

# ソースコード解析に関する実践

## —公的年金財政検証プログラムの解析と応用—

石田 朗大<sup>†1</sup> 寺西 教高<sup>†1</sup> 日隈 美奈子<sup>†1</sup>

<sup>†1</sup> (株) シーエーシー

設計書や仕様書などのドキュメントが十分に整備されていないシステムについて、その利用や改修を行うためにソースコードの解析、ドキュメントの整備を行う場合、目的によっては必ずしもシステム全体を解析する必要がない場合も多い。課題に対してシステムの必要な部分のみを解析するといったことは、実務ではよく行われるが、その具体的な手法について体系的にまとめられたものはなかなか見受けられない。本稿では、公的年金財政検証プログラムを対象として、いかにして短期間でソースコードの解析を行い、プログラムの改修を行ったかについて、その実践事例を示す。設計書や仕様書を網羅的に作成するのではなく、目的に合った解析を集中的に行うことで、プログラム解析の工数削減を行うことができた。

### 1. はじめに

長期にわたって運用されているシステムでは、設計書や仕様書などのドキュメントが十分に整備されておらず、ソースコードを解析しながら利用・改修している場合が多い。そのため、ソースコードを自動的に解析して設計書を生成するためのソフトウェアやそのようなサービスも多く提供されている。しかしながら、実際には、システム全体についての解析は必ずしも必要でないこともあり、その場合、ブラックボックス化しても問題ないモジュールと詳しく解析すべきモジュールを区分した上で、後者にリソースを集中する方法は極めて有効である。

厚生労働省の公的年金財政検証プログラム（以下、「財政検証プログラム」と呼ぶ）は、そのソースコードがすべて公開されており、前提条件を変更した場合の試算などを理論的には実施することが可能である。しかしながら、ソースコードのみが公開されており、設計書や仕様書、利用手順等は公開されていない。さらに、前提条件の変更にはソースコード自体にも手を入れる必要がある。そこで（一社）年金総合研究所では、財政検証プログラムを解析し、主に研究者を対象として、研究ニーズに合ったより利用しやすいツールを開発することを目指した研究会（年金財政シミュレーションツール開発研究プロジェクト）を2013年4月に発足させた。筆者らは、この研究会にシステムの専門家として参加しており、プログラムの解析と機能追加のための改修を主として担当している。この公開されている財政検証プログラムは、まさに、「設計書や仕様書などのドキュメントが十分に整備されておらず、ソースコードを解析しながら利用・

改修している」典型例となっている。この年金財政シミュレーションツールの機能については、稲垣、小澤、寺西、日隈、石田[1]が発表しているが、そのためのソースコードの解析やドキュメントの整備方法といった実務については触れていない。

本稿では、このツールの作成過程において、いかに財政検証プログラムの全体像を把握し、詳しく解析すべきモジュールを抽出・解析して短期間での機能追加のためのプログラム改修にこぎつけたか、システムの専門家の視点からその手法について述べる。第2章では、財政検証プログラムの概要と特徴を示し、第3章では、具体的なソースコードの解析方法について述べるとともに、作成したドキュメント類を示す。最後に、第4章において、この方法論に関する評価と類似の他業務への応用などについて考察する。

以降、財政検証プログラム全体を「プログラム」、5つのパートに分かれている各部を「サブシステム」、各サブシステムに含まれる関数やサブルーチンを「モジュール」と呼ぶこととする。

### 2. 公的年金の財政検証プログラム

#### 2.1 公的年金の財政検証とは

昨今、少子高齢化の進行する社会において、公的年金に対する不安感は大きく、若い人の間では、自分は将来十分な年金を受給できないのではないかと、あるいは年金制度が破たんするのではないかなどといった懸念が広まっている。こうした懸念を払しょくし、公的年金制度を健全に運営していくことは政府としての責務であるこ

とから、公的年金制度を所管している厚生労働省では、5年ごとに公的年金に関する財政シミュレーションを実施し、長期的な財政の持続可能性の確認を行っている。これを「財政検証」と呼んでいる。

