Detalhamento de um Framework Vertical para Sistemas Contábeis

- Referência: An Object-Oriented Framework for Accounting Systems

Sistemas Contábeis

- Numa definição apressada, sistemas contábeis representam o fluxo de dinheiro numa organização
- Exemplos:
  - Folha de pagamento
  - Contabilidade geral
  - Vendas
  - Investimentos (portfólios de ações)
  - Orçamento
  - Sistemas bancários
  - Finanças pessoais
- Coisas que tais sistemas têm em comum:
  - Contas
  - Transações envolvendo uma ou mais contas
    - A palavra "Transação" *não* é usada aqui no sentido de operação atômica em BD
- Definição mais larga
  - Representam mudanças no tempo de categorias de objetos
    - Mesmo envolvendo dados não financeiros
  - Exemplos:
    - Inventário (numa loja)
    - Inventário (na manufatura)
    - Dados de um censo
    - Alguns dados meteorológicos
    - Registro de notas de alunos
  - "Contas" e "Transações" podem representar os aspectos dinâmicos de tais sistemas
- Características essenciais

- Forma de acessar transações por data
  - Ex. Quantos itens foram comprados este mês?
- Habilidade de criar grupos de contas
  - Ex. Quais são as vendas por região?
  - Tais grupos contêm outras contas
  - Eles combinam informação das contas inclusas
    - Ex. Total de algum valor
  - Transações não são aplicáveis diretamente a grupos mas a contas individuais

Descrição de um Framework para Sistemas Contábeis

Contas e Transações

- As duas classes básicas do framework são Conta e Transação
- Uma conta:
  - Mantém um registro de transações lançadas para ela
    - São contas que armazenam transações
  - Pode calcular valores de atributos para faixas de datas
- Uma transação:
  - Registra um evento *numérico* significativo, numa certa data, sobre objetos de uma categoria relevante ao sistema

Papeis

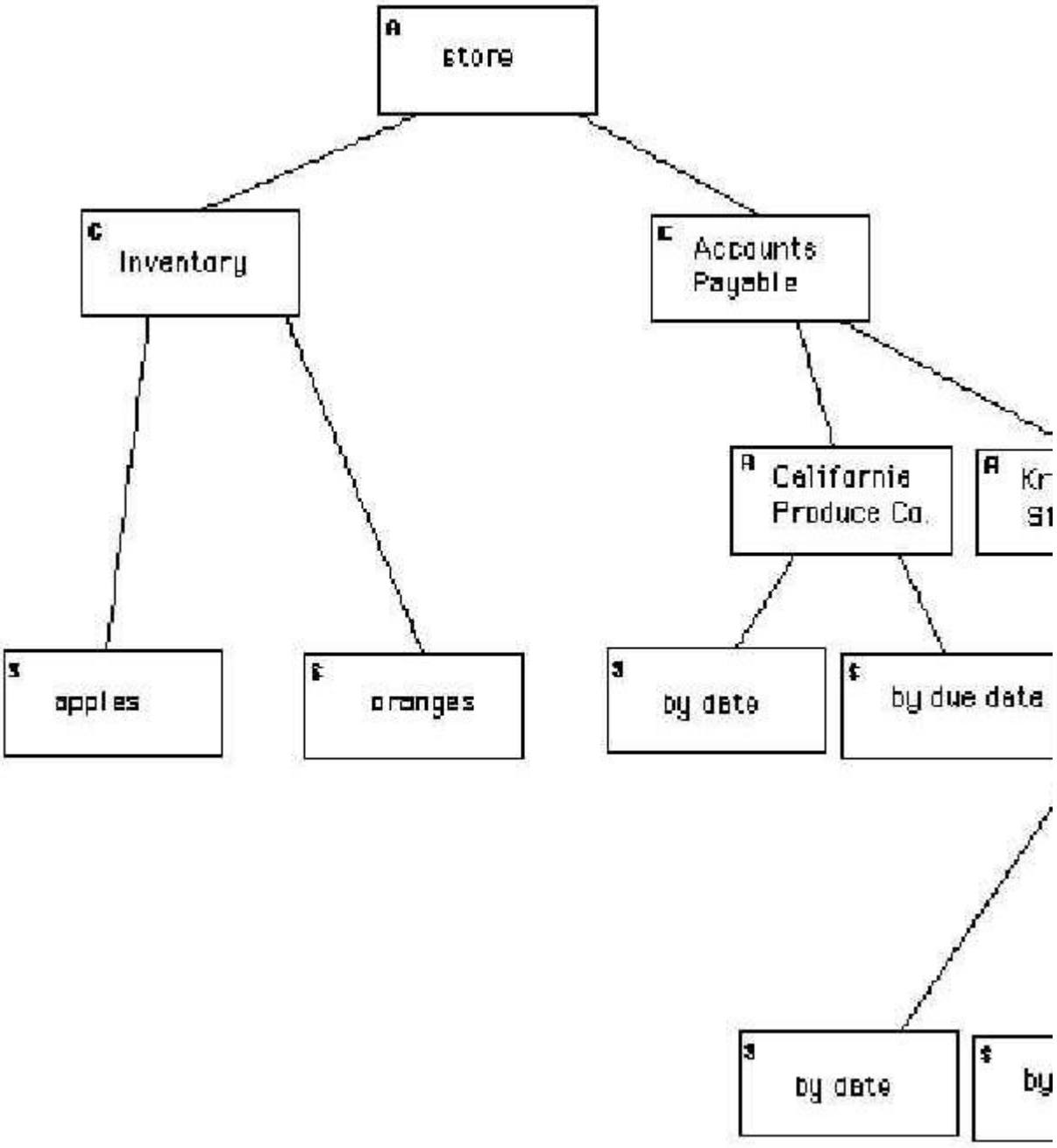
- Programador de aplicações (usando o framework pronto)
  - Determina o tipo de contas e suas relações para um domínio especial de problema
    - Exemplo: Inventário
      - Contas para itens de inventário

