システム資源間関連情報の自動収集と確認手段の構築～迅速な影響調査による開発生産性の向上を目指して～

Ｔ＆Ｄ情報システム（株）

■ 執筆者Ｐｒｏｆｉｌｅ ■

内藤　隆太郎

2018 年　現在 事業五部　営業システム課
サブマネージャー

竹内　翔吾

2018 年　現在 事業五部　営業システム課

重松　利弥

2018 年　現在 事業五部　営業システム課

■ 論文要旨 ■

生命保険業界を取り巻く環境は大きく変化し、ＩＴ技術を活用した保険会社の新たな競争がはじまっている。当社は開発サイクルの短縮化と開発ボリュームの拡大に対応するため、様々な生産性向上策に取り組んでいる。

当社開発部門の生産性向上策の一つとして、現状のシステム資源の影響調査作業（システム改修に伴い影響が及ぶ資源を特定する作業）の効率化を図ることで、開発工程全体の生産性向上が期待できた。現状の影響調査作業を分析した結果、３点の課題「資源ごとに異なる影響調査手順」・「既存の資源検索ツール機能の不足」・「複雑な資源に対する影響調査の属人化」があり、生産性の阻害要因となっていることが判明した。そこで、当社で利用するすべてのシステム資源間関連情報を自動収集し、検索結果を出力する「資源検索システム」構築により、課題を解決したことに加え、影響調査の効率化・生産性の向上を実現した。

■ 論文目次 ■

■ 図表一覧 ■

１．はじめに

Ｔ＆Ｄ情報システム株式会社（以下、当社）は、大同生命保険株式会社（以下、大同生命）・太陽生命保険株式会社・Ｔ＆Ｄフィナンシャル生命保険株式会社の３社を中核とするＴ＆Ｄ保険グループ唯一の情報システム会社として中核各社のシステム運用・開発を受託し経営戦略の実現に取り組んでいる。

生命保険業界は少子高齢化の進展、医療技術の進歩やライフスタイルの変化により、これまでの死亡保障だけでなくガンなどの重大疾病や就業不能時の生きるための保障など多様化したニーズへの対応が求められている。また、健康状態で加入後にも保険料が変動する生命保険や健康に関するビックデータをＡＩによりリスク分析したうえで、生命保険の引受を拡大するなどＩＴ技術を活用した保険会社の新たな競争がはじまっている。このような環境の変化に順応するため、生命保険各社は短期間に新商品やフィデューシャリー・デューティー（顧客本位の業務運営）に基づく新たなサービスの提供に取り組んでいる。

私たちが担当している大同生命は、中小企業を中心とする法人向け市場に特化した保険商品の販売をコアビジネスとしている。生命保険各社のサービス競争に対応するため、中小企業経営者の多様化した保障ニーズに対する新商品の開発や、健康サービス、高齢者対応、ビックデータの分析などデジタル開発に取り組んでいる。

当社においては、開発サイクルの短縮化と開発ボリュームの拡大に対応するため、平成２８年度より次の３つの生産性向上策に取り組んでいる。

（１）システム設計の全体最適化
　　類似する業務システムやプログラムを統合することで効率を上げる取り組み。
（２）プログラムのシンプル化
　　複雑化したプログラム構造を簡素で保守しやすいプログラムに見直す取り組み。
（３）開発環境の効率化
　　開発環境の複数化やシステム資源の影響調査を機械化することで各作業の省力化を図る取り組み。

本論文では「（３）開発環境の効率化」施策のひとつとして影響調査の生産性向上、品質の確保を目的に開発した「資源検索システム」における課題とその対策や工夫点について述べる。

２．影響調査における課題

要件分析工程から外部設計工程（以下、上流工程）における調査漏れや設計不備は下流工程の手戻りに繋がることがあるため、システム資源の影響調査は正確かつ迅速に対応することがポイントとなる。上流工程の作業のうち約６割を占める影響調査の作業効率を高めることで、開発工程全体の生産性向上が期待できた。