この財政検証を行うプログラムは、複雑な年金制度を反映し、複雑で大規模なものとなっている。我が国の公的年金制度は、基礎年金と報酬比例年金からなる2階建ての仕組みであり、職業によって加入する制度が異なっている。具体的には、厚生年金保険のほか共済組合や国民年金など多くの制度に分かれており、年金の給付要件や年金額の算定式、保険料水準なども多様である。これらの制度は、基本的には一元化される方向ではあるが、過去に存在していた制度における加入状況が将来の年金額に反映されること、制度間での財政調整の仕組みも存在することなどから、財政検証のためのシミュレーションでは、数多くの種類の入力データに基づいて、複雑で大規模な計算を行う必要がある。

このプログラムはかつては公開されていなかったが、年金制度の研究者の間にあった計算結果に対する不信感を解消する観点などから、だれでも計算が再現できるよう、2004年財政再計算から一般に公開されるようになった。直近の2009年財政検証のためのプログラムのソースコードと基礎データは、Webサイト上[2]に公開されている。また、いわゆる設計書や仕様書は公開されていないが、財政検証の計算手順の概要や主要な計算式などは、「平成21年財政検証結果レポート」[3]としてまとめられている。しかし、厚生労働省の検証結果の再現は可能であるが、入力データや出力データの仕様が不明であることなどから、前提条件を変更したシミュレーションの実行や、出力結果の検証をするなどといった、研究者が望むような利用しやすいプログラムにはなっていない。

## 2.2 財政検証プログラムの概要

公的年金の財政構造は図1に示すとおり、「国民年金勘定」、「厚生年金勘定」、「基礎年金勘定」の3つの勘定から構成されており、それぞれの勘定の収支見通しが政府の財政検証結果として公開されている。しかしながら、「基礎年金勘定」は基礎年金給付費の負担を国民年金勘定と厚生年金勘定で按分するために設けられものであり、財政検証プログラムでは、独立したサブシステムとせず、国民年金の収支計算の一部として取り扱われている。

一方、基礎年金給付は、全国民共通のルールで算定さ

れる給付であり、基礎年金勘定を通して一括して支払われるものであるが、その給付費の推計は、1985年改正前の制度に対応した旧国民年金グループ（現在の区分では第1号・第3号被保険者）と旧厚生年金グループ（同第2号被保険者）に分けて計算されている。具体的には、前者は「国民年金の給付費推計」、後者は「厚生年金の給付費推計」のサブシステムでその計算が行われている。いわば、銀行合併が行われたときに、全体のシステムを再開するのではなく、2つのシステムを併存させ、ブリッジ接続する形と類似した構造となっている。そのため、サブシステム間でのデータの受け渡しが多くなっている。