        - Contas para fornecedores
    - Exemplo: Corretor de investimentos
        - Contas para clientes (portfólios)
        - Contas para instrumentos de investimentos (ações, papeis, fundos, ...)
    - Determina o tipo de transações e como lançá-las nas contas
    - Determina os atributos a manter em cada conta e em cada transação
    - Desenvolve uma interface para o sistema
- Criador de contas
    - Cria contas para um sistema particular (= deployment da aplicação)
        - Ex. Que contas específicas de inventário são mantidas?
            - Canários, cadeiras, lápis, lâminas de aço, ...
    - Dá nomes às contas
- Usuário da aplicação
    - Faz a entrada de dados
    - Lança transações

Contas

- Contas simples
    - Contas do nível mais baixo (folhas da floresta de contas)
    - É apenas aqui que se armazenam transações lançadas
- Contas compostas
    - Contêm contas "homogêneas"
        - As contas podem não ser idênticas mas têm um conjunto de nomes de atributos em comum
        - Podem também ter outros atributos não compartilhados entre as contas
        - Útil quando os atributos da conta composta são somatórios de valores das contas filhas

- Contas agregadas
  - Contêm contas não homogêneas
- Uso típico
  - Tipicamente, há uma única conta agregada no topo
    - A estrutura da conta é uma árvore, embora o framework permita ter uma floresta
    - É o ponto de entrada no sistema contábil para lançar transações
      - Toda transação é lançada aí
  - Contas compostas são usadas para manter um *diário*
  - Contas agregadas são usadas:
    - Quando os filhos armazenam informação diferente; ou
    - Quando uma transação deve ser lançada para mais de um filho
      - Exemplo: Contas a pagar
        - Há contas organizadas por data de vencimento
          - "Quanto devo à empresa XYZ?"
        - Há contas organizadas por data de compra
          - "Quanto compramos este mês?"
        - Na figura abaixo, S=Simples, C=Composta, A=Agregada

