形式仕様のための読書案内

山 﨑 利 治

1. はじめに

これはソフトウェア開発の形式的方法についての読書案内で，未経験の実務家を形式的方法へ誘うのが目的である．1970 年代のプログラミング方法論は自然に 80 年代の形式的方法へ推移した．90 年代はその実用化時代で，Z，B，VDM，RAISE などそれぞれの形式的方法がソフトウェア開発の現場で多くの実績を積んできている．これらの方法については教科書や入門書も多く出版されているので，ここでは本に限って紹介したい．ぜひ一冊は読んで仲間うちで議論し実用化を図ってほしい．また，ここに挙げた方法は仕様作成とその実現という二つの側面をもっているが，とくに仕様化技術の改善が急務であるとの考えから紹介は仕様面に限りたい．

形式とは実体，たとえば，ボルトとナットという場合，そのボルトとナットの形状をいい，鉄やプラスチックなどの材質は問わないことを指す．仕様記述の場合では，記述言語は構文と意味をもつが，意味に言及しないで構文のみで記述対象を議論することである．これは命題論理や述語論理に対して命題計算や述語計算という形式的体系が構成できる事実に対応している．つまり，この形式性によって対象のもつ性質の発見，解析，検証など対象についての理論展開が容易にできるわけである．

方法とは「理性を正しく導き科学の真理を求めるための」方法，あるいは，「自然解明に対する正しい指示としての」方法をいい，一時的な術や方便ではなく対象についての客観的な認識と論理に基いた目標達成への道程をいう．

形式と方法の二つの用語をもつ形式的方法は，したがって，まずソフトウェア開発のための理論をもち，この土台の上に理論の適用手段としての技術があり，さらにこの技術を円滑に適用するために道具を用意し，技術を選択利用する手段をもっていなければならない．形式的方法はこのように技術であるので，経験の蓄積としての技能ではなく，技術としての教育を可能にするものである．

Z，B，VDM，RAISE など，それぞれ一つの完結した方法を産業界が注目し採用したのは，それらの記法や構造化機構が優れていることのほかに，ソフトウェア開発が多数要員の共同作業に依存している現実がある．共同作業では不可避であったさまざまな約束事項の取り決めに要する時間や労力が方法の共有によって不要になるからである．また，これらの方法には教育資料やソフトウェアツールが豊富である点も大きい．以下に，Z，B，VDM，RAISE の順に本を紹介し，形式的なオブジェクト指向方法や形式的方法に対する基礎教養書にも触れる．

2. Z

Z［zéd］は 1970 年代よりオックスフォード大学のプログラミング研究班で育成されてきた．Z は日本ユニシスにとって馴染み深い方法である．社内教育を試み，社内業務の一部に対して Z 仕様を書き C によって実現し予期した成果をえた例もあるか

らである．しかし，現状では普及しているとはいえないので広い実用を願って数冊の本をとりあげてやや丁寧に紹介したい．はじめにZの特徴から集合と論理について一言注意しておきたい．

Zでは，宣言部と公理部の二つをもつみやすい箱型のスキームが記述単位の基本である．宣言部には実体の名前と型を，公理部には実体のもつ性質を書く．実体とは記述対象そのもの，あるいは，その一部である．ここで実体は集合や関係あるいは関数などで書き，その性質は述語として書くので，集合や関数など数学上の概念や述語などの論理学上の概念がZでは陽に現れる．このような概念は数学の論理学のという大仰なものではないが慣れる必要がある．

集合や関数は中学や高校の数学で周知の概念であるが，Zは型付きの集合を扱うので，どんな集まりがZでいう集合になり，あるいは，ならないかを知らねばならない．それはZの入門書や教科書で学ぶことになる．

