

**FORMULARIOS ELETRONICOS: DEFINIÇÃO
E MANIPULAÇÃO AUTOMÁTICA DA INTERFACE DE USUÁRIO**

José Palazzo Moreira de Oliveira - Duncan Dubugras Alcoba Ruiz

Universidade Federal do Rio Grande do Sul

Porto Alegre - Brasil

RESUMO

É apresentado o conceito de Formulário Eletrônico bem como um modelo conceitual para a definição estrutural de formatação e de restituição de formulários. O mecanismo de geração da interface de usuário é descrito com sua possibilidade de apresentação automática de formulários eletrônicos. Este projeto está sendo desenvolvido, no Curso de Pós-graduação em Ciência da Computação da Universidade Federal do Rio Grande do Sul.

Palavras-chave: automação de escritórios, formulários eletrônicos, banco de dados generalizados, interface de usuário, programa especialista.

1. Introdução

Os Sistemas de Informação tem sido utilizados, na maior parte das vezes, em atividades ligadas à administração tradicional. Estes sistemas são apoiados por SGBDs convencionais baseados em modelos que descrevem e manipulam dados simples e pouco estruturados tais como inteiros, reais e cadeias de caracteres. Recentemente tem sido desenvolvidos esforços para integrar à estes sistemas outros grupos de atividades com características particulares. Estas novas áreas de aplicação apresentam necessidades diferenciadas de manipulação de dados.

Como consequência, o desenvolvimento de novas pesquisas no campo de Banco de Dados tem-se direcionado, principalmente, para a inclusão de uma maior parcela da semântica da realidade no modelo de dados. Surge, então, a necessidade de dispor de novos tipos de dados associados à esta extensão da semântica. Estes estudos levaram ao desenvolvimento de Sistemas de Gerência de Bancos de Dados Generalizados (SGBDG). A Automação de Escritórios é uma das áreas que mais tem-se destacado nas aplicações de SGBDG. A AE trabalha com os tipos de dados documentos, formulários e tempo.

Dentro da automação de escritórios um dos aspectos mais importantes é o uso dos formulários como elementos de transferência de informação e de controle. O formulário é o elemento privilegiado para a transferência de dados dentro de uma organização. Os formulários tem, também, um papel preponderante na estrutura de controle das operações. Este controle é processado pelo desencadeamento de atividades pela chegada de um formulário a uma estação de trabalho satisfazendo certas

condições pré-estabelecidas baseadas em seu conteúdo.

Dentro de uma visão conceitual, um formulário é um tipo particular de documento generalizado [BRV 83]. No presente trabalho, entretanto, devido a sua importância particular o tipo formulário será considerado como um tipo independente, isto é, tendo definição própria. Os conceitos desenvolvidos para os documentos foram adaptados ao caso particular dos formulários.

2. Os Formulários Eletrônicos

O objetivo de um sistema integrado de formulários é a eliminação completa, dentro do ambiente de uma organização, do formulário impresso sobre papel.

Os formulários são um elemento comum nas atividades cotidianas. Eles permitem o intercâmbio ordenado e não ambíguo de informações. Estas características são devidas, principalmente, a sua estrutura ao mesmo tempo estruturada e de fácil interpretação. Além disto as convenções para o uso de formulários são bem conhecidas e de uso universal. Por estes motivos os formulários são largamente utilizados nos sistemas de aplicações de AE.

Em um sistema informatizado, os formulários existem e são armazenados de forma digital. Estes formulários são denominados **Formulários Eletrônicos**.

O presente projeto integra os trabalhos sobre Modelos de Dados com aqueles ligados a AE e visa o estudo dos diferentes aspectos envolvidos na utilização de formulários em um ambiente automatizado. Um objetivo importante a ser atingido é a integração dos conceitos desenvolvidos em modelos de dados generalizados ([Cod 79], [SNF 80], [HM 81], [KL 82], [ACO 85]) nas aplicações de escritório.

O primeiro objetivo consiste em garantir a integração dos conceitos desenvolvidos em SGBDG com as atividades de escritório de forma a manter os procedimentos naturais de manipulação de formulários. Este é um fator importante pois da transição suave, sem rupturas dos procedimentos atualmente adotados, depende a aceitação de um sistema orientado ao usuário final.