財政検証の計算過程は、図2に示すとおり、まず、全

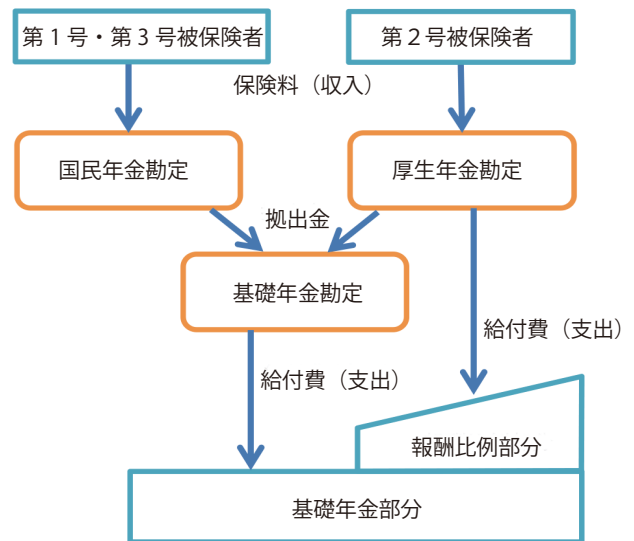


図1 公的年金の財政構造

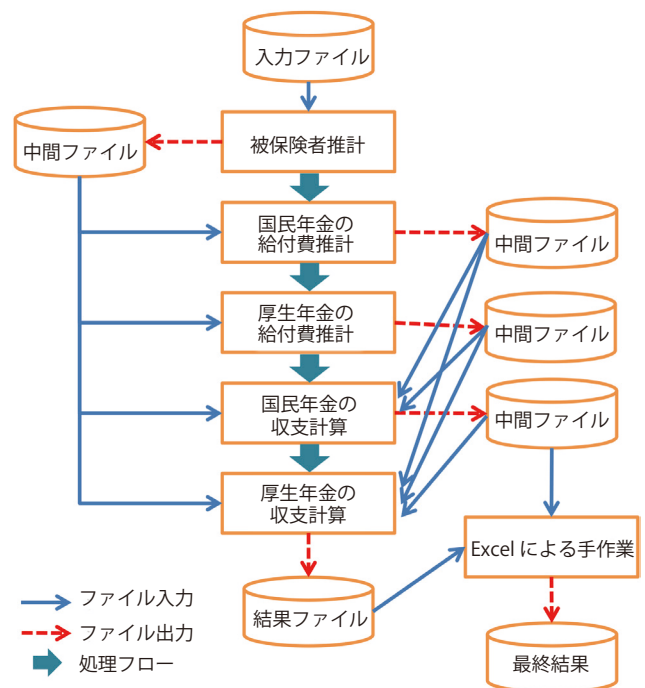


図2 財政検証プログラムのサブシステム関連図

体の基礎となる被保険者推計を行った後、国民年金と厚生年金に分けて給付費が計算される。前述のように、厚生年金の給付費には基礎年金相当部分が含まれることから、この給付費推計結果が中間ファイルを通して国民年金の収支計算に引き継がれ、基礎年金勘定の按分ルールに従って、厚生年金勘定と国民年金勘定の拠出金負担額が、「国民年金の収支計算」内で計算される。

国民年金勘定では、この拠出金負担額と国民年金独自の給付<sup>☆1</sup>が支出であり、保険料や国庫負担などが収入であるが、これらの収入と支出に基づいた収支計算が「国民年金の収支計算」で行われる。厚生年金勘定では、同様に、拠出金負担額と厚生年金独自の給付（報酬比例年金など）が支出であり、保険料や国庫負担などが収入であるが、これらの収入と支出に基づいた収支計算が「厚生年金の収支計算」で行われる。

各サブシステムは、図2に示す順番で実行する必要があるが、サブシステム間でのデータの受け渡しが多いことがこのプログラムの1つの特徴である。基本的に、後続のサブシステムは、先行するすべてのサブシステムの結果を入力データとして利用している<sup>☆2</sup>。この膨大な受け渡しデータの仕様が示されていないことが、プログラムを解析する際のポイントの1つであったが、上記のように公的年金の財政構造と財政検証プログラムの構造が事前に分かっていたことから、重点的に解析すべきポイントの絞り込みを容易に行うことができた。

プログラムはUNIX上で動作するFORTRANまたはC++で記述されており、ステップ数や入出力ファイル数などは表1に示すとおりである。入出力ファイルは、ほとんどがCSV形式のテキストファイルであるが、総サイズが約1GBにおよぶ膨大なデータを含むファイルとなっている。ただし、これらのデータのうち、後続のプログラムで利用するデータや結果として利用しているデータは一部に限られる。さらに、プログラムの改修を繰り返すうちに、不要となったソースコードが消去されないうまま残されているものもあった。

また、国民年金の収支計算など一部のシミュレーションでは、このサブシステムから最終的な結果が得られるわけではなく、エクセルのワークシートを利用した追加の手作業が必要となっていた。