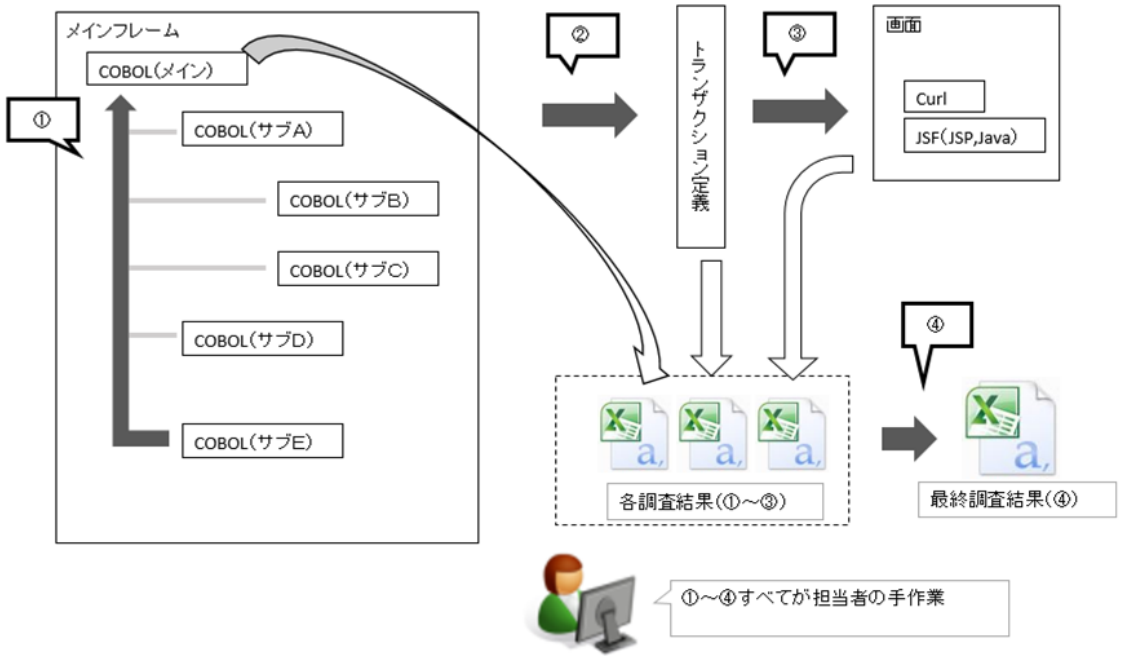
複数の開発部門の影響調査作業を分析した結果、以下３点の課題「資源ごとに異なる影響調査手順」・「既存の資源検索ツール機能の不足」・「複雑な資源に対する影響調査の属人化」があり、生産性の阻害要因となっていることが判明した。

（１）資源ごとに異なる影響調査手順

システム資源の影響調査で利用するツール・手順は、画面プログラム、帳票定義体、そしてビジネスロジックにあたるＣＯＢＯＬプログラムなど個々の資源ごとに異なっている。従来はＣＯＢＯＬプログラムの開発が中心であり、システム資源の種類が少ないことから、その資源に特化したツールを提供してきた。近年ではワークフローシステムなどのオープン系の開発により、システム資源種類が増加・複雑化し、調査担当者が利用するツール・影響調査手順の習得負荷が高くなってきた。

（２）既存の資源検索ツール機能の不足

各資源ごとに提供されている既存の資源検索ツールでは２種類までの関連するシステム資源の影響を把握できるが、３種類以上のシステム資源の関連を紐付けて検索する機能はなかった。このため各ツールの調査結果を人手で関連付ける作業が必要となり、調査に多くの時間を割くとともに、見落としなどのミスを誘発するリスクがある。現行調査の一例としてＣＯＢＯＬプログラムと関連する画面プログラムを特定する場合は、４段階の手順を踏む必要がある。当該作業プロセスを図１に示す。



＜現状の各資源間の関連調査プロセス＞

① ＣＯＢＯＬプログラムのサブプログラムから最上位のプログラムを検索

② 最上位のプログラムからトランザクション定義（＊１）を検索

③ トランザクション定義を使用する画面プログラムを検索

④ 上記①～④の各ツール検索結果を手作業で関連付け、最終調査結果を作成

＊１：画面プログラムからＣＯＢＯＬプログラムを呼び出す際の中間インターフェース

図１　現状の各資源間の関連調査

（３）複雑な資源に対する影響調査の属人化

実装時の自由度が高い画面プログラムは、度重なるプログラム変更や機能追加でシステム構造が複雑化しており、現行のツールでは検索が困難な資源があった。これらの資源の調査は、画面実装スキルの保有者がソースコードを分析する場面が多く、経験者の実装ノウハウ・スキルに依存していることが課題である。

３．資源検索システムの目的と概要

３．１　資源検索システムに求めること

影響調査における課題を解決するため、以下２点のあるべき姿を定義した。

・各資源で異なっていた調査手順を統一し単純化することで、ツールの習熟コストの低減と影響調査の効率化を図り、生産性・品質の向上を図る。

・調査担当者のスキルに依存せず、システム改修に伴う影響調査結果の品質を担保する。

上記のあるべき姿を実現するため、当社で利用するすべてのシステム資源間関連情報（画面で使われているトランザクション名、プログラム名など）を自動収集（以下、解析）し、検索結果をアウトプットする仕組みを構築する。資源検索システムを構築するにあたり、前述の課題への対応方針について、概要を表１にまとめる。

表１　課題と対応方針の概要

従来の課題	資源検索システムの対応方針
個々の資源ごとにツール・手順が異なり、利用負荷が高い。	すべての資源が検索可能な「資源検索システム」を構築し、手順の統一と単純化を図る。
各ツールの調査結果を人手で紐付ける作業が必要で調査負荷が高い。	３種類以上のシステム資源を関連付けする機能を追加することで、影響調査時間の短縮を図る。
複雑な資源の調査は経験者の実装ノウハウ・スキルに依存。	精度が高い解析機能を構築することで、正確な影響調査結果の出力を実現し、経験者のスキル依存脱却を図る。

