論文・解説

17

設計出図データチェックシステムの開発
Development of Engineering Release Data Validation System

中本正義*1　川崎俊司*2　宮原美智子
Masayoshi Nakamoto　Shunji Kawasaki　Michiko Miyahara

要　約

設計出図情報の精度を維持するため出図担当者がデータをチェックする必要があり，出図処理の効率に大きな影響を与えていた。今回，電子出図システムの一機能として出図情報の自動チェックシステムを開発，ベテラン出図担当者の出図情報チェックノウハウをルールとして文書化・定式化し，その定式化したチェックルールを解析・評価するチェックエンジンを開発した。設計出図データチェックの自動化が可能になり，出図期間の短縮・出図情報の精度向上に寄与するのみでなく，チェックルールとチェックエンジンとを分離したことにより，ルールの変更・追加に柔軟に対応可能になった。本稿では，電子出図システムの一機能として開発した設計出図データチェックシステムについて，その概要を紹介する。

Summary

Conventionally, engineering release data, such as parts structures known as “Bill of Material,” was checked by people in charge to maintain the accuracy of data. We improved the efficiency of this operation by newly developing an automatic check system as a chain of electric & electronics engineering release system. We documented human knowhow as rule, and developed a check-engine to evaluate/analyze the rules. The automatic check of engineering release data contributed to reduce the time and improve the accuracy of engineering release. Together, the division of the check rule and the check engine made it easy to add/revise rules. This paper introduces the outline of newly developed engineering release check system.

1．はじめに

一般に，設計出図時には図面以外に部品構成・部品情報・使用条件等さまざまな情報が出図される。従来，紙による出図を行っていた時には，提出された紙情報を出図管理部門のベテラン担当者がチェックし，出図情報の整合性や入力ミス等を発見し設計者にフィードバックすることで，出図情報の精度が保たれていた。

開発期間短縮を目的にデジタル化が進み，出図情報も大幅に変化・対応している中，チェック時間や発見したミスを修正する，いわゆる手戻りの時間損失が浮き彫りになってきた。そこで，出図情報のチェックを設計者が情報を作成・保存する時や提出する時に，自動で行うことができれば，ミスの早期発見・修正が可能になり，出図の早期化と品質精度向上が期待できると考えた。

今回，出図管理部門のベテランのデータチェックノウハウをまず言葉でルール化し，それをコンピュータで処理できる形に再構成して，ルールDB（データベース）として登録した。次にそれらのルールを解析し，出図データに適用して結果をフィードバックするエンジンを開発することで，設計出図データの自動チェックシステムとして適用した。

2．システム構成

2.1　電子出図システムとの関連

電子出図システムは，設計者の出図情報デジタル作成支援を行い，上司承認を経て出図管理部門に提出する基本のWebベースシステムである。

設計出図データチェックシステムは，電子出図システム

＊1，2　車両レイアウト・CAD部
Vehicle Layout Engineering & CAD Dept.