☆1 付加年金や寡婦年金などがある。ただし、全体の給付費から見ると少額であるので、図1の公的年金の財政構造のフローでは省略している。

☆2 厚生年金の給付費推計サブシステムのみ、先行する一部のサブシステム（国民年金の給付費推計）の出力ファイルを利用していない。

表1 サブシステム一覧

サブシステム名	モジュール数	総ステップ数	入力ファイル数	出力ファイル数
被保険者推計	29	4,777	60	35
国民年金の給付費推計	32	8,760	18	9
厚生年金の給付費推計	66	15,348	80	33
国民年金の収支計算	30	4,958	6	10
厚生年金の収支計算	69	10,516	18	3
合計	226	44,359	182	90

## 2.3 年金財政シミュレーションツール開発研究プロジェクトの課題

筆者らが参加している年金財政シミュレーションツール開発研究プロジェクトでは、財政検証プログラムを容易に再現できるようにするほか、経済前提や出生率などの前提条件を変更したときのシミュレーション、年金の支給開始年齢の引上げなど制度改正を行ったときの財政影響試算などが課題とされた。しかしながら、これらの課題のうち、制度改正を行った場合のシミュレーションを行うためには、計算ロジックに踏み込んだプログラムの修正が必要なほか、2014年には厚生労働省がこのプログラムの最新版を公開することが予定されていたため、当プロジェクトの作業の手戻りを防止する観点などから、当面、計算ロジックには踏み込まず、以下の4つを課題として設定した。これらの課題については、システムの専門家である筆者らが担当することとなった。

### (1) シミュレーション実行の簡易化

財政検証プログラムは、5つのサブシステムごとに、コマンドラインから各種パラメータを入力しながら順次実行していく必要がある。利便性の向上のため、ワンクリックで実行可能なインターフェースの開発を課題とした。

### (2) 出生率や死亡率を変更できる機能の追加

主要な計算前提である出生率、死亡率について、利用者が作成した任意のファイルをもとにシミュレーションを実行できる機能の開発を課題とした。

### (3) エクセルによる出力帳票作成機能の追加

財政検証プログラムの出力はすべてCSVファイルのため、財政検証結果レポートに掲載されている結果表と照らし合わせるためには、手動で結果整形を行う必要があった。そこで、財政検証結果レポートの結果表と同じ様式での帳票を作成する機能の開発を課題とした。



#### (4) 公開されていないデータを出力する機能の追加

年齢階級別の年金受給総額や受給者数等は、最終結果の計算のためにプログラム内で計算はしているが、出力結果としては出ておらず、その値も公開されていない。そこで、プログラム内のその計算をしている変数から値を取り出し、結果として出力する機能の追加を課題とした。

### 3. 財政検証プログラムの解析

#### 3.1 解析の目的と手法

プログラムを解析する手段として、ソースコードを自動的に解析して設計書を生成するソフトウェアやサービスが提供されている[4],[5]。このようなソフトウェアやサービスの目的は、主に仕様書のないレガシーコードの流用や保守のためであり、各関数のロジックや呼び出し構造の解析までをプログラム全体で行っている。

これに対して、本プロジェクトにおけるプログラムの解析の目的は、利便性の向上や中間結果の取得、あるいは、年金制度の仕組みを一部変更したときの財政影響評価のためのプログラム修正である。

財政検証プログラムは入出力ファイルの数が膨大であることから、その分析が当プロジェクトの鍵であると判断し、まずファイル一覧兼CRUD図の作成を行った。その結果、サブシステム単位では生成と参照のみで、更新や削除は行わず、ファイルの数は多いがプログラム自体はシンプルな構造であることが分かった。

プログラム構造やファイルの入出力が複雑に入り組んでいるシステムの場合、プログラム修正の対応開始後に想定外の問題・調査漏れ等のリスクを避けるため、プログラムの前後関係にも十分に留意した広範囲の解析が必要となってくる。しかし、財政検証プログラムの場合、おむねプログラム全体の構造や仕組みは事前に理解できており、またファイルの入出力やプログラムの構造がシンプルであることが事前調査で判明したこと、および、すでに公開されているプログラムと同じ結果が得られるという検証が可能であることから、前述のようなリスクはほとんどないと判断した。そこで、個々の課題に応じてキーワードを設定し、原則として、そのキーワードを1つも含まないモジュールについては解析対象外と判断した。

