

コーポレート (統合) ディクショナリ

— 情報処理工場の生産管理 —

嶋 田 正 裕

株式会社 ソフトウェア・エージ
コーポレート・ディクショナリ開発室

コンピュータ利用の裾野が広がり、その利用の仕方も多様化、高度化してきている。増大する一方のバックログの解消のため、第四世代言語を導入するなどシステム開発手法自身も多様化している。また、企業活動において、情報の重要性はますます高まりつつあり、この情報を扱う中核である情報処理部門の役割の重要性は今更言うまでもない。

ところが、他部門では当然のことのように行われている情報管理も、それが専門であるはずの情報処理部門自身は、まだまだ人手に頼っているのが現状である。

そこで我々は、情報処理部門を情報処理工場としてとらえ、その生産物である「データ」の生産管理とデータ処理の為に生産機械である「プログラム」の生産管理を行うために、コーポレート・ディクショナリ (統合ディクショナリ) という概念を導入して、情報処理部門の真の生産性向上を計っている。

本稿では、情報処理工場における機能を3つに分け (開発センタ、運用センタ、情報センタ) それぞれの機能、およびそれぞれが管理する情報を提示し、これを行うための製品 (プロダクト) についてもその概要を記述している。

CORPORATE DICTIONARY

— Production Control in the Data Processing Factory —

Masahiro Shimada

Software AG of Far East, Inc.

CORPORATE DICTIONARY Development Department

7-2 Yaesu 2-chome, chuo-ku, Tokyo 104 Japan

As information becomes more and more important for business, needless to say that the Data Processing (DP) department, the core of information, takes an important role in the organization.

If we regard DP department as a DP factory, it produces products 'data' and machines 'programs'. For the production control in this factory, we form a conception of CORPORATE DICTIONARY in order to improve productivity.

The DP factory is divided into three functions; development center, operation center, and information center. These functions are stated in this paper and also a software product to practice them is outlined.

1. はじめに

我々は、情報処理部門を情報処理工場としてとらえ、その生産管理を行おうと考える。その際管理対象となるものは「データ」とデータ処理を行うための「プログラム」である。

データには流れがあり、データと共にその流れも管理する必要がある。またプログラムは、情報処理工場における生産機械に相当するが、この情報処理工場では生産機械も手作りで作成している。またその修繕や改良も行っている。

この情報処理工場の生産は多種少量生産型であり、さらに生産機械自身の生産も行っているのだ、これらの生産管理は簡単ではない。でき上がったプログラムを本番運用するための管理も必要である。我々はコーポレート・ディクショナリ(統合ディクショナリ)という概念を導入し、データの流れの管理と、プログラム生産の管理、運用管理を一元的に把握、情報処理部門の生産性の向上を実現をする。

2. コーポレート・ディクショナリの必要性

コンピュータの高速化、メインメモリ・外部メモリ価格の低下、端末の性能向上、コンピュータ・ネットワーク技術の向上等により、利用可能なコンピュータ資源が拡大し、さらにコンピュータ利用も単なる事務処理の効率化、合理化から企業戦略への活用へと変化している。したがって、コンピュータ化される業務(システム)が著しく増加し、この結果プログラム数、使用するデータ項目、扱うデータ量、実行されるジョブが急激に増加している。また、外部環境の変化により、既存プログラムの保守作業も増加している。この様に増加する一方の中で、情報処理部門は依然としてこれらの情報を人手によって管理しているのが現状である。これでは激しく変化する環境に対応することができない。たとえば、あるプログラムを変更したいが、誰がいつ作ったのか、これに関連するプログラムは何か、あるいは使用しているデータ項目は何か、データ項目を変更したらどこに影響するか、また実行時間はどう変化するのか、使用できるリソースは十分か等の情報を知っていなければならない。しかし、管理する情報の規模が大きくなれば、人的能力の範囲を越えるものになる。

情報処理部門を開発センタ、運用センタ、情報センタという3つのセンタ機能として捉え、これらのセンタで扱う情報をコーポレート・ディクショナリとして一元管理し、人手による管理が限界に達している情報をデータベース化することにより、真の情報処理部門の効率化を達成できる。ただ、情報を手作業でディクショナリに登録し、管理するだ