の一機能として追加され，作成・提出される出図情報の整合性・入力ミスのチェック等を行って，出図期間短縮・出図情報の品質向上に寄与している。

2.2　データチェックシステムのシステム構成

Fig.1に本システムの構成概要を示す。データチェックエンジンと呼んでいる部分が，データチェックルールの解析・出図データへのルールの適用の主要部分を占めている。

電子出図システム以外のシステムからの利用も可能になるよう設計しているため，チェック用の出図情報をデータチェックシステム内に一時保持する形になっている。

外部比較DBは，弊社の場合，MIDASと呼ばれる設計情報を生産情報としてホストコンピュータ上に保存しているデータを使用している。

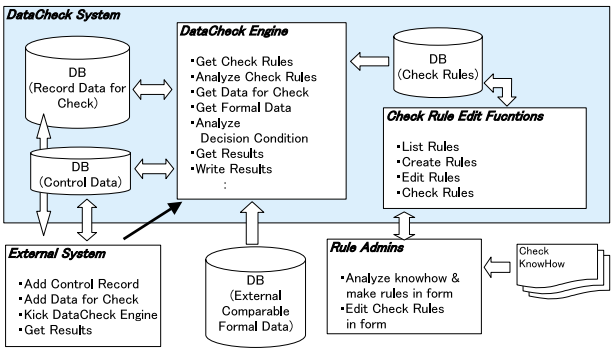

Fig.1　DataCheck System Structure Outline

3．データチェック処理

3.1　データチェック処理の流れ

(1)　チェック対象データの格納

複数のシステムから利用できることを考慮し，チェック対象データはシステム内に一時保存する。ただし，エラー発生時，元データにもエラー情報が反映できるように必要な元データ情報もあわせて保持する。

また，複数のチェック要求が，並行して発生する可能性があるため，制御用の管理DBのレコードも作成する。

(2)　データチェックエンジンの起動

チェックしたいデータ量に応じてオンライン処理とオフライン処理とを選択起動できる。簡単な少数の入力チェックの場合を除いて，データチェック処理に時間がかかるため，基本的にはオフライン処理を行う方が良い。

データチェック結果は基本的に対象レコードごとに記録されるとともに，全体として不適合の件数情報を記録する。

3.2　データチェック処理結果の取得

(1)　オンライン処理

オンライン処理では，Webベース処理になる。そのため処理結果は，該当データの該当レコードに記録されるとともに，Web上にも結果が表示され，整合性や入力チェックの結果がすぐ得られることになる。

(2)　オフライン処理

オフライン処理では，Webベース処理から一旦離れ，サーバ上で独立して処理される。ユーザには処理終了時，あらかじめ登録されたメールアドレスに，処理終了および処理結果の通知メールが発信される。ユーザは，そのメールを待つか，Web上で進捗チェックの機能を用いて，処理完了かどうかおよび，エラーの有無を確認できる。

出図情報の提出時のチェックの場合には，エラーがなければそのまま提出処理を継続し，ユーザがあらためて提出処理を行う必要はない。エラーがある場合は，ユーザにその旨のメールが届き，ユーザが修正後，再提出を行う。

4．データチェックルールの表現

4.1　データチェックルールの定形表現

データチェックルールをコンピュータで解析できる形にいかに定式化するかということが，一番大きな課題である。このシステムでは一部の例外を除き，主として次の基本要素を組み合わせることで，ルールの表現を実現している。

(1)　設計出図情報の取得（複数レコード可）

検索条件を指定して，部品構成・部品情報等の設計作成情報を取得する。

(2)　参照情報の取得（複数レコード可）

検索条件を指定して，登録済み情報等を取得する。

(3)　繰り返し制御

あまり複雑な制御は行わず，(1)(2)等で得られた複数の情報に対して，以下を繰り返す制御を行う。

(4)　演算式

得られた情報に対して，エラーとするかどうかの判定を行う式を記述する。演算で使用できる記号，演算子および関数は次の通りである。

(,)，＝，<，>，<=，>=，<>，IN，AND，OR，NOT，＋，－，＊，/

CDate，CInt，Mid，Right，Left

例えば，あるデータD1～D4についてD1とD2が一致しかつD3とD4が一致しない時エラーにしたい場合には，次のように記述し，式の値がTrueになった時エラーとする。

（D1＝D2）AND（D3<>D4）

4.2　データチェックルールの例

実際にどのようにチェックが行われるか，具体的な事例で説明する。以下の例は，設計者が部品を新設し，部品番号を設定した時，その部品番号が既に使用されたものであれば，他の部品との部品番号のダブりになるためエラーを発生させ，部品番号の変更を促す状況を想定している。

