

既存パッケージのSaaS化に向けた課題と解決策

前田和俊*
鈴木 剛*

Challenges and Solutions for Existing Packages toward SaaS

Kazutoshi Maeda, Takeshi Suzuki

要 旨

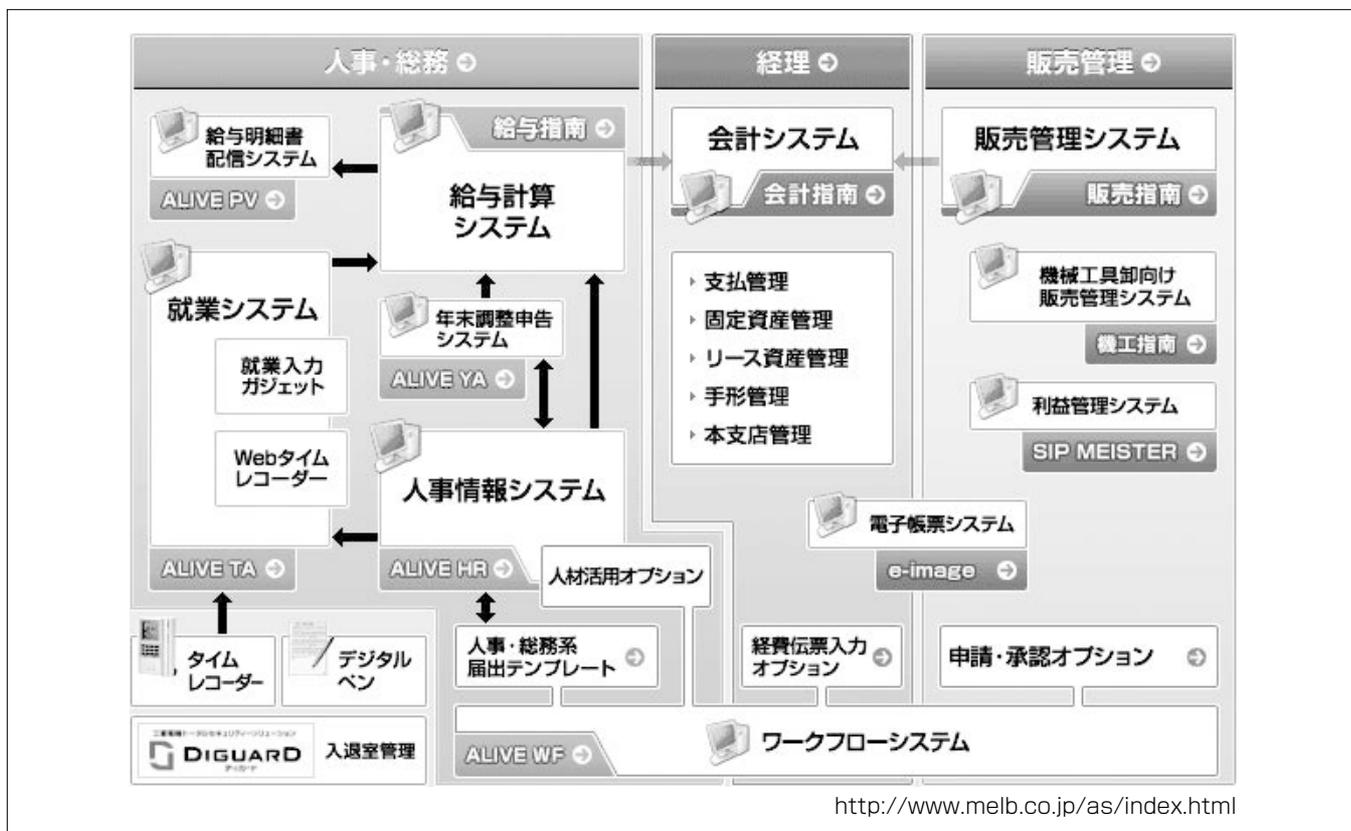
(株)三菱電機ビジネスシステム(MB)では、人事・総務系業務パッケージの“ALIVEシリーズ”と、基幹系パッケージの“指南シリーズ”を開発・販売している。

MBではSaaS(Software as a Service)化に向けた様々な技術検討を実施しており、本稿では、両シリーズのユーザーインターフェースを実現するために開発したフレームワークと共通ソフトウェアコンポーネントを述べ、SaaS化に向けたこのフレームワークの課題と解決策について述べる。

