

VPN サービス設備管理に関する小規模ツールから 本格システムへの移行事例

大江 哲也 下村 零 川島 岳志
横山 真生 岩崎 智

NTT コミュニケーションズ株式会社 ネットワーク事業部
〒100-8019 東京都千代田区内幸町 1-1-6

E-mail: {t.ooe, rei.shimomura, ta.kawashima, mas.yokoyama, s.iwasaki}@ntt.com

あらまし VPN サービスの設備管理業務に対する小規模ツールの導入から本格システムへの移行において、コストを抑制しながらサービス拡大、オペレーションプロセスの変化に適時に対応したシステム開発事例を紹介する
キーワード サービス拡大、プロセス変化、システム移行

Case study of migration from small scale tool to large scale system on VPN service

Tetsuya OE Rei SHIMOMURA Takeshi KAWASHIMA

Masao YOKOYAMA and Satoshi IWASAKI

Network Division, NTT Communications Corporation

1-1-6 Uchisaiwai-cho 1-chome Chiyoda-ku, Tokyo 100-8019 Japan

E-mail: {t.ooe, rei.shimomura, ta.kawashima, mas.yokoyama, s.iwasaki}@ntt.com

Abstract This paper shows a case study of system migration from small scale tool to large scale system designed for service-node management for VPN service. On this migration we achieved the cost control of the system development while the service expanding and the operation process changing.

Keyword service expansion , process change, system migration

1. はじめに

IP 技術の発達により、通信市場には安価で安定した VPN サービスが投入されるようになった。従来の専用線サービスから VPN サービスへのマイグレーションが進み、通信キャリアにはより安価でより高品質の VPN サービスの提供が求められている。

このような環境下、通信キャリアのオペレーションサポートシステム（以下 OSS と呼ぶ）には業務コストの低減と共に、次々と投入される新たなサービスに柔軟かつ即時に対応できることが要求される。

NTT コミュニケーションズでは、VPN サービスに対する幅広いお客様のご要望に応えるため「統合 VPN」ソリューションを提供している。「統合 VPN」ソリューションでは「Arcstar IP-VPN」、「e-VLAN」、「Group-VPN」、「OCN VPN」を提供し、お客様のご要

望されるネットワークの規模、用途に応じた選択を可能としている。

本報告で紹介するシステムは Group-VPN の設備管理業務に対応する OSS である。

Group-VPN サービスは、大規模企業を対象とする Arcstar IP-VPN と小規模企業を対象とする OCN VPN の間を補完するサービスとして位置づけられ、他社との競争関係においてはサービスの導入、立ち上げが急がれたサービスであった。

本報告では Group-VPN サービスの設備管理業務における OSS について、3段階の Step (Step1: 小規模ツール, Step2: プロトタイプ, Step3: 本格システム) を経て小規模ツールの導入から本格システムへの移行し、サービス拡大、オペレーションフローの変化に柔軟に対応した事例を紹介し、本移行を検討するための

契機となった条件，移行による効果，移行におけるポイントについて述べる。

以下，第2項で本ツール，システムの主な機能について述べ，第3項では小規模ツールにおける運用について述べる．第4項においては本格システム化を見据えたプロトタイプ実現の具体例を示し，第5項では本格システムへ移行するにあたっての条件，最後に第6項で小規模ツールから本格システムへの移行におけるメリット，移行におけるポイントについて考察する。

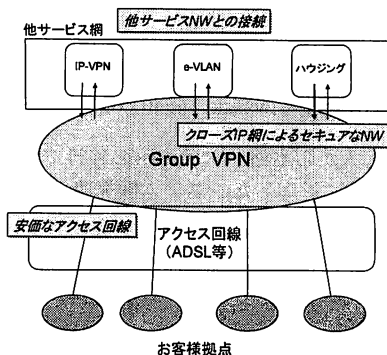


図1. Group-VPN サービス

2. 対象ツール，システムの主な機能

本ツール，システムが対応する業務プロセスの範囲は e-TOM^[1] (運用レベル2プロセス) における以下のプロセスである。

- ・「オーダー処理プロセス」
- ・「問題処理プロセス」
- ・「サービス設定・起動プロセス」
- ・「リソース供給・割当プロセス」
- ・「リソース管理・運用準備プロセス」