けでは、うまく機能しない。入力忘れや、'めんどくさい'等で、情報の抜けが発生し、ディクショナリの精度に信頼性はなくなり、しまいには、使われなくなってしまう。これでは、意味がない。そこで、抜けをなくす手段が必要になってくる。それには、人手による登録作業をできるだけ少なくし、管理システム側で、自動収集可能な情報は、自動的に取得するツールが必要である。しかし、完全に人手による入力をなくすわけにはいかない。したがって、この抜けをなくす手段が必要である。

たとえば、

- プログラムの情報が登録されていなければ開発ができない
- データ項目の定義がなければファイルが作れない。しかしあれば自動的ファイルにできる
- 実行ジョブの定義がなければ本番業務が実行できない。しかし登録されていれば自動的に実行される

等の仕組みを作り、ディクショナリに登録すれば、後々便利に利用でき、かつ大幅な生産性の向上につなげることができる。

コーポレート・ディクショナリ、およびこの情報を活用したシステムは以上の様な自動的に情報を収集する機能と、ディクショナリ情報に基づいて機能するディクショナリベースのシステム群である。

3. 情報処理工場における3つの機能

情報処理部門のはたす役割を見てみると、これには3つの側面がある。図1に情報処理部門の機能の3つの側面を示す。

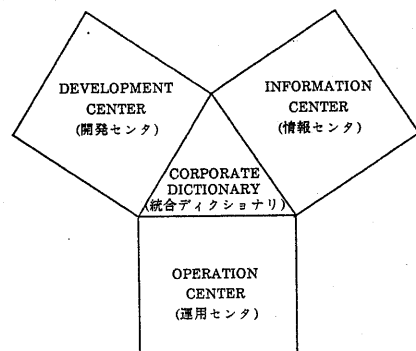


図1 情報処理部門の3つの機能

(1) 開発センタの管理機能

開発センタの機能は、情報処理工場の生産機械であるプログラム自身の生産とその管理を行うことである。アプリケーションシステムのデザイン、プログラム開発、テスト、本番運用、保守という一連の流れを管理する。具体的には次に示す様な管理である。

- システム開発のプロジェクト管理
- プログラム・ライブラリ管理
- プログラム開発進捗管理
- テスト管理
- 開発保守履歴管理
- 開発資源管理
- プログラム資産管理
- データディクショナリ管理

(2) 運用センタの管理機能

運用センタの機能は、日常のコンピュータの運用と管理を行う事である。情報処理工場の直接の生産物であるデータ(情報)の流れの管理も行う。すなわちデータの入力から出力までの一連の流れと、処理の流れを管理する事であり次に示す機能がある。

- 入力データスケジュール管理
- 入力データ実績管理
- 入力データ形式の管理
- ジョブスケジュール管理
- ジョブ実行実績管理
- リソース管理
- 出力(帳表)スケジュール管理
- 出力形式の管理
- 出力実績管理

(3) 情報センタの管理機能

情報センタは、企業の情報を企業の現場の人達すなわちエンドユーザに情報を提供し、エンドユーザが自由にデータを利用できる環境を提供することである。従って、このセンタでの管理機能は次のものがある。

- データ・ディクショナリ管理
- 企業情報の管理
- データ・セキュリティ管理

現在のところ3つのセンタ機能は、互いに独立し、分離されているのが実情である。しかし互いに独立し、別々に情報を管理していたのでは真の生産性向上は実現できない。三つのセンタ機能が互いに連絡し、情報の相互交換を行い、情報も共有できるような統合システムが必要である。これを実現するのがコーポレート・ディクショナリであり情報処理部門全体を統合管理する情報の集まりである。

4. 情報処理工場を管理するための情報

コーポレート・ディレクショナリは、2で述べたように、情報処理工場としての情報処理部門を、開発センタ、運用センタ、情報センタとして捉え、それぞれのセンタ機能に対応した管理情報をディクショナリとして統合したものである。各センタで管理する情報は次のとおりである。

