

# 企業環境の変化に対応する システム間データ連携基盤

河井弘安\*  
三浦 隆\*  
草場信夫\*

Infrastructure for Multi System Linkage in the Cloud Computing Era

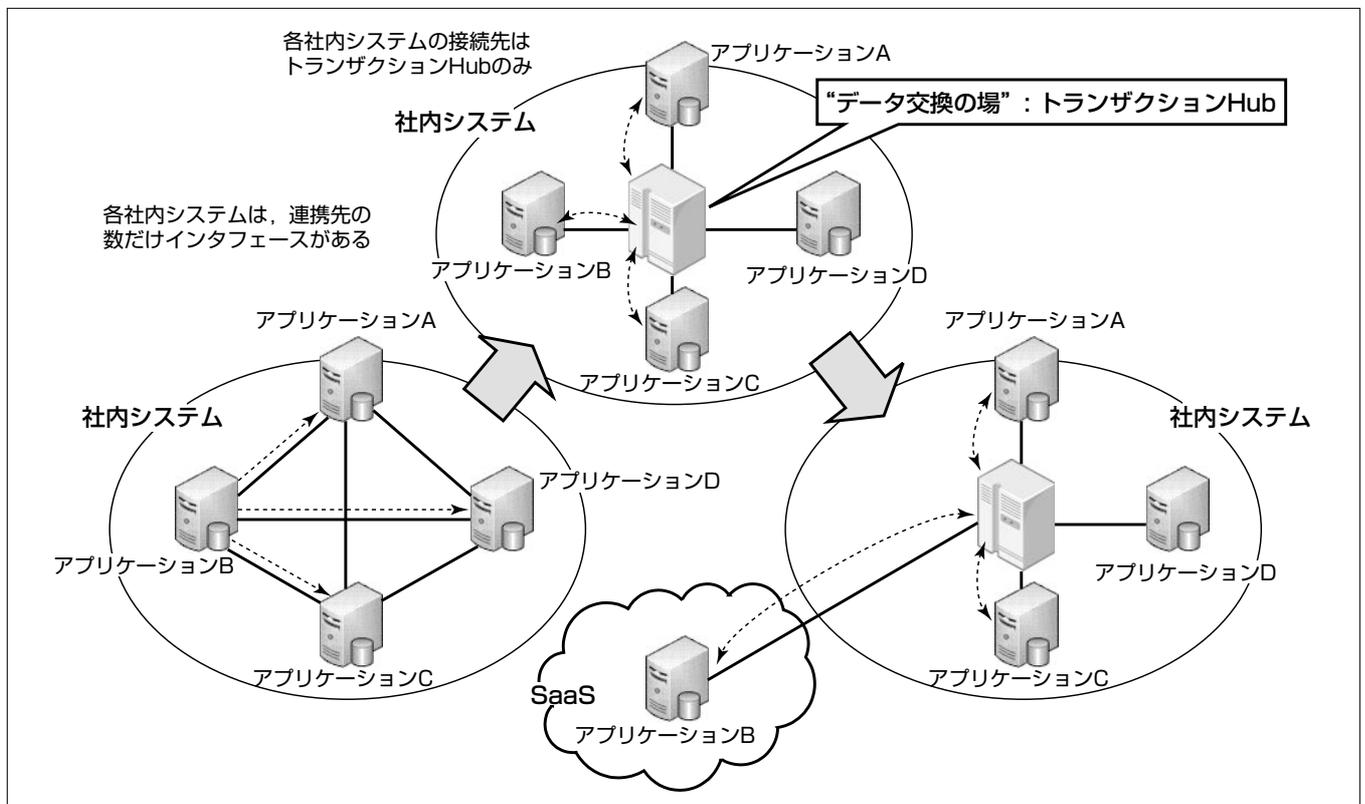
Hiroyasu Kawai, Takashi Miura, Nobuo Kusaba

## 要 旨

近年、業務システムにクラウド、特にSaaS(Software as a Service)の利用が拡大している。SaaSの利用企業数を示す直接的な統計はないが、SaaSの市場規模が年率20%で拡大していることから<sup>(1)</sup>、間接的にユーザー企業のSaaS利用の拡大が進んでいることが分かる。しかし、社内システムのアプリケーションが相互に依存しているために、業務システム上のアプリケーションを単純にSaaS化すると、その連携が分断されてしまう。また、SaaS化するアプリケーションに連携するすべての業務アプリケーションを修正しなければならないなどの問題に直面する。