O segundo consiste em desenvolver uma estrutura conceitual consistente para a definição dos diferentes níveis de especificação de formulários. Uma definição clara dos níveis de representação de formulários é essencial para assegurar a independência de dados e de formas de apresentação.

3. A arquitetura em níveis

O conceito básico utilizado para a definição de um modelo conceitual para a definição e manipulação de formulários é a decomposição desta atividade em um conjunto de níveis. Cada nível é caracterizado por suas funções e permite a decomposição do problema de modelagem em diversos sub-conjuntos isolados.

Este modelo em níveis foi proposto, inicialmente, para a representação de documentos em [BRV 83]. O mesmo permite uma grande independência de dados e de dispositivos físicos na modelagem e desenvolvimento de sistemas de aplicação. Uma das suas principais vantagens é a independência entre a definição das operações sobre o esquema (operadores de tipo) e sobre as ocorrências de formulários.

A figura 1 apresenta a arquitetura em níveis e as etapas do processo de criação de um formulário. A geração de um formulário é composta pela descrição do tipo de formulário, de suas características de formatação, pela criação de um formulário eletrônico e, finalmente, pela reprodução de um formulário em um meio físico.

Um grupo de formulários que possuam a mesma estrutura é identificado pelo correspondente tipo de formulário. O conceito de tipo de formulário é análogo ao de tipo de dados tal como empregado em linguagens de programação. A descrição do tipo de formulário é realizada através da linguagem de definição de dados associada ao Modelo Sintático.

A geração de uma ocorrência de um formulário corresponde à associação de um conjunto de dados (Informações Textuais) ao modelo sintático. Isto é representado, no banco de dados, pela inicialização de uma área e pelo armazenamento dos dados correspondentes à estrutura definida no modelo sintático. Uma ocorrência de um conjunto de dados associada a um modelo sintático é denominada Formulário Eletrônico.

Um formulário possui características que não são associadas ao seu conteúdo mas ligadas à forma de apresentação. A associação dos formulários eletrônicos as regras de apresentação lógica denomina-se formatação. Estas regras de apresentação são por exemplo: realce de um campo ou representação de um grupo de campos associados.

As regras de apresentação podem ser divididas em dois grupos: as independentes das restrições impostas pelas dimensões lógicas das páginas e aquelas vinculadas ao formato da página ou janela de exibição. Por exemplo: um texto poder ser formatado em duas colunas (parte puramente lógica) e estas colunas devem ser representadas em uma página com 96 caracteres de largura.

A regras de formatação são lógicas e, portanto, independentes do sistema físico de restituição. Sua modificação não altera o conteúdo do formulário e, dependendo da aplicação,

um mesmo formulário eletrônico pode ser associado a diferentes conjuntos de regras de formatação.

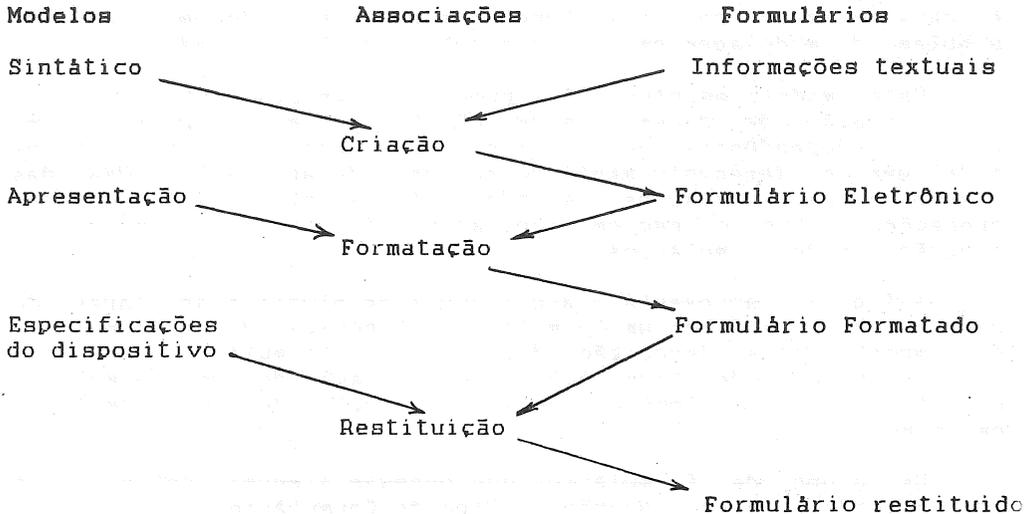


Fig 1 O modelo hierárquico para a representação de formulários.

