

損害保険料率算出機構

国民生活に密着した損害保険については、社会・公共的な観点から、公正で妥当な保険料率の算出を通じて安定的な保険の提供が確保される必要があります。このため、損害保険料率算出機構では、会員である保険会社からデータを収集し、適正な参考純率と基準料率を算出、会員保険会社に提供しています。その中で自動車保険に関する自動車保険統計システム「ACTION」は、ハードウェア更改にあたり、初期/ランニングコストの削減、従来以上のシステム性能向上などをめざして、UNIX から Windows へインフラ移行することに。統合開発環境として Micro Focus Visual COBOL を導入。新システムでは、コストや処理時間の短縮およびプログラム品質の向上が見込まれています。



Overview

損害保険は、社会の安全・安心を支えるインフラの一つです。この損害保険の保険料率を適正な水準で維持することを通じて、損害保険業の健全な発達と、保険契約者等の利益保護を実現するために制定されたのが、「損害保険料率算出団体に関する法律」です。この法律に基づいて設立されたのが損害保険料率算出機構で、会員である保険会社から大量のデータを収集し、精度の高い統計に基づく適正な参考純率と基準料率を算出、会員保険会社に提供しています。

同機構の主要業務は、上記の料率算出業務、自賠責保険の損害調査業務およびデータバンクの3つです。

自賠責保険に関しては、公正で適正な保険金の迅速な支払いが行われる必要があるため、同機構において、自賠責保険の損害調査を行っています。

データバンク業務については、各種保険に関するデータを収集して統計の作成や各種の調査研究を行い、これらの成果を会員保険会社や社会に提供しています。

Challenge

機構の開発したシステムの一つに、「ACTION」と呼ばれる自動車保険に関する自動車保険統計システムがあります。毎月、会員保険会社から契約情報、保険金支払情報などのデータを収集、機構内でデータを処理した上で料率計算の基礎データ作成を行います。バッチ処理プログラムとして構成されており、もともとこれはメインフレームで稼働していました。1年分が約1億レコードに迫るデータ量で、それを1度に数年分扱う大規模システムであることから、他のシステムに影響を及ぼさないよう2003年にUNIX環境へ移植、その後、ハードウェア更改を経て今日まで活用しています。

2017年4月、同機構では4回目のハードウェア更改となる「ACTION 4」に向けて動き出すことになり、システムが抱えていた3つの課題解決にも取り組むことにしました。

1つめは、初期/ランニングコストの削減です。このシステムにはRDBMSが導入されており、プログラムはSQLが7割、COBOLが2割、C言語が1割という構成でした。しかし、システムを改めて分析した結果RDBMSである必要がないと判断したため、RDBMSを撤廃してCOBOLの標準ファイルに変更し、SQLで書かれたプログラムをCOBOLにリライトすることにしました。それは情報システム部内のテクノロジースキルを集約するという意味でも重要な決断でした。また、コストの観点からUNIXからWindowsへとOSも変更することになりました。近年は情報システム部内でWindowsシステムが増えてエンジニアが親しんでおり、Windowsサーバーの性能の向上を実感していたのがその大きな理由でした。

2つめは、従来以上のシステム性能の実現とそれによる業務効率アップです。現状は直近複数年分のデータで料率計算を行ってききましたが、さらに統計精度を上げようとデータ量を数倍に増加するという計画が持ち上がっていました。こうしてデータ量は増加する一方であるため、高い性能を望んでおりました。

3つめは、テスト環境の充実です。従来はUNIXマシンスペックの制限により、テスト環境を確保できない状況でした。これを機にテスト環境も合わせて増強し、さらなるプログラム品質向上をめざすことになりました。

Solution

今回の移行プロジェクトでは大きなプログラム変更も入るため、情報システム部はハードウェアベンダーが所有する検証センターで、

At a Glance

- **Industry**
料率算出団体
- **Location**
本部：東京都新宿区
- **Challenge**
ハードウェア更改を契機としたRDBMS撤廃
バッチシステム性能の向上
テスト環境の充実
- **Solution**
+ SQLをCOBOLへリライト
+ Micro Focus Visual COBOL導入
- **Results**
+ システムコスト半減
+ モデルプログラムの処理時間は1/5以下に短縮
+ 十分なテスト環境の確保
+ COBOLへのプログラム集約によるメンテナンス効率向上