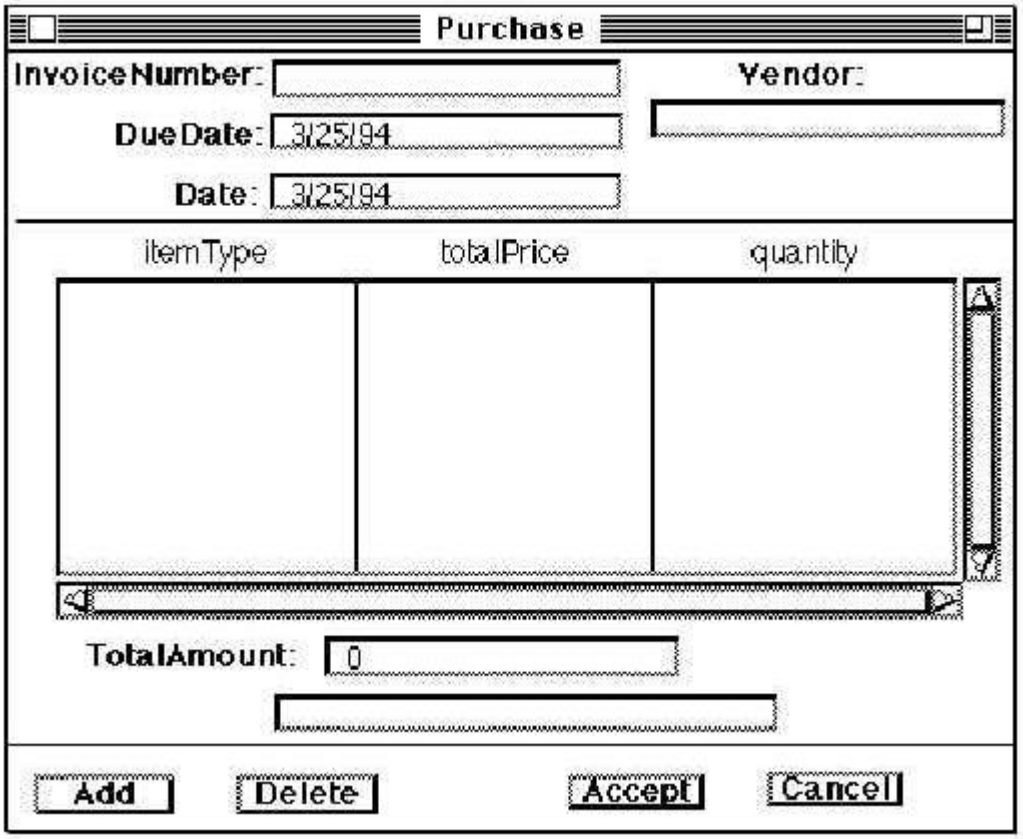


Transações

- O lançamento de uma transação inicia na raiz de uma árvore de contas
  - Normalmente uma conta agregada
- A transação desce a árvore até chegar a uma ou mais folhas
- Cada conta intermediária decide como lançar a transação "para baixo"
  - Se for uma conta composta, uma subconta é escolhida para receber a transação

