

UM EXEMPLO DE SOFTWARE INTEGRADO DE 4a.
GERAÇÃO PARA O DESENVOLVIMENTO DE SISTEMAS
DE INFORMAÇÃO

Elisa Hatsue Moriya

nº 23

Abril
1986

UM EXEMPLO DE SOFTWARE INTEGRADO DE
4ª GERAÇÃO PARA O DESENVOLVIMENTO DE
SISTEMAS DE INFORMAÇÃO

Elisa Hatsue Moriya
Departamento de Informática
Centro de Tecnologia
Universidade Estadual de Maringá
Caixa Postal 331
87020 - Maringá - Paraná
BRASIL

RESUMO: São discutidas as características que definem um instrumental de 4ª geração para o desenvolvimento de Sistemas de Informação, sendo apresentadas 7 referências bibliográficas sobre esse assunto. O papel desse instrumental no atendimento da dinâmica empresarial atual é analisado. Os recursos de um exemplo desse instrumental são apresentados. O exemplo escolhido refere-se ao IDMS (Integrated Data Management System), que é integrado por vários componentes, cada um dos quais é discutido no trabalho. Quinze referências bibliográficas sobre esse instrumental são apresentadas, sendo fornecido um resumo do seu conteúdo. Assim, são discutidos o IDD (Integrated Data Dictionary) que é o dicionário de dados que forma o núcleo do sistema; e ADS/O (Application Development System/on line) que é a ferramenta para desenvolvimento de aplicação para processamento imediato de transação; a OLQ

(on line query) que é a linguagem de consulta principalmente para projetistas; a OLE (on line english) que é a linguagem voltada ao usuário final e CULPRIT que é o gerador de relatórios utilizado pelos projetistas. É fornecida referência de um trabalho da autora, no qual se descreve uma aplicação desse instrumental no desenvolvimento de um sistema real.

PALAVRAS-CHAVE: Software de 4ª geração, processamento imediato de transação, linguagem de consulta, gerador de relatórios, usuário final, registro mestre, registro membro, esquema, sub-esquema.

1 - INTRODUÇÃO

A dinâmica empresarial atual, solicita dos sistemas de informação, muitas vezes informações que não haviam sido previstas quando da especificação do sistema. Em ambientes tradicionais de desenvolvimento de sistemas, o usuário que necessita repentinamente de algum relatório ou mesmo alteração de um que já exista, acaba sujeitando-se à disponibilidade da equipe de desenvolvimento que nem sempre pode atendê-lo de imediato.

É desejável portanto, ter disponíveis técnicas e ferramentas que permitam aos usuários terem um atendimento mais expedito feito pelos próprios analistas sem a necessidade de envolvimento de equipes de programadores.

A comunicação com a máquina é feita através de "linguagens" (ou ferramentas) hoje mais potentes, mas que para atingir tal estágio, sofreram uma evolução desde as primitivas linguagens de máquina. O conceito fundamental atrás de uma linguagem não procedimental [Co 85], é a transferência da preocupação com o fluxo de programas do especialista em processamento de dados para o software. É especificado o resultado desejado ao invés de especificar as ações necessárias para obter aquele resultado.

Uma linguagem de 4ª geração tem como uma das principais características [Ma 81], permitir ao analista obter resultados mais rápidos do que ao especificar programas. O analista trabalha junto ao usuário interativamente, até que satisfaça as exigências dele. Existem linguagens de 4ª

geração voltadas ao usuário, que permitiriam a ele obter por si mesmo, os resultados desejados desde que os dados tenham sido devidamente estruturados.

No entanto, [Gr 85], as linguagens de 4ª geração não são a solução para os problemas que comumente aparecem na implementação de sistemas de informação administrativos tradicionais. Deve-se verificar até que ponto a produtividade ganha, compensa a despesa decorrente deste tipo de software pois "uma linguagem de 4ª geração exige uma organização de 4ª geração".

Um estudo criterioso deve ser feito quando da escolha de uma linguagem de 4ª geração [Ti 85], para utilização em uma determinada instalação, sendo o ADS/O uma delas que é parte integrante deste trabalho.