３．２　システム概要

システム資源間関連情報を抽出するためには、機械的なソースコードの解析が必要となる。資源種類は 69 種類・資源数は数十万件存在するため、検索時にリアルタイムに資源を解析する方式は、品質面・性能面において現実的な方式ではなかった。

そのため、システム資源間関連情報を機械的に解析した結果をあらかじめデータベースに格納し、そのデータベースから検索する方式とした。資源検索システムの概要・機能の関連について表２および図２で示す。

表２　資源検索システムの概要

No	機能名	機能概要	開発技術
1	検索・表示機能 （オンライン）	○検索条件にもとづき、リポジトリから抽出したデータ（検索結果）を表示する。 ○表示された対象資源の選択により、システム資源の補足情報（バージョン情報・ソース・ﾄﾞｷｭﾒﾝﾄ）を表示する。	Excel VBA
2	解析機能 （バッチ）	システム資源を解析し、資源同士の紐付き情報を抽出。抽出結果はＣＳＶ形式で出力する。	Curl、Java
3	リポジトリ登録機能 （バッチ）	上記２の情報をもとに、リポジトリ（資源関連データベース）に対し登録（全件洗替え）する。	Oracle11g

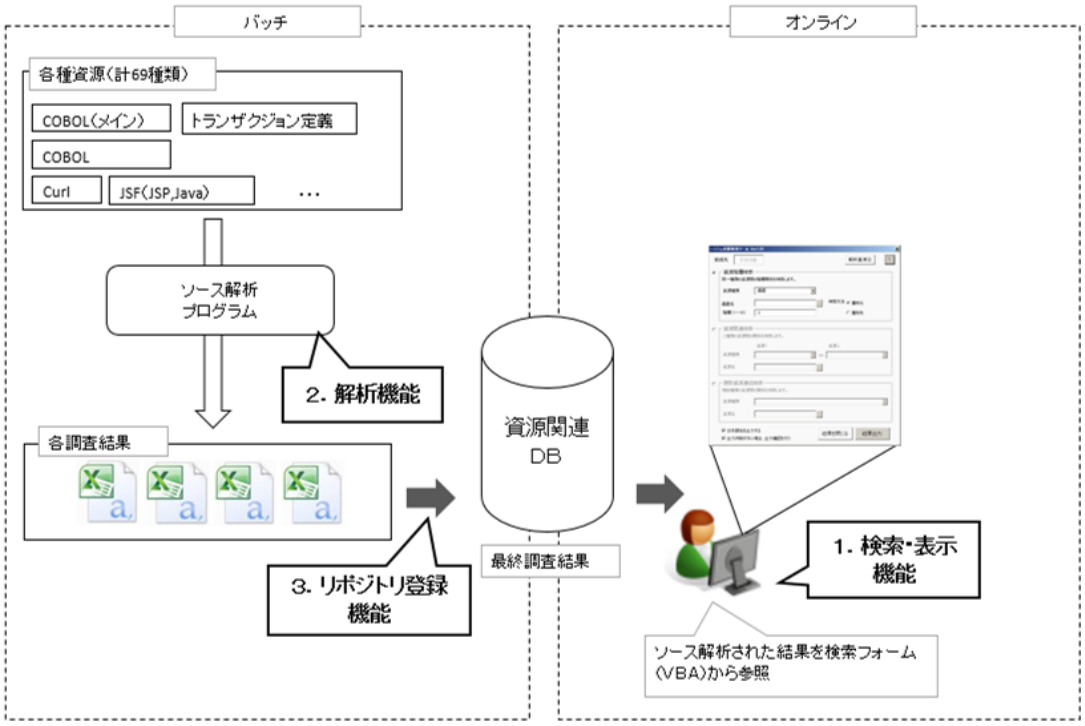


図２　資源検索システムの機能の関連

３．３　機能概要

当社 69 種類のシステム資源を対象に、生産性への貢献度が高い解析結果の確認手段を選定し、以下表３の機能を開発することに決定した。

表３　提供機能一覧

No	機能名	機能概要
1	検索・表示機能	
1-1	資源の単純検索	資源名および資源種類を入力・選択し検索する
1-2	資源間の関連検索	解析した結果（ソース情報、定義情報より資源間の関連性を抽出した結果）から、３種類以上のシステム資源などの関連付けを検索する
1-3	検索結果のＥｘｃｅｌ出力	Ｅｘｃｅｌに出力し調査結果として使用する
1-4	ソースビューア	検索結果の資源からソースファイルを起動する
1-5	ドキュメントビューア	検索結果の資源からドキュメントを起動する
1-6	変更履歴ビューア	検索結果の資源から変更履歴を起動する
2	解析機能	
2-1	静的情報の解析	機械的にプログラム文を特定し、資源名を抽出する（手続き型言語のＣＯＢＯＬプログラムなどに導入）
2-2	動的情報の解析	プログラム疑似実行の実行ログ（実行されたプログラム文の並び）から資源名を抽出（オブジェクト指向言語の画面プログラムに導入）