このような問題を解決するために、三菱電機インフォメーションテクノロジー株(MDIT)は、ETL(Extract Transform Load)ツールを活用して、クラウド技術の適用を支えるデータ連携基盤であるトランザクションHubをユーザー企業向けに構築した。トランザクションHubは、

“もの”の移動をとらえるトランザクションデータを蓄積して、そのデータを再利用する業務アプリケーションとの間で交換する“データ交換の場”である。トランザクションHubを社内システムに導入することにより、業務アプリケーションはトランザクションHubを経由してデータを交換する。その結果、業務アプリケーションは連携先業務アプリケーションの影響を受けない独立した部品となる。このように、トランザクションHubにより相互依存性を排除された社内システムでは、レガシーシステムからオープン系システムへのダウンサイジング、システムの交換、新たなシステムの追加、SaaS化などを容易に実現できる。トランザクションHubを導入することにより、SaaSを含めた選択肢の中から、企業環境の変化に柔軟に対応できる最適な社内システムを構築できる。



## データ連携基盤としてのトランザクションHubによるアプリケーション連携の変遷

MDITが構築したトランザクションHubを社内システムに導入することにより、業務アプリケーション相互の連携を疎結合にする。疎結合になることにより、業務アプリケーションは独立した部品となり、連携先のアプリケーションの変更や取替えによる影響を受けなくなる。このような社内システム連携を構築することで、業務アプリケーションのSaaS化を容易に行える。

### 1. ま え が き

近年、業務システムにクラウド技術を利用する企業数は増加傾向にある。クラウドには種々の定義があるが、本稿では、いわゆるSaaSを対象とする。

SaaSとは、業務を遂行するために外部企業が提供するソフトウェアのサービスを利用すること、業務そのものを外部企業に委託することを言う。サービス提供ベンダー数の増加やネットワーク環境の改善などによって、以前に比べてSaaSを容易に利用できるようになった。しかし、単純に業務システム上のアプリケーションをSaaS化すると様々な問題が発生する。その1つがデータ連携である。

本稿では、MDITがETLツールを活用して、データ連携基盤をユーザー企業向けに構築した事例について述べる。このデータ連携基盤は、社内システムにクラウド技術を採用する際に直面するデータ連携の問題を解決する。

### 2. 業務アプリケーションのSaaS化の進展と問題点

業務システムにクラウド、特にSaaSの利用が拡大している。SaaSの利用企業数を示す直接的な統計はないが、SaaSの市場規模は、年率20%で拡大している。これは、間接的にユーザー企業のSaaSの利用の拡大が進んでいることを示していると言える。また、ネットワーク帯域の拡大がこれを後押ししている。ここ数年で企業内ネットワークや社外ネットワークの回線の速度が飛躍的に向上した。従来は遠距離通信に専用線を使い帯域を確保していたが、現在ではインターネット網でも十分な通信速度が得られる。

ユーザー企業の情報システム部門によるコストメリット追求も、クラウド技術利用拡大に寄与していると言われている。自社にとって独自性の強いシステムか、より汎用(はんよう)的なシステムかといった視点でのシステムの選別が進み、企業の独自性が少ない業務アプリケーションは外部のサービスを利用するか、又は業務そのものを外部企業に委託する流れが加速している。

このようにクラウド技術利用の障壁はかなり低くなっているが、業務アプリケーションを単純にSaaS化すると大きな問題に直面する。

1つ目の問題点は、業務アプリケーション間の連携が分断されてしまう点である。今日の企業の業務は、1つの業務アプリケーションでは完結せずに複数のアプリケーションから構成される。例えば、図1に示すように3つのアプリケーションA、B及びCを利用して業務を行っている場合を考える。アプリケーション間の連携数は、連携先の数だけ存在する。この状態のままでアプリケーションCと同等の機能を持つアプリケーションDというサービスを利用すると、SaaS化前の社内システム(アプリケーションAとC、BとC)との連携が分断されてしまう。

2つ目の問題点は、業務アプリケーションが連携先のアプリケーションに依存しているために、連携先のアプリケーションがSaaSに変更された場合、連携元アプリケーションも修正しなければならないことである。連携元アプリケーションに依存するとは、連携元のアプリケーションが連携先の情報を保持し、それに基づいて処理していることを言う。具体的には相手先サーバのアドレスや、データレイアウトなどである。