Várias técnicas podem ser utilizadas para o desenvolvimento de aplicações relativas ao processamento imediato de transações [Rw 85], tais como: linguagens de programação de alto nível, sistemas de definição de telas, geradores de aplicações e ambientes integrados de desenvolvimento. O ADS/O pode ser considerado como um instrumento desse tipo.

Alguns critérios devem ser estabelecidos quando da escolha da linguagem a ser utilizada, [Ru 85], bem como o aspecto produtividade faz a utilização de linguagens de 4ª geração no desenvolvimento de aplicações.

Este trabalho tem por objetivo, apresentar um exemplo de instrumental de 4ª geração disponível no mercado, que pode ser utilizado no desenvolvimento de sistemas. 0

instrumental aqui estudado é o IDMS, cujo diagrama geral com as componentes está ilustrado na figura 1.

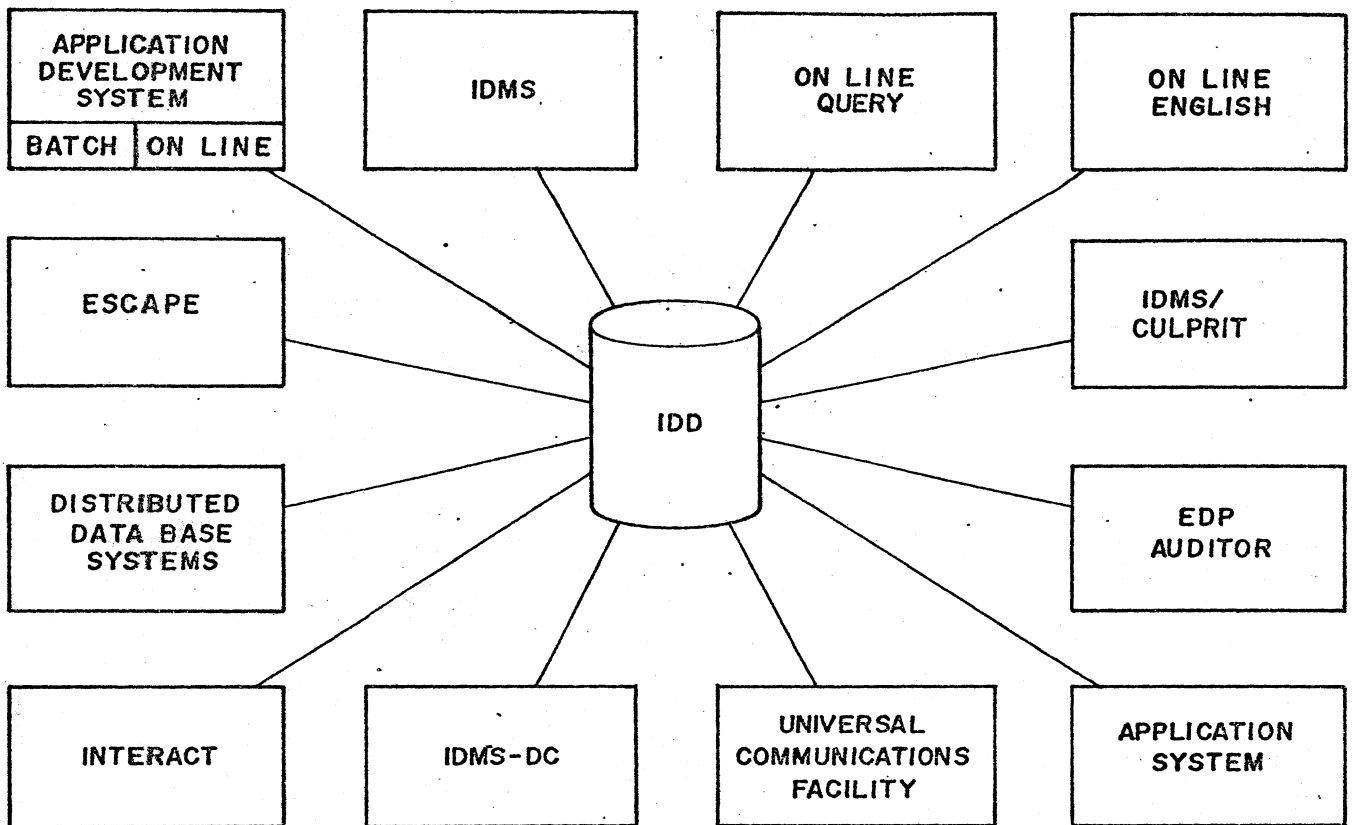


Figura 01 - Software Integrado de Administração de Dados.