次章以降では、資源検索システムの根幹である検索機能および解析機能をどのように工夫して実装したかを詳細に述べていく。

４．開発内容と工夫点

４．１　検索機能

要件分析・外部設計の工程では、毎週定例会を設け、システムの利用者となるシステム部門（以下、ユーザー）に対し、検索機能を実装したプロトタイプ画面を確認し議論を交わしながら仕様を確定していく進め方とした。ユーザーインターフェース（以下、ＵＩ）に関する要望はプロトタイプに反映し、ユーザーでの操作確認を繰り返すことで、真にユーザー視点に適した仕様を反映させることができた。本章では、資源検索システムの機能のうち、ユーザーからの要望が多かった３機能の工夫点および、既存の資源検索ツールで課題となっていた検索応答性能確保のための対策を述べる。

４．１．１　ユーザーインターフェースの工夫点

（１）利用頻度の高い検索パターンの提供

ユーザーが検索したい情報の範囲を収集し、資源間の関連検索は３種類の検索機能提供で決定した。（図３を参照）

①　階層検索

単一種別のシステム資源について、最大 15 階層まで関連情報を出力する。呼び出しの親子関係が存在するＣＯＢＯＬプログラムや、画面遷移という形式で相互に関連しあう画面プログラムなどが対象となる。

②　関連検索

異なる種別のシステム資源同士の関連情報を出力する。画面プログラムとトランザクション定義の関連情報などが対象となる。

③　連結検索

３種類以上（②の範囲を拡大）の異なる種別のシステム資源同士の関連情報を出力する。ＣＯＢＯＬプログラム、トランザクション定義、画面プログラムの関連情報などが対象となる。ユーザーからの要望を収集し、連結検索対象の資源の組み合わせパターンを整理し、影響調査で主に利用する 10 種類のパターンで決定した。

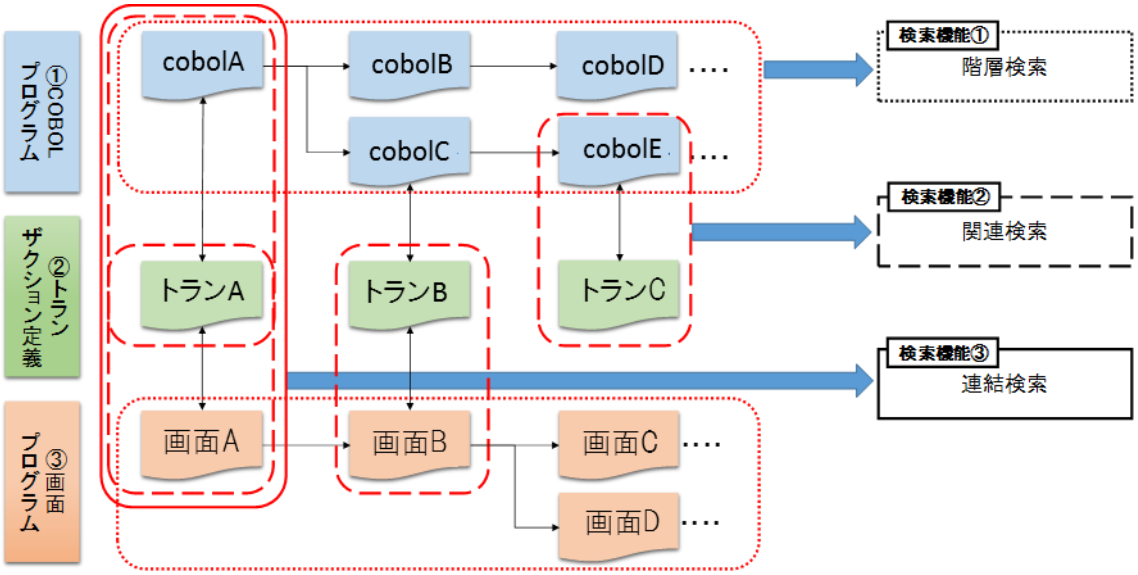


図３　資源間の関連検索機能

検索画面のＵＩは、ユーザーが検索条件を入力する前に、上記検索範囲を選択できるよう、検索条件の入力領域を３種類の領域に分けて、その領域を選択する画面構成とした。（図４を参照）また、ユーザーの誤入力を防止するよう、現在選択している領域以外は非活性とした。