図1の例では、アプリケーションAとBはSaaS化前のアプリケーションCに依存した設計となっているため、何らかの方法でSaaSベンダーが提供するアプリケーションDと連携するための修正が必要となってしまう。このことは、SaaS/ASP(Application Service Provider)利用実態調査の結果にも現れている<sup>(2)</sup>。SaaS採用前後で“一度採用すると他サービスに移行しにくい”という問題点が上位に挙げられている。これは、社内アプリケーションが、利用するSaaSに依存した連携を構築しているために、他のサービスへ移行すると、連携する社内アプリケーションを修正する多大なコストが必要になることを示していると言えよう。

### 3. データ連携基盤の構築

あるユーザー企業では、将来の社内システムのクラウド技術活用を想定して、業務アプリケーション間の“データ交換の場”の必要性を感じていた。そこで、ホスト上の販売物流システム、生産管理システム、原価計算システムをダウンサイジングするに当たり、MDITは業務アプリケーションの“データ交換の場”であるデータ連携基盤、トランザクションHubを構築した(図2)。このトランザクションHubで交換されるトランザクションデータとは、“もの”の移動をとらえるデータである。“もの”とは製商品や原材料といった物だけでなく、人の異動、金銭の移動も含み、企業内には伝票データや人事異動データなど様々なトラン

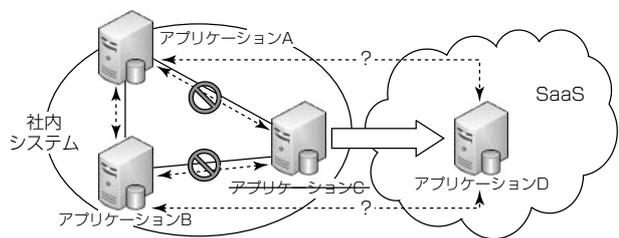


図1. 業務アプリケーションのSaaS化の問題点

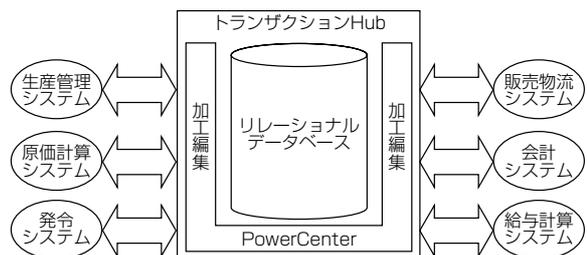


図2. データ連携基盤：トランザクションHub

ザクションデータが存在する。今回構築したトランザクションHubは、製商品や原材料の移動情報を蓄積する。

“データ交換の場”とは、1つのコンピュータ内での複数のプログラムのやり取りであればデータベースである。複数のプログラムは、データベースを介してデータをやり取りして相互の依存性を排除している。この考えを複数のコンピュータ上にある複数のアプリケーションがデータを交換する状況に応用して、“データを交換する場”がトランザクションHubである<sup>(3)</sup>。このデータのやり取りのために、トランザクションHubは、データを格納するサービス、及び取得するサービスを提供する。

### 3.1 トランザクションHubの特長

トランザクションHubは、社内システムに2つの疎結合を実現する。ここで言う“疎結合”とは、接続相手に依存しない連携のことを言う。具体的には、業務アプリケーション間の疎結合と業務アプリケーションとトランザクションHubとの間の疎結合である。

#### (1) 業務アプリケーション間の疎結合

業務アプリケーション間の疎結合は、業務アプリケーションが連携するときにトランザクションHubを介することで実現する。連携元のアプリケーションは、あらかじめ決められたフォーマットでトランザクションデータをトランザクションHubに格納する。格納されたトランザクションデータを再利用するアプリケーションは、必要なデータを必要なタイミングでトランザクションHubから取得する。データを格納、利用する業務アプリケーションはトランザクションHubのみと非同期で連携する。

格納するトランザクションデータは、可能な限り明細データとした。これは、トランザクションデータを再利用するアプリケーションの必要とするデータの粒度が各々異なるためである。例えば、原価計算システムでは月次で集計されたデータでよいが、在庫管理システムでは日々の製商品、原材料の動きをとらえる必要がある。

トランザクションデータを格納するとき、データはすべて挿入のみとする。トランザクションデータを再利用するアプリケーションは、損益計算システムなど会計系システムも含まれている。追跡可能性と検証性を高めるため、修正や訂正などは伝票処理と同様に赤黒修正を行う。