(1) 開発センタで管理する情報

開発センタでは、生産機械であるプログラムの生産管理および保守のための情報管理を行う。プログラムの生産管理を少し広げて、適用業務システム開発管理として捉えてみると、システムの企画から始まり基本設計、業務設計、内部設計、プログラム開発、テスト、本番移行、そして評価という一連の流れを管理する事に対応する。これらを管理するためには、次の情報が必要である。

- システムの情報
- 開発プロジェクトの情報
- システム開発の工程管理情報
- プログラムの情報
- システム開発環境の情報
- テスト管理の情報
- プログラム保守(修正履歴)の情報
- プログラムの実績情報

以上の情報が、システム開発管理ディクショナリ(SYS-DICT)、データ・ディクショナリ(D-DICT)で管理される。

(2) 運用センタで管理する情報

運用センタでは、日常のコンピュータ運用管理に関する情報として、次の情報を管理する。

- 実行ジョブの情報
- 実行順序の情報
- スケジュールの情報
- 使用資源
- 実行結果の情報

これらの情報は運用管理ディクショナリ(A-DICT)として管理される。

(3) 情報センタで管理する情報

情報センタでは、企業で扱う全情報に関して次のようなデータ定義情報を管理する。

- データの名前、意味
- データの属性
- データの発生源

- 他データとの関係
- データの所有者、利用者

これら情報は、データ・ディクショナリ (D-DICT) で管理される。

5. 登録する情報と自動取得情報

ディクショナリに蓄積される情報には、登録する情報 (カタログ情報) と、自動取得情報 (学習情報) がある。

カタログ情報とは、人間が手で登録する「静的」な情報であり、学習情報とは、コーポレイト・ディクショナリを使うシステムが実行された実績から学び取り自動的に蓄積する「動的」な情報である。図2にコーポレイト・ディクショナリの全体図を示す。

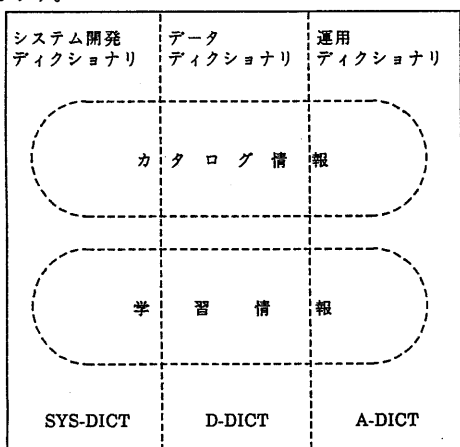


図2 コーポレイト・ディクショナリの全体図

システム開発管理ディクショナリ (SYS-DICT)、データディクショナリ (D-DICT)、運用管理ディクショナリ (A-DICT) は互いに情報を共有し、コンピュータシステム全体を管理する。

学習情報の入手は、実行実績を知る立場にあるオペレーティングシステム (OS)、プログラムエディタ、コンパイラ (あるいはプレコンパイラ) から得ることができる。エディタからは例えば次のような情報を入手できる。

- プログラム毎のエディット回数
 - 修正ステップ数
 - 使用プログラム・ライブラリ等
- コンパイラからは次の情報を入手できる。

- プログラム毎のコンパイル回数
- プログラムと使用サブルーチンのクロスリファレンス情報
- プログラムと使用ファイルおよびその中の各項目のクロスリファレンス情報 等

OSからは次の情報を入手できる。

- ジョブ毎の使用CPU時間
- エラップス時間
- メモリ使用量
- 使用ファイルの容量
- I/O回数
- 実行日 等

また、ディクショナリを使用するシステムが自動収集する情報として、開発開始/終了テスト回数、テスト・カバレッジ情報などが含まれる。

これら自動収集された実績情報は、いろいろな目的で利用できる。

たとえば現状を定量的にとらえることができるため、次のような質問にも正確な回答を下せる。

- 開発プロジェクトの進捗状況はどうか
- あるプログラムについてデバッグはどの位進んでいるか
- 誰がいつ何の目的で作成したプログラムか
- 見積りと比べ実際の開発工数はどの位かかったか
- 開発完了後の修正量はどの位か
- 誰がいつそのプログラムを修正したのか
- 共通サブルーチンを変更したいが、これを使っているプログラムはどれとどれか
- ファイル定義を変更したいが、影響を受けるプログラムはどの位あるか
- 使用頻度の高い (あるいは低い) プログラムはどれか