No item 2, são fornecidos detalhes sobre o IDMS. No item 3, descrevem-se as características do dicionário de dados - IDD. No item 4, são descritos os recursos da ferramenta para desenvolvimento de sistemas on line - ADS/0. As características da linguagem de consulta OLQ, são discutidas no item 5. Os recursos da linguagem voltada ao usuário final OLE, são descritas no item 6 e os detalhes do gerador

de relatórios CULPRIT, no item 7. As demais componentes da figura 1, são sucintamente descritas a seguir.

- IDMS-DC - é o monitor de teleprocessamento
- Distributed Data Base System: é o sistema que permite programas de aplicação sendo executados em diversas CPU, terem acesso e atualizar base de dados comuns.
- Application Development System/Batch: permite centralizar e simplificar as tarefas de validação de arquivos de transação e atualização de base de dados.
- Universal Communications Facility: possibilita o desenvolvimento de aplicações on line sob monitor de teleprocessamento.
- Escape: é um conjunto de interfaces que permite utilizar DL/1 com IDMS.
- Interact é o sistema que pode ser utilizado para edição de texto.
- EDP - Auditor: é um conjunto de rotinas que permite executar tarefas de auditoria.
- Application System: são aplicações desenvolvidas para finalidades específicas.

Um exemplo de utilização desse instrumental no desenvolvimento de um sistema real pode ser encontrado em [Mo 86], que inclui também uma avaliação da utilização de recursos de métodos de análise e projeto no desenvolvimento de sistemas.

2 - CARACTERIZAÇÃO DO IDMS

O IDMS é um "software" gerenciador, de Base de Dados voltado a um dicionário de dados.

Uma base de dados IDMS tem as seguintes componentes: *esquema*, *sub-esquema* e "*Device Media Control Language - DMCL*".

Um esquema é a descrição completa da base de dados contendo nomes e descrições de todos os arquivos, registros, áreas e relações estabelecidas entre registros denominadas *sets*.

Um sub-esquema é uma visão da base de dados que satisfaz as necessidades de dados de um grupo de programas ou aplicações. Pode constituir-se de toda ou parte das descrições de elementos, registros, sets e áreas definidas no esquema.

DMCL é a componente utilizada para definir tamanho de "buffer", de área, de páginas e opcionalmente "journal file" que é utilizada para recuperação de informações.

Para descrever o esquema, sub-esquema e DMCL quando da definição de uma base de dados, tem se disponível a DDL (Data Description Language), [C&83f].

A análise e projeto de uma base de dados, envolve atividades de identificação de elementos, registros, arquivos, funções e sets. Nas relações que são estabelecidas entre registros, sempre é identificado o registro mestre e o registro membro, podendo ser um dos seguintes tipos: 1:1, 1:n e n:m, sendo que no caso n:m é dado um tratamento espe-

cial através da "junction record". Uma vez que os elementos, registros e sets foram acomodados satisfazendo às restrições IDMS, deve ser desenvolvido o diagrama de estrutura que contém informações relativas aos registros, como por exemplo: comprimento, modo de localização, modo de armazenamento; informações relativas aos sets, tais como: ordem de ocorrência dos registros membros, forma como um registro é conectado ou desconectado de um set e informações relativas à área que será atribuída ao registro.

O IDMS permite que as informações sejam mantidas em arquivos VSAM (Virtual Storage Access Memory) sem que sejam definidos como Base de Dados, [C83f]. Em relação ao aspecto físico, são apresentados conceitos de página, área, arquivo e diferentes modos de localização que podem ser utilizados para acesso aos registros, [C83e]. São apresentadas sugestões para o cálculo do tamanho de páginas e determinação da posição dos ponteiros quando da especificação de SET LINKAGE (ligação entre registros mestre e membro) e opções de segurança que podem ser consideradas na definição do sub-esquema, para impedir o acesso de pessoas não autorizadas, bem como a sintaxe e cláusulas válidas dos comandos da DDL (Data Definition Language) utilizados para definição de esquema, sub-esquema e DMCL, em [C83f].

