

業務ルールベースの仕様記述を実現する統合CASE：
AA/BRMODELLER

2M-4

冲山 智

佐野 慎一

藤川 泰之

富士通株式会社 SDAS開発部

1. はじめに

統合CASEツールAA/BRMODELLERは、業務の仕様を宣言的なルール（業務ルール）で表現することにより、保守性、品質、生産性の向上を可能にした。本稿ではこの業務ルールによる仕様記述の手法とそのためのものである。

2. 仕様記述の問題

利用者と開発者間で仕様に対する認識を一致させるため、仕様記述には、厳密性、網羅性、そして、開発者と利用者双方からの理解容易性が同時に求められる。しかし、このような仕様記述を実際に作成し保守し続けることは容易でない。実現方法を指定する詳細設計書あるいはプログラムソースでは、技術的な知識を持たない利用者には理解が難しいし、利用者用の仕様書をより詳細に記述するという方法では、保守コストが急増しこれにも限界があるからである。

3. 仕様記述のための戦略

AA/BRMODELLERでは、上記の問題を解決するために、次のような戦略を取っている（図1）。

●業務仕様と制御仕様の分離^{1) 2)}

仕様を、業務の仕様に依存して決まる部分（業務仕様）と、実現方式に依存して決まる部分（制御仕様）

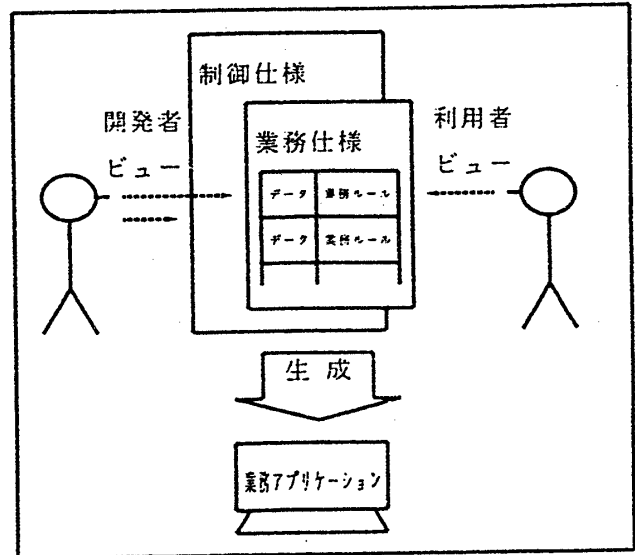


図1. AA/BRMODELLERの戦略

）に分離する。業務仕様の記述は、実現方式に依存しない形式で表わし、技術的な知識がなくても業務仕様を理解できるようにする。

●宣言的なルールによる業務仕様記述

業務仕様を宣言的なルール（業務ルール）で記述することにより、利用者に理解しやすい仕様表現を実現する。業務ルールは、機能毎にまとめるのではなく、データ毎にまとめて整理する。これにより、見通しのよい、変更に対応できる仕様記述を実現し、保守性を向上させる。

●業務ルールからのプログラム生成^{3) 4)}

業務仕様を表現する業務ルールからプログラムを直接生成する。これにより、生産性と品質の向上を可能にする。

次に、業務仕様を記述する単位となる、業務ルールについて説明する。

AA/BRMODELLER: An Integrated CASE Tool with a Business Rule-Based Specification Technique.
Satoru Okiyama, Shin'ichi Sano & Yasuyuki Fujikawa. Fujitsu Ltd. SDAS Development Department.

4. 業務ルール

業務ルールは、静的データ構造に関するものと、プロセスに関するものの二種類に分けられる。以下ではこれらの業務ルールにより、業務仕様がどのようにして宣言的に表現されるかを説明する。

●静的データ構造に関する業務ルール

静的なデータ構造を規定する制約条件は、業務を特徴づける重要な仕様と言えらる。例えば、「従業員番号は10000以上90000以下である」、「商品サイズは、"S", "M", または, "L" である」などである。これらの制約条件を表現する業務ルールは、関連するデータ定義の一部として記述する(表1)。

従業員番号	@ ≥ 10000 かつ @ ≤ 90000
商品サイズ	@ = "S" または @ = "M" または @ = "L"

表1. 静的データ構造に関する業務ルールの例

このようにして定義されたルールは、そのデータを使用している各プロセスに取り込まれ、入力チェックなどのロジックを生成するために利用される。

●プロセスに関する業務ルール

プロセスの業務仕様は、事前条件と事後条件(precondition, postcondition)として表現可能と考えられる。例えば、「受注入力プロセス」において「商品の受注数は、その商品の在庫数よりも多くてはならない」、「出荷プロセス」において「商品の在庫数は、出荷前の在庫数よりも、出荷数だけ減っていないなければならない」などである。これらの制約条件は、プロセスで使用するデータ項目に対するチェックのルールや更新のルールとして定義する(表2)。定義された情報は、入力チェックやDBの更新ロジックを生成するために利用する。

(チェックのルール)

データ項目	処理	条件
受注数	エラー	@ > 在庫数

(更新のルール)

データ項目	処理	条件
在庫数	@ - 出荷数	

表2. プロセスに関する業務ルールの例

これらの業務ルールは、利用者により直接検証され、そのままプログラムになるため、品質の高いアプリケーションの開発が可能である。

また、業務仕様の変更が発生した場合には、制御仕様は無視して、該当する業務ルールのみを変更すればよい。すなわち、業務ビュー上での保守が可能になり、保守性を向上させることができる。

4. 結論と課題

業務ビュー上での業務ルールによる仕様記述手法により、アプリケーションの保守性、生産性、品質の向上が可能となった。今後は、蓄積されつつある業務仕様記述の資産を効率的に再利用するための枠組みが必要となろう。

参考文献

- 1)吉岡他：業務モデル再利用の実際。
ソフトウェア再利用シンポジウム，情報処理学会
1992.
- 2)橋本：ソフトウェア開発の自動化技術：CASE
FUJITSU, 44. No2, 1993.
- 3)橋本他：SDAS統合CASEによるシステム開
発。FUJITSU, 45. No3, 1994.
- 4)佐野他：複数の言語に対応したプログラム自動生成を
実現する統合CASE：AA/BRMODELLER。
情報処理学会第51回全国大会論文集，1995.