Para ser percebido por um usuário o formulário formatado deve ser materializado em um dispositivo de saída. Esta atividade é denominada *restituição*. Um formulário é *restituído* de acordo com as especificações do equipamento (tipo de caracteres, vídeo inverso ou negrito etc). A *restituição* consiste na interpretação das regras de apresentação em função das características físicas do equipamento utilizado.

4. O modelo sintático de formulário

A descrição do tipo de formulário é feita pelo *Modelo Sintático*. Um modelo sintático constitui-se na descrição de um tipo de formulário. Um formulário é composto por um *cabeçalho* que é utilizado para a descrição externa e pelo *corpo*.

O corpo de um formulário é definido pela enumeração de suas propriedades. Cada propriedade é representada por meio de um campo definido sobre um tipo de dado semelhante aos da linguagem Pascal. Estes tipos correspondem ao conceito de *tipos simples*, *tipo renomeado* ou *tipo estruturado* [Pal 84].

Um tipo de formulário contém campos e grupos de campos, definidos sobre tipos de dados. Os grupos campos podem ser considerados entidades características tais como descritas em

[Cod-79] e utilizadas para a representação de Formulários Abstratos em [Col 85].

As diversas operações disponíveis para a definição dos tipos de formulários são constituídas por operadores atuando sobre um ou mais tipos. Os operadores de tipo permitem a definição de formulários, a partir dos tipos de base ou de tipos anteriormente definidos.

A definição sintática de um formulário compreende ainda os textos de fundo (cabecinhos, títulos de grupos de atributos, instruções e valores possíveis de atributos) e parâmetros.

Os identificadores de tipos são incluídos em um dicionário de dados onde, além da definição correspondente, são armazenadas informações sobre o uso, origem e outros dados administrativos. Desta forma é possível manter a uniformidade de descrição dos campos dos diversos tipos de formulário.

O processo de definição dos formulários é facilitado pela existência de uma função de apoio (help) que permite, a qualquer momento, a obtenção de informações sobre os campos dos formulários.

4.1 Declaração de um Formulário:

A declaração de um formulário é a associação de um identificador com uma definição de dados. Esta estrutura define todas as características do formulário e é suficiente para permitir sua restituição e manipulação.

Exemplo:

```
TYPE
  MOD70 [ATUALIZACAO DO CADASTRO DE FITAS]:
    <tipo-formulario>;
```

Sintaxe associada:

```
<declaracao-tipo-formulario> ::= <identificador>
    ['<titulo>']:' <tipo-formulario>
```

O <identificador> obedece as mesmas regras da linguagem PASCAL. O <titulo> corresponde ao nome externo do formulário.

4.2 Definição de um Formulário:

O elemento sintático <tipo-formulario> engloba toda a especificação dos elementos constantes de um formulário que constituem o "corpo do formulário". Estes elementos pertencem a três categorias: estrutura, apresentação e operações.

Exemplo:

```

...:Formulario
    estrutura

        <conjunto-elementos-estrutura>

    fim;

    apresentacao

        <conjunto-apresentacoes>

    fim;

    <def-operacoes>

fim;

```

Sintaxe associada:

```

<tipo-formulario> ::= 'Formulario' <def-formulario>
                                     'fim'

<def-formulario> ::= <def-estrutura>
                    <def-operacoes> <def-apresentacao>

<def-estrutura> ::= 'estrutura'
                  <conjunto-elementos-estrutura> 'fim;'

<def-apresentacao> ::= apresentacao'
                    <conjunto-apresentacoes> 'fim;'

```

Na <def-estrutura> é feita a definição do tipo formulário, ou seja, os campos constantes do formulário sendo definido.

Na <def-apresentacao> é feita a especificação de como os mecanismos de formatação e restituição devem agir a nível de interface de usuário.

Na <def-operacoes> é realizada a definição das operações que podem ser executadas sobre o formulário.

A definição do tipo em <def-estrutura> obedece aos mesmos mecanismos disponíveis na linguagem PASCAL existindo também operadores de tipo definidos em <tipo-derivado>. Estes modelos refletem as necessidades de manipulação de dados do usuário, ou seja, é definido um modelo que represente com fidelidade a realidade modelada.