As sugestões de diretrizes para elaboração do projeto da Base de Dados, bem como aspectos relativos ao desempenho, privacidade e segurança, definição de área, página, etc... que devem ser considerados no projeto são tratados em [C83c].

Alguns comandos utilitários que podem ser utilizados pelo administrador da base de dados (DBA), para criação e/ou manutenção de uma base de dados, são tratados em [C283e].

3 - O DICIONÁRIO DE DADOS - IDD

Os recursos dos dados que fluem em uma organização devem ser mantidos o mais atualizado possível, cuidando também dos aspectos integridade e flexibilidade. O IDMS oferece uma ferramenta - IDD, Integrated Data Dictionary - que é o sistema utilizado para armazenamento e manutenção dos recursos de dados no depósito central, denominado dicionário de dados, cujas características, benefícios e conceitos são apresentados em [C&82c].

Os aspectos relativos às etapas que poderiam ser seguidas quando do início da utilização de um dicionário de dados: planejamento, implementação e avaliação, de modo que possam ser melhor aproveitados os recursos disponíveis, são discutidos em [C&84b].

No dicionário de dados são armazenados ocorrências de componentes denominadas *entidades*, que podem ser divididas em: entidades básicas, de teleprocessamento e especiais.

As entidades básicas correspondem por exemplo o elemento, registro, arquivo, sistema, usuário, programa, etc...

As entidades de teleprocessamento correspondem às componentes do processamento imediato como por exemplo mapa, linha, terminal lógico, terminal físico, destino, etc...

As entidades especiais correspondem por exemplo a módulo de carga, entidades definidas pelo usuário, etc...

Uma ocorrência de entidade é uma instância específica de uma particular entidade. Assim, pode-se ter por exemplo a entidade ELEMENT e a ocorrência NOME-CLIENTE.

O dicionário de dados pode ser utilizado na fase de análise e projeto de sistemas, uma vez que para os dados obtidos através de entrevistas, usuários, fluxo de informações, podem ser definidas informações como fonte dos dados, características físicas, relações, frequência de acesso, etc... Para cada um ou conjunto desses dados pode ser identificada uma entidade.

O IDD tem como componente a Data Dictionary Description Language (DDDL) que é utilizada para definir as entidades no dicionário de dados, [Cø83d].

O IDD oferece ainda recursos "on line", que podem ser utilizados para atualização de descrições de ocorrências de entidades no dicionário de dados. Estão disponíveis para utilização por exemplo os comandos FIND, END, SWAP, PRINT, DISPLAY, que são discutidos também em [Cø83d].

No entanto, a alimentação do dicionário de dados não deve ser feita indiscriminadamente. Devem ser estabelecidos padrões a serem seguidos como por exemplo, convenções para dar nomes a entidades, definição de rotinas padrão para que os sistemas que delas necessitem possam usá-las sem duplicar a definição de rotinas, estabelecimento de cronogramas para o pessoal de produção, para que os dados necessários já estejam efetivamente atualizados, assegurar que somente pessoas autorizadas tenham acesso às informações, o que pode ser feito através de senhas e atribuição de níveis de auto-

rização para ter acesso, são discutidos com mais detalhe em [C&84b]. O IDD permite ainda que sejam definidos sinônimos para nomes de entidade, oferecendo assim uma maior flexibilidade, pois uma única ocorrência da entidade no dicionário de dados, pode ser referenciada por diferentes nomes de acordo com as necessidades dos usuários.

Uma vez que os dados tenham sido armazenados no dicionário de dados, eles se tornam imediatamente disponíveis. Assim, se o usuário desejar, o IDD tem como componente o Data Dictionary Report (DDR) que permite ao usuário obter diretamente do dicionário de dados, diferentes tipos de relatórios. Alguns desses relatórios, como por exemplo relatório por programa, por usuário, por registro, relatório sumário, relatório com descrições do elemento, etc. São denominados relatórios-padrão e são fornecidos quando da instalação do sistema. No entanto, é dada ao usuário a flexibilidade de criar relatórios próprios quando a aplicação assim exigir, como pode ser visto em [C&82c].