論理学の面では，命題計算や述語計算の概略を知っていれば十分である．

真偽が決定できる文や式を命題という．「5+3=8」は真の命題であり，「12は素数である」は偽の命題である．命題に対して演算を導入してより複雑な命題が構成できる．たとえば，pやqで命題を表すとき，¬p（否定，pではない），p∨q（選言，pであるかまたはqである），p∧q（連言，pでありかつqである），p→q（含意，pならばqである）などである．これらの複合命題を論理式ともいうが，その真偽は式を構成する命題と演算によって定まる．たとえば，¬pはpが真のとき偽，pが偽のとき真と定義し，p∨qはpとqが共に偽のときに限って偽と定義する．p∨¬pやp→(q→p）のようにpやqの真偽にかかわらず真となる式もあり，恒真式という．

さて論理学は推論あるいは論証に関する学であった．推論とは前提といういくつかの論理式から帰結という一つの論理式を導くことである．妥当な推論とは前提がすべて真なら帰結も真であり，またそのときに限っていう．複雑な推論には「証明」がほしい．そこで証明をつぎのような形式的な手続きとして考える．まずいくつかの恒真式をえらび論理公理とする．つぎに，前提pとp→qから帰結qが推論できるという三段論法を推論規則とする．有限個の論理式の集まりをΓとする．Γから論理式pが推論できることの証明をつぎのような式の列が存在することとする．ただし，列の最後はpであり，間の式は公理であるかΓに含まれるか，あるいは，それ以前のいくつかの式を前提として推論規則を適用したときのその帰結の一つになっている．この証明は式の形式だけに依存したものになっている．つまり式の内容によって定まる真偽が形式面だけから決定できることになる．このような体系を命題計算という．命題計算にとって，妥当でない推論の証明は存在してはならないという健全性と，妥当な論証は必ず証明できるという完全性が大切であるが，この二つは命題計算では満たされる．

「5+x=8」のように変数xを含む式はxの値によって真偽が変化する．このような式や文を述語という．対象のもつ性質や対象間の関係なども表現できるように述語を導入して命題論理を拡張したものを述語論理という．その特徴は述語を命題に対応させる演算∀x（すべてのxに対して）や∃x（あるxが存在して）の導入である．述語論理に対しても命題論理の場合とおなじように述語計算が構成できる．

以下，Z についての入門書や必携などを紹介する．

［PST］B. Potter, J. Sinclair and D. Till, *An Introduction to Formal Specification and Z*. Prentice Hall, 1991. xiii＋304 pp. 田中武二監訳，ソフトウェア仕様記述の先進技法：Z 言語．プレンティスホール・トッパン，1993. xiv＋321.

Z に関する英文の書物は随分多いが，これは現在邦訳のある唯一の本である．読者として専門学科の学部 1，2 年生あるいは経験を積んだソフトウェア実務家を想定し，まったく予備知識なしに一歩一歩学習できる形式仕様と Z に関する入門書である．この本では仕様記述問題で有名な図書館問題を Z で書いていて，読了すれば Z の仕様が書けたり読めたりするはずである．入門書のなかには大変難しいものもあるが，これは文字どおりの入門書といえ一人で懐手して読める本である．

構成は 3 部からなり，第 1 部 51 ページは前口上と準備，第 2 部 123 ページは Z 記法の包括的な解説，第 3 部 108 ページは実現その他である．

第 1 章ではソフトウェア工学の側面からみた形式仕様について一通り説明する．

第 2 章は形式ばらない論理と集合への入門である．まず，集合である．集合とは「直観または思考の，明確な互いによく区別される対象を一つの全体にまとめたもの」である．そしてその対象を集合の要素という．といってもあまり判然としないだろう．具体例による理解が一番である．国を対象としてその上の集合を考える．たとえば，ベネルクス＝＝｛ベルギー，オランダ，ルクセンブルグ｝といった具合である．これは要素を明示的に並べて集合を定義したものであり，外延的定義という．ほかにも EEC，NATO，北欧などを定義する．これらの上の演算として，合併，共通部分，差などを扱い，演算間に成立する諸法則を示す．要素をもたない集合として空集合 ϕ をも導入する．それから集合間の関係，含まれる，真に含まれる，等しい，等しくないを定義し練習問題にはいる．命題と述語を型どおり説明したあと再度集合に戻る．要素の性質を述語で与える集合の内包的定義，組，集合の集合である巾集合などを説明する．わかりやすい説明のなかにも Z で許容する集合になるように細心の注意を払っている．