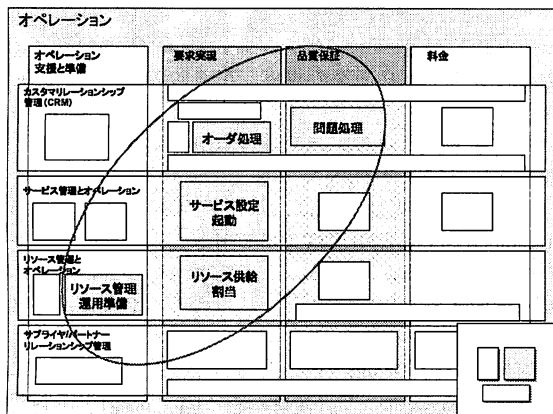


図2. 本システムの適用範囲

各プロセスでの業務は次のように定義される。

2.1. オーダー処理プロセス

「オーダー処理プロセス」では顧客から要求されたサービスを処理する。

「オーダー処理プロセス」で顧客の要求サービスが選択された後，「リソース供給・割当プロセス」により通信サービス提供装置（以下，装置と呼ぶ）へ顧客ごとのサービスの割付が行われる。また，顧客ごとの要求サービスが受付されてから提供開始するまでの進捗を管理するプロセス管理を行っている。

2.2. 問題処理プロセス

「問題処理プロセス」では顧客に対するサービス品質を保証するため，保守部門が必要とする情報の提供を行う。

装置の故障等でサービス異常が生じている顧客への連絡に必要な情報や，その他サービス品質を維持するために必要となる保守運用情報の提供を行う。また，装置の計画工事において，当該装置に割当られる顧客へのサービス中断等の事前工事連絡の情報提供も行う。

2.3. リソース供給・割当プロセス

「リソース供給・割当プロセス」では顧客が要求したサービスを提供するために，装置の物理ポート，論理ポート等（以下，リソースと呼ぶ）への割当を適時に行う。

顧客の要求サービス（「オーダー処理プロセス」を予め準備されたリソース（「リソース管理・運用準備プロセス」）へ割り当てることでサービス提供を実現する。リソース割当時に当該装置設定（「サービス設定・起動プロセス」における設定情報生成）で必要となる各種パラメータの自動生成を行う。各種パラメータのうち，有限のパラメータ資源（VLAN-Tag，グローバルIPアドレス等）については，パラメータ払出し，開放，再利用の管理を実現する。

2.4. サービス設定・起動プロセス

「サービス設定・起動プロセス」では顧客から要求されたサービスを適時実現するため装置への設定情報（コンフィグレーション）生成を行う。

当該設定情報は前項の「リソース供給・割当プロセス」において，顧客の要求サービスをリソースに対して割り当てることで生成が可能となる。

2.5. リソース管理・運用準備プロセス

「リソース管理・運用準備プロセス」ではリソースの管理を行い，需要に応じた準備を行う。

本プロセスの実行により，顧客ごとの要求サービスのリソースへの割当（「リソース供給・割当プロセス」）が可能となる。また，タイムリーな運用準備を可能とするため設備ごとの割当情報を提供し，リソースに対するプロビジョニング業務をサポートする。

3. 小規模ツール

Group-VPNサービスの開始時期では市場規模も小規模であり顧客からの申込も少数であった。また、サービスメニューのバリエーションも少数に限定されており、2項で示した各種プロセスも単純であった。一方、その各種プロセスにおいては、必ずしも完成された処理フローが適用されておらず最終的に効率的なプロセスに達するまでの間、活発なPDCAサイクルによる改善が続けられた。また、OSSに対して迅速な機能提供の要求が強いこともGroup-VPNサービス開始時期の特徴であった。