4 - A FERRAMENTA PARA DESENVOLVIMENTO - ADS/O

É desejável uma ferramenta de fácil manipulação para o desenvolvimento de aplicações "on line". Para tanto, o IDMS tem disponível a ferramenta Application Development System/on line - ADS/O.

Uma aplicação pode ser composta de um ou mais *diálogos*. Um diálogo pode ser entendido como uma unidade de trabalho ADS. A cada diálogo está associado um *mapa*, um ou mais processos, sub-esquema e registros que devem estar armazenados no dicionário de dados; para que o diálogo possa ser gerado e conseqüentemente a aplicação ser executada.

Para que se possa proceder à definição de um mapa é necessário que os registros de trabalho e tabelas a serem utilizados, estejam definidos no dicionário de dados. Uma vez que os registros e as tabelas tenham sido definidos, o projetista pode dar início à definição do mapa propriamente dita, utilizando os recursos de On Line Mapping (OLM), cujos detalhes podem ser encontrados em [C82b] e [Mo86], que compreende o preenchimento de um conjunto de telas, onde são definidas características inerentes a cada campo que tenha sido "desenhado" no mapa. Ainda em [C82b], podem ser encontrados detalhes para definição de mapas no modo batch, com a sintaxe dos comandos a serem utilizados bem como aspectos relativos à manipulação de erros e edição automática, que devem estar definidos no dicionário de dados, as considerações que devem ser feitas quando da definição, alteração ou utilização de elementos, registros ou tabelas utilizadas pelo

mapa.

Os processos que compõem um diálogo podem ser de dois tipos: *pré-mapa* e de *resposta ao mapa*, sendo eles quem define o processamento a ser executado.

A cada diálogo pode estar associado um processo *pré-mapa* e um ou mais processos de resposta ao mapa. Um processo *pré-mapa* é caracterizado por efetuar todo o processamento necessário antes que o mapa seja exibido ao usuário. Já o processo de resposta ao mapa, executa o processamento de acordo com uma lógica definida, considerando os dados digitados pelo usuário.

A operação de exibir as informações das áreas dos registros para a tela é denominada "mapout" e a operação de transferir os dados digitados na tela para as áreas dos registros, é denominada "mapin".

Para a definição do módulo fonte de um processo, pode ser utilizada uma combinação de comandos que definem, por exemplo, operações aritméticas, de alteração de mapa, de estruturas de seleção e iteração, para ter acesso à base de dados, etc... estabelecendo assim a lógica inerente a ele. Maiores detalhes sobre a sintaxe e utilização desses comandos podem ser encontrados em [C284a].

Após a definição do mapa e dos processos, deve-se gerar o diálogo. Para gerar um diálogo utiliza-se o AD SG. Deve ser preenchido um conjunto de telas contendo informações do nome do diálogo, do sub-esquema utilizado, do mapa associado, dos processos *pré-mapa* e de resposta ao mapa e a chave de controle associada ao processo. Maiores detalhes podem

ser encontrados em [C083a, C084a].

Os erros que ocorrerem durante a compilação dos processos podem ser corrigidos utilizando os recursos disponíveis no IDD para processamento "on line". Esses mesmos recursos combinados ou não com o reprocessamento do sub-esquema, esquema ou mapa, podem ser utilizados quando houver necessidade de fazer alguma modificação em diálogos que já existam. Note-se no entanto, que neste último caso, não há necessidade de preencher o conjunto de telas acima referido, basta que sejam feitas as alterações necessárias e o diálogo regenerado, para que as alterações efetuadas tornem com efeito.

Para a execução das aplicações, o sistema de execução ADS/0 carrega os módulos de carga exigidos, interpreta e processa os dados digitados no terminal, de acordo com as definições estabelecidas.

O projetista pode também obter relatórios relativos às aplicações ADS/0 como consta em [C084a].

ser encontrados em [C283a, C284a].

Os erros que ocorrerem durante a compilação dos processos podem ser corrigidos utilizando os recursos disponíveis no IDD para processamento "on line". Esses mesmos recursos combinados ou não com o reprocessamento do sub-esquema, esquema ou mapa, podem ser utilizados quando houver necessidade de fazer alguma modificação em diálogos que já existam. Note-se no entanto, que neste último caso, não há necessidade de preencher o conjunto de telas acima referido, basta que sejam feitas as alterações necessárias e o diálogo regerado, para que as alterações efetuadas tornem com efeito.