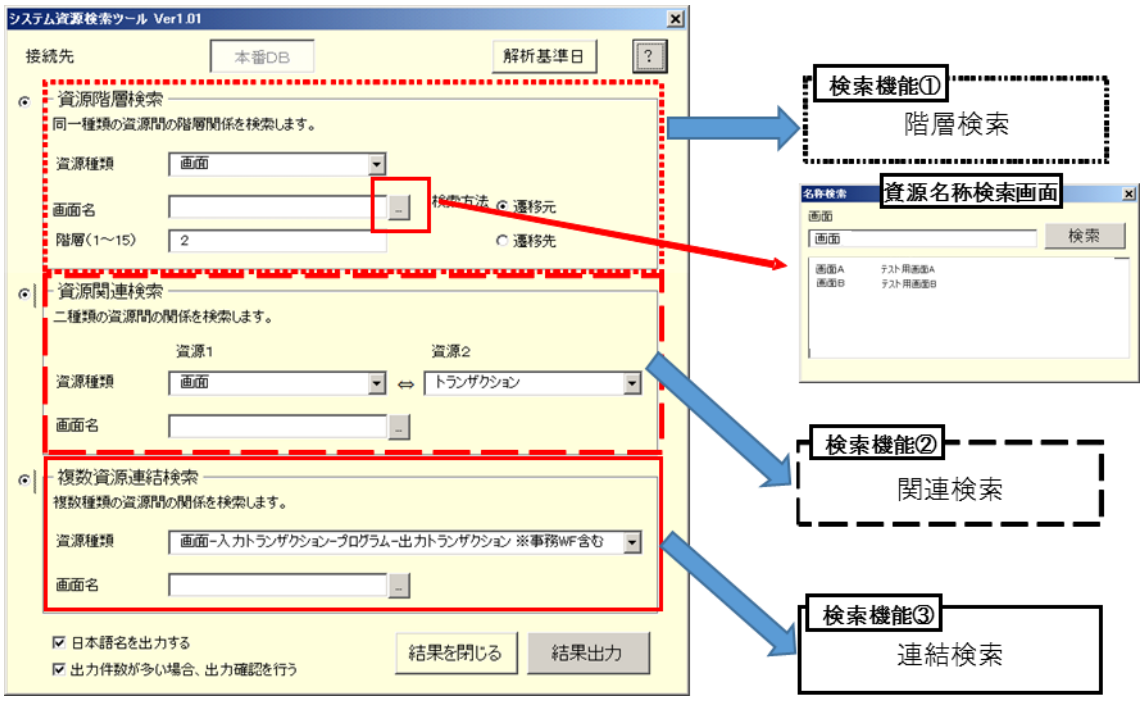


図４　資源間の関連検索画面

３種類の入力領域での共通の検索条件は、資源種類（ＣＯＢＯＬ・トランザクション定義・画面など）と資源名称からなる。ユーザーが具体的な資源名称を記憶していない状況でも、正しい資源名称が入力できるよう、入力補助機能として、資源名称検索画面から特定の資源を選択する機能と、あいまい検索機能を設けている。

（２）出力結果２次加工の容易さ

検索後の出力結果のインターフェースは、２次加工が容易なＥｘｃｅｌ形式とした。前述の資源間の関連検索機能からの出力結果イメージは図５のとおりとなる。この機能を活用することで、現状の各資源間の関連調査の手順を短縮（図１のプロセス①～④を４段階から１段階に短縮）することができた。

No.	区分	プログラム名	プログラム名(日本語名)	トランザクション名	トランザクション名(日本語名)	画面名	画面名(日本語名)
1	プログラム	ProgramA	プログラムA				
2	トランザクション			transactionA	トランザクションA		
3	Curl画面					GAMENA	画面A
4	Curl画面					GAMENB	画面B
5	Curl画面					GAMENC	画面C
6	Curl画面					GAMEND	画面D
7	Curl画面					GAMENE	画面E

図５　検索内容の出力結果

また、システム資源Ａとシステム資源Ｂが相関しているという情報だけではなく、Ａのどの部分とＢが紐づいているか把握できれば影響調査の一助につながる。画面間の関連情報を検索した場合、検索結果の補足情報欄（図６参照）に対し、画面遷移処理の実装箇所（イベント・メソッド）を出力している。このように、詳細情報を検索結果に出力することで、影響調査の更なる効率化を図っている。

No.	画面	画面(日本語名)	遷移先	遷移先(日本語名)	補足情報
1	画面A	遷移元画面A	画面1	遷移先画面1	イベント:BUTONA_Action ハンドラ:handlerA
2	画面B	遷移元画面B	画面2	遷移先画面2	イベント:BUTONB_Action ハンドラ:handlerB
3	画面C	遷移元画面C	画面3	遷移先画面3	イベント:BUTONC_Action ハンドラ:handlerC
4	画面D	遷移元画面D	画面4	遷移先画面4	イベント:BUTOND_Action ハンドラ:handlerD
5	画面E	遷移元画面E	画面5	遷移先画面5	イベント:BUTONE_Action ハンドラ:handlerE
6	画面F	遷移元画面F	画面6	遷移先画面6	イベント:BUTONF_Action ハンドラ:handlerF

図６　補足情報のイメージ

（３）ソースビューア機能との連携

検索結果（Ｅｘｃｅｌ出力）の資源名を選択することで、本番環境または開発環境で動作している最新のソースファイルを閲覧することができる。（図７を参照）

システム導入以前は、ソース管理サーバから全ソースを取得し、変更履歴から本番または開発環境で動作する資源の抽出作業が必要であったが、導入後は、検索結果の資源名を選択する操作のみで、最新のソースファイルを閲覧することができるので、調査時間短縮につなげることができた。

