

岡本 環

富士通コミュニケーションシステムズ株式会社

1. はじめに

私達が開発している伝送装置は、組込み型ソフトウェアを内蔵し、外部から入力されたコマンドに従って制御される。従来は、装置に入力するコマンド仕様(以下、コマンド仕様書)を元に、ソフトウェアを手作業で開発し、試験工程では実際にコマンドを入力して仕様通りに動作しているかどうかを確認していた。しかし、短期間での開発を要求されるようになった現在、全ての作業を手作業で行うことは開発遅延や、試験カバレッジの低下、ケアレスミスによるソフトウェアの品質低下の原因になり兼ねない。そこで、ソフトウェア開発作業全般の手作業を減らし、開発作業の自動化を実施した。

従来

コマンド仕様書

```

USER-LOGIN:<TID>:<AID>:<CTAG>:<POS>;
        user id    password
USER-LOGOFF:<TID>:<AID>:<CTAG>;
        user id
ED-SYS:<TID>:<AID>:<CTAG>:<Keyword=Domain>;
        null      AUTOP=Y/N
        AISPENA=Y/N
        AAT=0/25/10
  
```

手作業で作成



手作業

試験データ

```

USER-LOGIN:TID1:FWITSU:CTAG:FWITSU;
ED-SYS:TID1:CTAG:AUTOP=Y;
ED-SYS:TID1:CTAG:AUTOP=N;
ED-SYS:TID1:CTAG:AAT=0;
ED-SYS:TID1:CTAG:AAT=25;
ED-SYS:TID1:CTAG:AAT=10,AISPENA=Y;
~
  
```

手作業で作成

ソースの構成は同じ

ソース

USER-LOGINのソース
 コマンド"USER-LOGIN"は<AID>にuser idを入力できる。
 コマンド"USER-LOGIN"は<POS>にpasswordを入力できる。

ED-SYSのソース
 コマンド"ED-SYS"は<AID>は省略のみ許す。
 コマンド"ED-SYS"は<Keyword=Domain>にAUTOPを入力できる。
 そのときのDomainは"Y"または"N"である。
 ~

2. 作業の自動化

コマンド仕様書は、“文章”で記述されている為、ツールで使用する“データ”として扱いにくい。そこで、コマンド種別、入力可能なパラメータ等をキーにして、“文章”を“単語”に分割して管理し、データベース化する。データベースは、Excel を使って作成し、ツールは Excel の VBA で作成した。

このデータベースから自動生成可能になったものは、

①ソース、②コマンド仕様書、③試験データである。

コマンドデータベース

コマンド種別	AID	POS	Keyword	Domain
USER-IN	user id	passwd	なし	なし
USER-OFF	user id	なし	なし	なし
ED-SYS	省略	なし	AUTOP	Y/N
			AISPENA	Y/N
			AAT	0/25/10

入力

コマンドデータベース
(Excel)

出力

ツールA → ソース
 ツールB → コマンド仕様書
 ツールC → 試験データ
 (全てVBA)

2. 1. ソースの自動生成

入力されたコマンドが仕様と一致しているかどうかを判断するソースは、コマンド毎に入力できるパラメータが異なるので、コマンド毎に存在する。しかし、判定内容は異なってもソースの構成は同じであるので、判定内容の部分にデータベースから読み出したデータを埋め込み、ソースを生成することが可能である。

データベース

種別	AID	POS	Keyword	Domain
ED-SYS	null	なし	AUTOP	Y/N
~				

Baseソース

コマンド"aaa"は、<AID>に"bbb"を入力できる。
 コマンド"aaa"は、<Keyword=Domain>に"ccc"を入力できる。
 そのときのDomainは"ddd"である。

ED-SYSのソース

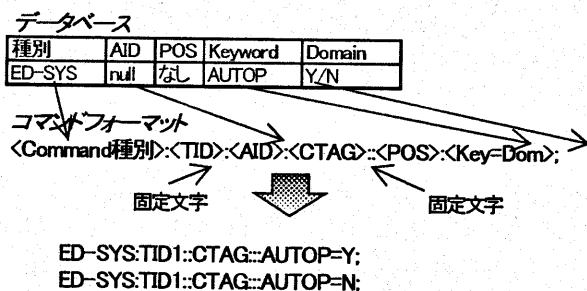
コマンド"ED-SYS"は、<AID>に"null"(省略)を入力できる。
 コマンド"ED-SYS"は、<Keyword=Domain>に"AUTOP"を入力できる。
 そのときのDomainは、"Y"または"N"である。

2. 2. コマンド仕様書の自動生成

コマンドの基本フォーマットに従い、データベースにあるデータを埋め込み、コマンドの文字列を生成する。

2. 3. 試験データの自動生成

コマンドの基本フォーマットに従い、データベースにあるデータを様々に組み合わせながらつなぎ合わせて、コマンド文字列を成形する。これをテキストファイルに出力し、自動試験で読み込ませる試験データに利用する。



3. 効果

下記に、作成した各ツールの規模、作成に要した工数、ツールによる工数削減効果をまとめる。

ツール	ツールの規模	ツール作成工数	生成物の規模	実作業工数
ソース生成	7.6 kstep	181.7 人H	52.4 kstep	726.8 人H 1.0 人H
コマンド仕様書生成	0.9 kstep	39.5 人H	328 コマンド	79 人H 0.1 人H
試験データ生成	0.4 kstep	11.4 人H	7552 コマンド	158 人H 0.1 人H

※ 実作業工数欄の上段: 改良前(従来) 下段: 改良後

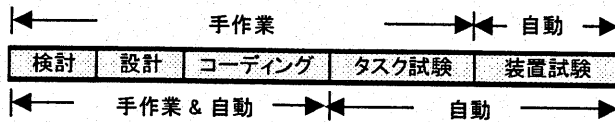
試験工数	
手動試験(従来)	1048.8 人H
自動試験	79 人H

上表の通り、作業を自動化することによって、作業者の負荷を軽減することが出来た。同時に、自動作業をしている間に、他の作業を手がけることが出来るので、作業の幅が広がり、かつ手作業によるケアレスミスがなくすることが出来るので、ソフトウェアの品質向上につながることでもできた。また、一つの入力情報(データベース)から複数の出力情報(アウトプット)を得ることが出来るので、各アウトプットにおける記述内容の矛盾・不一致を容易、かつ確実になくすることができた。開発工程の面から見ても、下図のように自動化できる作業が増えて

いる。

これらにより、今後の開発作業に寄与するところが大い。

従来の作業



4. 今後の課題

各アウトプットの自動生成ツールは既に運用済みであり、特にコマンド仕様書自動生成ツールについては他プロジェクトでも運用している。しかし、ソース自動生成ツールは、他のプロジェクトで使用するには多少手を加える必要があり、現状はプロジェクト間での汎用性が低い。しかし、非常に効果的なツールであるので、汎用性を高め、他のプロジェクトでも容易に使用できるよう、改良していきたい。