ALIVEシリーズで使用するフレームワーク“radish”には、ユーザーインターフェースを受け持つ各種のコンポーネ

ント群があるが、パッケージとしてイントラネット環境での使用を前提としている。事前検証によって、ネットワークトラフィックに関する課題があり、パッケージをそのままSaaS化することは困難であることを確認した。解決策としてコンポーネントの軽量化に取り組んでおり、その試作評価結果について述べる。

また、指南シリーズは操作の応答性能を重視して、クライアントサーバ方式を採用している。クライアントサーバ方式のSaaS化に向けた課題と取組みについて述べる。



<http://www.melb.co.jp/as/index.html>

“ALIVE・指南シリーズ”の構成

幅広い業種・業務で培ってきたノウハウと技術力を基盤に、ユーザーの抱える現場のニーズを解決するシステムを提供している。効率的・機能的な業務改善を目的としたパッケージであり、パッケージ単独の導入はもちろん、必要なパッケージを自由に組み合わせて最適なシステムにカスタマイズすることも可能である。

*株)三菱電機ビジネスシステム

1. ま え が き

ALIVEシリーズは、総務系の業務を対象とした人事・総務のトータルシステムで、2002年から販売を開始した。

指南シリーズは、経理系の業務を対象とした会計・給与計算システムと、販売管理業務を対象とした販売管理システムで、1998年から販売を開始した。

両シリーズとも各種機能追加、法改正対応、オープン化を進めるための開発言語の変更を行い、顧客ニーズにあわせ商品性を向上させてきた。

本稿では、既存パッケージのALIVEシリーズと、指南シリーズを概観し、ALIVEシリーズのSaaS提供に向けた課題と解決策を中心に、両パッケージのSaaS化への取り組みについて述べる。

2. 既存パッケージの概要

2.1 ALIVEシリーズ

ALIVEシリーズは、人事・総務部門の業務効率化と、戦略的人材活用をサポートする業務パッケージである。

ALIVEシリーズの構成は次のとおりで、ALIVE TAの就業入力画面を図1に示す。

- ①ALIVE TA：就業システム
- ②ALIVE HR：人事情報システム
- ③ALIVE WF：ワークフローシステム
- ④ALIVE PV：給与明細書配信システム
- ⑤ALIVE YA：年末調整申告システム

2.2 指南シリーズ

指南シリーズは、経営活動をサポートする基幹系パッケージである。

指南シリーズの構成は次のとおりである。

- ①会計指南：会計システム
- ②給与指南：給与計算システム
- ③販売指南：販売管理システム

3. ALIVEシリーズのSaaS化

この章では、ALIVEシリーズのシステム概要、ALIVE



図1. ALIVE TAの就業入力画面

シリーズで利用するフレームワークradish、及びSaaS化におけるradishの課題と解決策について述べる。

3.1 ALIVEシリーズの構成

ALIVEシリーズは、社内イントラネットを前提としたWebシステムである。社内のサーバにインストールしたOS(Operating System)上でALIVEを稼働させ、利用者はブラウザからHTTP(HyperText Transfer Protocol)プロトコルでALIVEにアクセスする。図2に、現状のALIVEのイントラネット構成を示す。

SaaSでは、インターネット経由の利用が前提となる。データセンターの仮想マシンにインストールしたOSの上でALIVEを稼働させる。利用者はブラウザからインターネットを経由しALIVEにアクセスする。ALIVEでは人事データを取り扱うため、通信経路上での盗聴などを防止する必要があり、HTTPS(HTTP over Secure Socket Layer)で通信を行う。図3に、ALIVEのSaaS構成を示す。

3.2 ALIVEシリーズのフレームワークradish

radishは、ALIVEシリーズで使用しているMBが開発したJava^(注1)フレームワークである。

radishの中に、Webシステムのユーザーインターフェースを受け持つradish Web UIコンポーネント群があり、その中の1つに入力補助機能を提供する入力コンポーネントがある。HTML(HyperText Markup Language)のInputタグのテキストボックスは入力値の型や範囲をチェックする機能を持たないが、radish Web UIの入力コンポーネントによって、JavaScript^(注1)による入力値チェックやフォーマット変換の機能を付加する。

(注1) JavaとJavaScriptは、Oracle Corp.の登録商標である。



図2. ALIVEのイントラネット構成

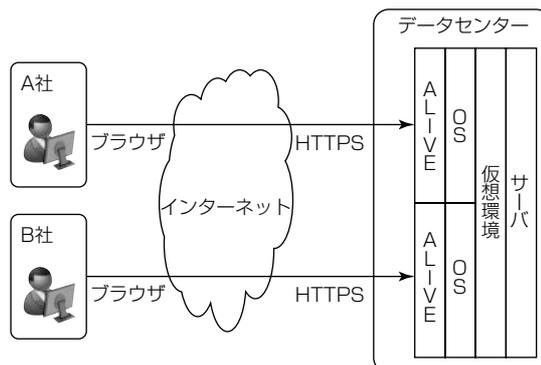


図3. ALIVEのSaaS構成

入力値のチェックはAjax (Asynchronous JavaScript and XML (eXtensible Markup Language) : サーバとの非同期通信)でサーバで処理を行っている。そのため、サーバ側に固有のチェック処理を記述し、テキストボックスから呼び出すことで、ビジネスロジックにかかわるチェック処理を、HTMLに記述することなく実現可能としている。

