5L-6

ドキュメント管理という目で見た時のSmalltalkについて

稲垣 彰[1)]，石橋 勝典[2)]，菅野 文友[2)]
([1)]エルコー㈱，[2)]東京理科大学 工学研究科)

1．はじめに

今日，ソフトウェア開発環境が完全に定着し，ソフトウェア開発におけるドキュメント化も，その立場を確固たるものにしている．しかし，開発思想の違うSmalltalk-80（以下Smalltalk と略す）においてはどうであろうか．

現在，ゼロックス社のSmalltalk 上で，設計支援のシステムを開発中であるが，ドキュメント化に疑問を感じ，今回の考察を行なった．

Smalltalk というソフトウェア開発環境を，ドキュメント管理という視点から考察する．

2．ドキュメント管理について

2．1 一般的なドキュメント

従来からの一般的なドキュメントについて整理すると，次のようになる．

○仕様書について
①基本仕様書
・ＢＳ（Basic Specification）
②機能仕様書
・ＦＳ（Functional Specification）
③設計仕様書
・ＤＳ（Design Specification）
④製造仕様書
・ＭＳ（Manufacturing Specification）

また，開発の流れから考えると，次のような分類が可能である[2]．
①プランニングの経過－設計仕様書
・ＦＳ
・ＧＦ（General Flowchart）
②コーディングの過程－設計書
・ＤＦ（Detailed Flowchart）
・ＣＳ（Coding Sheet）
・ＤＳ
③デバッギングの過程－取扱説明書
・ＯＭ（Operation Manual）
④プログラム・テストの過程－保守書
・ＭＭ（Maintenance Manual）

これらは，基本的にペーパ・ウェアといえる．

3．Smalltalk におけるドキュメントについて

3．1 Smalltalk の環境

ドキュメンテーションとは，「与えられた主題に関する文書の集まり．Ｃ6230」[3] である．つまり，情報が定義されていて，参照できることであるから，そのメディアがペーパである必要はない．

Smalltalk の環境は，次のようにいわれている．「Smalltalk は，視覚的な対話方式を採用したプログラミング環境である．また，Smalltalk は，パーソナル・コンピューティング・ビジョンでいわれているように，ユーザが利用できるシステム内のすべての構成要素を見たり操作したりする時，意味のある形で表示できるように設計されている……」[4]．文章の集まりとはいえないかもしれないが，同等の機能を果しているといえる．

Smalltalk といえども，開発の要求定義は不可欠である．しかし，それ以降では，その設計方法によって変わってくる．Smalltalk は，プロトタイピングに大きな威力を発揮する．机上で論理を十分確認した上で作っていくのではなく，モジュールごとに確認しながら作っていく形態をとっている．この方法では，ペーパとしては残りにくい．これがまた，Smalltalk を覚えにくいものにしている．

3．2 Smalltalk が持っている機能

Smalltalk が保持している特徴の中で，ドキュメントに関連したクラスをまとめる．

①ブラウザ(Browser)
テキストエディタを使用してメソッドを定義したり，修正したりするビュー．
・クラスの階層構造を示す．
・あるオブジェクトへのメッセージインターフェースの参照．
・オブジェクトの内部状態に関する情報の参照．
・いろいろな変更管理，ｅｔｃ．

②インスペクタ(inspector)
・オブジェクトの内部状態に関する情報の参照．
・オブジェクトのインスタンス変数の参照，
ｅｔｃ．

これらの中に，まだたくさんのコマンドがある．

3．3 結論

Smalltalk といえども，初めの地固め（基本仕様，機能仕様のレベル）が必要である．しかし，設計，製造，保守に関してはマシン一体という環境自身がドキュメント内蔵といえる．

ペーパ・ウェアとしては弱い．しかし，いちいち紙に打出す必要がないというポリシーがあるためで，マシンの中にドキュメントが入っていると考えると，

①食い違いが起こらない．システムを直すことがそのままドキュメントを直すことになる．
②いつでもすぐ見ることができる．
③必要なものを探すことができる．
④変更の履歴が残っている．

An evaluation of Smalltalk-Oriented document administration

Akira INAGAKI[1)] , Katsunori ISHIBASHI[2)] , Ayatomo KANNO[2)]
[1)]ELCO CO.,LTD. , [2)]Faculty of Engineering , Science University of Tokyo