第 3 章では第 2 章で学んだ集合や述語を仕様記述に利用してみる．簡単な両引きの用語辞書を計算機で実現することを考え，この仕様化を図る．辞書は相互に対訳となる用語対の集合であり，空集合にこの用語対を加えていけば構成できる．また，どちらかの言語の用語に対して訳語を求めるには，その用語を含む対を見出し対の他方を取りだせばよいことになる．こうして辞書の世界に集合を導入する．このような仕様化は対象をその状態とその上に作用する演算群によって定義する抽象データ型の考えに沿ったものである．

第 4 章から Z 記法の本格的な解説がはじまる．Z 言語は数学言語とスキーマ言語に分けられる．数学言語からはじめる．数学言語は対象自体や対象間の性質の記述に使うものである．Z に登場するすべては型をもっている．そこで，まず，型の意義をラッセルの逆理によって説明し，そのあと Z の型体系を具体的に展開する．整数などの組み込み型やその内容を指定しないで型として与えた与集合型（あわせて基本型と

いう)，集合の集合を与える巾集合型，組の集合を与える直積型と順に例示する．なにかをZに登場させるには名前と型を書けばよい．これが宣言である．ここで宣言の書き方と集合や述語の進んだ書き方を学ぶ．仕様記述単位をZの構文ではパラグラフといっているが，そのうち大域変数（定数）を導入する公理箱，総称的定義を与える総称箱などを学ぶ．総称的というのは型引数を伴った型のことで，たとえば，空集合の型は，本来，型Tを引数としてもち，Tを具体的に与えてはじめてその型の空集合が定まるもので，Zの型体系はこのような型を許している．

第5章も数学言語の続きである．関係と関数を学ぶ．関係とは演奏家とその演奏家がひく楽器との対応のように二集合の直積集合の部分集合をいい，関数とは都市とそれが存在する国との対応のように，関係でその対応先が一意的であるときにいう．これらの概念の説明も例を多くあげ懇切丁寧である．関数の書き方としてラムダ記法を述べたあと，列をとりあげる．列とは自然数区間からある集合への関数と定義するが，集合の要素を単に有限個並べた順序列である．これも仕様記述によく利用する重要な概念であり習熟しなければならない．以上は数学の道具箱といって宣言しないで仕様のなかで自由に使えるものである．

第6章はスキーマの書き方である．スキーマ言語は仕様を構造化しまた合成するためのものである．スキーマは仕様記述の基本単位で宣言，型，述語，関係などとして使う．スキーマに対する演算によって複雑なスキーマが簡単に書け，これをスキーマ計算といっている．スキーマの典型は抽象データ型の状態や演算の定義である．ここでは仕様記述課題として有名な図書館問題をとりあげ，スキーマを書いていく．説明はわかりやすい．ここまで読んでくれば大方の仕様は書けるはずである．

第7章は3章で説明した対訳辞書をしっかりZで書きなおし，仕様記述にさいしての必要な細かな注意を与える．章末に仕様の一般的な構造を提示する．仕様を書いていれば自然に体得することではあるが，このまとめはたいへん貴重である．

第8章は書いた仕様からなにが推論できるかを扱う．いままで水面下にあった形式推論の体系がここで顔をだす．書いた仕様に矛盾があればどんなプログラムでもその仕様を満たす正しい実現になるから，この章の内容は大切である．述語としてのスキーマもここで学ぶ．

第9章以降は実現に関わる話題を扱っている．仕様の正しい実現はきわめて重要な問題であるが仕様記述面に注目しているのでここでは割愛する．

［ZRM 2］J.M. Spivey, ***The Z Notation A Reference Manual***. Prentice Hall, 1992 (2 nd ed.). xii＋158 pp.

