

Design de Arquitectura

SALA 1

- Rafael Palma - 201600649
- Alexandre Coelho - 201701820
- José Castanhas - 201901389
- Luís Carvalho - 201702393
- André Nascimento - 201400141

Índice

- Estilos de arquitetura
 - Structure
 - Shared Memory
 - Messaging
 - Adaptative Systems
 - Distributed Systems

Structure:

- Component-based
- Monolithic application
- Layered
- **Pipes and filters**

Shared Memory:

- Database-centric
- **Blackboard**
- Rule-based

Messaging:

- **Event-driven**
- Publish-subscribe
- Asynchronous messaging

Adaptative Systems:

- **Plug-ins**
- Microkernel
- Reflection
- Domain specific languages

Distributed Systems:

- Client-server (multitier architecture exhibits this style)
- Shared nothing architecture
- Space-based architecture
- Object request broker
- **Peer-to-peer**
- Representational state transfer (REST)
- Service-oriented
- Cloud computing patterns

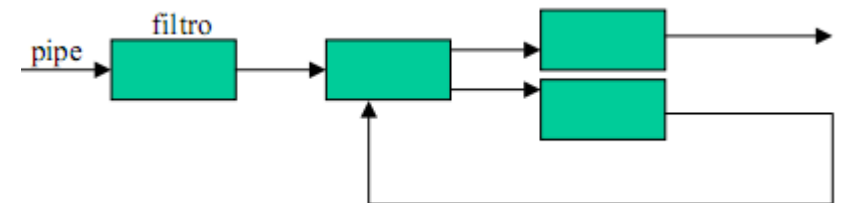
Pipes and Filters Architecture

Em engenharia de software, esta arquitectura encontra-se num estilo orientado a dados.

- Composto por uma cadeia de elementos de processamento dispostos de tal forma que a saída de cada elemento é a entrada do próximo, ou seja, os programas (filtros) processam as entradas recebidas e geram uma nova saída.

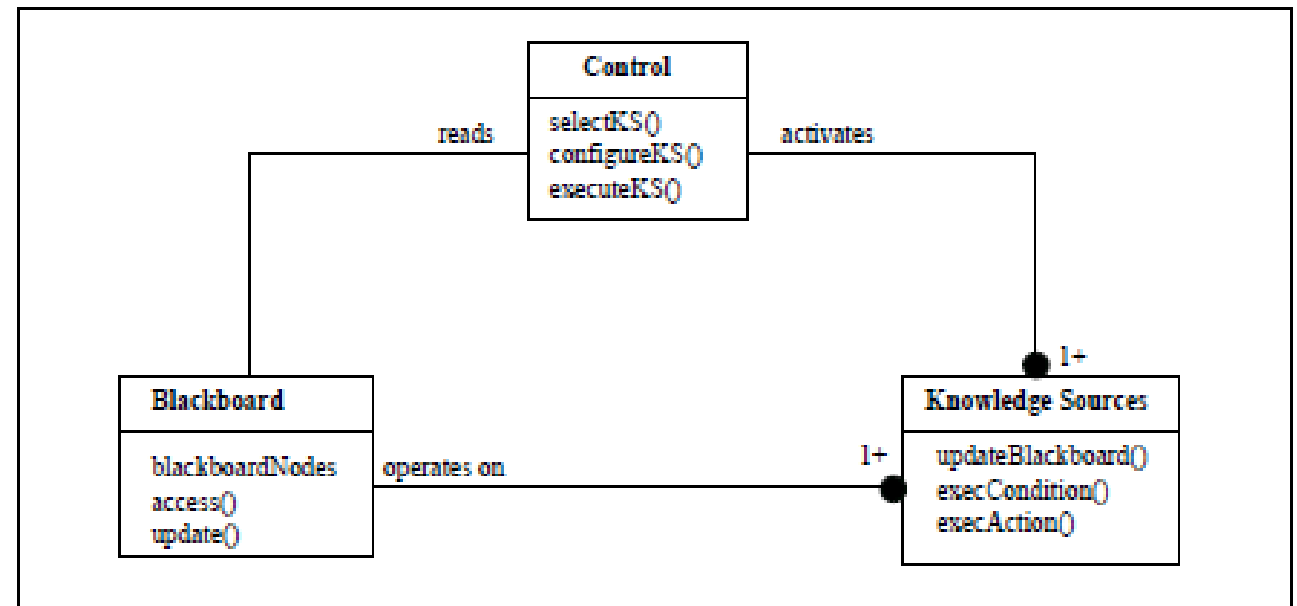


Um bom exemplo deste padrão de arquitectura é a shell de Linux, onde existem uma quantidade de pequenos programas com funções singulares, que podem ser encadeados na mesma linha de comando.