本報告で紹介する本格システムへの移行の3つのStepのうち最初のStep（小規模ツール）では、サービス開始時期におけるサービスの変化、プロセスの変化に迅速に対応できる業務サポートツールが要求された。その要求に応えるため、小規模ツールは「業務処理を行うオペレータ自身による変更が容易に行える」表計算ソフトの活用により業務サポート機能が実現された。

その主たる機能、およびその運用における問題点を次に示す。

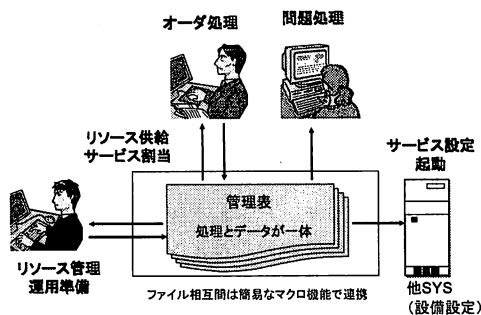


図3. 小規模ツール

3.1. 小規模ツールの機能

① オーダ処理プロセスにおける機能

顧客の要求サービスの登録およびそのサービスの実現までの処理状況が手動管理により行え、情報共有が可能である。実現方法としては、管理表としてデータ保持ができる簡易な方法を用いた。

② 問題処理プロセスにおける機能

問題処理に特化した機能を実装せず、データを共有、参照することにより「問題処理プロセス」を実現した。

③ リソース供給・割当プロセスにおける機能

「リソース管理・運用準備プロセス」により予め準備されるリソースに対して、「オーダー処理プロセス」により登録された顧客の要求サービスを手動で割り当てる。管理表内のセル間に関連付けを行う簡易な方法で

実現した。

④ サービス設定・起動プロセスにおける機能

「オーダー処理プロセス」、「リソース共有・割当プロセス」により投入されたデータを基に設定情報抽出を行う。本情報抽出においては表計算ソフトのマクロ機能により機能を実現した。

⑤ リソース管理・運用準備プロセスにおける機能

「リソース管理・運用準備プロセス」に特化した機能を実装せず、データをオペレータ間で共有、参照することによりリソースの使用状況を定期的に確認し、サービス需要に対応する運用準備を行った。具体的には、装置のフィールド工事実施後、管理表へ割当可能なリソースを手動により登録することにより実現した。

3.2. 小規模ツールの問題点

小規模ツールはサービス開始当初における環境の変化（新規サービスメニューの追加や装置の設定値の追加、変更等）に対して、オペレータ自身による機能の変更が容易に可能であり、環境の変化に柔軟かつ迅速に対応ができる反面、以下の問題点が存在した。

① 同時利用が困難

管理表として、ファイル単位での管理を実現していたため、サービス開始後暫く経過すると、オーダー数の増加と同時に装置の増設も活発に行われ、「オーダー処理プロセス」処理時に「リソース共有・割当プロセス」の結果をリアルタイムで共有することが困難となった。また、他オペレータと本ツールの同時利用性に制約があり、オペレーションの分散が図れないでいた。

② 高度なチェック機能の実装が困難

新サービスメニューの拡大に伴い複雑な投入情報チェックが必要となった。そのため、オペレーションにおいては誤投入が多く、投入データの正確性を担保するため有スキル者による相互チェックを実施するなど、大きな人的コストを要した。小規模ツールでは高度なチェック機能を実装することが困難であり、十分なチェック機能を実現できなかった。