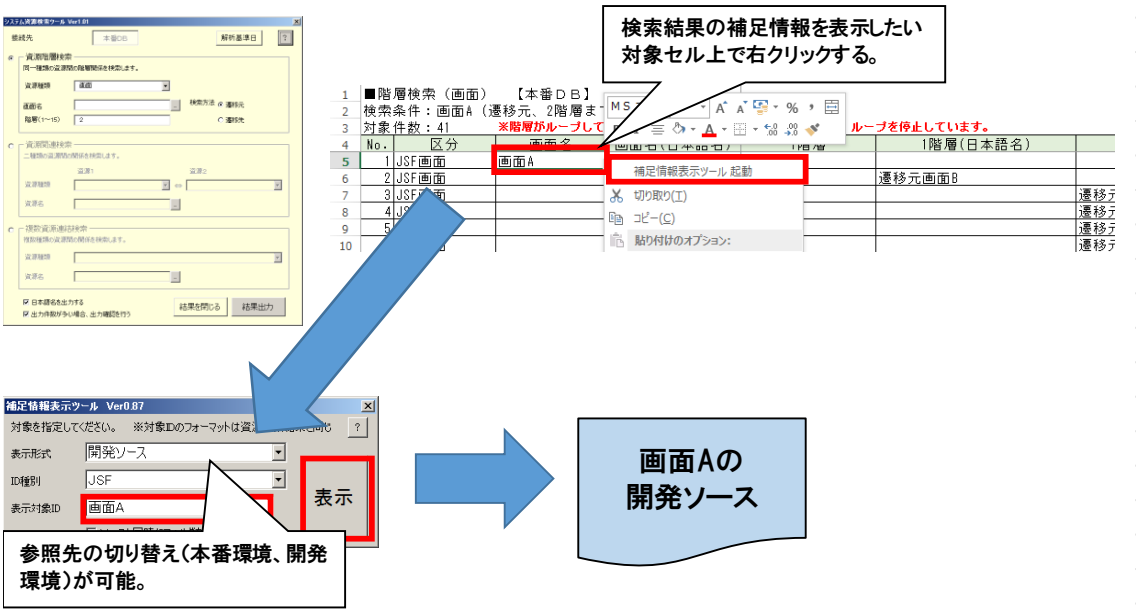


図７　ソースビューア機能の操作イメージ

４．１．２　検索性能確保に向けた対策

既存の資源検索ツールは、資源の階層数に比例して検索性能が劣化するケースがあり、性能面の改善を求める要望は多かった。影響調査結果を迅速に閲覧できるよう、資源検索システムの検索性能確保の対策を講じた。検索性能の目標値は、５秒以内で約千件の検索結果を表示することを目標とし、データ取得処理・データ出力処理の２点の性能改善に取り組んだ。

（１）データ取得処理の性能対策

データ取得処理は、Ｏｒａｃｌｅ１１Ｇから使用可能である階層ＳＱＬの技術を利用することで性能問題の解決を図っている。階層ＳＱＬの技術を利用することで大きく変わったことは、解析結果を登録するデータベース構造である。既存のデータベース構造は、システム資源間のすべての関連情報を保持する「経路列挙型」であった。一方、今回は単純な２つのデータの組み合わせによりデータを保持する「隣接リスト型」とした（図８参照）。階層ＳＱＬから、隣接リスト型で保持されているこれらのレコードを動的に紐付けることでデータベース内の効率的な検索を実現している。さらに、隣接リスト型は、経路列挙型と比較しレコードの挿入や更新の際に発生する処理量が軽減されるため、データベース検索と登録において従来よりも性能向上が見込めた。

経路列挙型

親ID	自分ID	
（なし）	A	←
A	B1	←
A→B1	B2	←
A	C1	←
A→C1	C2	←
A→C1→C2	C3	←

隣接リスト型

親ID	自分ID	
（なし）	A	←
A	B1	←
B1	B2	
A	C1	←
C1	C2	
C2	C3	

※Ａが変更される場合に修正が必要な対象は色付背景のレコード。経路列挙型は修正が全レコードに及ぶのに比べ、隣接リスト型では３レコードに収まっている。

図８　データベース構造の違い

当技術の導入により、既存ではタイムアウトしてしまうような検索条件においても、今回はタイムアウトすることなく検索時間を大幅に向上できた。既存の資源検索ツールと資源検索システムの性能比較を表４に示す。同一の検索条件で階層検索を行い、検索実行から検索結果取得までの時間を比較した。既存のツールは、資源の階層数とデータ件数の増加に比例して性能が劣化しているが、資源検索システムで性能面の改善を図ることができた。

表４　性能比較表

No	テストコード	階層	件数	検索所要時間（秒）	
				既存	導入後
1	プログラムA	15	7176	タイムアウト	6.30
2	プログラムB	5	233	12.21	1.38
3	プログラムC	3	30	4.98	1.58

隣接リスト型のデータ生成イメージを図９に示す。ある複数のシステム資源Ａ～Ｃ３がお互いに相関している場合、それぞれの関連情報は隣接リスト型のテーブルに親子の関係で登録される。