という，理想的なドキュメント形態を取っているといえるかもしれない．
一方，ペーパ・ウェアの弱さは，デザイン・レビューを実行しにくいものにしていると思われる．

4．ペーパ・ウェア・ドキュメントについて

Smalltalk には，ソースプリントの機能があるが，とても見やすいとはいえない．今回はパソコンを利用し，漢字出力とキャラクタ罫線を生かして，ソースリストを出力してみた．以下に例を示す．

```
TModel subclass: #TSM
    instanceVariableNames: 'mono point oldPoint '
    classVariableNames: 'PicNameCollect PictureCollect '
    poolDictionaries: ''
    category: 'Trans2'!


!TSM methodsFor: 'initialization'!

initialize
    "イニシャライズ"

    point ← nil.!

initialize:ss
    "イニシャライズ:絵の初期化"

    | nam |
    nam ← 'forms/xx',ss,'.form'.
    picture ← Form readFrom: nam.! !

!TSM methodsFor: 'menus'!

aMenu
    "メニューの設定"

    ↑ ActionMenu
        labels: ' accept '
        lines: #()
        selectors: #(accept)! !

!TSM methodsFor: 'accessing'!

act: aPoint
    | xx yy |
    "選択された点を計算する"

    point ← (aPoint - (40@40) grid: 80@80) / (80@80) + (1@
    xx ← self point x .
    yy ← self point y .
    self monoset:((self PicNameCollect at:yy) at:xx).
    self mono displayAt:100@100.

    self changed: #reverseForm!

"-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

TSM class
    instanceVariableNames: ''!


!TSM class methodsFor: 'instance creation'!

new
    "インスタンス生成"

    ↑ super new initialize! !

!TSM class methodsFor: 'class initialize'!

initialize
    | tmpCollect tmpCollect2|

    "TSM initialize"

    PictureCollect ← OrderedCollection new.
    PicNameCollect ← OrderedCollection new.
    1 to: 4 do: [ :y |
        tmpCollect ← OrderedCollection new.
        tmpCollect2 ← OrderedCollection new.
        1 to: 5 do: [ :x |
            tmpCollect add: (Form readFrom: 'forms/xx'
            tmpCollect2 add: ('xx', x printString , y
        PictureCollect add: tmpCollect.
        PicNameCollect add: tmpCollect2.]! !

TSM initialize!
```

図４．１　Smalltalkの出力

```
クラス名          : TSM
スーパークラス名  : TModel
インスタンス変数名 : mono point oldPoint
クラス変数名      : PicNameCollect PictureCollect
プールディクショナリ:

カテゴリ名        : Trans2
カテゴリ内の他のクラス名 : TSC
                    TSU

< クラスメソッド >
   【instance creation】
      new
   【class initialize】
      initialize

< インスタンスメソッド >
   【initialization】
      initialize
      initialize:ss
   【menus】
      aMenu
   【accessing】
      act: aPoint

┌<< TSM >> ----------< クラスメソッド > --------
├┬< new > 【instance creation】
│ │ "インスタンス生成"
│ │
│ └  ↑ super new initialize
│
├┬< initialize > 【class initialize】
│ │ | tmpCollect tmpCollect2|
│ │
│ │ "TSM initialize"
│ │
│ │ PictureCollect ← OrderedCollection new.
│ │ PicNameCollect ← OrderedCollection new.
│ │ 1 to: 4 do: [ :y |
│ │       tmpCollect ← OrderedCollection new.
│ │       tmpCollect2 ← OrderedCollection new.
│ │       1 to: 5 do: [ :x |
│ │           tmpCollect add: (Form readFrom: 'forms/xx', y printStrin
│ │g , x printString , '.form').
│ │           tmpCollect2 add: ('xx', x printString , y printString).]
│ │
│ │       PictureCollect add: tmpCollect.
│ └        PicNameCollect add: tmpCollect2.]
└

┌<< TSM >> ---------< インスタンスメソッド > --------
├┬< initialize > 【initialization】
│ │ "イニシャライズ"
│ │
│ └  point ← nil.
├┬< initialize:ss > 【initialization】
│ │ "イニシャライズ:絵の初期化"
│ │
│ │ | nam |
│ │ nam ← 'forms/xx',ss,'.form'.
│ └  picture ← Form readFrom: nam.
│
├┬< aMenu > 【menus】
│ │ "メニューの設定"
│ │
│ │ ↑ ActionMenu
│ │       labels: ' accept '
│ │       lines: #()
│ └        selectors: #(accept)
```

図４．２　変換した出力

5．おわりに

今後のSmalltalk に対する期待としては，新しく覚えようとする人たちが理解しやすいように，メッセージの日本語化が強く望まれる．

参考／引用文献

［１］　石部公男,青井精一:「システム設計入門」,同文館,(1986).
［２］　菅野文友:「ソフトウェア・エンジニアリング」,日科技連出版社,(1979).
［３］　三浦新,津田義和,狩野紀昭,大橋靖雄:「ＴＱＣ用語辞典」,日本規格協会,(1985).
［４］　Adele Goldberg,David Robson:「SMALLTALK-80 -言語詳解-」,ｵｰﾑ社,(1987).
［５］　Adele Goldberg:「SMALLTALK-80 -対話型プログラミング環境-」,ｵｰﾑ社,(1987).