③ 条件指定による検索抽出機能の実装が困難

データの抽出には表計算ソフトの検索抽出機能を利用したため、抽出の都度、オペレータが問題処理のためのデータや装置情報のデータを必要な形式に手動で加工していた。

3.3. 小規模ツールの改善

上記3.2項に挙げた問題点

① 同時利用が困難

② 高度なチェック機能の実装が困難

③ 条件指定による検索抽出機能の実装が困難

により、サービス開始後暫く経過し、オーダー数が増加するに従い、効率的な運用を行えなくなるといった問題を生じることとなった。しかし、サービスの拡大によ

る各プロセスの変化は落ち着く傾向にはあるものの依然活発であり、OSS の改変に対する要求は強かった。

そこで、筆者らは本格システム化を見据えプロトタイプとして、小規模ツールをデータベースソフトによりシステム化することとした。

4. プロトタイプ

プロトタイプではアプリケーションソフト（事務用データベースソフト）を活用し、処理アプリケーションとデータベースを分離した。表計算ソフトファイルに比べオペレータ自身による自由な改変は不可能となったが、現場の有スキル者による機能改変の容易性は確保した。それにより、サービス拡大、プロセス変化に柔軟に対応しつつ 3.2.項で挙げた問題点への対処を実施した。

4.1. 同時利用の問題の改善

処理アプリケーションから分離したデータベースを導入し共有することで、処理アプリケーションからの同時アクセスを実現した。

異なるプロセスの同時実行により、各プロセスにおける待機時間削減を可能とし、効率的なオペレーションを実現した。さらにデータベースの共有化により同一プロセス内でのオペレーションの分散を可能とし、オペレーションの効率化（同一プロセス並行動作により処理能力を 300% 拡大）を実現した。

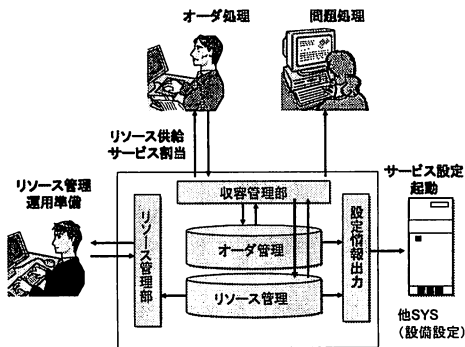


図 4. プロトタイプ

4.2. 高度なチェック機能の実装における問題の改善

処理アプリケーションの GUI 化、及び高度チェック機能の導入により、オペレータによる誤投入を防止する機能を実現した。その結果、有スキル者による相互チェックを廃止することで「オーダー処理プロセス」、「リソース共有・割当プロセス」における運用コストの削減（約 40% 削減）を達成した。

4.3. 条件指定による検索機能の実装における問題の改善

処理アプリケーションへの検索抽出機能の実装により、問題処理のためのデータや装置情報のデータ抽出、データ加工等におけるオペレーションを単純化し、その作業に要する人的コスト（約 10%）を抑制した。

4.4. プロトタイプの問題点

プロトタイプでは上記に挙げた問題点を解決することでトータルとして業務効率向上（約 40%）を実現したが、更なるサービスの拡大と共に以下の新たな問題が発生した。

① 処理能力

サービス需要の更なる拡大に伴う処理データ量の拡大に対して、アプリケーション、データベースの処理負荷が増大し、結果、オペレーション効率を悪化させるという事態となった。その原因として以下の要素が挙げられる。

- ・アプリケーションソフトにおけるアドオン機能の利用がアプリケーション開発を容易にする反面、必ずしも効率的な内部処理を行っていないこと
- ・データベースが大規模な容量を想定した機能を具備していないこと

② トランザクション管理の未実装

トランザクション管理が強固でなかったため、システム高負荷時や通信不安定時にデータの正確性を担保できなかった。

また、他プロセスに独立対応して構築された個々のシステムとの連携がサービス拡大とともに必要となった。システム間連携においては、システム間のデータ不整合を防ぐために高信頼なデータベース間通信を行う必要があったが、プロトタイプではデータの正確性を確保することができずシステム間連携を実現することができなかった。