Para a execução das aplicações, o sistema de execução ADS/O carrega os módulos de carga exigidos, interpreta e processa os dados digitados no terminal, de acordo com as definições estabelecidas.

O projetista pode também obter relatórios relativos às aplicações ADS/O como consta em [C284a].

5 - A LINGUAGEM DE CONSULTA - OLQ

O acesso às informações armazenadas em estruturas IDMS, através de consultas pode ser feito utilizando a OLQ. Essas consultas muitas vezes apresentam tal nível de complexidade que exigem uma "navegação" pela estrutura, para que se possa obter as informações necessárias.

O sistema conversacional On Line Query - OLQ - permite que se tenha acesso diretamente às informações de uma base de dados ou dicionário de dados, as quais podem ser formatadas em relatórios impressos ou exibidas diretamente na tela.

Para obter as informações necessárias, pode-se utilizar um dos seguintes tipos de comandos ou combinação deles: comandos de controle e ajuste, de recuperação de dados, de processamento de relatórios e de definição de caminhos e q-arquivos. Maiores detalhes sobre esses comandos podem ser encontrados em [Cl83j] e [Mo86].

Um caminho é constituído de vários comandos OLQ. A ordem em que eles são executados determina a "rota de busca" das informações em um sub-esquema de um registro mestre para um registro membro e vice-versa.

Um q-arquivo é constituído de vários comandos OLQ e de parâmetros e deve ser identificado por um nome pelo qual será referenciado posteriormente. Um q-arquivo pode ser catalogado no dicionário de dados, podendo assim ser referenciado e executado quantas vezes forem necessários, fornecendo os parâmetros convenientes para que melhor atendam as necessi-

dades do usuário.

Caminhos e q-arquivos permitem ter acesso a diferentes tipos de registros, enquanto que os comandos de recuperação de dados permitem ter acesso a um tipo de registro. As informações recuperadas, normalmente satisfazem algum critério especificado nos comandos de recuperação de dados.

Os comandos de processamento de relatórios, proporcionam uma maior flexibilidade para formatação de relatórios, pois têm disponíveis comandos que permitem efetuar cálculos sobre determinados campos numéricos, ordenar a saída por um determinado campo, determinar o maior ou o menor valor, a média dos valores, o total, estabelecer pontos de quebra, etc...

A recuperação de informações pode ser feita uma a uma ou em conjunto. No primeiro caso, as informações são exibidas em linhas horizontais e no segundo caso, em colunas. Para ambos os casos, OLQ tem disponível comando que permitem "avançar telas" para que o usuário tenha acesso às informações que não couberam numa única tela.

6 - A LINGUAGEM VOLTADA AO USUÁRIO - OLE

Uma característica de instrumental de quarta geração é oferecer recursos que permitam ao usuário final, obter as informações necessárias o mais independentemente possível. Assim o IDMS tem disponível a OLE que é uma linguagem voltada para o usuário final.

On Line English - OLE é uma ferramenta IDMS que permite aos próprios usuários terem acesso diretamente aos dados, utilizando uma linguagem mais natural, sendo as suas características e vantagens discutidas em [C&83h].

O vocabulário utilizado pelos usuários finais, está definido no "lexicon" sendo que uma parte desse vocabulário - "lexicon raiz" contém termos mais frequentemente utilizados, tais como: quanto, onde, quem, quais e é fornecida quando da instalação da OLE. A outra parte - "lexicon da aplicação", contém termos específicos da aplicação, tais como: nomes de arquivos, campos, agrupamentos de dados que serão utilizados pelo usuário final, permitindo assim um diálogo fluente entre o usuário e a OLE.

Na definição do "lexicon da aplicação", devem ser definidos os prováveis sinônimos para os nomes de campos que serão utilizados pelo usuário final. Assim, para uma aplicação que envolva o campo "empregado" por exemplo, seria interessante definir "funcionários", "pessoas" e "trabalhadores", como sinônimos. Um outro aspecto que poderia ser considerado é a definição de grupos. Assim, poderia por exemplo ser definido o grupo "lugar" como composto de "rua, cidade, esta-

do" com o sinônimo "endereço". Essas considerações e outras que podem ser encontradas em maiores detalhes em [C283i] dão maior flexibilidade para que o usuário construa as sentenças nas suas consultas.