以上より、それぞれの課題の内容に合ったモジュール、変数の絞り込みを行い、必要なモジュールを集中的に解析し、それ以外の部分をブラックボックスとして取り扱

うことにした。この作業により、モジュールや変数の解析工数を大幅に削減することができた。

そしてこれらの解析内容を元に、ユーザビリティの向上と機能の追加などを行うことで、プログラムの改修作業を効率的に行うことができた。

#### 3.2 プログラム解析の実践

##### 3.2.1 入出力ファイルの整理

財政検証プログラムは、サブシステム間で受け渡しをする入出力ファイルの数が多くだけでなく、データ量が膨大であること、後続のサブシステムでは利用しないデータが各ファイルの先頭の部分に含まれていることがあること、CSVのテキスト形式で数字の羅列になっているものが多いこと、ネーミングルールが一律でないことなど、入出力ファイルを見ただけでは、どのようなデータのファイルか判別することが困難なプログラムとなっている。さらに、入力ファイル名が、サブシステムのモジュールに直接記述されているケースがあるなど、慎重に入出力ファイルの解析を進めていく必要があった。

入出力ファイルは全部で234個あり、サブシステムごとに入力・出力されるファイルの数は表1のとおりである。

まず、入出力ファイル名の一覧を作成し、5つのサブシステム名をタイトルに持つ列を追加した。そして各サブシステムで入力されるファイルには「In」、出力されるファイルには「Out」と記載することで、入出力の関係を示したマトリックスであるCRUD図をエクセルで作成した(図3)。通常のCRUD図では「生成(Create)」、「参照(Read)」、「更新(Update)」、「削除(Delete)」の4つを用いるが、財政検証プログラムではファイルの更新と削除は行わないため、参照と生成のみですべてを表現することができ、ここでは、参照を「In」、生成を「Out」と記載した。

このCRUD図により、サブシステムと入出力ファイルの関係が明確になり、またエクセルのフィルタ機能を使

ファイル名	(1) 被保険者 推計	(2) 給付費 推計 の 年金	(3) 給付費 推計 の 年金	(4) 国民 年金 の 計算	(5) 厚生 年金 の 計算
waku1000-00.csv	Out	-	-	-	-
waku1000-01.csv	Out	-	In	-	-
DOKUZI400-[ECON]-1000-1830-02.csv	-	Out	-	In	-
KOKUKAITE-[ECON].csv	-	Out	-	In	In

図3 CRUD図の作成例

い、解析対象とする入出力ファイルやサブシステムに絞り込んだ入出力関係を閲覧できるようにした。

たとえば、被保険者推計サブシステムの列で「Out」をフィルタリングすることで、被保険者推計サブシステムで出力されたファイルがその後のサブシステムで入力されているかを簡単に検索することが可能になった。そして、入出力ファイル内にヘッダがあるものはその内容などを参考に、簡単にファイル内容を記載した。

CRUD図の必要性については、よく語られ、また理解されているはずであるが、実際の開発現場ではシステム規模が大きくなるほど、作成されていないケースが散見される。本プロジェクトでは入出力ファイルの整理、解析が重要であったため、ここで作成したCRUD図はこの後の解析で大変有効に活用できた。

CRUD図を作成したことで、以下のことが分かった。

- 234個のファイルのうち、約60%にあたる144個は、出生率や人口推計といった計算に用いる基礎率のファイルであり、入力として使われているのみである。
- 90個の出力ファイルのうち51個は、出力されるだけで後続のサブシステムに入力されない。
- ファイルが出力される個所が複数のサブシステムにまたがることはなく、ファイル出力を行う関数もまとまっている。