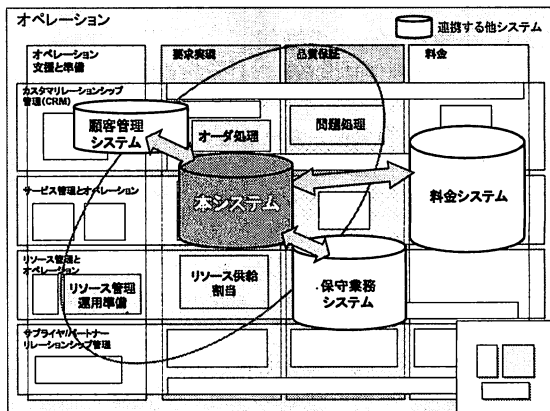


図 5. 他システム連携の実現

③ セキュリティ対応機能の未実装

SOX（サーベンス・オクスリー）法の導入に伴い社内業務システムの企業内内部統制の強化が求められ、特に不正アクセス、データ改竄に対するセキュリティ管理を可能とする仕組みが要請された。

筆者らはこれらの課題を解決すべく、プロトタイプから本格システムへの移行を検討した。

5. 本格システムへの要求事項と対応

本格システムの要求事項として以下の点を挙げた。

- ① 処理能力を拡大し、大量データに対し高速処理を提供できること
 - ② データベースアクセス時のトランザクション管理を実現しデータの正確性を確保し、他システムとの連携を実現すること
 - ③ セキュリティを強化し、不正アクセス、データ改竄の検知を可能とすること
- また、プロトタイプでの問題点への対応に加え、移行時に次の事項に留意した。
- ④ サービス拡大、プロセス変化に対する即応性の確保
 - ⑤ システム移行のコストを最小限に抑えられる方式

5.1. 高速処理への対応

データベースを事務用データベースソフトから専用DBMSへ移行することにより、容量の拡大、データ処理能力の拡大を図った。

また、処理アプリケーション部分をプログラミング言語化することでデータベースソフトのアドオン機能から脱却し、より高速な処理を可能とした。

5.2. トランザクション管理の実現

DBMSの具備するトランザクション管理の活用により、システム高負荷時、通信不安定時におけるデータ正確性の確保を実現し、他システムとのシステム間連携を実現した。

5.3. セキュリティ管理機能の強化

処理アプリケーション部分においてオペレータの権限による利用可能なシステム機能の制限を実現すると共に、オペレータによるデータ操作時のLog記録機能を実装し、不正アクセスやデータ改竄の検知を可能とした。

5.4. 開発容易性の確保

小規模ツールからの移行性、開発環境の充実度を考慮し、システムのプラットフォームとしてMicrosoft®.Net Frameworkを採用した。Microsoft®.Net Frameworkが持つフレームワークの活用、開発環境の利用により開発容易性を確保した。

5.5. システム移行コストの抑制

システム移行コストをツール機能自体の移行コス

ト、及び本格システムの実行を移行する際に係る教育コストに分け、それぞれの抑制を検討した。

① 本格システムへの機能移行コスト

機能移行コストの抑制に対して、データベースソフトウェアとのデータ移行性及び処理アプリケーションのプログラム移行性を重要視し、DBMS移行ツールを利用した。

② 本格システムへの移行に伴う教育コスト

システム移行に伴う再教育コスト抑制に対して、プロトタイプでのGUIを本格システムへ移植することでオペレーションプロセス変更への影響を極力排除した。

6. 小規模ツールから本格システムへの移行におけるメリット、移行におけるポイント

小規模ツール開発から本格システムへの移行によるメリット、およびポイントをまとめると次のようになる。

6.1. 小規模ツールから本格システムへの移行におけるメリット

① 環境変化への即応性

サービス開始時期、拡大時期においては、オペレーション方法の変化、及びPDCAサイクルによるプロセスの変化に柔軟に対応する必要がある。なお、プロセス変化と並行してサービス提供、サービス拡大を継続しなければならないことから、OSS開発の対応も迅速に行われなければならない。その点で、サービス開始時期における小規模ツールによる対応は即応性に優れた特性があり有用である。

