

システム構築支援ツールの試作

7G-6

坂田 豊
NTT情報通信網研究所

小田 良雄 安達 信一
NTTソフトウェア株式会社

1. はじめに

オンラインシステムのインストール作業の1つにシステムの動作条件を定義した制御文ファイルの作成がある。この制御文ファイルの規模は通常のオンラインシステムで数MBとなり、正しく設計するには専門家の持つ知識が不可欠である。

専門家の知識とは個々の製品の数千ページにわたる膨大なドキュメントやドキュメントに記載されないような各製品間のノウハウであり、これらを修得するには数年を費やす。一方、専門家が関係しないインストールの場合、1システムの構築作業が5~10人で3~4ヶ月に及ぶこともある。

ノウハウをドキュメントで管理する方法もあるが、ドキュメントの持つあいまい性のため品質の低下が想定され、またメンテナンスも行われなくなる。

上記の問題点の解決をねらい、専門家の持つ知識を体系化し、技術者間で共用できる様にツール化し試作したのが本ツールである。

本書は

- ・ 専門家の知識とは何か
- ・ その知識をどの様に体系化(表現)するか
- ・ 試作ツールの構成および特徴
- ・ 使用結果と今後の課題

の順で書かれている。

2. 専門家の知識について

制御文ファイルの作成では数千のパラメータを設定する必要がある。そこでの専門家の知識は次の様なものがある。

- ① 多数のパラメータを分類^{*1}する知識
- ② パラメータ間の相関関係をチェックする知識
- ③ 可変パラメータの設定値を決定する知識

本ツールでは①と②の知識が体系化される。

*1 パラメータは以下のように分類される

タイプA: 定型的なものでシステム設計者が意識しない
タイプB: 性能条件や運用条件によって変化するもので安全サイドの固定値を設定しておけば動作は保証できる
タイプC: システムの機能を定義するものでシステム設計者が直接意識する

3. 知識の体系化

専門家の知識を以下の様な形で体系化し、表現した。

(1) ひな型

パラメータの分類結果に関する知識をテキスト形式の制御文ファイル中に集約する。

制御文ファイルの形式は図1の通りであり、システム対応に設計するパラメータは(タイプCのパラメータ)は'\$'記号で囲み、パラメータ名を付与した。

その他の部分は各製品UPの入力形式のまま管理されている。

ツール上での形は図1の様になる。

```
FMT=P=$00$, 'T', 'N', 'N'
FMT=P=$01$, AA61,00,$02$, $03$ LINE01
```

図1. ひな型の例

(2) デザインシート

タイプCのパラメータを表形式に整理したものである。整理に当たっては、製品間で相関のあるパラメータを1つのテーブルにまとめる事により、設計者に分かりやすくすると共に製品間の整合ミスを無くしている。

シート名: TABLE01

ファイル名 \$00\$	ファイル形式 \$01\$	ファイル容量 \$02\$	データ形式 \$03\$
FILE00	MB	500	SS
FILE01	MB	500	SS
FILE02	MB	500	SS
FILE03	NMB	500	DD
FILE04	NMB	500	DD

図2. デザインシートの例

(3) 変換ルール

デザインシートで定義されたデータをどのように制御文に埋め込むかを定義する。

```
COPY LINE01 E(TABLE01): (a)
REP UPDATE LINE01:$00$ D(TABLE01:$00$): (b)
REP UPDATE LINE01:$01$ D(TABLE01:$01$):
REP UPDATE LINE01:$02$ D(TABLE01:$02$):
REP UPDATE LINE01:$03$ D(TABLE01:$03$):
```

図3. ルールの例

図3の①はひな型の修正位置を示し、②はデザインシート名と項目名を示している。

4. ツールの構成

4-1. 設計目標

ツールの設計に当たっては下記の設計目標を置いた。

- (1) 体系化した知識（ひな型、デザインシート、ルール）は変更が生じたときは容易に修正出来る事。
 制御文のパラメータとはもともとシステム対応に設計するためである。このため固定パラメータ（タイプA、B）とシステム対応に設計するパラメータ（タイプC）の境界は適用事例を重ねて修正する必要がある。
- (2) 体系化した知識の信頼性、品質を容易に確認するためにそれらを利用して制御文ソースを生成する機能を実現する。

4-2. ツールの構成

ツール全体はパソコン上に構築した。（図4）

- ① 固定パラメータとシステム対応に設計するパラメータを管理するひな型ファイルは、テキスト形式で作成して市販エディタで作成・修正する。
- ② デザインシートデータによるひな型の修正規則を管理するルールファイルは簡単な言語で記述されており、ひな型ファイルと同様に市販エディタで作成・修正する。
- ③ デザインシートはテーブル形式でできており、そのフォーマットに従ってシステム対応のデータをパソコン入力できる必要がある。このためフォーマット定義機能を備えた市販データベースプログラムを使用する。
- ④ 上記の3種のファイルを使用してSG制御文のソースを作成するのが図3中の変換プログラムである。変換プログラムはルールファイル中のルールを順次解釈してデザインシート中のデータをひな型ファイルにアップデートしてゆく。

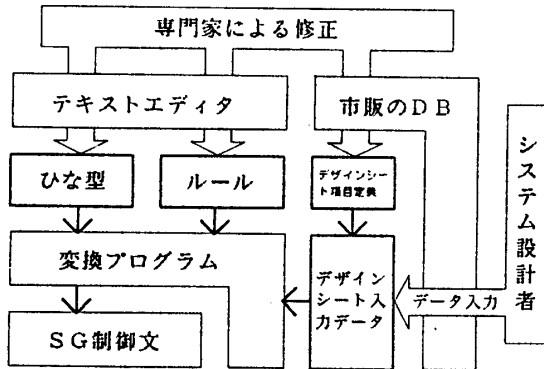


図4. ツールの構成

4-3. 変換ルールのメンテナンス性と簡易言語構成

- (1) 簡易言語のわかり易さ
 変換ルールは下記の2機能をベースとし、数個の拡張命令で構成した。
 ① デザインシートのエントリー分ひな型行をコピーする機能。（例：図3の(a)のCOPY命令）
 ② デザインシートの指定項目をひな型行の指定位置にアップデートする機能。（例：図3の(b)のUPDATE命令）

また構文は下図の様な単純な形式とした。

命令部 + ひな型指定 + デザインシート指定

ひな型指定：（制御文ファイル名：行番号：項目名）
 デザインシート指定：（デザインシートテーブル名：項目名）

- (2) 非手続き性
 変換ルールの手続き性が強くなると修正が困難になる為、言語にGOTO文を無くすと共にIF文も制御文の特殊ケースでの使用を想定した限定した機能とした。

5. おわりに

本ツールを実システムインストール作業に適用し下記の効果を確認している。

- (1) 従来、各製品毎に全パラメータについてドキュメント調査・設計を行ってきたが、本整理によりシステム毎に設計が必要なパラメータは約1/6に減少でき設計作業の効率化が計れている。
- (2) 本ツールを使用したインストール作業で検出した不具合は、ひな型、デザインシート、ルールをそのつど修正して行く形でインストール作業の信頼性が回を重ねる毎に高くなっている。

本ツールで実現している制御文設計方法はシステム建設の初期の試験環境の作成には有効であるが、総合試験さらにサービス開始とシステム建設の工程が進むと今まで固定にしていたパラメータの再設計が必要な事も多い。この部分の設計方法が今後に残された問題である。

*： 獨 鈞、 渡 辺、 坂 田 の パ タ ー ン 化 シ ス テ ム 構 築 支 援 ツ ー ル の 適 用 結 果 報 告
 情報処理学会第44回全国大会