以上のように、ファイルの数は多いが、そのファイルを入出力する構造については非常にシンプルであることが分かった。

### 3.2.2 サブシステムの解析

次に、すべてのモジュールのリストを作成し、それらのモジュールを利用しているサブシステムを整理した。そして、各モジュールの内容について、その内容を表すと考えられるコメントをリストに追加した。入出力ファイルと同様にネーミングルールが一律でないため、モジュール名はほとんど参考にならなかった。このリストは、各モジュールに1行程度の簡単なコメントを付しただけのものである。

複数のサブシステムで利用している共通のモジュールはなく、また、サブシステム内でのモジュール間のコール構造は単純であり、基本的に順番に処理をする構造となっていた。したがって、コールグラフは作成していない。

上記で作成したモジュール一覧をもとに、2.3節で述べたプロジェクトの各課題に合わせて、解析すべきモジュール、変数の絞り込みをしながら解析を行った。各課題に対応するために行ったサブシステムの解析手順は以下のとおりである。

#### (1) シミュレーション実行の簡易化

基本的には、5つのサブシステムを連続して実行するコマンドファイルを作成すればよいが、すべて実行するまでに計26個のパラメータ入力が必要であり、かつそのパラメータに関する説明がないという問題があった。

そこで、まずはC、C++で標準入力に用いる、書式付き入力関数“scanf”、標準入力ストリーム“cin”といったキーワードを含むモジュールを解析対象とし、パラメータの意味の特定を行った。その結果、入力内容にかかわらずプログラム内に直接書かれた特定のファイルしか読み込めないものがあることが分かった。

#### (2) 出生率や死亡率を変更できる機能の追加

ファイルの入力は、入力ファイル名がプログラムの中に直接記述されており、外部から指定できるようにはなっていなかった。

入力ファイル名を外部入力のパラメータで与える形に変更する必要があるため、プログラムの中で入力ファイルである出生率、死亡率等のファイル名が直接記述されている部分を特定した。

#### (3) エクセルによる出力帳票作成機能追加

まず、出力するシミュレーション結果表を作成するのに必要な出力ファイル（CSV形式）を入出力ファイル一覧から選出し、紐づけを行った。ファイルから読み取れる情報から結果表の多くは作成可能だったが、ファイルだけでは項目内容が読み取れないもの、端数処理の違いにより厚生労働省が発表しているものと値が合わないものがあり、そういったものは出力ファイルを作成しているモジュールから変数をさかのぼり、端数処理方法の解析を行った。

#### (4) 公開されていないデータを出力する機能の追加

必要な変数名は財政検証結果レポートに記載されていることから、それを元にシミュレーションの機能の流れなど全体像を把握し、プログラム処理の上流から順に解析を行おうとした。しかし、配列などの説明が必ずしもプログラムの内容と一致していないこと、同じ変数名が何度も使い回され計算処理が複雑であること、目的とは関係ない情報を見極めるのが難しいことなどから、時間がかかりすぎると判断し、断念した。

そこで、(3)の場合と同じように、年金受給総額や受給者数が出力されているファイルからその出力処理部分をたどり、変数をさかのぼっていくことで目当ての計算処理部分を特定し、解析を行った。

以上のように、(1)、(2)では、目的に必要な入力処理に関するモジュールを、(3)、(4)では出力処理と、その出力内容に関係する一部の計算処理に関するモジュールを解析するだけで、課題に必要な情報を得ることができた。

3.1節で述べたように、ファイルの入出力構造がシンプルであり、それぞれのファイルがどのサブモジュールで入出力されているかは容易に判断でき、また複数個所から出力されていないことが分かっていたことから、このような解析個所を集中的に調査するという手法をとることができた。