この本は毎日Z仕様を書いたり読んだりするソフトウェア実務家の必携で，非形式的ななかにも厳密にZ記法を定義している．毎日Zに直面していれば不要ともおもえるが，いずれにせよ言語の厳密な定義はどこかに存在しなければならないし，また，ソフトウェアツールにとっては記法の定義が格別に重要である．そこでこの本はZ記法の事実上の規格となっている．入門書ではないが，集合や関数に抵抗がなく命題や述語にも馴染みがあり，抽象データ型やオブジェクト指向などを見聞していれば

楽に読める本である．

第 1 章は誕生日帖を例に Z 記法を説明する．23 ページの短い章でスキーマ計算から正しい実現のための洗練計算までを扱う．

第 2 章は Z の背後にあるスキーマや型などの概念を説明する Z 言語の意味論である．

第 3 章は Z 言語自体の記述である．この記述様式はプログラム言語に普通に見られるものであり，構文範疇にしたがって宣言，述語，式などと説明がつづく．型に型変数を含む総称スキーマや，木などにみられる再帰構造を楽に書けるようにした自由型を簡潔にしかし精密に記述している．

第 4 章は数学道具箱である．集合，関係，関数，自然数，整数，列，多重集合と順に一定の様式で定義する．これらが仕様記述に際して自由に使えるわけである．

第 5 章はソフトウェア仕様を書くための便宜上の約束事を提示する．3，4 章でみた Z 記法による記述は構造化した数学理論の展開にすぎない．ソフトウェア仕様としては対象を抽象データ型とみなして，その状態と演算とをスキーマに書けばいいので，そのための細かな注意を与え，また命令型の逐次プログラム向きに段階的洗練についても述べている．

第 6 章には Z 言語の具体構文を変形バッカス記法によって与えている．

このあと章立てしないで，第 1 版との相違を述べ，用語小辞典と索引を付けている．この用語辞典は有用である．

小冊子でもあるので座右におき，一度は全体に目を通しておきたい本である．

［SUZ］J.M. Spivey, *Understanding Z*. Cambridge, 1988. viii＋131 pp.

副題「仕様言語とその形式意味論」をもつこの本はおなじ著者の［ZRM 2］が非形式的な意味記述であったのに対して形式的な意味記述を行ったものである．Z を教える先生や Z のソフトウェア・ツールをつくるひとが読むべき本である．Z 言語の仕様が Z で書けてこそ Z が立派な仕様記述言語である証になるから，それを実行する．記述の枠組みは表示的意味論である．論理で式の意味解釈は本質的に付値関数であった．その関数は式を数，真理値，関数など数学的対象の適切な領域へ写像する．この関数はまたフレーゲの原理，つまり，合成式の意味は構成部分式の意味の関数として表せるを満たした．Z の表示的意味定義もプログラム言語のそれと同様で，具体的には構文領域，抽象構文，意味領域，意味関数をそれぞれ定義する．問題は意味領域である．Z の集合をまず定義しなければならない．Z 集合の宇宙はツェルメロ・フレンケル流の公理的集合論から置換公理と選択公理を除いたものである．型は仕様の読み書きを容易にするほかに技術的な理由もあって必要であった．そこで与集合型，巾集合型，直積集合型，スキーマ型をおく．型の意味は台集合である．スキーマの意味は，結局，抽象データ型の意味とおなじ多種代数の族である．型付き集合を扱うので意味領域を領域方程式の解として定義することはない．また，意味関数の再帰的定義でも特別の注意によって解の存在や一意性を保証できるよう工夫し，意味領域は完備半順序集合などではなく集合で済ませる．話は技術的にすぎるが理解が深まるのも事実で