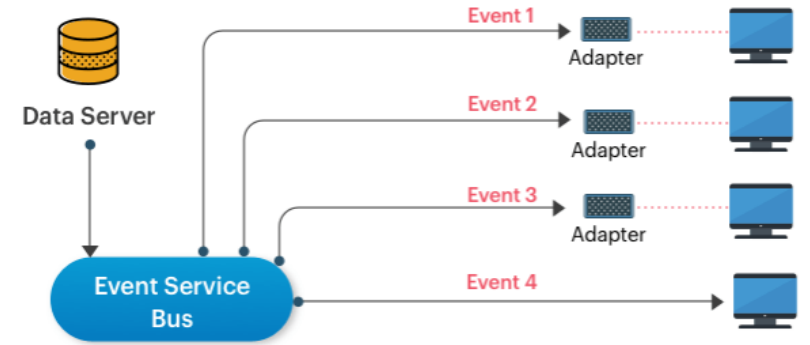


Blackboard Architecture

- Aplicação iterativa onde as fontes realizam updates á blackboard, com o objetivo de encontrar um fim para uma solução.
- Existem 3 grandes componentes, sendo estes:
 - Blackboard.
 - Fontes de conhecimento.
 - Componente de controlo.



Event-driven Architecture



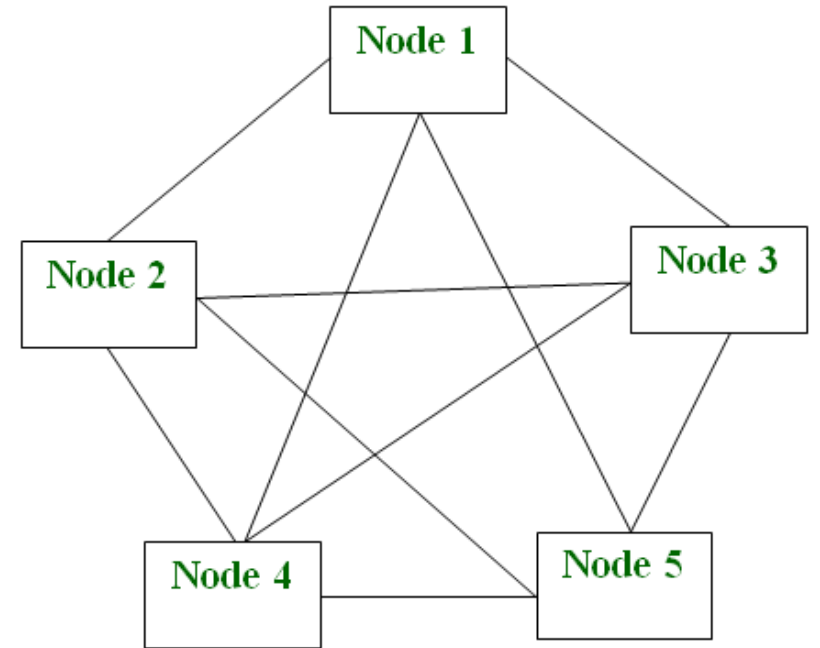
- Retrata um padrão de arquitetura de software de alta performance e ágil;
- É constituído de componentes que recebem e processam eventos assincronamente;
- Ajuda com *user interfaces* e é benéfico para aplicações onde blocos de dados individuais interagem apenas com poucos módulos.
- Um bom exemplo de uma boa implementação desta arquitetura é um site de e-commerce, onde o website tem de reagir a diversas fontes quando existe uma procura alta;
- Constitui-se de duas topologias:
 - **Mediator** - É utilizado quando diversos passos/eventos necessitam de ser agrupados através de um intermediário central;
 - **Broker** - É utilizada para agrupar eventos sem utilizar intermediários (Mediator).