このように、あらかじめ決められたフォーマットの明細レベルのトランザクションデータを、業務アプリケーション間で非同期に共有することによって、データを格納する側はだれがいつデータを使用するかを意識する必要がなくなる。データを再利用する側も、だれがいつデータを格納したかに依存しない。

#### (2) 業務アプリケーションとHubとの疎結合

トランザクションHubは、業務アプリケーションとの疎結合を実現するために3階層アーキテクチャを採用した

(図3)。データストア層、Hub層、インタフェース層である。データストア層にはリレーショナルデータベースを採用した。Hub層は、トランザクションHubの主機能の実装を行い、文字コード変換、データマッピング、データの粒度変換を行う。この機能の実現には、ETLツールであるPowerCenter<sup>(注1)</sup>を利用した。インタフェース層は、格納要求サービス、取得要求サービスのAPI(Application Program Interface)を提供する(図4)。このAPIは、業務アプリケーションからトランザクションHubとの連携の実装やHubそのもの実装を隠蔽(いんべい)する。アプリケーションは、このAPIを通してトランザクションHubにデータの格納や取得を要求するので、トランザクションHubがどのようなシステムかといったことや、どのようにデータを連携しているかを意識しなくてよい。

3階層アーキテクチャは、トランザクションHubそのものの拡張性、保守性を向上させる。トランザクションHubの機能を階層化して階層間のデータの授受の方式を決めることで、それぞれの階層の実装方式を変更しても他の階層への影響を最小限に抑えることができる。例えば、インタフェース層をファイル交換によるバッチ処理で実現しているとき、将来の要求に対してデータベースとの直接連携やWebサービスなどのリアルタイム連携への変更が短期間で実現できる。業務アプリケーションからのトランザクションデータの粒度の変更要求に対しては、Hub層の修正のみで対応できる。このように3階層アーキテクチャを採用することによって、トランザクションHubへの変更要求に対して柔軟に対応できる高品質なシステムを提供した。

(注1) PowerCenterは、Informatica Corp.の登録商標である。

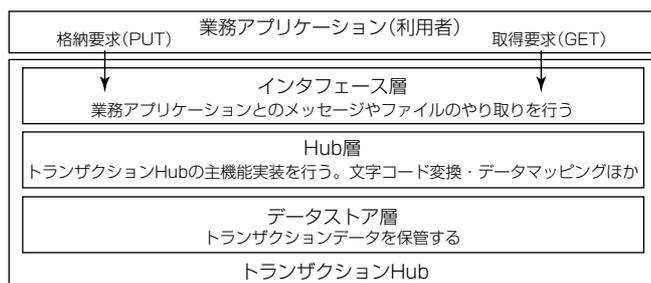
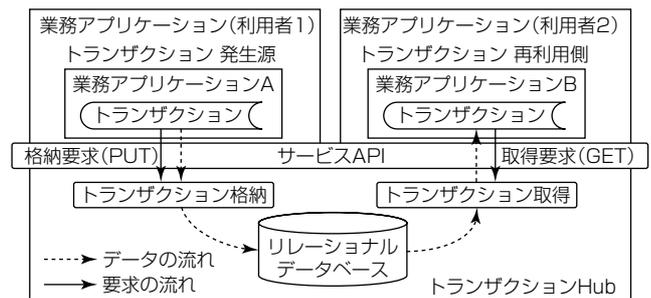


図3. トランザクションHubの3階層アーキテクチャ



API : Application Program Interface

図4. サービスAPIによるトランザクションHubの隠蔽

3.2 サービス指向アーキテクチャとの違い

一般的にSOA(Service Oriented Architecture：サービス指向アーキテクチャ)やESB(Enterprise Service Bus)は次の特徴を備えていると言われている。

- ①標準化されたインターフェースで内部を隠蔽する
- ②個々のサービスは独立して稼働する
- ③サービスの実装方法や稼働しているプラットフォームを問わない
- ④実際にサービスがどこにあっても関係ない

