

Z 記述の視覚化に関する考察

97B525 今泉 文吾

指導教員 野呂 昌満

1 はじめに

ソフトウェア開発の初期段階において要求分析の成果物である要求仕様書が明確でなければ、開発者と依頼者の間に理解の不一致が生じてしまい、実際に開発されたものが要求されたものとは異なったものになってしまうなどの問題が生じる。仕様記述方法の1つとして、数学的理論に基づく形式手法が知られている。形式手法を用いて仕様を記述することで、仕様を作る側と見る側の意思の疎通に矛盾点や誤解が少なくなることが期待できる。しかし一般に知られている形式手法では、仕様書はあらかじめ決められた数学的表現で記述されるので、仕様書を見る側も数学的知識を持っていなければ理解しにくいものになってしまう。

本研究の目的は、「形式的な仕様記述を絵によって表現することで直観的に視覚に訴え、理解の支援をすること」である。対象者は、ソフトウェア開発に携わる者や、学校などの教育の場を想定しているため、ある程度の数学的知識は持っていることを前提とする。

仕様記述言語としては比較的分かりやすく、普及している Z を用いる。最終的な目標は、形式仕様記述言語 Z で記述された仕様の理解支援ツールの作成である。本研究では、ツールの作成に先立って仕様の視覚化の確立を目指す。

2 形式仕様記述言語 Z

形式的仕様記述言語 Z は、集合論や述語論理など、数学の標準的な概念を基礎としている。特徴として次のことがいえる。

- 意味は明白で、基礎をなす数学に照らし合わせれば常に曖昧さを取り除くことができ、矛盾点の早期発見に役立つ。
- 簡潔さを損なうことなく、正確に表現できる。
- 数学の証明技法の応用が可能で、仕様記述の厳密な証明ができる。

2.1 Z の仕様記述例

簡単な誕生日帳の例を挙げて Z の記述方法を紹介する。

[NAME, DATE]

```
BirthdayBook
known : PNAME
birthday : NAME ↔ DATE
known = dom birthday
```

3 Z 記述の視覚化

Z の仕様で使われる数学的記述を視覚化する。

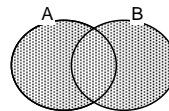
3.1 集合

Z で仕様を書くときに使われる集合の宣言には、所与の集合, free type definition, 集合の変数などが挙げられる。集合を絵にするときは教科書などでもよく使われるように、円で表すことにした。以下の絵で known は NAME の集合であることを表している。



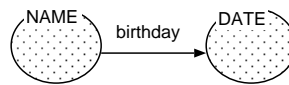
3.2 集合に関する演算

Z で使われる集合の演算には主に、和集合, 差, 積集合の3つが挙げられる。以下は和集合について考えた場合の絵 (AUB) である。色で塗られている部分はそれが演算の結果で得られる範囲であることを示している。



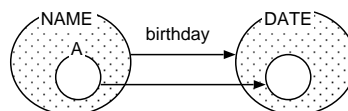
3.3 関数

Z で扱われる関数は部分関数である。関数にはその他にも特別な場合として全域関数, 単射, 全射, 全単射が挙げられる。関数は集合と集合の関係によって表されるので, 部分関数を以下のような絵にする。これは誕生日帳において birthday は NAME から DATE への写像であることを示している。



3.4 関数に関する演算

関数に適用できる演算は主に定義域や値域の減算や限定, 割り込みである。関数も集合であることから, 差や和集合なども適用することができる。以下は誕生日帳から A という NAME の集合を定義域減算した絵である。



4 考察

一目見ただけで全ての人々が共通の理解を得ることは不可能であると考えます。直観的に理解するための目指すべき絵というのは、絵を見せ補足の説明をした後で全ての人々が共通の理解を得られるような絵のことです。ただし説明が多くなるとそれらを覚えることが困難になってしまい、結局理解するのに時間がかかってしまうので、説明は少なければ少ない程よいと考えます。

4.1 静止画に関する考察

冒頭では、「対象者はある程度の数学的知識を持っていることを前提とする」と述べた。ここでいう「ある程度」とは、3章に挙げたぐらいの基本的な数学的記法を知っていることを指す。集合が何かも知らないような人や、逆に数学の知識が豊富な人は対象外とする。