Exemplo:

```

...: Formulario
    estrutura
        nro-serie [NRO. SERIE]:String(6);
        tamanho[TAMANHO]:('P','M','G');
        nome-resp[NOME DO RESPONSAVEL]:String(20);
    fim;
fim;

```

Sintaxe associada:

```

<conjunto-elementos-estrutura> ::= <elemento-estrutura> |
    <elemento-estrutura> ';' <conjunto-elementos-estrutura>

<elemento-estrutura> ::= 'lista (' <min> ',' <max> ') de '
    <declaracao-elemento> |
    'caso' <conjunto-casos> 'fim' |
    <identificador-tipo-formulario> '.estrutura' |
    <declaracao-elemento>

```

Através de <elemento-estrutura>, é possível criar novos formulários utilizando definições já existentes, estipular um intervalo do número de ocorrências possíveis e, também possibilidades sintáticas alternativas para um mesmo formulário (caso).

```

<declaracao-elemento> ::= <identificador '[' <titulo>']':
    <elemento>

```

```

<elemento> ::= <tipo-simples> | <tipo-restrito> |
    <tipo-composto> | <tipo-derivado>

```

```

<tipo-simples> ::= 'Inteiro' | 'Real' | 'Booleano' |
    'String (' <inteiro-sem-sinal> ')

```

A <declaracao-elemento> é semelhante à declaracao de variáveis na linguagem PASCAL. A diferença existente é a associação de um <titulo> ao identificador do campo para servir de identificação do campo ao usuário.

Os <tipo-simples> possuem características de armazenamento, manipulação e formatação padrões e, por conseguinte, não possuem especificação de número de posições (exceto os do tipo String) e restrições de integridade: são assumidas as características do equipamento no qual os formulários serão manipulados.

A definição de <tipo-restrito> engloba dois tipos de dados existentes na PASCAL:

```

<tipo-restrito> ::= <tipo-escalar> | <tipo-intervalo>
<tipo-escalar> ::= '(' <conjunto-elementos-escalares> ')'
<conjunto-elementos-escalares> ::= <elemento-escalar> |
    <elemento-escalar> ',' <conjunto-elementos-escalares>
<elemento-escalar> ::= <string> | <constante-numerica>
<tipo-intervalo> ::= '(' <constante-numerica> '..'
    <constante-numerica> ')'
```

A utilização desses tipos obedece os mesmos padrões da linguagem PASCAL.

```

<tipo-composto> ::= <tipo-arranjo> | <tipo-registro>
<tipo-arranjo> ::= 'arranjo (' <inteiro-sem-sinal>
    ')' de ' <tipo-basico>
<tipo-registro> ::= 'registro' <conjunto-tipos-basicos>
    'fim'
<conjunto-tipos-basicos> ::= <declaracao-tipo-basico> |
    <declaracao-tipo-basico> ';' <conjunto-tipos-basicos>
```

O uso de <tipo-composto> torna extremamente flexível a definição de formulários incluindo tabelas de campos e hierarquias.

TYPE

MOD30[ATUALIZACAO DE HORARIOS]:

Formulario

estrutura

```

cod-func[CODIGO FUNCIONARIO]:num-dc-func;
transacao[TRANSACAO]: (1..3);
mes-ref[HORARIO DO MES]: ('JAN', 'FEV', 'MAR', 'ABR',
    'MAI', 'JUN', 'JUL', 'AGO', 'SET', 'OUT', 'NOV', 'DEZ');
data-preench[DATA]:data6;
regime[REGIME]: (1,2);
benef-sem[BENEFICIO SEMANAL]:hora4;
carga-hor[CARGA HORARIA CONTRATUAL]:hora4;
horario-intencao[HORARIO DE INTENCAO]:
    arranjo (14) de horario;
horario-obrigatorio[HORARIO OBRIGATORIO]:
    arranjo (14) de horario;
```

fim;

onde:

type

 horario:

 registro

 inicio[INICIO]:hora4;

 duracao[DURACAO]:hora3-duracao;

 dia[DIA]:(2..7);

 fim;

Neste exemplo, ficam claras as vantagens da utilização de tipos como os <tipo-registro>s e <tipo-composto>s.

5. O modelo de apresentação

Nesta estrutura sintática é feita a especificação de parâmetros de formatação e textos para orientação do usuário a nível de manipulação via interface de usuário.