さらに、現状を踏まえて、今後の予測つまり推論を行うことができる。例えば、次のような推論を行える。

- これからスタートする開発プロジェクトはどの位の開発期間を要するか
- 納期どおりに開発完了可能か
- あるアプリケーションの本番移行にあたって、必要な資源はどの位か、現状で十分か否か

このように逐次、最新の情報を学び取り蓄積していくことによって、実績をベースにした信頼性の高い推論を行うことができる。

6. コーポレート・ディクショナリを取り巻くシステム

コーポレート・ディクショナリに蓄えられた様々な情報を有効に活用していくことにより、情報処理部門の真の生産性向上・効率化を実現できる。

図3にコーポレート・ディクショナリをとりまく情報処理部門の効率化のためとシステムの全体像を示す。

SYS-DICTに対応するシステム開発ディクショナリ・システム、D-DICTに対応するデータ・ディクショナリ・システム、およびA-DICTに対応するジョブ・ディクショナリ・システムは、ディクショナリ・システムの基本機能であるカタログ情報の定義、変更等を行うメンテナンス機能と、ディクショナリ情報からシステム開発の進捗状況、プログラムと使用データのクロスリファレンス、ジョブと使用プログラムの関係等の各種ドキュメントを作成する検索機能を提供する。

上記3システム以外の各システムは、コーポレート・ディクショナリの情報を使用して実行するシステムである。これらのシステムは実行結果から収集される学習情報をコーポレート・ディクショナリにフィードバックする機能も持っている。

図4は、図3のシステムを、ソフトウェア・エンジニアのプラダクトに置き換えて記述したものである。

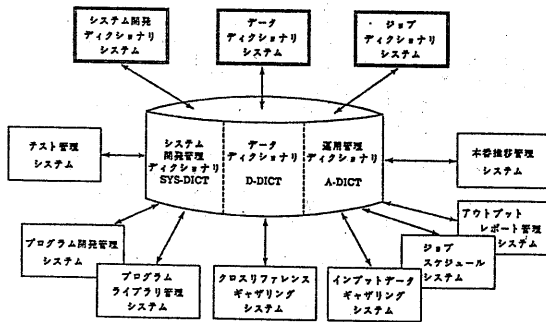


図3 コーポレート・ディクショナリとシステム概要

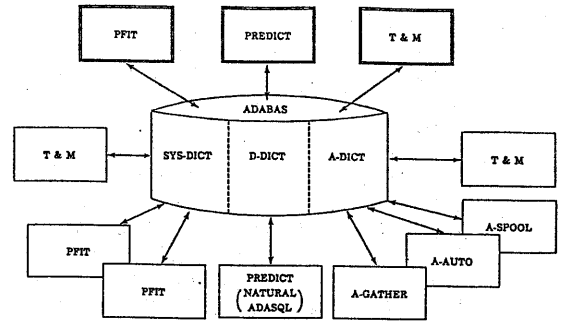


図4 コーポレート・ディクショナリとプラダクト

(1) PFIT(プログラム開発管理/ライブラリ管理システム)

PFITは、システム開発・保守を統合管理するプラダクトである。コーポレート・ディクショナリ内の特にシステム開発ディクショナリ(SYS-DICT)と密接に関係し、開発センタ機能の土台となるものである。

情報処理部門で生産する製品—プログラム、画面定義、JCL、環境定義等—をすべて「モジュール」として捉え、それらが生まれてから消滅するまでの「ライフサイクル」を統合的に管理し、開発・保守・運用を効率的に行うためのツールである。

(2) T & M(テスト管理システム)