(1)　文章によるルールの表現例

ルールを文章化すれば，次のように表現される。

「新設と宣言された部品番号が，登録済み情報に存在すればエラー」

(2)　ルールの定型表現の例（ここでは，文章で表現）

1） 新設部品として設定されている情報を抽出
　実際には，あらかじめ登録されている出図情報一覧から抽出したい情報（例では部品番号）を選択し，抽出条件として新設であることを設定する。
2） 指定された部品番号を登録済み情報の中で検索
　あらかじめ登録されている，登録済み情報一覧から検索すべき情報（例では部品番号）を選択し，抽出条件として1）で取得した情報を設定する。
3） 登録済み情報が存在すればエラー
　2）の検索結果で部品が存在すればエラーとする。
演算式で書けば，2）の検索結果をD2として
　　D2<>"
と表現され，式の値を評価してTrueであればエラーとする。

5．データチェックルールの解析と式の評価

データを取得する部分は，テーブル・フィールド・検索条件を設定すれば取得できる。演算式の評価は，構文解析を行って要素に分解した後で，行う。

5.1　構文解析

構文解析には，演算子順位構文解析法を使用した。
使用した演算子順位表をTable 1に示す。
式の文字列から値や演算子等の要素を切り出し，演算子順位表を基に構文解析して，一つずつの要素に分離するとともに順位付けを行う。構文解析の詳細は，インターネット上にも情報が多いので，そちらに譲ることとする。

Table 1　Order of Operators

	&	=	+	−	*	/	(	)
&	G	L	L	L	L	L	L	G
=	G	G	L	L	L	L	L	G
+	G	G	G	G	L	L	L	G
−	G	G	G	G	L	L	L	G
*	G	G	G	G	G	G	L	G
/	G	G	G	G	G	G	L	G
(	L	L	L	L	L	L	L	E
)	G	G	G	G	G	G	E	G

&：AND, OR, IN
G：Greater than, L：Less than, E：Equal

5.2　式の評価

構文解析の結果は，逆ポーランド記法（演算子を非演算子の後に記述する方法）になっている。逆ポーランド記法では，計算にスタックを用いるのが一般的である。スタックとは，LIFO（後入れ先出し）式のデータ形式で，データは最初から積み重ねられていき，取り出す時は一番最後に入れられたデータから順に取り出される。式の値を評価するには，単純に構文解析の結果に従って実際の値をスタックに貯めていき，演算子が来たらスタックから値を取り出して演算し，結果をスタックに貯めるという操作を繰り返せばよい。

5.3　構文解析と式の評価の例

例えば，4.1であげた次式の場合
　（D1＝D2）AND（D3<>D4）
構文解析結果は次のようになる。
　D1，D2，＝，D3，D4，<>，AND
式の評価は，次のようなステップで行う。
(1) D1をスタックに入れる
(2) D2をスタックに入れる
(3) ＝なのでスタックから値を取り出し（この場合，D1，D2），評価結果をスタックに入れる（R1）
(4) D3をスタックに入れる
(5) D4をスタックに入れる
(6) <>なのでスタックから値を取り出し（この場合，D3，D4），評価結果をスタックに入れる（R2）
(7) ANDなのでスタックから値を取り出し（この場合，R1，R2），評価結果をスタックに入れる
(8) 最後の値が式の値となる

6．まとめ

ベテランのチェックノウハウを文書化し，更に，定式化して解析適用可能な形に発展させ，設計出図情報の自動チェックに道を開くことができたと確信している。今回の定式化の手法だけでは，結果的に大変長いルール記述になって表現困難になるケースもある。そのようなケースには，専用の解析関数を用意して適用するという手法も取り入れている。この場合，関数化してしまった部分のルールの変更は困難になってしまうという問題があり，今後，ユーザ定義関数を用意することが必要になると考えている。
また，今回の開発では，情報システム部門ではなく，出図管理部門に在籍するシステム開発チームが開発を担当した。オンサイト開発（ユーザ部門に常駐での開発）という形により，業務担当者とシステム開発担当者との間で密なコミュニケーションが取れ，システム開発が大幅に効率化されたと確信している。基幹システム以外のシステム開発の一つの形として，オンサイト開発という方向もあるのではないかと考えている。

■著　者■


中本正義


川崎俊司