Exemplo:

 apresentacao

 DOCUMENTAMOD70:

 globais

 cabecalho moldura enfatizado [UFRGS CPD];

 fim;

 nodo nro-serie: esquerda moldura;

 nodo tamanho: centrado moldura

 #P - PEQUENO, M - MEDIO, G - GRANDE#;

 nodo nome-resp: moldura;

 fim;

 fim;

Sintaxe associada:

Para:

 apresentacao

 DOCUMENTAMOD70:

 <construtor-apresentacao>

 fim;

 fim;

```

<def-apresentacao> ::= 'apresentacao'
    <conjunto-apresentacoes> 'fim;'

<conjunto-apresentacoes> ::= <apresentacao> |
    <apresentacao> ';' <conjunto-apresentacoes>

<apresentacao> ::= <identificador> ':'
    <construtor-apresentacao>

```

Em cada <instancia-apresentacao>, é especificada uma forma de localizar e representar os campos. Um mesmo tipo formulário pode ter uma a "n" apresentações diferentes. Cada representação estará associada a, pelo menos, uma operação sobre o formulário. A vinculação das apresentações com as operações é feita em <lista-operacoes>.

Exemplo:

```

DOCUMENTAMOD70:
    globais
        cabecalho <forma-apresentacao> [UFRGS CPD];
    fim;
    ...
    fim;

```

Sintaxe associada:

```

<construtor-apresentacao> ::= 'globais'
    <conjunto-apresentacoes-globais> 'fim;'
    <conjunto-apresentacoes-nodos>

<conjunto-apresentacoes-globais> ::= <apresentacao-global> |
    <apresentacao-global> ';' <conjunto-apresentacoes-globais>

<apresentacao-global> ::= <apresentacao-cabecalhos>
    <apresentacao-textos>

<apresentacao-cabecalhos> ::= <empty> | 'cabecalho'
    <forma-apresentacao> '[' <texto> '];'

<apresentacao-textos> ::= <empty> | <conteudo-texto>

```

Em <apresentacao-global>, são definidas sentenças que titulam o formulário (<apresentacao-cabecalhos>) e textos orientadores ao usuário (<apresentacao-textos>). Os textos orientadores são visíveis ao usuário via os mecanismos de "help".

Exemplo:

DOCUMENTAMOD70:

```

...
nodo nro-serie: <def-posicao> <forma-apresentacao>;
nodo tamanho: <def-posicao> <forma-apresentacao>
                #P - PEQUENO, M - MEDIO, G - GRANDE#;
nodo nome-resp: <def-posicao> <forma-apresentacao>;

```

Sintaxe associada:

```

<conjunto-apresentacoes-nodos> ::= <apresentacao-nodo> |
<apresentacao-nodo> ';' <conjunto-apresentacoes-nodos>
<apresentacao-nodo> ::= 'nodo' <nome-nodo> ':'
                        <def-posicao> <forma-apresentacao>
                        <apresentacao-textos>

```

Na <apresentacao-nodo> são especificadas orientações à interface de usuário a respeito de posicionamento e apresentação. Em <conteudo-texto> são definidas orientações ao usuário sobre o campo, acessíveis através dos mecanismos de "help".

Sendo que:

```

<def-posicao> ::= <empty> | 'centrado' | 'direita' |
                'esquerda'
<forma-apresentacao> ::= <contorno> <apresentacao>
<contorno> ::= <empty> | 'moldura'
<apresentacao> ::= <empty> | 'secreto' | 'enfaticado'
<conteudo-texto> ::= '#' <texto> '#'
<texto> ::= {sequencia de caracteres exceto ']' e '#'}

```

A <def-posicao> orienta a interface de usuário quanto a disposição do campo no plano de representação do formulário; a <forma-apresentacao>, quanto ao padrão de apresentação (formatação e restituição).

6. O processo de formatação e de restituição

A partir da definição sintática e das opções escolhidas na definição do modelo de apresentação é possível formatar os dados representados por um formulário eletrônico e posteriormente restituir o formulário formatado em um meio físico. O processo de restituição é vinculado ao dispositivo físico utilizado e

consiste no mapeamento dos dados para uma estrutura física bidimensional equivalente a um formulário tradicional sobre uma folha de papel.