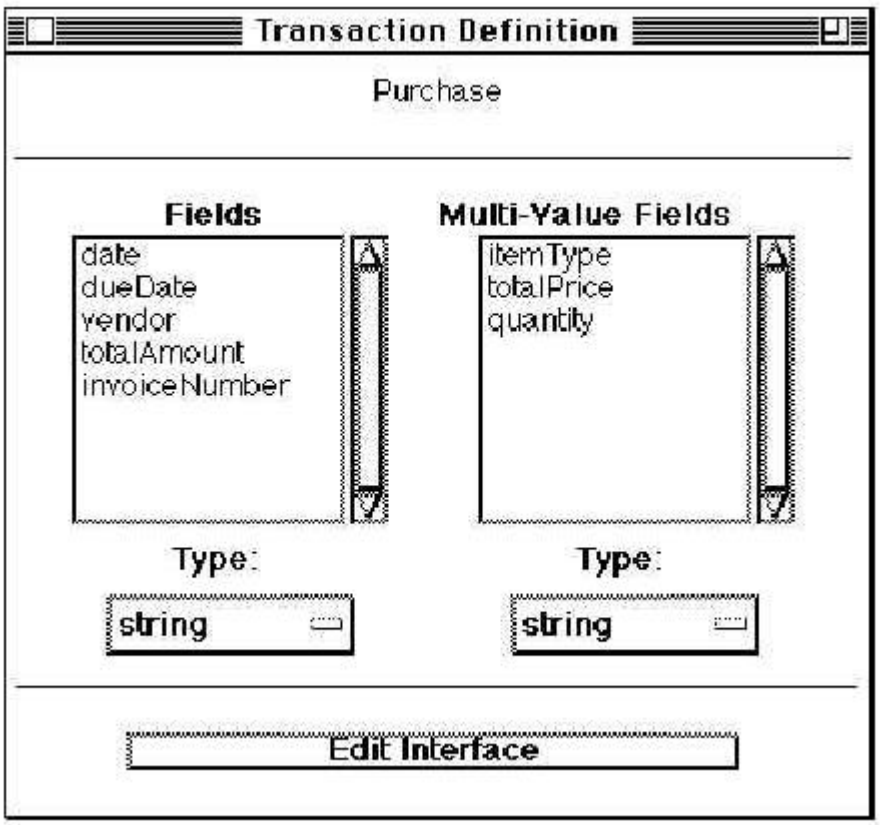
- Se for uma conta agregada, a transação é convertida em uma ou mais transações, lançadas em subcontas

- Por que desdobrar uma transação em mais transações?
  - Situação mais comum: a informação contida na transação é de interesse de várias subcontas
  - Exemplo:
    - Se uma fatura (a pagar) contiver vários itens de inventário, uma transação separada seria criada para cada item de inventário e outra transação seria criada para lançamento na conta do fornecedor
- Transações podem conter campos com valor único e campos multivalorados
  - A figura abaixo mostra um exemplo com ambos os casos



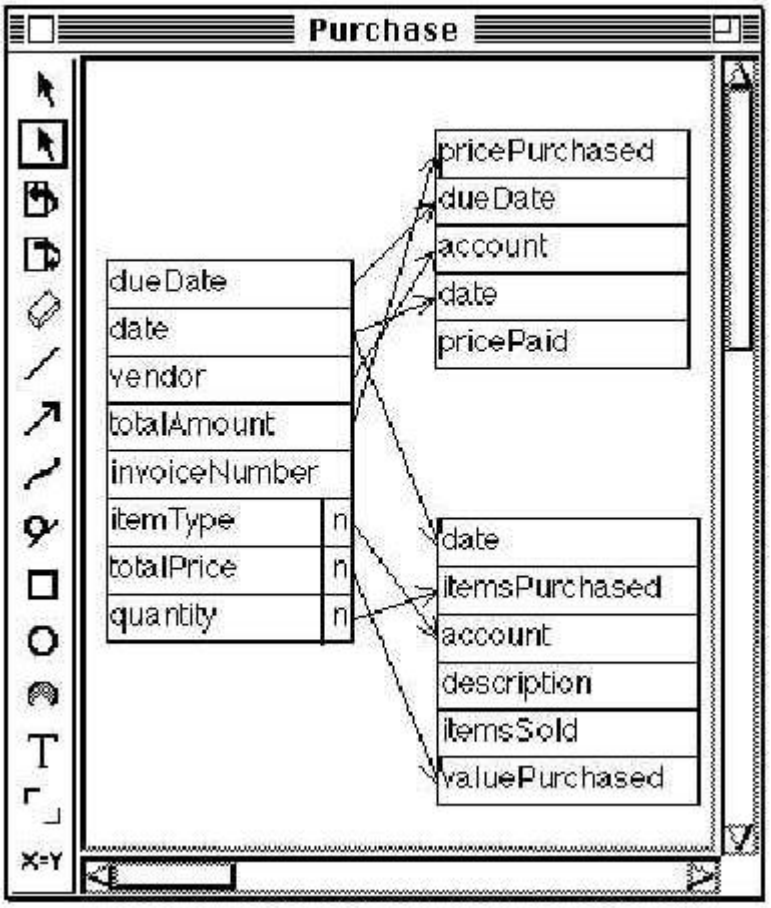
- A criação de um novo tipo de transação envolve:
  - A definição dos campos individuais e multivalorados