実際に解析したモジュールの本数は表2のとおりである。詳細な分析が必要なモジュールは、相当程度絞り込まれていることが分かる。また、各工程に実際にかかった工数を表3に示した。CRUD 図の作成等には7人日、各課題に関連するプログラム解析には23人日、改装・実装には42人日必要であり、全体で72人日の工程が必要であった。解析すべきプログラムの絞り込みをしなかった場合には、プログラム解析工数がモジュール数に比例するとして推計すると、プログラム解析に121人日必要と推計されることから、全工程で98人日増加して170人日となる。一般的な手法を採用した場合の42.4%の工数に削減できたと考えられる。

### 3.2.3 まとめ、考察

今回の解析で最も重要である234個もの入出力ファイルの整理、解析には、ファイルの入出力を1つずつ調べ、

表2 解析を行ったモジュールの本数

サブシステム名	モジュール数	解析モジュール	比率
被保険者推計	29	4	13.8%
国民年金の給付費推計	32	7	21.9%
厚生年金の給付費推計	66	11	16.7%
国民年金の収支計算	30	10	33.3%
厚生年金の収支計算	69	11	15.9%
合計	226	43	19.0%

表3 成果物作成及び課題にかかった工数

成果物	ファイル・モジュール総数	作成工数
入出力ファイル一覧	234	5.0人日
モジュール一覧	226	2.0人日

課題	プログラム解析	改修・実装
(1) シミュレーション実行の簡易化	3.0人日	20.0人日
(2) 出生率や死亡率を変更できる機能の追加	5.0人日	10.0人日
(3) エクセルによる出力帳票の追加	5.0人日	9.0人日
(4) 公開されていないデータを出力する機能の追加	10.0人日	3.0人日

CRUD 図として整理するのに5人日という、モジュール一覧作成の倍以上もの工数がかかったが、モジュール解析を行う前に十分な整理ができたことで、その後の解析を効率よく行うことができた。

また本プロジェクトでは、厚生労働省が公開している結果の正確性の評価については目的対象外であったことから、現行のプログラムが正しく動作していることを前提にできたため、計算結果に影響するモジュールの解析を行う必要がなかったことも、短時間で作業することができた要因でもある。今後、計算プロセスの修正にかかわる課題が挙げられた場合には、計算部分に関するモジュールの解析が必要となる。

今回解析を行う上で苦勞した点としては、次のことが挙げられる。

入出力ファイルの整理をする際には、

- 1つのCSVファイルが複数個所で書き込まれるなど、データの内容からみて、本来別々のファイルとすべきものが1つのファイルにまとめられているケースがあった。
  - さらに、それらの複数のデータ群に、ヘッダ情報がある項目とない項目があり、データ群の切り分けを行うために、プログラム解析まで行う必要があるものがあった。
- プログラム解析を行う際には、
- 長年、多くの人によって改修され続けたプログラムであるため、コーディング手法が統一されていないことがあった。
  - 最大5次元にもおよぶ配列が多数あり、しかも添字の意味が不明なものもあった。
  - デッドロジックとなっている部分があり、不要な情報が多かった。

これらのことは、システム開発を行う時点で入出力ファイルのフォーマット、データ内容を整理し、入出力関係を明らかにすること、コーディング規約を明確に設定することなど、開発時に注意すべき点としても同様のことがいえるであろう。

## 4. おわりに

設計書や仕様書などのドキュメントが十分に整備されておらず、ソースコードを解析しながら利用するシステムは少なくない。ソースコードを自動的に解析して設計書を作成するソフトウェアやサービスは、いくつかの会社が提供しているが、基本的には、システム全体を網羅



的にカバーするサービスであり、高価であったり、必ずしも意図したドキュメントが作成されたりするとは限らない。

本プロジェクトでは、厚生労働省が公開している財政検証プログラムのソースコードを元に、設計書や仕様書に相当するドキュメントの作成とそれらを用いたプログラムの改修を試みた。システムの構造が公的年金財政の仕組みそのものであることから、プログラム全体の構造や仕組みが事前に理解できたことや、入出力ファイルの数は多いがその処理構造はシンプルであることから、ドキュメントを網羅的に作成するのではなく、目的に沿った解析を行い、必要な部分に特化したドキュメント化を行っても、少ないリスクと短い期間で各課題が解決できたことは、前章で述べたとおりである。