視覚化の対象としているのはソフトウェア開発における依頼者と開発者のコミュニケーションや教育の場面であり、そのどちらの場合でも「ある程度」の数学的知識を持っていると想定し、視覚化してきた。そのため絵の中にいくつか数学の記述を採り入れた。そうすることで、絵から曖昧さや理解の不一致を取り除くことができるし、新しい記述を増やした時よりも見た者が理解しやすいものになる。全てを絵で表現したのでは見る人によってさまざまな理解を持ってしまいますので、共通の理解が得られるような絵を定義するのがとても難しくなってしまう。

また全ての絵には一貫した統一性を持たせた。そうすることでより説明を少なくすることができる。数学的記述の一つについて絵を考える際、それまでと同じ方法で記述すると矛盾や無理が生じてしまうことがある。それによって以前に書いた絵を変更しなければならなくなる。すると、それにとまって全てを書き直さなければならない場合が頻繁に起こる。そういうことの繰り返しによりよい絵を生むと考える。

4.2 述語論理の視覚化

Zの特徴は数学の標準的な概念である集合論や述語論理を基礎としていることであつた。しかしこれまでは、集合論しか視覚化できなかつた。述語をどうやって視覚化することは今後の大きな課題である。

=は演算の時に、よく使われる。しかし=は本当に視覚化する必要があるのだろうか。何度も言うが、視覚化するのは直観的理解の促進であるので、理解しているものをわざわざ視覚化する必要はない。=はそのままで容易に厳密な理解を得ることができる。しかし、=が使われる時は必ず前述と後述がある。その内のどれとどれが等しいかと、聞かれると記述によってはよく分からないことも考えられるので、そういう意味での表現は必要になる。その他にも「かつ」、「または」、「含意」、「同値」として知られている $\wedge, \vee, \Rightarrow, \Leftrightarrow$ などをどう視覚化するか、今後

の課題として残っている。

4.3 動画による視覚化

これまで静止画で仕様を表現してきたが、絵には動きを持った動画がある。静止画で研究を進めて来ると同時に、動画で表すことも検証して来た。動画は、単純に静止画をいくつも組み合わせて作る絵のことなので、静止画1枚を見るよりも、動画を1つ見た方がより深い理解を得ることができる。例えば、和集合のように前の状態から後の状態に変化がある、つまり動きのある操作は動画で表すことがより効果的であることが分かった。これを1枚の静止画で表そうとするのは困難なことで、いいと思って書いた絵も全ての人々が共通の理解を得ることは難しい。逆に動きのないもの、変数の宣言などはその変数が何を保持しているか表すだけで、事は足りるので動きを付けて想像力をかき立てるよりも静止画の1枚に納めた方がすっきりする。スキーマは状態と操作を表すことができる。状態は静止画で表し、操作は動画で表すのがより望ましいと考える。

5 おわりに

Z言語は数学の概念である集合論と述語論理を基礎としている。これまでに、Z記述の全てを視覚化するという目的で視覚化を進め、Zの要素のうち集合論について絵を定義した。しかし定義仕切れなかつた演算がまだ残っている。また、述語論理についてもまだ未定義である。研究を進めていく中で、絵を用いて数学記述を視覚化することで直観的に仕様記述を理解することができることが分かった。絵には静止画と動画があり、表す対象によって使い分けることでより深い理解を得ることができる。今回、定義した絵で万人が共通の理解を得ることができるかと言うとそうとは言い切れない。自分に分かりやすいと思って書いた絵が他人にとっては分かりづらいということは、よくある話である。定義した絵の評価を常に行っていかなければならない。

今後の課題として以下が挙げられる。

- 定義していない演算や論理式の視覚化
- 理解支援ツールの設計、試作

謝辞

本研究を進めるにあたり、一年半御指導いただいた野呂昌満教授に深く感謝致します。また有益なアドバイスをいただいた、張漢明先生、大学院の熊崎敦司さん、池内仁さん、宗宮健仁さん、青木俊介さんに深く感謝致します。

参考文献

- [1] J.M.Spivey: *The Z Notation A Reference Manual Second Edition*, 1992.