A atividade de formatação é equivalente à realizada, nos sistemas tradicionais, pelo analista de formulários. O "lay-out" de um formulário é definido a partir dos dados a serem manipulados e é orientado por uma série de regras adotadas por uma organização ou regras estéticas que levam a uma aparência agradável e que propiciam uma utilização facilitada.

A aptidão para desenvolver este trabalho é adquirida ao longo de um aprendizado lento e quase sempre não metódico. Dependendo muito de qualidades artesanais é um processo laborioso e de resultados não assegurados.

Pode ser traçado um paralelo entre a realização do "lay-out" de um formulário e a datilografia de um texto. A separação das palavras para permitir o ajustamento da margem direita é uma tarefa para datilógrafas bem treinadas. Os editores de texto simples resolvem este problema incluindo espaços entre as palavras e hifenizando alguma palavra mais longa. Desta forma a tarefa de justificação é automatizada liberando o digitador das decisões ligadas a apresentação do texto.

Entretanto as decisões tomadas por uma boa datilógrafa não se limitam ao conteúdo de uma linha mas levam em conta toda a aparência de uma página. Neste caso as diferentes decisões elementares são dependentes das demais decisões já tomadas, isto é, as decisões são dependentes do contexto.

Em um sistema automatizado é possível representar-se as diferentes regras de formatação e o processo decisório dependente do contexto de tal forma que cada página apresente a melhor configuração de acordo com certos critérios estéticos. Este método é utilizado pelo editor COBATEF [PUN-85]. O conjunto de regras e de procedimentos necessários para realizar esta tarefa formam um sistema especialista que realiza o processo de decisão que de outra forma seria desenvolvido pelo redator do documento.

No caso de formulários eletrônicos a transferência destas atividades de geração do "lay-out" para um sistema especialista pode gerar uma melhoria considerável na qualidade do interface de usuário. A economia de tempo é também apreciável pois esta constitui-se em uma fase longa na concepção de um formulário. Por outro lado a possibilidade de serem realizadas operações de consulta envolvendo dois ou mais formulários eletrônicos e gerando um novo tipo, não definido no esquema, apresenta uma necessidade de geração automática de "lay-outs".

Em um sistema que permite a manipulação de formulários de forma não limitada aos tipos definidos no esquema a possibilidade de formatação automática é essencial. Os tipos novos são gerados

por meio de operações sobre os tipos já existentes criando novos tipos para os quais não existem definições, em forma explícita, das regras de formatação.

6.1 Arquitetura do gerenciador do interface

A arquitetura do gerenciador de interface foi desenvolvida a partir dos conceitos empregados para a construção de sistemas especialistas. O conhecimento necessário para a execução da sua atividade é representada por um conjunto de regras que são interpretadas para produzir a configuração do formulário a ser restituído.

Cada possibilidade de formatação está sendo codificada em forma de regras que permitem selecionar uma ação a partir do estado parcial da restituição do formulário e das diferentes possibilidades de apresentação disponíveis para uma determinada estrutura de dados. As regras tem a seguinte estrutura:

`<estado_form_X> & <estrut_dados_A> --> <rotina_Y>, <coef_mérito>`

onde `<coef_mérito>` é um coeficiente indicando uma avaliação quantitativa sobre a qualidade da solução obtida pela execução da `<rotina_Y>`. A arquitetura deste sistema é representada na figura 2.