ある．職業的ソフトウェア技術者であれば読んでおきたい本である．

［DIL］A. Diller, *An Introduction to Formal Methods*. John Wiley, 1990, xxi＋309 pp.
［INC］D.C. Ince, *An Introduction to Discrete Mathematics and Formal System Specification*. Oxford, 1988. xii＋349 pp.
［JAC］J. Jacky, *The Way of Z Practical Programming with Formal Methods*. Cambridge, 1997. xviii＋350 pp.
［LIG］D. Lightfoot, *Formal Specification Using Z*. Macmillan, 1991. xi＋169 pp.
［MPZ］M. McMorran and S. Powell, *Z Guide for Beginners*. Blackwell, 1993. viii＋247 pp.
［RAT］B. Ratcliff, *Introducing Specification Using Z: A Practical Case Study Approach*. McGraw-Hill, 1994. xii＋308 pp.
［WDZ］J. Woodcock and J. Davies, *Using Z Specification, refinement. and proof*. Prentice Hall, 1996. xvi＋391 pp.
［JBW］J.B. Wordsworth, *Software Development with Z A practical approach to Formal Methods in software enginneering*. Addison-Wesley, 1992. xii＋334 pp.

以上はいずれも入門書である．このほかにも出版されている本があるがまだ見ていない．［DIL］は第2版がでているがこれも見ていない．［DIL］は読みやすい本である．［INC］は多分Zについての最初の本である．［JAC］の著者は癌の放射線治療器の制御プログラムを書いているひとでその面の記述に特色がある．［LIG］は小冊子で電車のなかで読める．［MPZ］も丁寧な本である．［RAT］と［JBW］は実務的な例によってZを解説していてそのぶん実務家には親しみやすいかもしれない．［WDZ］は正面から推論の形式体系を説明するが，すこし骨っぽいだけ結局よく身につくとおもえ推奨する．スキーマのスキーマへの埋め込み（promotion）の解説はこの本が一番わかりやすかった．

プログラム言語の場合とおなじで，仕様言語を覚えても必ずしもうまく仕様が書けるわけではない．よく書けた記述例を読む必要もあるに違いない．つぎはその目的に適った本である．

［BOW］J. Bowen, *Formal Specification and Documentation Using Z A Case Study Approach*. Thomson, 1996. xvii＋302 p
［HAY］I. Hayes（ed.）, *Specification Case Studies*. Prentice Hall, 1987. 332 pp.

［HAY］はすこし古い気がするが改訂版がでているからそちらはそうでないかもしれない．［BOW］は極めて有意義な本である．手短にZの解説をしたあと，ネットワークサービス，UNIXの文章整形ツール，Transputerの命令，Xウィンドウシステムなどを題材にその仕様記述を試みている．付録とした，Zについてのさまざま

な情報とその源，文献案内と文献表は貴重である．Z についてはここまでとする．

3. B

B は Z を生み育てた J.-R. Abrial によって開発され，とくにフランスで実績のある方法である．Z の弟分である B は Z に比べて実現をより強く意識していて，仕様作成から設計，プログラム作成までを同一言語によって行う．仕様記述単位を抽象機械といい，これを B 抽象機械記法 B　AMN で書き，Z の数学言語に相当するものは一般代入言語 GSL で書く．もちろん，GSL の使える範囲は仕様，実現のそれぞれの段階で違い，B AMN の構造化機構も仕様，設計，コード化の段階によって細かく規定されている．本は 3 冊出ている．

［JRA］J.-R. Abrial, *The B-Book : Assigning programs to meanings*. Cambridge, 1996. xxxiv＋779 pp.

［KLB］K. Lano, *The B Language and Method : A guide to practical formal development*. Springer, 1996. viii＋232 pp.

［JBW］J.B. Wordsworth, *Software Engineering with B*. Addison Wesley, 1996. xv＋331 pp.

