産業情報専攻科の第 1 期生、井澤正人君の課題研究を元に加筆したものです。

本文中の会社名、製品名などは、各社の商標または登録商標です。™, ©マークを省略させていただきました。

文献

- (1) 深津, 増井: "リッチクライアントの革命児 Ajax 見参", Software Design, Vol.242, No.6, pp.84-88(2005)
- (2) 伊藤, 増井, 深津: "Ajax-JavaScript によるユーザビリティ革命", WEB+DB PRESS, Vol.27, pp.111-142(2005)
- (3) Google サジェスト日本語版, <http://www.google.co.jp/webhp?complete=1&hl=ja>
- (4) goo ラボ エリア情報検索実験, <http://map.labs.goo.ne.jp/>

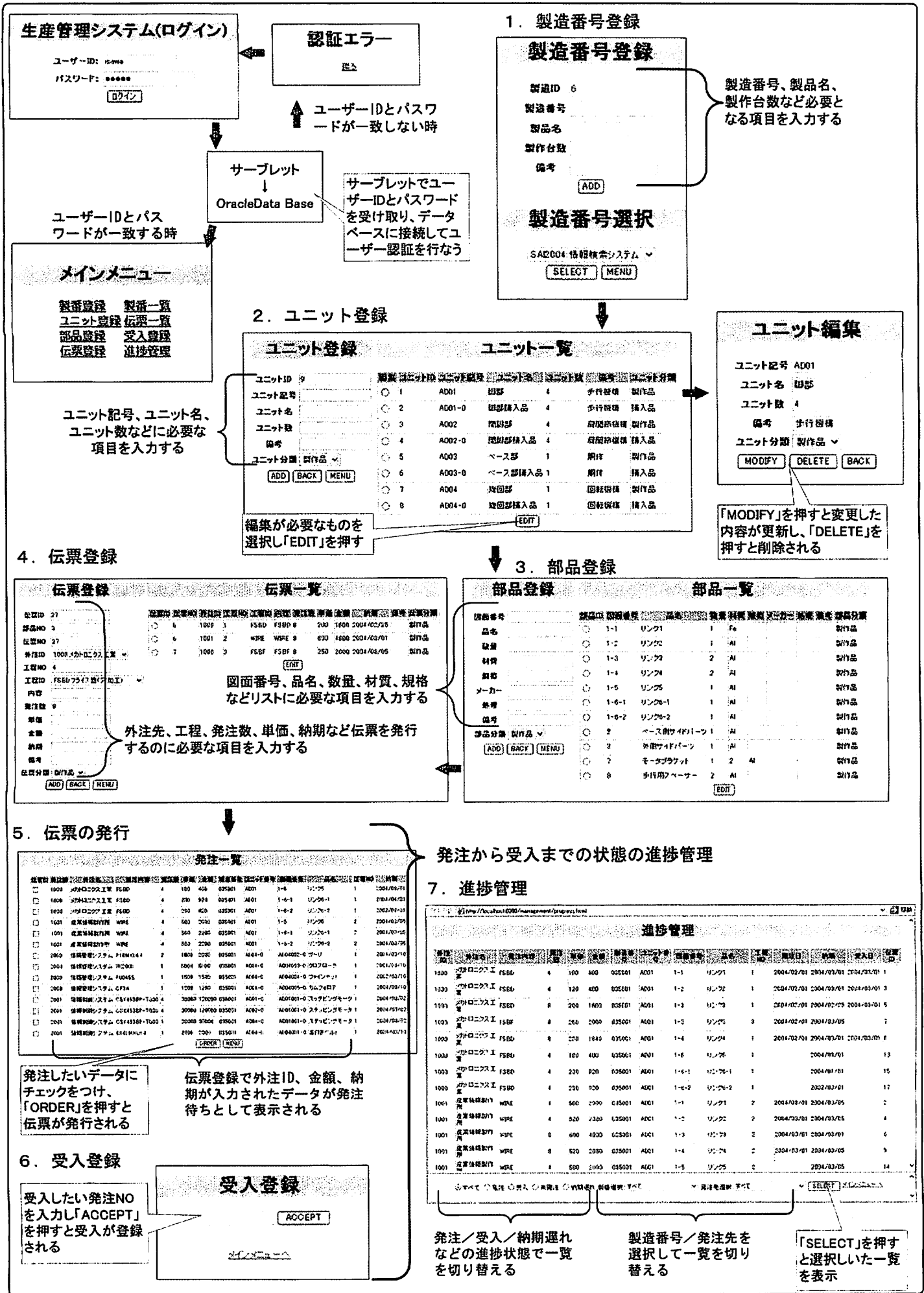


図7 システム構成
Fig. 7 System constitution