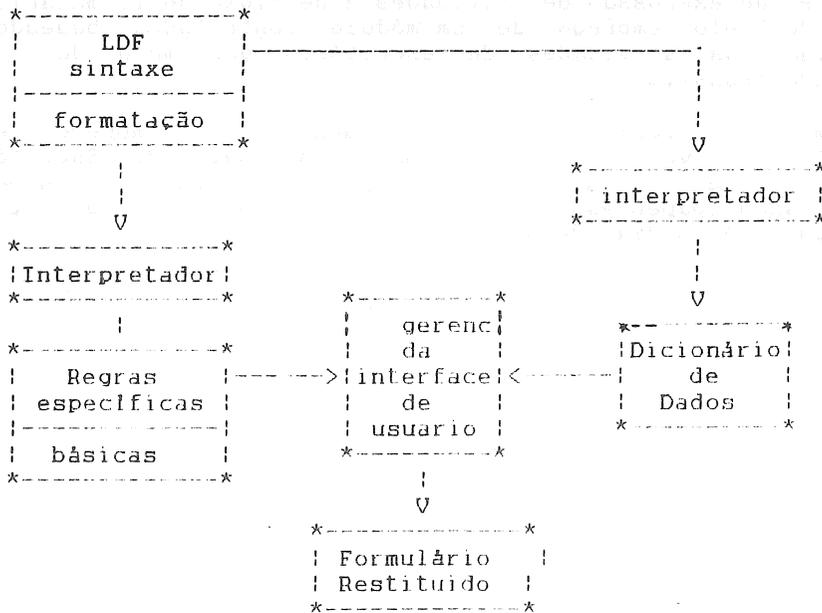


Fig 2: Arquitetura do sistema de restituição

Para um determinado formulário, a definição de formatação permite o interpretador da Linguagem de Definição incluir novas regras à base de conhecimentos. Estas regras estarão ligadas a exclusivamente este tipo de formulário. Quanto interpretadas, em conjunto com as regras básicas, permitem a formatação completa do formulário.

7. Conclusões

Em nossos estudos sobre Automação de Escritórios, tornou-se evidente que o sucesso de um sistema integrado de AE está intimamente ligado a possibilidade de automatizar as funções de apoio liberando as pessoas para as atividades criativas. A solução trivial de automatizar as tarefas de baixo nível sem ser exploradas as possibilidades evoluídas da automação levam a sistemas pouco integrados que produzem resultados fracos para o aumento da eficácia do trabalho.

O projeto de Formulários Eletrônicos [Pal 86] destina-se ao desenvolvimento, tanto no plano conceitual quanto no prático, de soluções para atividades de AE. O acesso aos recursos distribuídos de computação é obtido por uma rede de microcomputadores e por um servidor de formulários. A interface de usuário é gerenciada por um sistema especialista. Finalmente o controle de execução de atividades e de fluxo de formulários é garantido pelo emprego de um módulo controlador baseado na descrição das atividades de escritório por meio de redes predicado-transição.

Em nosso ponto de vista, somente pelo emprego destas metodologias evoluídas os sistemas de Automação de Escritórios poderão atingir o nível de difusão e aceitação imaginados a alguns anos (paperless-office) e não confirmado pela evolução da automação até os dias de hoje.

BIBLIOGRAFIA

- [BRV 83] G. Bogo, H. Richy, I. Vatton, "Proposition de Modèle pour la Normalisation des Documents", Relatório de pesquisa TIGRE n 3, março 83, Institut National Polytechnique de Grenoble, França.
- [Cod 79] EF Codd, "Extending the Relational Database Model to Capture more Meaning", ACM Transactions on Database Systems, v 4, n 4, pag ???, dez 79.
- [Col 85] C. Collet, "Les Formulaires Multi-Media", Séminaire INFORSID '85, Luchon, França, mai 1985.
- [HM 81] M. Hammer & D McLeod, "Database Description with SMD: a Semantic Database Model", ACM Transactions on Database Systems, v 6, n 3, pag 351, set 81.
- [KL 82] I. Kowalski & M. Lopez, "The Document Concept in a Database", Proceedings da ACM SIGMOD Conference, Orlando, Flórida, USA, 82.
- [Pal 84] J. Palazzo Oliveira, "Le Modèle de Données et sa Représentation Relationnelle dans un Systeme de Gestion de Bases de Données Generalisées", Tese de Doutorado, Institut National Polytechnique de Grenoble, 1984, França.
- [Pal 86] J. Palazzo Oliveira, "Electronic Forms, project description.", Relatório de pesquisa CPGCC/UFRGS n 47, jun 86.
- [PJN-85] A. Peels, N. Jansen & W. Nawijn, "Document Architecture and Text Editing", ACM TOIS, V 3, N 4, OCT 85, pag 347.
- [SNF 80] C.S. Santos, E.J. Neuhold & A.L. Furtado, "A Data-type Approach to the Entity-Relationship Model", em Entity Relationship Approach to System Analysis and Design, ed P.P. Chen, North Holland, 80.
- [Tsi 82] D. Tsichritzis, "Form Management", Communications of ACM, v 25, n 7, jul 82.
- [YHS 84] S.B. Yao, A.R. Hevner & Z. Shi, "Formanager: an Office Forms Management System", ACM Transactions on Office Information Systems, v 2, n 3, pag 453.