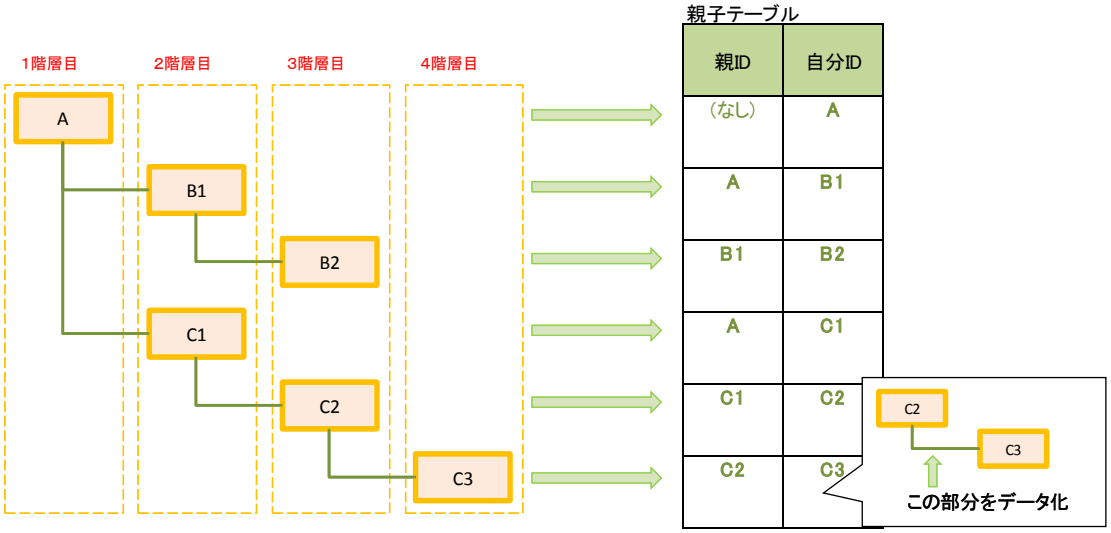


図９　隣接リスト型のデータ生成イメージ

（２）データ出力処理の性能対策

既存の資源検索ツールはＷｅｂシステムであるため、大量のデータを表示する際に、ｈｔｍｌ描画に時間を要する問題があった。大量データ出力の性能向上を図るため、今回はＥｘｃｅｌＶＢＡの高速なデータ出力技術を用いて、取得した値を加工せずＥｘｃｅｌへデータ展開する方式とした。図１０は、階層検索の取得結果をＥｘｃｅｌに出力したイメージである。階層ＳＱＬで取得した階層・資源情報（左図）をもとに、Ｅｘｃｅｌへデータ展開する際は、列単位に階層と資源を出力する形式（右図）とした。

＜階層SQLで取得したレコード＞

階層	区分	プログラム名
0	本番	プログラムA
1	本番	プログラムB
2	本番	プログラムC
3	本番	プログラムX
1	本番	プログラムD
2	本番	プログラムE
3	本番	プログラムF
3	本番	プログラムG
2	本番	プログラムH

＜アウトプットフォーマットに沿って展開した出力結果＞

No.	区分	プログラム名	1階層	2階層	3階層
1	プログラム[本番ライブラリ]	プログラムA			
2	プログラム[本番ライブラリ]		プログラムB		
3	プログラム[本番ライブラリ]			プログラムC	
4	プログラム[本番ライブラリ]				プログラムX
5	プログラム[本番ライブラリ]		プログラムD		
6	プログラム[本番ライブラリ]			プログラムE	
7	プログラム[本番ライブラリ]				プログラムF
8	プログラム[本番ライブラリ]				プログラムG
9	プログラム[本番ライブラリ]			プログラムH	

図１０　データ取得結果をＥｘｃｅｌに出力したイメージ

４．２　解析機能

解析精度が現状より低い精度である場合、検索結果に対する信頼性が損なわれる。そこで、精度が高いプログラム解析手法を確立し、その手法にもとづき自動解析プログラムを構築した。さらに高い正確性が維持されるよう、開発規約が遵守される仕組みを構築することとした。

４．２．１　高い正確性確保に向けた解析手法の確立

システム資源間の関連する情報は、以下①・②のプログラム文指定の資源名を抽出することで求められる。

① 資源名を定数で指定した資源の呼び出し処理を実装したプログラム文

② 資源名を変数で指定した資源の呼び出し処理を実装したプログラム文

既存の資源検索ツールにおいて、①の資源名の抽出は定数であるため特定が容易である。しかし、②の資源名の抽出は製造物ごとに「変数へ資源名を代入するプログラム文」の定義が異なるため、複雑性が高い資源は検索できなかった。そのため、これらの資源の影響調査は、実装スキルを保有する経験者に依存することが多かった。

これらの課題を解決するため、開発する自動解析プログラムは、静的情報の解析または動的情報の解析の２点の手法により構築することで、効率的な自動解析と正確な解析結果の出力を実現した。

（１）静的情報の解析手法

機械的に上記①・②のプログラム文を特定し、資源名を抽出する解析手法であり、手続き型言語のＣＯＢＯＬプログラムなどに導入した。②の抽出処理は、資源の難易度・複雑度に応じた専用の処理を構築する必要がある。そのため、難易度・複雑度ごとに分類した資源単位で、自動解析プログラムの実行およびチューニングを繰り返すことで、段階的に自動解析プログラムの精度を向上させた。また、難易度・複雑度が高いプログラムは、実際の動作を確認することで、自動解析プログラムの解析結果の正確性の裏付けをとった。

