

金融機関におけるデータベースシステムと 導入に当たっての問題点

野崎 英一
(富士銀行)

I [データベースシステム導入の必要性]

1. 構勢変化への対応

- (1) 情報利用のタイミング改善 → 意思決定の迅速化
- (2) 使い易い情報の提供 → 経営レポート作成負担の軽減
- (3) 弾力的な情報の提供 → プラニングの向上
- (4) 要求ベースの情報提供 → ユーザー自身による利用
- (5) 総合的・例外的情報の提供 → 業務横断的データの利用

2. 人件費の上昇

- (1) システム開発コストの削減 → 最少のマンパワーで最大の効果
- (2) メンテナンスコストの削減 → 初率的なシステム維持

3. 情報量の増大

- (1) データの重複管理の排除
- (2) データ収集コストの削減
- (3) データの保全
- (4) データの信頼性
- (5) レス・ペーパー化 → ファイリング効率

4. 組織の巨大化

- (1) 情報利用体制の確立
- (2) 共通データの利用

II [金融機関におけるオンライン適用分野]

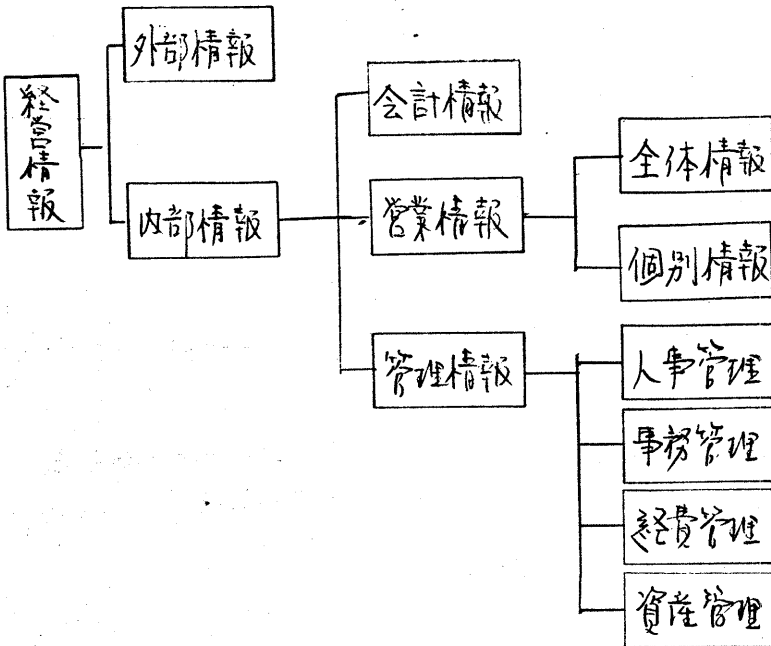
1. 事務処理におけるオンライン

- ・ 全店全科目業務処理
- ・ 顧客管理

2. オンラインによる情報利用

3. マネージメントサイエンス

経営情報、概念的分類

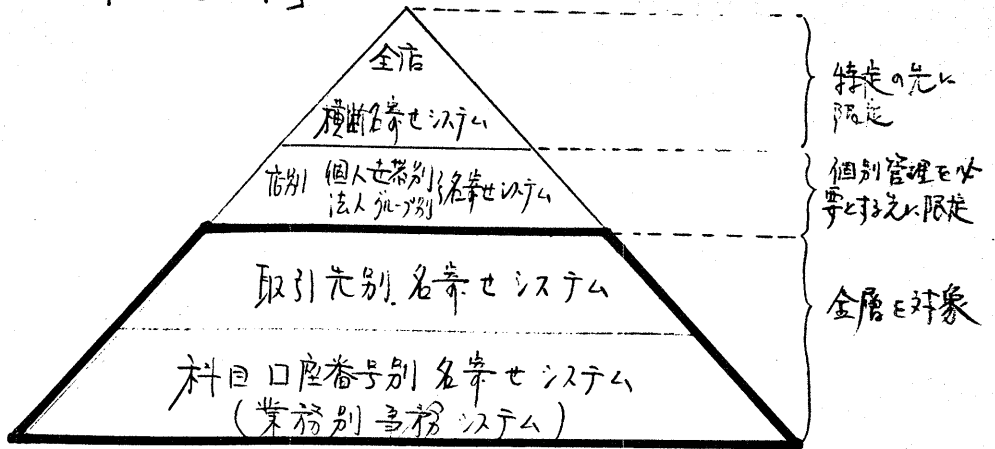


Ⅲ [データベースシステム導入に当たりの問題点]