- A definição do tipo de cada campo (int, String, ...)
- A figura abaixo mostra uma interface possível para a criação de transações



Lançamento de Transações

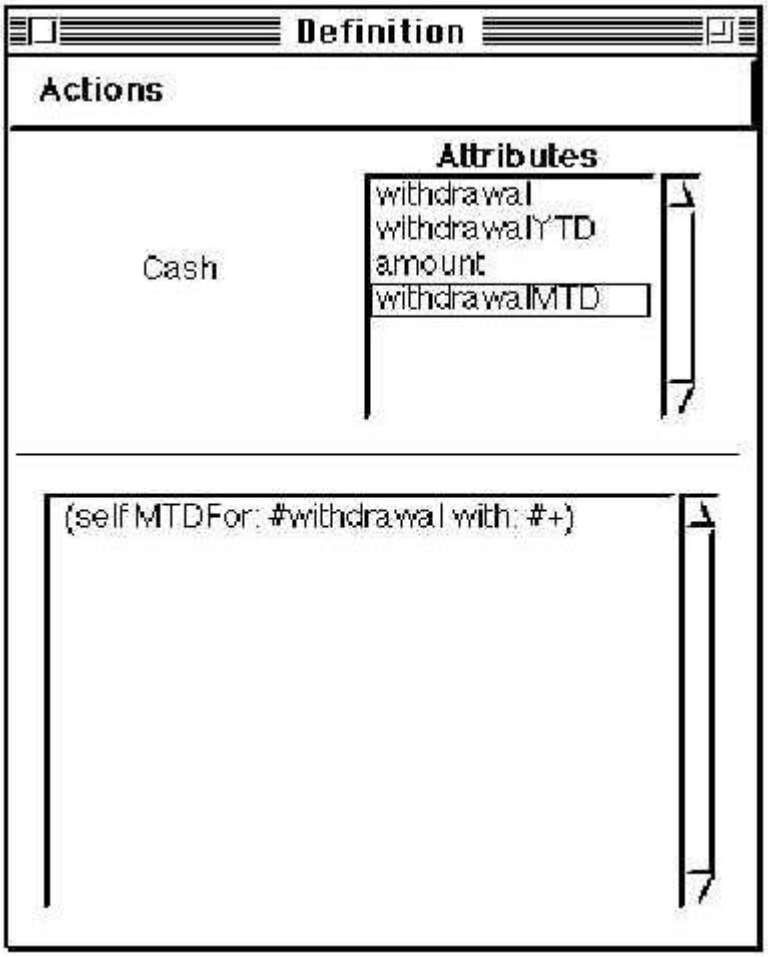
- Cada conta deve decidir como tratar uma transação
- Contas simples e compostas nunca alteram a transação
- Para contas agregadas, especifica-se um mapeamento de campos da transação original para os campos das novas transações
  - Define-se como calcular o valor de cada campo de cada nova transação
- Uma interface possível para a definição do mapeamento aparece abaixo
  - "n" significa um campo multivalorado, o que gera várias transações, uma para cada valor



- Paa contas compostas, a transação deve ter um campo chamado "Conta" que indica em qual subconta lançar a transação

Atributos de contas

- Permitem armazenar informação agregada sobre transações
- Envolvem expressões aritméticas combinando outros atributos, campos de transações, constantes, datas, etc.
- A figura abaixo mostra a definição de um atributo
  - withdrawalMTD = Saques do mês até hoje (Month To Date)
  - A expressão obedece ao padrão self<faixaDeTempo>For:#<nomeDeCampo>with:#
  - onde
    - faixaDeTempo pode ser "total", "MTD", "YTD"
    - nomeDeCampo é o nome de algum campo da transação
    - operador pode ser +, *, min, max



Detalhes de projeto

- Podem ser vistos na referência

frame-6 programa anterior próxima