O usuário final utiliza a OLE de forma interativa. Assim, quando é digitada no terminal uma consulta, a OLE tentará entendê-la e se todos os termos forem identificáveis, as informações serão recuperadas e exibidas ao usuário, no terminal. Por outro lado, se a consulta contiver termos não identificáveis, serão exibidas mensagens solicitando ao usuário uma complementação das informações até que todos os termos utilizados possam ser identificados e então possam ser recuperadas as informações desejadas.

Além das consultas tradicionais, o usuário tem disponível alguns processos identificados por palavras-chave definidos no "lexicon" que executam funções específicas. Através desses processos, pode ser obtida uma saída ordenada por um determinado campo ou executar processamento sobre campos numéricos. Tais processamentos permitem calcular porcentagem, totais com quebras, cálculos estatísticos, etc... Pode também ser especificado para que seja desenhado o gráfico de barras, tendo assim uma representação gráfica dos resultados obtidos.

As ilustrações das diferentes formas em que podem ser elaboradas as consultas, forma como a OLE as interpreta e também os processos pré-definidos, podem ser encontrados em [C283h].

7 - O GERADOR DE RELATÓRIOS - CULPRIT

CULPRIT é uma outra ferramenta IDMS que permite gerar relatórios que podem ser impressos ou não. As características, benefícios bem como vários exemplos de relatórios, que procuram demonstrar através de discussões a flexibilidade para formatação de relatórios que os recursos disponíveis oferecem aos usuários que utilizam essa ferramenta, proporcionando assim um significativo aumento de produtividade são discutidos em [C281]. Dentre os exemplos apresentados, consta um em que o mesmo relatório foi gerado através de um programa codificado em COBOL e também utilizando CULPRIT, onde é evidente a redução do número de comandos utilizados.

A estrutura de um programa CULPRIT compreende de parâmetros para definição do perfil, da entrada, do processo, da saída, da área de trabalho e de campos com índice. Maiores detalhes sobre parâmetros e facilidades permitidas podem ser encontrados em [C283b].

Os parâmetros de definição do perfil, permitem especificar por exemplo, formato de impressão de "dump"; formato de impressão da hora de processamento; número de linhas/página; nome de usuário, etc ...

Os parâmetros de definição de entrada, descrevem o arquivo de entrada, especificando por exemplo, tamanho do registro; tipo de arquivo; tamanho de bloco; os campos a serem utilizados por esse programa com os respectivos nomes, posição inicial, tamanho e posições decimais; critérios de seleção a serem considerados para cada registro de entrada, etc...

Os parâmetros de definição de saída, descrevem o relatório especificando por exemplo, tamanho do registro, tamanho de páginas; a indicação de que os campos serão ordenados ou não; caracter de edição para descrever linhas cabeçalho, detalhe e total.

Os parâmetros de definição de área de trabalho, definem as áreas de trabalho necessárias a um programa, podendo ser atribuídos valores iniciais a elas.

Os parâmetros de definição de índices, permitem descrever nomes de campos com índices que serão referenciados posteriormente durante o processamento.

Os dados aos quais um programa CULPRIT tem acesso, podem estar armazenados em arquivos convencionais ou em uma base de dados IDMS, podendo neste último caso, serem definidos caminhos e registros lógicos.

Um programa CULPRIT pode eventualmente fazer referências a arquivos de entrada que sejam constituídos de dados de mais de um arquivo, tendo sido assim definidos previamente. Uma outra facilidade permitida é a catalogação de módulos desenvolvidos pelos usuários, que quando desenvolvidos em COBOL são tratados como sub-rotinas. São permitidas também catalogação de definição de saídas, processos frequentemente utilizados que podem ser copiados diretamente.