1. 総合化の限界

	事務処理 オンライン	オンラインによる情報利用
処理態様	<ul style="list-style-type: none"> ・単純照会・更新 ・定形的 ・通帳・判定帳票 	<ul style="list-style-type: none"> ・複雑検索 ・非定形的 ・汎用
処理時間	<ul style="list-style-type: none"> ・即時処理 	<ul style="list-style-type: none"> ・数秒 - 数分
トランザクション	<ul style="list-style-type: none"> ・膨大 	<ul style="list-style-type: none"> ・少量
ファイル容量	<ul style="list-style-type: none"> ・膨大 	<ul style="list-style-type: none"> ・ "
ファイル構造	<ul style="list-style-type: none"> ・単純 	<ul style="list-style-type: none"> ・複雑
利用者	<ul style="list-style-type: none"> ・営業店オペレーター 	<ul style="list-style-type: none"> ・主として本部スタッフ

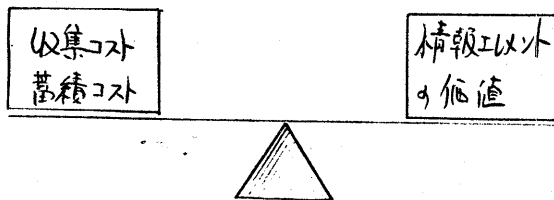
[統合化別 CIF]



2 データギャザリングシステム

(1) データギャザリングの問題

- 金 } がかり過ぎてはいか
- 時間 }
- 体力 }
- 集めたデータは十分活用されているか
- 活用状況が自動的に把握できる仕組みができていますか



(2) データが十分活用されない場合の原因

- データの利用目的があいまい
- データの信頼性が欠けている
- インプット負担が大きい
- データ提供者がインプットをしながらも何の痛痒も感じない
- データの新鮮度が欠ける

(3) 効率的に精度の高いデータを安定的に収集するには

- データを無暗に集めない
- ユーザーと一体になって、利用目的を明確にし情報エレメントの価値分析を行なったうえで収集すべきデータを決める

- データが自然にはいってくるようにする。
→ オペレーショナルなオンラインシステムに組込むなど、データの発生したところから無理なく自然に収集できるようにする。
- データ提供者にメリット感を与えよ
→ 受益者にインパクトさせる
- 評価の結果をつける
→ インパクトをしないと思わせる仕組みにする
- データの活用推進をはかる
→ 活用する段階でエラーに気がいっただけ訂正できるようにする
→ データの制約をユーザーに明示して正しい使い方をしてもらう。
- 余り負担のかからないエッジシステムを確立する
→ 例. コンピュータから定期的に精査表を還元し、該当箇所だけ記入すれば、それだけでインポート帳票とほぼいっぴきターナラウンドシステムの採用など

3 データベースの構築運営

(1) 汎用ソフトウェアの導入に当たっては自社ニーズに合ったものを!