（２）動的情報の解析手法

疑似的にすべてのプログラムを実行し、その実行ログ（実際に実行されたプログラム文の並び）を解析する手法であり、プログラム構造の複雑性が高い画面プログラム（Ｊａｖａ）に導入した。オブジェクト指向言語には継承などの概念が導入されており、実行時に決定される要素が多く含まれるため、静的な解析手法では正確な解析が困難である。そのため、画面クラスを疑似的に実行するリフレクション機能（Ｊａｖａ標準ライブラリ）を利用して、出力された画面の実行ログから、呼び出した資源（画面やトランザクション定義など）を特定することで、正確な解析結果を得ることができた。

４．２．２　開発規約が遵守される仕組みの整備

効率的かつ高品質の保守開発作業のため、当社では統一的な開発規約を周知徹底している。ただし、複雑性が高い画面資源においては、遵守すべき規約が多岐にわたるため、規約に沿わない資源が今後発生する可能性があり、これらの資源を正確に自動解析することは難しいと考えられた。

そこで、画面開発環境・ツールに開発規約の遵守をチェックする機能を組み込んだ。具体的には、当機能が画面プログラムのコンパイル時に、コーディング規約（約 200 個）をチェックし、開発規約に準拠していないソースコードをコンパイル不可とした。これにより、開発担当者に開発規約を遵守させ、画面の保守性を向上させるとともに、資源検索システム解析の正確性を維持することができた。

５．導入効果

システム開発時より真にユーザー視点に適した機能を提供できるよう工夫し「高い開発・保守生産性」の実現を図った。資源検索システム導入効果は以下のとおり。

（１）影響調査における時間短縮・調査工数の削減

資源検索システムを導入することにより、最も多い影響調査のパターン（以下の例を参照）において、約 30 分要していた調査時間が約３分に短縮された。

＜システム導入前後の影響調査プロセスの比較（例）＞

○現状の各資源間の関連調査プロセス（所要時間約 30 分）

① ＣＯＢＯＬプログラムのサブプログラムから最上位のプログラムを検索
② 最上位のプログラムからトランザクション定義を検索
③ トランザクション定義を使用する画面プログラムを検索
④ ①～④の各ツール検索結果を手作業で関連付け、最終調査結果を作成

○システム導入後の各資源間の関連調査プロセス（所要時間約３分）

① ＣＯＢＯＬプログラムの最上位のプログラムを検索条件に連結検索機能を利用し、最終結果を出力

（２）調査の正確性向上によるスキル依存の脱却

資源検索システムにより機械的に調査結果を出力することで、調査担当者のスキル不足による調査漏れ、誤りがなくなる。また、従来はシステム資源の影響調査で利用するツール・手順が個々の資源ごとに異なっていたが、資源検索システムでは共通のインターフェースで調査結果が出力されるため、調査担当者のスキル依存から脱却できた。

（３）その他（定性評価）

資源検索システム導入後に集まったユーザーの声を定性評価としている。意見内容を集約すると以下のようになり、ユーザーに受け入れられていると考えられる。

・既存の資源検索ツールと比べ数秒で結果が返りストレスが無い。
・ＵＩがシンプルであるために直感的に操作できる。
・不定期な利用でも操作方法を一度覚えれば再度覚えなくて済む。
・すべての検索結果のアウトプットがＥｘｃｅｌで統一されているので利用しやすい。
・検索結果から資源のソースがすぐ閲覧できるので便利である。

６．今後に向けて

2018 年３月にシステムリリースを行い、現時点で活用しながら引き続きチューニングおよび仕様改善を実施している。

以下の取り組みについて、随時対応可能なものから実施しリリースを行っている。今回の開発のように都度、プロトタイプで確認を行いユーザー要望に沿っているか確認しながら開発を進めているため、現在も認識相違などの手戻りなく進捗している。

＜チューニング、仕様改善（例）＞

① ＵＩ：階層検索の階層数の拡大、検索結果表示項目の追加

② 性能：表示レスポンスの更なる改善

７．おわりに

資源検索システム導入後、一定の評価を得られたことは開発メンバーおよびユーザーの協力の結果だと考えている。開発に関わった全員に感謝したい。また、導入後に寄せられたユーザーからの声は肯定的な意見がほとんどであったことは非常に喜ばしかった。声に出さずともこれまでの影響調査に苦労していた開発者が多いこともわかり、今回の開発は効果あるものだったと実感することができた。

しかしながら、今回の開発内容からのチューニングを行わなければ何十年後には開発にまつわる環境も大きく変わり、後世の開発者に今回と同じ課題を与えてしまう。常に「高い開発・保守生産性」を実現できるように、今後も引き続き資源検索システムの質を向上させていきたい。

今後ともさまざまなシステムサービスの実現を通じて、将来にわたって大同生命のお客さまに「最高の安心」と「最大の満足」をお届けし続けたい。

８．参考文献

[1]OracleDatabaseSQL 言語リファレンス（階層問合せ）
https://docs.oracle.com/cd/E16338_01/server.112/b56299/queries003.htm