トランザクションHubもこれらの特徴を持っているが、大きな違いは、連携するデータフォーマットを共通化した点にある。“もの”の移動情報であるトランザクションデータのレイアウトはどうあるべきかを突き詰め、汎化(はんか)を行い、フォーマットを決定した。汎化は、単に各業務アプリケーションが使用するフォーマットの最小公倍数を取るのではなく、トランザクションデータのモデル化から行った。具体的には、“もの”の移動を5W1Hでとらえる。何を、いつ、だれが、どこに、なぜ、どのように“もの”を移動したかを表現する。この汎化モデルは、移動を表現するFromとToの情報が含まれているため、UML(Unified Modeling Language)などを使用して表現すると、左右対称のきれいなクラス図となる。トランザクションデータの共通レイアウトを採用することで、業務アプリケーションはデータを再利用する他の業務アプリケーションに依存することなく、データをトランザクションHubに格納できる。

4. データ連携基盤の効果

将来のSaaS化を見据えた社内システムにおいて、業務アプリケーションの相互依存性をなくして、連携するデータのレイアウトを共通化するトランザクションHubの活用は大きな効果をもたらす。

(1) 業務アプリケーションの部品化による効果

業務アプリケーションが疎結合になることによって、業務アプリケーションは独立した部品となる。部品化することによって、その業務に最適な部品(アプリケーション)を利用することができる。パッケージソフトウェア、独自に開発したプログラムなど、適材適所に導入することを可能にする。

部品化した業務アプリケーションは、修正や入替えを行っても他の業務アプリケーションには影響しない。したがって、業務アプリケーションのSaaSへの移行も容易に行うことができる。2章で述べた、業務アプリケーション連携の分断や1つのアプリケーションの変更が連携先に波及するというような問題は発生しない。

その結果、社内システムをレガシーシステムからオープンシステムへ、オープンシステムからSaaSへと段階的に緩やかな移行を可能にする(図5)。業務アプリケーション

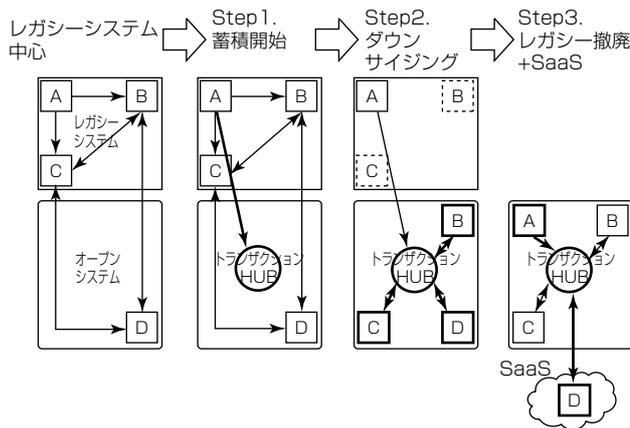


図5. レガシーシステムからSaaSへの段階的に緩やかな移行

を一気にSaaS化するのではなく、まずはトランザクションHubでデータを一元的に管理する基盤を構築する。次に、業務アプリケーションの連携をトランザクションHubを経由させる。業務アプリケーション間の依存関係を排除した上で、SaaS化するシステムを見極めて外出しにするというステップを踏むことが可能となる。

(2) PowerCenterの利用の効果

トランザクションHubにPowerCenterを活用することによって、データの流れを可視化できる。Metadata Manager<sup>(注2)</sup>によるメタデータ管理機能を利用してメタデータを横断的に収集分析し、データの出自や変換ロジックを把握してトランザクションHubの信頼性の向上を図ることができる<sup>(4)</sup>。

(注2) Metadata Managerは、Informatica Corp.の登録商標である。

5. むすび

MDITは、企業環境の変化に対応するシステム間データ連携基盤であるトランザクションHubを構築した。トランザクションHubは、業務アプリケーション間を疎結合にして部品化する。業務アプリケーションの部品化は、企業内業務アプリケーションを緩やかに段階的にSaaS化するというシナリオを可能にする。その結果、SaaSを含めた選択肢の中から、企業環境の変化に柔軟に対応できる最適な社内システムを構築できる。

参考文献

- (1) 国内SaaS/XaaS市場 2008年の分析と2009年～2013年の予測アップデート, IDC Japan (2009)
- (2) SaaS/ASP利用実態調査2009-2010, 日経マーケット・アクセス, 日経BPコンサルティング調査部 (2009)
- (3) 椿正明：名人椿正明が教えるデータモデリングの“技”, 翔泳社 (2005)
- (4) 高山茂伸, ほか：メタデータ管理で広がるデータ統合ソリューション, 三菱電機技報, 81, No.7, 477～480 (2007)