② 不要機能開発のリスク回避、システム開発コストの抑制

プロセスの変更において新たな機能が必要になる反面、一部機能が不要となった場合、かつその開発に大きなコストを要すると、その損失が大きくなってしまふ。この点において、本手法はサービス開始時期、拡大時期においてサービス需要動向を把握しつつ、本格システムへの移行必要性を検討できることから、投資リスクを回避できるメリットを持つ。

また、小規模ツールから本格システムへの移行においては、小規模ツール、プロトタイプにおいて確立した機能面での設計を効果的に本格システムへ移植することで、システム設計面におけるコスト、時間を大きく節約することが可能となる。

③ オペレーションを停止しない開発

小規模ツールから本格システムへの移行にあたり、例えばシステム運用開始時にプロトタイプの一部機能を外部呼出しにより連携するなど、現場でのオペレーションを停止することなく段階的に本格システムへ移行できる。これにより、オペレーションを停止しない継

続的なシステム開発が可能となる。

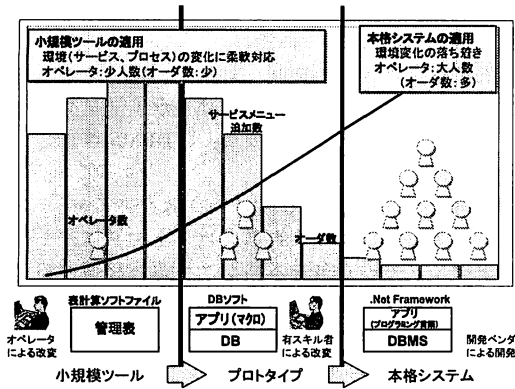


図 6. 本格システムへの移行イメージ

6.2. 小規模ツールから本格システムへの移行におけるポイント

小規模ツールから本格システムへの移行におけるポイントとして次の2点が挙げられる。

① 本開発手法の適用領域について

システム開発においては、サービス需要の確度の高い予想やプロセスの確定に基づき、要求仕様を策定し、開発を遂行することが一般的である。

本格システムをサービス開始当初より導入する場合はシステムの仕様策定から運用開始までのリードタイムが長くなる反面、小規模ツールへの開発投資、運用対処によるコストが不要となる。

従って、本手法の適用は新サービス開始時期においてOSS開発への要求条件が激しく変動し、早期の対応を迫られる場合に有効である。

② ツール機能の本格システムへの移行について

小規模ツール（プロトタイプ）は運用現場に近い環境で作成されるため機能の変更が容易な表計算ソフト、データベースソフト等の事務系ソフトを利用して構築されることが多い。小規模ツール（プロトタイプ）開発で事務系ソフトの汎用機能（アドオン等の機能）を多用した場合、本格システムへの機能移植において、汎用機能自身の移植が必要となるため、想定外のコストを要する場合がある。従って、本格システムへの移行を念頭においた小規模ツール（プロトタイプ）開発時の汎用機能の選択が本格システムへの移行効率化のポイントとなる。さらに、本格システム移行にあたっては汎用機能に対応するフレームワークが豊富に準備されているプラットフォームを選択することがより効率的である。

またサービス開始時期では迅速な対応を優先し、設計ドキュメント等を準備せずOSS対応を優先することが多い。小規模ツール（プロトタイプ）開発におけ

る有スキル者の知識を形式知とするために、環境変化の落ち着きに応じて、本格システム化を見越した設計ドキュメント等を整備しておくことが重要である。

7. おわりに

通信市場における厳しい競争環境下では、新サービスの急速な立ち上げ、及び日々変化するプロセスに対してOSSの迅速な対応が強く求められ、システム開発期間短縮化へのニーズは今後もさらに強まると予想される。

そのようなシステム開発のニーズに対して、本報告ではVPNサービスの設備管理業務における小規模ツール導入から本格システムへの移行の適用事例を紹介し、至急のサービス立ち上げが要求される場合において、対応柔軟性を有する小規模ツールによる対応とサービス需要の伸びに応じた本格システム化への移行が有用な手法であることを確認した。

文 献

- [1] ITU-T M.3050 eTOM enhanced Telecom Operation Map