図4に、時間入力コンポーネントの例を示す。1行目は入力途中の状態、8時53分の入力するために“8.53”と入力している。この状態でフォーカスを移動すると、2行目のように“08:53”と自動でフォーマット変換し表示する。3行目の“8.61”，4行目の“abc”のように時間表現としては不正な入力をした場合、テキストボックスの背景色を変え、入力値の不正を表現する。

3.3 SaaS化に向けた課題

SaaS化に向けた検証の一例として、ALIVE TAをインターネット経由で利用したとき、イントラネット内で利用したときの性能比較、パケット量の調査結果について述べる。

検証環境は、図3の構成で、仮想マシンはCPU (Central Processing Unit) : 1Core, メモリ : 4GBを割り当てた。インターネット経由の利用はMB社内から、イントラネット利用はデータセンター内の別仮想マシンからアクセスすることで、アクセス経路のみが異なる環境を構築した。

3.3.1 ネットワークトラフィックの課題

就業入力画面をインターネット経由で利用した場合、今回検証した環境では、イントラネット環境に比べ、最大2.5倍の応答時間を要した。インターネット経由で応答時間が長くなる原因として、回線速度やサーバ性能のほか、ネットワークトラフィックが多いこともあげられる。

現行のALIVEは3.2節で述べたとおり、コンポーネントでHTML標準のテキストボックスに機能を付加している。機能の実装にはJavaScriptを多用しているため、HTMLが巨大化している。

表1に、テキストボックスのみを表示したサンプルHTMLに関して、radishによる変換処理前、処理後のHTML容量を示す。テキストボックスが1個あるHTMLの容量は161バイトであるが、このHTMLをradish上で実行すると、テキストボックスのほか、formタグに対する初期処理などが追加され、3,266バイトとなる。

No	翌日	時刻(始業・終業)	
1	<input type="checkbox"/>	8.53	出勤
2	<input type="checkbox"/>	08:53	
3	<input type="checkbox"/>	8.61	
4	<input type="checkbox"/>	abc	

図4. 時間入力コンポーネント

JavaScriptで機能を付加したHTMLをクライアントに転送するネットワークトラフィックの削減が、SaaS提供する際の応答性能を維持するための課題となる。

3.3.2 ユーザビリティの課題

SaaSでは、システム導入作業が現状のパッケージ販売とは異なるため、導入時の操作指導に関しても対応が必要となる。より理解しやすいマニュアルを整備するとともに、より直感的に利用可能なユーザビリティへと改善する必要がある。

3.4 解決策

3.4.1 ネットワークトラフィック削減の解決策

SaaS提供するために、radish Web UIコンポーネントが生成するJavaScriptや機能の見直しをすることで、ネットワークトラフィックの削減を目指し、radishの改善版を試作評価した。

現行版のradishでは、コンポーネントの振る舞いに必要なJavaScriptを各コンポーネントが個別に生成している。この動的に生成しているJavaScriptのうち、共有可能な処理を関数として1つ生成し、各コンポーネントから呼び出すことで、JavaScriptの生成量を減らしHTMLサイズを縮小した。

表2に、radishによって機能付加したHTMLに関して、クライアントに出力される容量を示す。テキストボックスが1個の場合、改善版のHTMLは現行版の-52%と悪化している。しかし、テキストボックスが10個の場合は32%、20個の場合は50%と、テキストボックスの数が多くなるほど、共有化による効果が現れる。

実際のHTMLにはデザインに関連する記述や、表示文字列等が含まれるため、一概には言えないが、項目数の少ない画面で転送量が問題になることは少なく、改善による効果は大きい。

3.4.2 ユーザビリティの解決策

具体例として、ガジェットのようなシンプル機能を用意することで、SaaS導入時における操作指導などの利用者の負担を減らすことも可能となる。ALIVE TAパッケージの最新版では、就業入力ガジェット(図5)を追加している。