A. 実務的見地からみた汎用ソフトウェアの限界

- システム デザインの依存度大
- 機能とパフォーマンス
- 複雑加工と使い易さ
- データ構造の弾力性
- 機能保持
- 信頼性
- 採算性

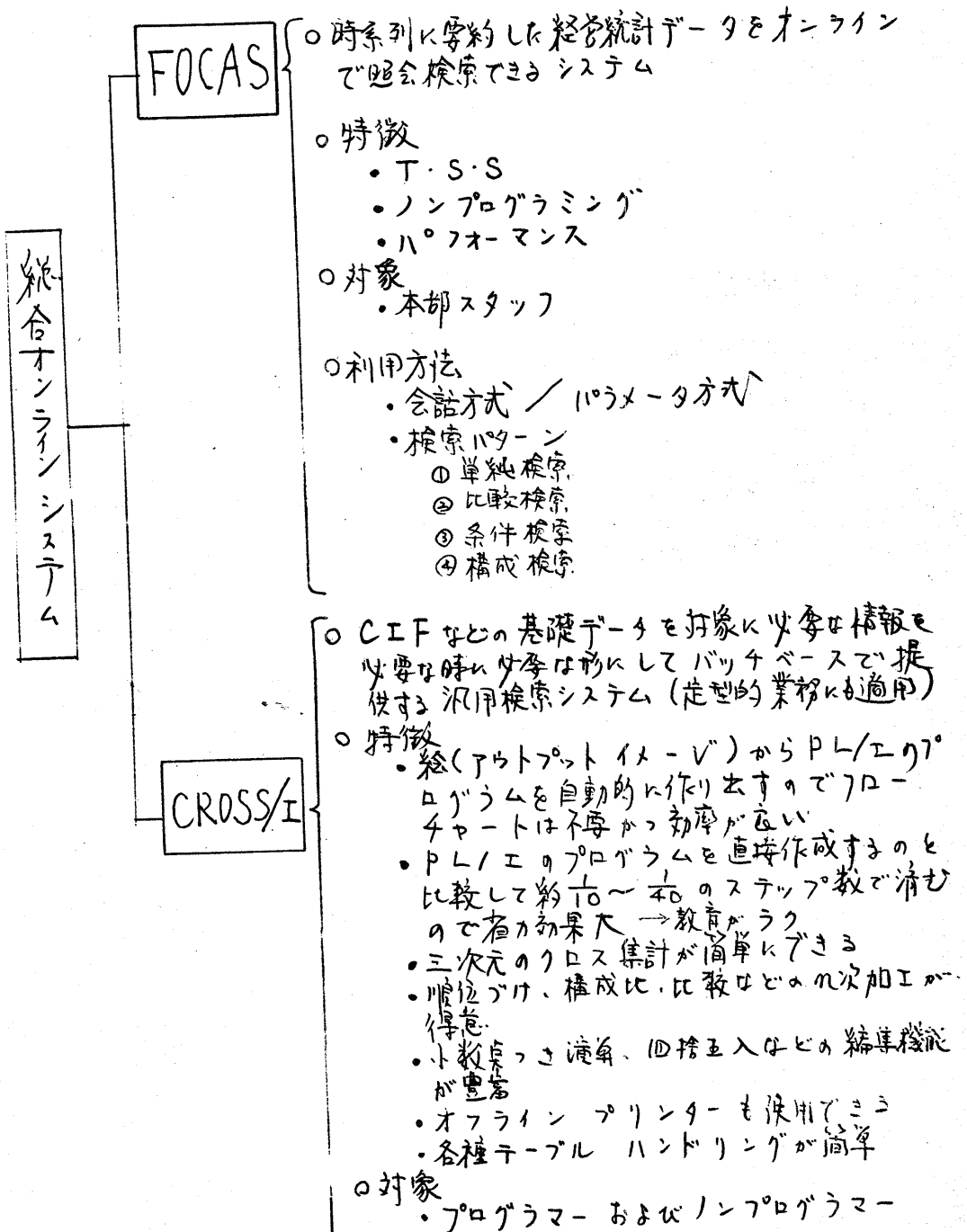
B. 目的のない何でもできる汎用化は逆甲

- あらゆる分野に汎用ソフトウェアを使うにせよ、対象領域を明確して、それぞれの利用目的と効果を十分カバーして、汎用ソフトウェアの適甲分野を決めて使うのが現実的
- ユーザーニーズの分析が不十分だと効率の悪いシステムとぼろ可能性が大

(2) オンライン/バッチ

→ 全てオンラインである必要はなくむしろバッチの領域も多い

[富士銀行の例]



CROSS / E の利用方法

6種類の入カシートにデータ名と加工内容を記入するだけで必要な情報がバッチベースで得られる

- 必須
- ① Fカード --- 検索ファイルの指定
 - ② Xカード --- X軸項目に関する指定
 - ③ Yカード --- Y軸項目
- オプション
- ④ Mカード --- メジャー項目に関する指定
 - ⑤ Hカード --- ヘッダに関する指定
 - ⑥ Vカード --- 中間値など不要項目に関する指定

(3) 総合化 / 分散化

→ 総合化は頭の中で、実態的には分散化を!

(4) 単純化 / 複雑化

→ 単純化した方が効率良い

(5) 原始データ / 集約データ

→ 古いデータを持つのは理想、現実には集約データも必要

(6) 重複データもときには必要

(7) 日次データ / 月次データ

→ 日次データ更新はCPU負荷大 必要最小限にとどめよべき

(8) 共通テーブルは大切に活用すべし

4 [情報検索システム]

(1) 情報検索システムこそ企業のニーズに合ったものを!

(2) 定型化 / 汎用化

→ 情報ニーズを分析したうえで 汎用化した方がシステムの弾力性
拡張性が高い

(3) レスポンスタイムはオンライン検索のカギ

(4) 使い易さ / 高度なデータ加工

→ 一度画面にアウトプットしたものを更に加工処理できるように
すべしと一層プログラミングの向上に役立つ

(5) アウトプット方式

ディスプレイにアウトプットする他、ダイレクトにプリンターに
アウトプットすることも必要

(6) 専用回線の効率的な使用

→ 同一回線 異種端末の利用

(7) 自動評価システムの開発

→ ユーザーの利用状況が自動的に把握できるシステムは、システム
の利用促進およびレベルアップに必要不可欠

以上