教典は［JRA］である．B のすべてが書いてある．副題は有名な Floyd の論文名を逆にしたものである．つまり，仕様が先でそれから正しくプログラムを作れというのである．800 ページを超えるこの持つにも重い本は，しかし，いきなり全部を読めというのではない，実務家，理論家，初心プログラマ，それぞれのひと向きに親切な導きの糸を提示しているのでそれに従って読めばよい．

［KLB］は副題に実用とあるように実務家向きの本であるが，形式仕様かオブジェクト指向に馴染んでいないと読みにくいかもしれない．この本は OMT を B AMN で書こうとしているので OMT や UML に飽き足らないひとにはとくにお勧めである．

［JBW］はおなじ著者の Z に関する本の姉妹編である．完全に実務家向きの本である．

4. VDM

VDM の歴史は古い．デンマークの D. Bjørner と英国の C. Jones の普及活動によって実務家が注目するようになった．VDM の起源はプログラム言語の仕様記述と処理系の自動作成にあった．多くのソフトウェアの仕様も言語のそれに見習って書けるからこの方法の学習も貴重である．つぎのような本がある．

［AI］D. Andrews and D. Ince, *Practical Formal Methods with VDM*. McGraw-Hill, 1991. xvi＋450 pp.

［B］J.C. Bicarregui et al., *Proof in VDM : A practitioner's guide*. Springer, 1994. xvi＋362 pp.

［BJ］D. Bjφrner and C.B. Jones, *Formal Specification and Software Development*. Prentice-Hall, 1982. x＋501 pp.

［JD］J. Dawes, *The VDM-SL Reference Guide*. Pitman, 1991.

［CBJ］C.B. Jones, *Software Development: A rigorous Approach*. Prentice-Hall, 1980. xv＋383 pp.

［CBJ 1］C.B. Jones, *Systematic Software Development using VDM*. Prentice-Hall, 1994（2nd ed.）. xiv＋333 pp.

［JS］C.B. Jones and R.C. Shaw（eds.）, *Case Studies in Systematic Software Development*. Prentice-Hall, 1990. xvii＋387 pp.

［LBC］J.T. Latham et al., *The Programming Processs An introduction using VDM and Pascal*. Addison-Wesley, 1990. xviii＋537 pp.

VDM の標準的教科書は［CBJ 1］である．そのあとで［JS］を読めば完璧である．実務家向きの［AI］もある．［LBC］はプログラミングの入門書であるが，プログラミングを要求分析，形式仕様，算法とデータの設計，実現の 4 段階に分け形式仕様を VDM で実現を Pascal で書く．電文解析や KWIC など年寄りには懐かしい課題を扱い，また，女性プログラマがボスのメモを読んで仕事をするという挿話を挟んで自然に VDM も Pascal も覚えるという寸法である．かなりなページをもつが内容はやさしく抵抗なく読める本である．［JD］は VDM の仕様記述言語 SL を解説したものである．［B］は正しい実現であると主張するための証明について述べたものである．［CBJ］は［CBJ 1］の前身でいまでは［CBJ 1］を読めばいいだろう．［BJ］は古典で，その初期にはどんな目的をもち，どのように利用していたかを知るのに役立つ．また表示的意味論で必要な領域など VDM の基礎理論についての解説も含んでいる．

5. RAISE

RAISE の仕様記述言語 RSL は VDM を基礎として仕様記述で試みられたあらゆる考えを取り入れた欲張ったものである．状態を表に出して演算を状態の変化で書くモデル指向でも演算を演算間の関係だけで書く性質指向でも書け，これを命令型，作用型，並行型のいずれかで書く．現在 2 冊の本が出ている．

［RLG］C. George, et al. *The RAISE Specification Language*. Prentice Hall, 1992. xix＋397 pp.

［RMG］C. George, et al., *The RAISE Development Method*. Prentice Hall, 1995. xvi＋493 pp.

とくに［RMG］は熟読玩味したい．入門編で，仕様には，悪いことは未来永劫にわたり起こらないという安全性や良いことはやがて起こるという活性を書くべきであるなどと述べ，技術編では開発，証明，翻訳に分け RAISE の方法を詳説しているからである．ここで開発とは仕様作成とその詳細化をいい，証明とはある条件の成立の根拠を厳密に証明することをいい，翻訳とは仕様のコード化をいう．