8 - BIBLIOGRAFIA

- C281 - CULLINANE DATABASE SYSTEM INC. - CULPRIT Report Generator: A management Summary, MACU-BR01-0582 Cullinane Database System Inc, Westwood, Mass., 1981.
- C282a - CULLINANE DATABASE SYSTEM INC. - Application Development System/on line - Summary Description, SADO-1282-1001 release 1.0. Cullinane Database System Inc, Westwood, Mass., 1982.
- C282b - CULLINANE DATABASE SYSTEM INC. - IDMS-DC/UCF - Mapping Facility User's Guide, TDDC-0310-2000, revision 0.0, release 2.0. Cullinane Database System Inc, Westwood, Mass., 1982.
- C282c - CULLINANE DATABASE SYSTEM INC. - Integrated Data Dictionary Summary Description, SDID-0682-3001, release 5.0. Cullinane Database System Inc, Westwood, Mass., 1982.
- C283a - CULLINANE DATABASE SYSTEM INC. - ADS/O User's Guide, TDAO-0310-1100, revision P.0, Cullinane Database System Inc, Westwood, Mass., 1983.
- C283b - CULLINET SOFTWARE INC. - CULPRIT User's Guide, TACU-0310-6102, revision 0.2. release 6.1. Cullinet Software Inc, Westwood, Mass., 1983.
- C283c - CULLINET SOFTWARE INC. - Database Analysis and Design, Eddb-0040-5701, revision 0.0. Cullinet Software Inc, Westwood, Mass., 1983.
- C283d - CULLINET SOFTWARE INC. - IDD-DDDL Reference Guide, TDID-0321-3010, Revision 1.0, release 3.0 Cullinet Database System Inc, Westwood, Mass., 1983.

- C283e - CULLINANE DATABASE SYSTEM INC. - IDMS System Overview, TDDB-0180-5502, Revision 0.2, release 5.5. Cullinane Database System Inc, Westwood, Mass., 1983.
- C283f - CULLINANE DATABASE SYSTEM INC. - IDMS - Database Design and Definition Guide. TDDB-0300-5521, revision 0.1, release 5.7. Cullinane Database Inc, Westwood, Mass., 1983.
- C283g - CULLINET SOFTWARE INC. - Introduction to the Application Development System/on line, EDAO-0020-1002. Cullinet Software Inc, Westwood, Mass., 1983.
- C283h - CULLINANE DATABASE SYSTEM INC. - On line English - Summary Description, SDDE-0383-2002. Cullinane System Inc, Westwood, Mass., 1983.
- C283i - CULLINET SOFTWARE INC. - On line English - Design and Definition Guide, TDDE-0230-3000, revision p.0, release 3.0. Cullinet Software Inc, Westwood, Mass., 1983.
- C283j - CULLINET SOFTWARE INC. - On line Query - Use's Guide, TDDQ-0320-3001. Cullinet Software Inc. Westwood, Mass., 1983.
- C284a - CULLINET SOFTWARE INC. - ADS/O - Reference, TDAO-0330-10000, revision P.0, revision P.0, Cullinet Software Inc, Westwood, Mass., 1984.
- C284b - CULLINET SOFTWARE INC. - Integrated Data Dictionary User's Guide. Cullinet Software Inc, Westwood, Mass., 1984.
- Co85 - COBB, R.H. - In Praise of 4GLS. Datamation: 31(4) 90-96, jul. 1985.

- Gr85 - GRANT, F.J. - The Downside of 4GLS. Datamation: 31(14)
99-104, jul 1985.
- Ma81 - MARTIN, J. & FINKELSTEIN, C. - Use of Fourth
Generation Language. In: Information Engineering.
Vol. 2. Savant Institute, 1981.
- Ma82 - MARTINS, J. - Application Development Without
Programmers Prentice Hall, New Jersey, 1982.
- Mo86 - MORIYA, E.H. - Uma Experiência de Desenvolvimento de
Sistemas de Informação Usando Instrumental de 4ª
Geração. Tese de Mestrado apresentada ao ICMSC-
USP, 1986..
- Ru85 - RUDOLPH, E.E. A Produtividade no Desenvolvimento de
Aplicações por computador. Data News, 20-27, ago.
1985.
- Rw85 - ROWE, L.A. - Tools for developing OLTP - Applications
Datamation: 31(15); 73-82, ago. 1985.
- Ti85 - TINNIRELLO, P. - Make a Sensible 4GL Selection
Computer Decision, 40-45, jul., 1985.