Plug-ins Architecture

- O plugin é usado para adicionar funções a outros programas, provendo uma nova funcionalidade. Um plugin leve que é usado sob grandes programas.
- Por exemplo num software hospedeira provê serviços que o plugin pode usar, incluindo uma forma da extensão registrada entre si mesma no software e um protocolo para a troca de informações entre eles.

Peer-to-Peer Architecture

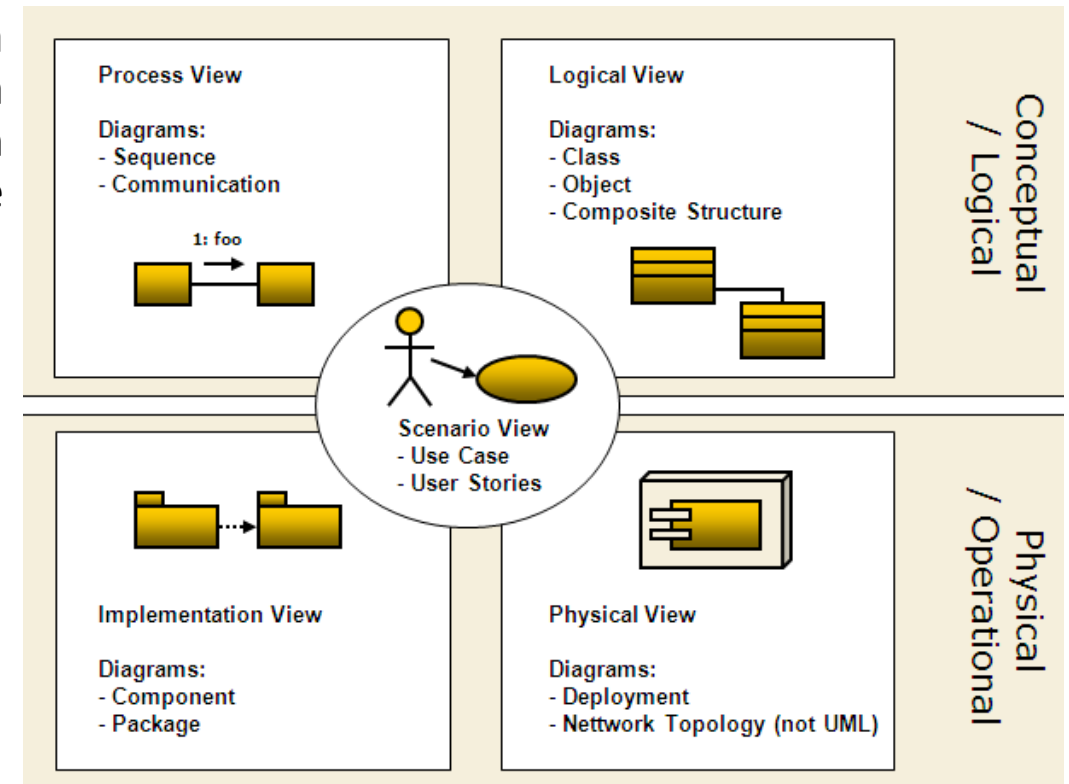
- Cada computador é um nó da rede
- Cada nó pode ser um cliente ou um servidor
- Mais conveniente para o armazenamento de objetos imutáveis(Ex.: Torrents)
- Baseada em distribuição horizontal onde cliente ou servidor podem estar fisicamente divididos, cada um opera a sua própria porção de dados, balanceando assim a carga.
- Características:
 - Descentralização – não necessita de um servidor central, todos os nós são iguais;
 - Heterogeneidade – como não temos a garantia que todos os nós ficaram ligados, recorre-se a replicação, a probabilidade de falhar diminui quando se usa objetos replicados



Vistas de Arquitetura de Software

No que toca às vistas podemos ter vistas diferentes da arquitetura, consoante o *scope* da aplicação que estamos a visualizar, o diagrama ilustra uma boa representação da mesma arquitetura de um sistema com vistas diferentes. Em cada vista podem ser utilizados diferentes diagramas de representação para as descrever:

- Vista de processo
 - Diagramas de sequência e de comunicação
- Vista lógica
 - Diagramas de classes, objectos, estrutura composta
- Vista de implementação
 - Diagramas de componentes e de pacotes
- Vista física
 - Diagramas de desenvolvimento, topologia de rede
- Vista de cenário
 - Diagramas de *Use Case*, *User Stories*





That's all Folks!