T & Mは、テスト管理と本番業務への移行をコンピュータ化したプラダクトである。運用管理ディクショナリ(A-DICT)とSYS-DICTを使用し、システム・テスト、ユーザ・テスト、運用テスト、本番移行への流れを管理し、移行処理を自動化することにより、手作業によるミスを防止し効率良くテスト、本番移行を行うためのツールである。

(3) A-GATHER, A-AUTO, A-SPOOL(インプット・ギャザリング/ジョブ・スケジュール/アウトプット管理システム)

A-GATHER, A-AUTO, A-SPOOLは、運用センタの機能である。データの流れの実行管理を行うプラダクト群である。この三つのプラダクトは、アプリケーション・ジョブ・スーパーバイザ(AS)^[3]の概念を実用化したものである。A-DICTと密接に関係し、データが情報処理工場に入力され、加工され生産物として出力されるまでの一連の流れを管理する。

図5にアプリケーション・ジョブ・スーパーバイザの機能を示す。

A-GATHERは、入力データのスケジュール管理、入力データの収集、および実績情報をA-DICTに自動登録するツールである。

A-AUTOは、ジョブスケジュールにしたがって、コンピュータ・リソースを監視し、ジョブを自動起動する機能、およびジョブの実行結果をA-DICTに自動登録するツールである。

A-SPOOLは、A-AUTOにより、自動起動されたジョブが出力した帳票を管理し、出力帳票の配送スケジュール、配送分類等により、出力機能とその実行結果をA-DICTに自動登録するツールである。

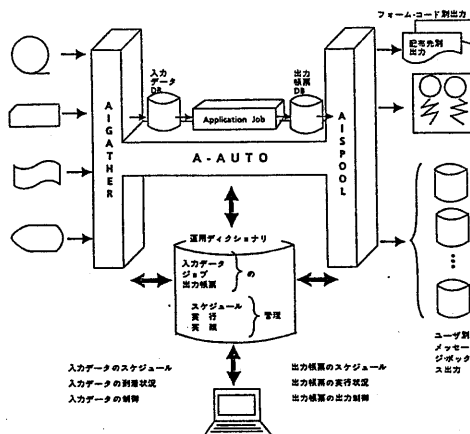


図5 アプリケーション・ジョブ・スーパーバイザの機能

(4) PREDICT(データ・ディクショナリ/クロスリファレンス・ギャザリング・システム)

PREDICTは、オンライン会話型データ・ディクショナリシステムである。

次にPREDICTの主な機能を挙げる。

- MAINTENANCE機能は、データ・ディクショナリ情報の定義、変更を行う。
- RETRIEVE機能は、各種データ・ディクショナリ情報の検索を行う。
- GENERATE機能は、DBMS(ADABAS)、OS(SAM, VSAM等)、第4世代言語(NATURAL)等へのディレクトリの作成を行う。

- INCORPORATE機能は、DBMS(ADABAS)、OS(COPY句)、第4世代言語(NATURAL)等からデータ・ディクショナリ情報の作成を行う。

また、PREDICTはNATURALおよび第3世代言語用SQLインターフェース(ADASQL)に関してプログラムで使用しているデータベースフィールド、使用画面、使用サブルーチン等についてのクロスリファレンス情報を自動収集する機能を提供している。

7. コーポレート・ディクショナリの構造

7.1 システム開発管理ディクショナリ(SYS-DICT)

SYS-DICTは、システム開発・保守に関する情報を管理するディクショナリである。情報処理部門で生産する製品—プログラム、画面定義、JCL、環境定義等—を「モジュール」としてとらえ、これらのモジュールの基本情報を管理する。図6にSYS-DICTのエンティティ(管理対象)を示す。

またデータディクショナリ(D-DICT)、運用管理ディクショナリ(A-DICT)との関係は、図に示したとおりであり、SYS-DICTのシステムエントリから、D-DICTのシステムエントリと関係づけられ、モジュールエントリからD-DICTのプログラムエントリと、A-DICTのジョブステップエントリと関係づけられる。