表1. HTML容量

テキストボックス数	処理前(byte)	処理後(byte)
1個	161	3,266
10個	684	9,895
20個	1,274	17,305

表2. radish改善によるHTML削減効果

テキストボックス数	現行版HTML量 (byte)	改善版HTML量 (byte)	削減量 (byte)	削減効果(%)
1個	3,266	4,972	-1,706	-52
10個	9,895	6,731	3,164	32
20個	17,305	8,701	8,604	50



図 5. 就業入力ガジェット

この機能によって、Windows Vista^(注2)以降のクライアントOSで、デスクトップに表示可能なガジェットとして、出勤・退勤を1クリックで入力できる。

(注2) Windows Vistaは、Microsoft Corp. の登録商標である。

4. 指南シリーズのSaaS化

この章では、指南シリーズで採用している入力コンポーネント“MBInput”とSaaS化に向けた取組みについて述べる。

4.1 指南シリーズの構成

指南シリーズは、クライアントサーバ方式を採用しており、利用端末ごとにソフトウェアをインストールする。

4.2 指南シリーズ共通コンポーネントMBInput

MBInputは、業務アプリケーションにおける項目入力の標準化を目的としてMBが開発した指南シリーズ共通の入力コンポーネントである。売上入力で使用される入力コンポーネントには、伝票情報を片手にテンキーのみで操作できる機能と、1伝票を数秒で登録可能とする性能が求められる。MBInputのコンポーネントは、上で述べたような入力業務で求められる機能と性能の両方を兼ね備えており、ユーザーインターフェースで重要な役割を担っている。

MBInputの特長を次に示す。

- ・複数の入力項目の形式を1つのコンポーネントでサポート
- ・指定された型に対応した入力値制御，入力けた数・範囲指定が可能
- ・Enterキーによるフォーカス移動
- ・フォーカス取得時の背景色変更機能
- ・優れた入力応答性

図6に、販売指南の売上入力の画面イメージを示す。入力項目はすべてMBInputで作成している。

4.3 課題と解決に向けた取組み

指南シリーズは、4.2節に示したとおり入力時のユーザーインターフェースが重要であり、Webシステムとして作り直す場合も、MBInputが持つ機能・性能のすべてを実現する必要がある。このためALIVEシリーズよりSaaS化の課題が多く、次で述べるように、早期提供開始、セキュ

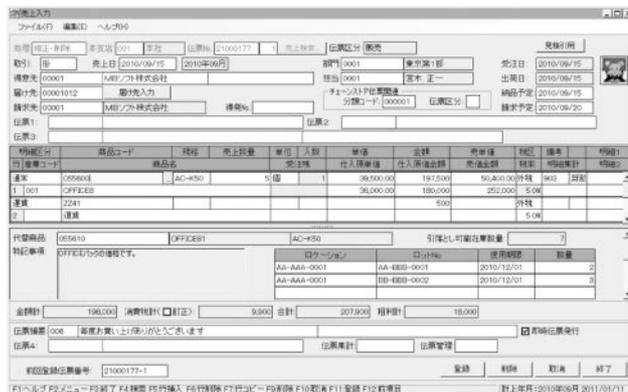


図 6. 販売指南の売上入力画面

リティ、コスト、機能・性能等の要件を満たすことのできる提供方式を検討している。

SaaSでは、インターネット経由でソフトウェアを利用するため、セキュリティや提供コストを考慮するとWebシステムの方が都合がよい。リッチクライアント方式を採用すれば機能・性能を満たせるが、対応開発のための期間とコストを要する。

機能・性能要件を満たした上で、顧客のSaaS化ニーズに俊敏に対応するため、現時点では、既存のクライアントサーバ方式の資産を活(い)かしてSaaS提供する方向で技術課題を抽出中である。

5. ソフトウェア以外の課題

SaaSサービスを提供する際に、ソフトウェア以外に次のような大きな課題が挙げられ、これらについても取組みを進めている。

5.1 サーバのサイジング

ALIVE TAなどの業務処理が特定の時間に集中する場合に、同一の物理サーバにいくつの仮想環境を動作させることが可能か検証し、ガイドラインを設定する必要がある。

5.2 サービス環境構築の迅速化

認証認可の設定、ポート、ファイアウォール等、仮想ネットワークの設定や仮想サーバを容易に設定できる環境やツールを準備し、サービス開始までの導入期間、コストを削減する必要がある。

6. む す び

SaaS提供には、本稿で述べた課題のほか、アプリケーションのマルチテナント対応、マルチテナントでSaaS提供する場合のセキュリティ課題、インターネット回線使用時の性能確保等、既存パッケージとは異なる技術課題が存在する。今後、これらの課題に取り組み、変化の激しい市場環境で、顧客ニーズに俊敏にこたえられる製品、サービスを提供していく所存である。