「マシンスペックは十分のはずなのに、性能が出ないことに違和感があって
マイクロフォーカス社に連絡を取りました。
すると、ボトルネックの検出方法やコンパイルオプションの設定方法など
COBOLのパフォーマンスチューニングヒントを
技術者が丁寧に教えてくれました。あのアドバイスがなかったら、
COBOLへのリライトは断念していたかもしれません。助かりました」

損害保険料率算出機構
情報システム部 システム開発グループ
プロジェクトマネージャー

www.microfocus.co.jp

SQLをCOBOLへリライトしたモデルプログラムを走らせてみました。当初、期待する性能が出なかったため、情報システム部 システム開発グループはマイクロフォーカス社へ相談します。プロジェクトマネージャーは当時のことを次のように振り返ります。

「マシンスペックは十分のはずなのに、性能が出ないことに違和感があって連絡を取りました。すると、ボトルネックの検出方法やコンパイルオプションの設定方法などCOBOLのパフォーマンスチューニングヒントを技術者が丁寧に教えてくれました。あのアドバイスがなかったら、COBOLへのリライトは断念していたかもしれません。助かりました」

結果的に、マシンスペックがそれほどハイエンドでなくても処理時間は従来の1/5以下になることが分かり、この結果から、情報システム部は当初の方針でハードウェア更改を進めることとし、プログラムメンテナンスのための統合開発環境として、Micro Focus Visual COBOLの導入を決定しました。情報システム部 システム開発グループはほぼ全員がCOBOLの扱いに慣れたエンジニアです。「ACTION」ではUNIXシステム時代からCOBOL開発環境としてMicro Focus Server Expressを活用してきました。今回はその流れを継続する形でMicro Focus Visual COBOLを選択したことになります。

システム開発グループのエンジニアは、自身が受けたCOBOL教育について次のように語ります。「それまでプログラミング経験はありませんでした。配属されてすぐに一週間のCOBOL基本研修を受講、部に戻って業務に即した形でCOBOLプログラムに対する習熟を深めていきました」

同様にCOBOLを学んだシステム開発グループの別のエンジニアは、この言語の利点を次のように語ります。「コマンドが分かり易い英語なので、敷居が低く感じました。大量データや複雑な処理を安定的に処理できるのが

COBOLの良さだと思います」

彼らの先輩エンジニアは“複雑な処理”という点について、次のように補足しました。「損害保険の世界では事故の種類ごとに計算式が変わったり、多くの項目を設定してその項目にまた複数の条件をかけて計算するような方式を取っているため、どうしても分岐は複雑になります。プログラム言語はこれらをうまく扱う必要があります」

システム開発グループ主任エンジニアの一人はメンバーの発言をまとめてこう結論づけます。「テクノロジースキル管理の観点から、検収が必要とする言語の数を絞りたいという考え、同じ言語でシステムを維持していくのがよいという考えがあり、当グループではCOBOLを有力なプログラムと捉えています」

Results

「ACTION 4」の開発は現在鋭意進められており、本番稼働は来年の予定です。その次期システムでは、コストが半減、処理時間もモデルプログラムが1/5以下になったように大幅な短縮が予想されています。

システム開発グループの別の主任エンジニアは、新システムへの期待を次のように語ります。「本番環境と同レベルのテスト環境が確保できるようになるため、今後はここで実現したかったテストをすべて行い、今以上にプログラム品質の向上を図っていきます。また、SQL部分も我々が最も習熟しているCOBOLに集約されるので、何かエラーが発生した際にも一層迅速に対応できるようになると考えています」

ハードウェア更改を機に、守るべきは守り、変えるべきは変えるという方針で課題解決を果たす「ACTION」担当チーム。Micro FocusのCOBOL製品は、自動車保険参考純率算出の土台をサポートするシステムを長年にわたり支え続けています。

ユーザープロフィール

損害保険料率算出機構

本 部：東京都新宿区

設 立：1948年11月
(前身 損害保険料率算出会、
自動車保険料率算出会が統合)

従業員数：2,250名(2018年4月1日現在)

事業内容：参考純率および基準料率の算出・
提供、自賠責保険の損害調査、データバンク

<https://www.giroj.or.jp/>

記載の会社名、製品名は各社の商標または登録商標です。
本ユーザー事例の内容は、2018年8月に作成したものです。
MFJ710-1808-2MA | © 2018 Micro Focus. All rights reserved.

新システムの処理フロー概要