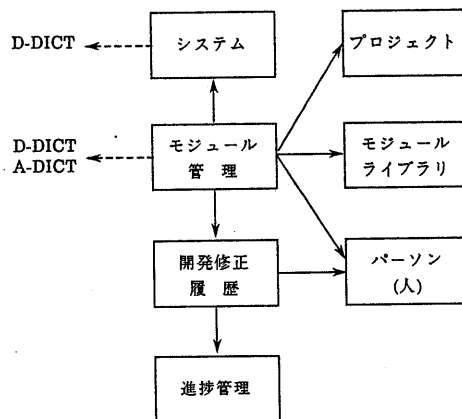


図6 SYS-DICTのエンティティ

7.2 運用管理ディクショナリ (A-DICT)

A-DICTは、コンピュータ運用に関する情報を管理するディクショナリである。一連の処理単位を「ジョブ」としてとらえ、また、一連の処理の実行順序を「ジョブ・ネットワーク」として管理する。この2つのエンティティを中心に、ジョブあるいは、ジョブネットワーク単位に、それ自身の情報、実行結果、使用リソース等が管理される。

また、A-DICTではジョブ、ジョブ・ネットワーク単位の開発、本番移行のための情報も管理しており、ジョブテスト、あるいは、ジョブ・ネットワーク単位によるシステム・テスト、運用テストの管理が行える。またジョブ、ジョブ・ネットワーク単位による進捗管理が行える。

図7にA-DICTのエンティティを示す。

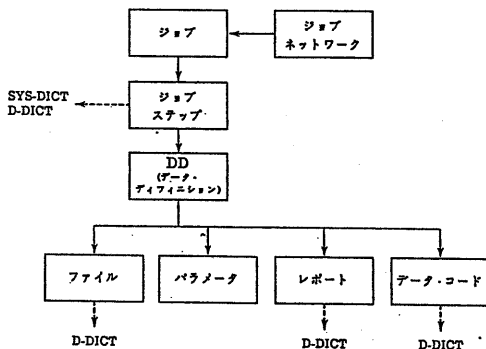


図7 A-DICTのエンティティ

ジョブ、ジョブステップ、DD、およびファイルエンティティは、ジョブに含まれるジョブ・ステップ情報、ステップで使用するリソース情報(ファイル、レポート、データコード)である。

また、レポート・エンティティは、帳票に関する情報であり、A-SPOOLにより管理される。データ・コードとは、A-GATHERで管理する情報であり、入力データに関する情報である。

データ・ディクショナリ (D-DICT) とSYS-DICTの関係は、ジョブステップ情報のプログラム名で、D-DICTのプログラム・エン트리、SYS-DICTのモジュール・エン트리と関係づけられる。ファイル・エン트리からは、D-DICTのファイルエン트리と、レポートエン트리からD-DICTのレポートエン트리、データコード・エン트리からはD-DICTのファイルあるいは画面エン트리へと関係づけられる。

7.3 データ・ディクショナリ (D-DICT)

