

7 Conclusões

Neste trabalho, introduzimos o conceito de *circuito de objetos* como metáfora de computação. Fazendo uma ponte com circuitos elétricos, construímos paulatinamente um modelo onde o fluxo de informação no circuito é representado por estruturas de dados complexas denominadas *objetos*. O foco deste trabalho inicial foi o de descrever completamente os elementos que compõem um circuito de objetos – dispositivos, barramentos e nós – e as interações que ocorrem entre eles. A descrição rigorosa do objeto foi intencionalmente deixada para uma segunda etapa, e os objetos presentes nos exemplos foram invariavelmente simples: números, listas e funções.

Esta aparente quebra de simetria na importância dada aos circuitos e aos objetos tem explicação. Objetos têm sido extensivamente utilizados em programação e são bem conhecidos para este fim; já circuitos, não, excetuando-se aí o caso de micro-programação. No que diz respeito à Ciência da Computação, a “novidade”, por assim dizer, está no circuito, e não no objeto. Por pragmatismo, optamos por manter o foco no que há de novo, de forma a manter o trabalho mais enxuto e de fácil compreensão.

Argumentamos que a abordagem de circuitos de objetos possui um nicho próprio onde sua aplicação excede em performance àquela obtida por outras tecnologias. Em particular, domínios com alto grau de concorrência e dinamismo, como simulação, tiram o máximo proveito de circuitos de objetos. Possivelmente, seu uso pode ser transportado com sucesso para outras áreas, ficando aí faltando uma comprovação experimental.

De uma forma geral, esperamos que a flexibilidade conferida ao modelo de circuitos de objetos graças à utilização de componentes de software estenda-se para outros domínios de aplicação, tornando circuitos de objetos um modelo de uso genérico. Existem indícios nesta direção. O encapsulamento de dispositivos permite que um sistema qualquer baseado em circuitos de objetos seja estendido de forma a adicionar novas funcionalidades, sem que nenhuma parte existente seja

modificada ou mesmo tenha implementação conhecida. Por exemplo, considere um sistema composto de três dispositivos: interface com o usuário, camada de negócios e banco de dados. A camada de negócios faz a ponte entre a interface com o usuário e o banco, tal como na Ilustração 1. A conexão entre os módulos foi simplificada.

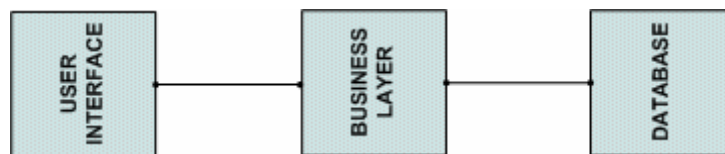


Ilustração 1 Um sistema típico em três camadas.

Suponha agora que seja preciso gravar em arquivo todas as ações do usuário no sistema. Idealmente, tudo que é preciso fazer é adicionar um novo dispositivo *em paralelo* que “intercepta” os sinais da interface, como mostrado na Ilustração 2:

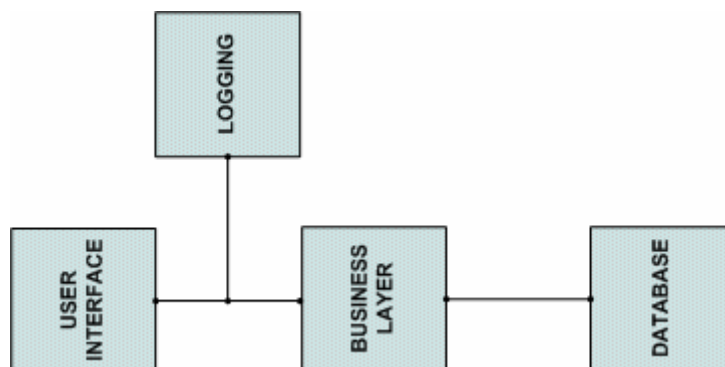


Ilustração 2 Adicionando capacidade de *logging* ao sistema.

Continuando o exemplo, se for necessário monitorar os objetos que seguem para o banco, possivelmente controlando seu tráfego, isto é tão simples quanto adicionar um novo dispositivo *em série* ao circuito original:



Ilustração 3 Adicionando um filtro de objetos.

Finalmente, é possível ainda trocar a camada de negócios, encapsulando a camada original num novo dispositivo:

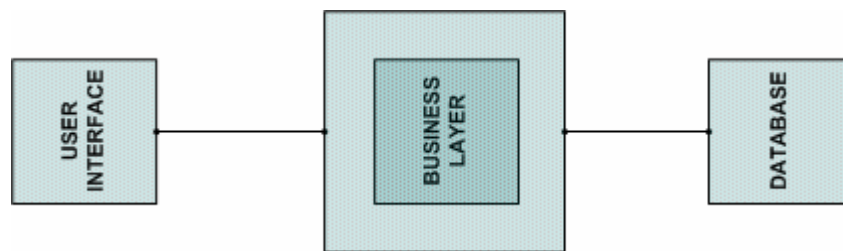


Ilustração 4 Encapsulando a camada de negócios.

Não há nada de surpreendente nestes diagramas; de fato, não seria estranho que um programador de uma linguagem OO elaborasse algo similar como parte da especificação do software, e a partir daí passasse a desenvolver as classes que integram cada módulo.

A diferença crucial é que, no caso de circuitos de objetos, a especificação também faz parte da implementação: basta que o designer detalhe sucessivamente os componentes que integram cada um dos dispositivos do nível imediatamente acima, e, ao final do processo, terá um circuito completo e funcional. Podemos então dizer que uma implementação de circuitos de objetos é uma especificação *executável* do sistema.

Por fim, a conclusão final deste trabalho é a de que as idéias apresentadas aqui são inéditas, até certo ponto inacabadas, e que somente uma análise mais criteriosa será capaz de julgar onde é melhor aplicá-las.