6. オブジェクト指向

OMT などのオブジェクト指向分析設計は拡張実体関連模型とステートチャートという世界観で課題を眺め，図と補足文によって課題背景を書きこの上に機能要求を加え分析といい，それを基にしてクラスの継承とクラス関数の遅延束縛によってコード化の効率向上を図る．ここでは仕様の概念は希薄である．一方形式的方法は世界観については無言であり，継承や遅延束縛については仕様記述単位の構造化と洗練の機構（引数化，強化拡張，弱化隠蔽，名前替え）によって明言しすべてを仕様記述の視点で整理する．つまりオブジェクト指向方法と形式的方法とは互いに補完関係にあるといえる．

そこで OMT などのもつ記述の曖昧さを嫌いオブジェクト指向分析や設計の成果を形式的に書こうとするひともいる．この面の本も何冊かでている．

［FUS］D. Coleman et al., *Object-Oriented Development The Fusion Method*. Prentice-Hall, 1994. xx＋313 pp.
横河ヒューレットパッカード株式会社カストマ教育センタ訳，オブジェクトオリエンテッド開発設計論：The Fusion Method. プレンティスホールトッパン，1994. xiv＋337 pp.

［SYN］S. Cook and J. Daniels, *Designing Object Systems Object-Oriented Modelling with Syntropy*. Prentice-Hall, 1994. xx＋389 pp.

［KL］K. Lano, *Formal Object-Oriented Development*. Springer, 1995. xiii＋422 pp.

［LH］K. Lano and H. Haughton（eds.）, *Object-Oriented Specification Case Studies*. Prentice Hall, 1994. xx＋236 pp.

［OOZ］S. Stepney et al.（eds.）, *Object Orientation in Z. Springer*, 1992. vii＋144 pp.

［FUS］と［SYN］はオブジェクト指向分析設計の伝統を継ぎ，内容は，明白にはいわないが，形式的そのものである．［OOZ］は Z の枠組みでオブジェクト指向技術をとりいれようとする試みを紹介している．［KL］はとくに Z＋＋と VDM＋＋をとりあげ，オブジェクト指向技術を復習しながら仕様記述を詳細に議論し解説している．したがってこの 1 冊でオブジェクト指向と形式的仕様化の両方が使えるようになる便利な本である．［LH］は手軽に読めるオブジェクト指向に基づく形式仕様のあれこれである．

7. 基礎教養書

本格的な本が読みにくいと感じられるかもしれない．これには多分二つの理由がある．一つは読みすすめても仕様を書くイメージがもてないこと，もう一つは数学や論理の詳細に疲れることであろう．後者は単に慣れの問題で虚心に読んで速く慣れるしかない．前者は少々むずかしい．そうであれば，つぎのどれかを是非実行してみてほしい．システムアナリストやデザイナなら，構造化分析の小仕様を非手続き的に書く

とすればどう考えるか，あるいは，OMT を知っていれば［FUS］か［SYN］を読んで，それらが要請する文書はどんな言語で書けばいいかを考えてみよ．プログラマなら，Oracle や SQL を使っていれば実体関連模型を集合と関係によって記述してみたり，Java や C＋＋を使っていればクラスに対してその仕様をどう書けばいいかを考えてみよ．クラス，パッケージ，モジュールなど呼び名はさまざまでもこれらはどれもみな抽象データ型である．抽象データ型はプログラム作成論で生まれ，仕様記述の議論で一般的に使われている用語である．ともあれ基礎教養が身につく本の一読はお奨めである．そのような本を以下に挙げる．

［JCC］J.C. Cleaveland, *An Introduction to Data Types*. Addison-Wesley, 1986. xii＋239 pp. 小林光夫訳，データ型序説．共立，1990. ix＋301 pp.

［GOR］M.J.C. Gordon, *The Denotational Description of Programming Languages An Introduction. Springer*, 1979. v＋160 pp.

［HAY］林　晋，プログラム検証論．共立，1995. xi＋211 pp.

［MOR］C. Morgan, *Programming from Specifications*. Prentice Hall, 1994（2nd ed.）. xv＋332 pp.

［ONO］小野寛晰，情報科学における論理．日本評論社，1994.xii＋297 pp.