特に、本プロジェクトでは入出力ファイルの整理が一番重要であったことから、CRUD図の作成を試みた。実際の開発現場ではシステム開発の規模が大きくなるほど、作成されていないケースが散見されるが、本プロジェクトではこのCRUD図が大変有効に活用できた。

題材としたのは、一般の業務システムではなく、科学技術計算に分類されるであろう財政検証プログラムである。ブラックボックス化できる部分が多く、かつ、それを容易に特定できるという特徴があったことは事実である。しかしながら、修正する可能性のない部分についてブラックボックス化を割り切り、全体の作業工程を大幅に短縮して、重要なモジュールの解析にリソースを集中することは、1つの方法論として考えられるのではないだろうか。

なお、本プロジェクトでは、新たな「公開されていないデータ」を出力する機能を追加したり、年金額算定式の変更といった計算プロセスにかかわる修正を行ったりするなど、新たな課題にとりかかる必要があり得る。その場合でも、追加解析が必要なモジュールの特定は容易であり、一部のブラックボックス化した部分のみ解析すればよいため、本稿の手法は特に有効であった。

**謝辞** 本稿を作成するにあたり、小澤正典（慶應義塾大学）からは、FORTRANで記述されている財政検証プログラムの一部についてC++に変換したプログラムの提供を受けた。稲垣誠一（東京工業大学）からは、公的年金制度や財政検証の仕組みなどについて多くのアドバイスを受けた。その他、本研究において多くの支援をいただいた関係者の方々に厚く感謝の意を表する次第である。もちろん、本稿にあり得べき誤謬等は、すべて筆者らの責任である。なお、本稿で示されている内容および意見は筆者ら個人に帰するものであり、(株)シーエーシーあるいは(一社)年金総合研究所の公式見解を示すものではない。

#### 参考文献

- 1) 稲垣誠一、小澤正典、寺西教高、日隈美奈子、石田朗大：公的年金財政シミュレーションツールについて、JARIP 会報:大会プロシーディングス特集号, Vol.1, pp.59-70 (2014).
- 2) 厚生労働省：将来の厚生年金・国民年金の財政見通し, <http://www.mhlw.go.jp/stf/seisakunitsuite/bunya/nenkin/nenkin/zaiseikensyo/index.html> (2014年3月12日現在)
- 3) 厚生労働省：平成21年財政検証結果レポート「国民年金及び厚生年金に係る財政の現況及び見通し」(詳細版), <http://www.mhlw.go.jp/topics/nenkin/zaisei/zaisei/report2009/mokuji.html> (2014年3月12日現在)
- 4) ガイオ・テクノロジー (株)：プログラム仕様書リバース作成・ソース解析サービス, [http://www.gaiotech.co.jp/product/dev\\_tools/pdt\\_progrevs.html](http://www.gaiotech.co.jp/product/dev_tools/pdt_progrevs.html) (2014年3月12日現在)
- 5) (株)【A HotDocument】：ドキュメント自動生成ツール【A HotDocument】, <http://www.hotdocument.net/> (2014年3月12日現在)

石田 朗大 (非会員) [ishidaak@cac.co.jp](mailto:ishidaak@cac.co.jp)

1984年生。2009年広島大学大学院先端物質科学研究科量子物質科学専攻修士課程修了。同年(株)シーエーシー入社。

寺西 教高 (非会員) [nteranis@cac.co.jp](mailto:nteranis@cac.co.jp)

1977年生。2001年関西大学商学部卒業。同年(株)シーエーシー入社。

日隈 美奈子 (非会員) [minakoh@cac.co.jp](mailto:minakoh@cac.co.jp)

1981年生。2005年東京理科大学大学院理学研究科物理学専攻修士課程修了。同年(株)シーエーシー入社。

投稿受付：2014年4月11日

採録決定：2015年3月19日

編集担当：安崎篤郎 ((株) 日立製作所)