(1) D-DICTのエンティティ

D-DICTは、三種類のエンティティを持っており、データ定義情報、プログラム定義情報、所有者/使用者の情報が蓄積されるディクショナリである。

● データ・エンティティ

データベース、ファイル、フィールドおよび、これらの関係、整合チェック情報。

● 処理エンティティ

システム、プログラム、モジュール(サブルーチン)、レポートおよび画面定義情報。

● 特殊エンティティ

キーワードおよび、ユーザ(使用者)、オーナー(所有者)、クロスリファレンス情報。

図8にデータ・ディクショナリ・エンティティを示す。

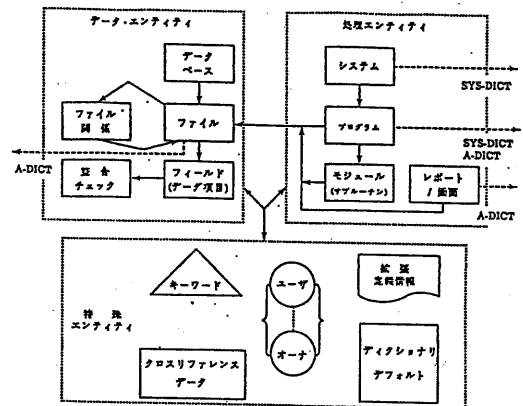


図8 データ・ディクショナリ・エンティティ

(2) 4層ファイル・モデル

データ・エンティティには、データベース、ファイル、ファイル間関係、フィールド、フィールド整合チェックが定義される。また、ファイルには、ADABAS, VSAM, SAM, ISAM、概念ファイル、スタンダードファイル、ユーザビュー等のタイプが有る。D-DICTの扱うファイルは、4層ファイル・モデルであり、4階層でデータ定義ができる。図9に4層ファイル・モデルを示す。

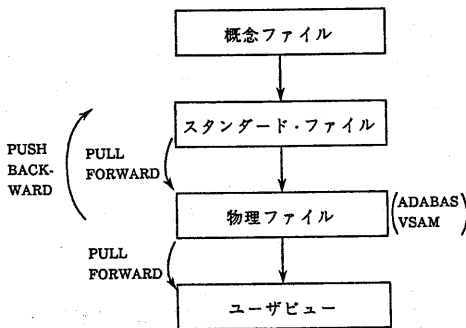


図9 4層ファイルモデル

概念ファイルは、データ・ベース・デザインの時に作業用に使用するファイルであり、必要なデータ項目を洗い出し横並びに定義したファイルである。

スタンダード・ファイルは、概念ファイルから関連するデータ項目単位にまとめ(正規化)したファイルである、企業における標準ファイルとなる。

物理ファイル(ADABAS, VSAM)、は効率を考慮し、スタンダードファイルをいくつかまとめ作り出す(PULL FORWARD)ファイルである。

ユーザビューは、物理ファイルからアプリケーションで使用するデータ項目だけが定義されるファイルである(これもPULL FORWARDされる)。

PULL FORWARDとは、上位階層のファイルからコピーされたデータ項目群であり上位階層のファイルが変更されれば、下位レベルのファイルも自動更新される。

すなわち、スタンダードファイルを変更すると下位レベルの物理ファイル・ユーザビューも全て、自動更新される。

また、既存の物理ファイルのデータ項目定義

からスタンダードファイルを構成する(PUSH BACKWARD)こともできる。

- (3) キーワードとクロスリファレンスデータ
キーワードは、D-DICT内に蓄積されている情報をエンティティ単位でなく、別の切り口(例えばプロジェクトコード、作業工番等で)、管理、検索する時に使用する情報である。例えば、プロジェクト単位に関係するエンティティに共通のキーワードを割り当てておけば、いつでも、プロジェクトに関係するエンティティを取り出すことができる。
クロスリファレンスデータは、第4世代言語NATURALおよびSQLインタフェースADASQLで自動収集された情報であり、プログラムと使用データ項目、使用サブルーチン、使用レポート、および画面とのクロスリファレンス情報である。

8. おわりに

高度情報社会を迎え、各企業において情報処理部門の重要性が高まっている。企業の情報処理の管理の中核として、ここで述べたコーポレート・ディレクショナルの必要性は言うまでもないであろう。

現在のコーポレート・ディレクショナルは、情報としてカタログ情報とシステム側で自動収集する学習情報が蓄積され、利用されているが、学習情報はまだ十分に活用されているとはいえない。今後は学習情報に知識データを加えてデータベースデザインツール、システムデザインツール、自動チューニング機能等、様々な分野に応用でき、情報処理部門の統合エキスパートシステムとして発展することになろう。

参 考 文 献

- (1) 石井義興: 第4世代テクノロジーによる解決策, ソフトウェア・エージェンシー
- (2) 末舛史郎: システム開発・保守の統合管理システムPFIT, ソフトウェア流通 1986年10月号
- (3) 石井義興、宮台功、高橋良和: アプリケーション・ジョブ・スーパーバイザ(AS)の概念, ソフトウェア・エージェンシー
- (4) 石井義興: システム部門エキスパート・システムへの道, コンピュータ・レポート 1986年1月号