［JCC］には抽象データ型についての詳説がある．抽象データ型は仕様記述の中心概念だから十分な理解が必要である．わかりやすい本なので是非読んでほしい．［GOR］は表示的意味論の実務家向きの本である．VDM はもともと表示的意味論を土台にしているし，この流儀の仕様記述は理解しておきたい．日本語の教科書も何冊かあるがいずれも理論家向きなので結局［GOR］が読みやすいだろう．［HAY］はプログラムの正しさを証明する話の詳論であり，Z の紹介も行っている．検証論は，プログラムを書いてから正しさを証明するのではなく仕様から正しくプログラムをつくれというように実務向きに発展する．［MOR］がそれで，いまではこれを洗練計算といっている．Z でも B でも VDM,RAISE などでも実現過程では［MOR］が基本である．［ONO］は仕様作成やプログラム作成で必要な論理について温習するのに向いている．これらは直接に仕様化技術を学ぶ傍ら読んでほしい本である．

8. おわりに

形式的方法として欧州育ちの四つを紹介したが，無論この他にも多くの形式的方法や形式的記述言語がある．後者の LOTOS は最初の国際規格でもあり，邦語の詳細な紹介書もある．対象の動的挙動を直接的に書こうという言語で交信プロセス理論を基礎にしている．目的は通信プロトコル記述であった．たとえば，ジャクソンシステム開発法の仕様記述に LOTOS は有効であるがその意味では現在では RAISE が代替する．

日本で強力に育成した形式的方法に Cafe Obj がある．これは代数的方法の代表でよいソフトウェアツールも整備され普及させたいものである．不幸にも Z に見られるような入門書や教科書がまだ出版されていないのでここではとりあげなかった．米

国育ちの Larch も代数的な仕様記述をとる方法でわかりやすい本もあるが割愛した．

おなじ課題に対しても意識や目的の差によって定式化に差がでるが，これがさらに選んだ記述言語によって歪まないか，また逆に素早く書けたりしないか．あるいは，方法には応用領域に依存して向き不向きがあるだろうか．方法の採用に選択肢があり，大勢の集団による開発を行っている場合にはこのような疑問はもっともである．手短な答えはそんな心配をしないでどれでもいいから使いなさいである．それぞれに巧く書けるに相違なく，もし問題があれば事例研究を眺めたり仲間と検討したりすればいいわけである．もう少しなにかいえというなら応用領域に付いて考えなければならない．つぎのような分野を考えるひとがいる．コンパイラなどの翻訳系，データベースと問い合わせの情報系，センサとアクチュエータを伴う制御系，人間機械系の界面，電子取引などの取引決済系，ワープロなどの著作支援系（Visual Basic などで作成する事務プログラムもここにいれていい），これらのいくつかが複合した系などである．このそれぞれに仕様記述に反映する特徴がある．Z はもっとも一般的で普遍的に書ける．逆にいえば，対象に依存して特別に書きやすくすることもない．翻訳系や情報系や著作支援系には表示的意味記述に倣って VDM か RAISE を選びたいし，制御系や界面なら CSP 流に考えて RAISE を採用してもいい．実現まで継ぎ目なく開発するなら，また，オブジェクト指向の OMT を考えるなら B を採用したい．というような面は確かにあるが，Z は浮動小数点演算プログラムや放射線医療器の制御プログラムの作成，オシロスコープの仕様や設計に使われ成功している．Z による仕様化技術を仲間うちで共有することの意味は大きいに違いない．大方の格別の研鑽を期待したい所以である．

執筆者紹介 山 崎 利 治（Toshiharu Yamasaki）

1957 年名古屋大学理学部数学科卒業．爾来日本ユニバック(株)，日本ユニシス(株)勤務，1993 年退職．著書に「プログラム言語」（昭晃堂，1989 年），「プログラムの設計」（共立出版，1990 年），共訳書に M.T. ベルティーニ＋Y. タリノー「構造的コボル教則本」（TBS 出版会，1977 年），M. ジャクソン「システム開発−JSD 法」（共立出版，1990 年）がある．