



Inteligência Artificial para Sistemas Autónomos Relatório de Projeto

Francisco Engenheiro, n.º 49428, e-mail: a49428@alunos.isel.pt

Professor: Luís Morgado

Abril de 2024

asasas[?]

1 Introdução

asasas[?]

2 Enquadramento Téorico

TODO: falar sobre agente, ambiente.. e principios de inteligencia artificial

Inteligência \Rightarrow (implica) Autonomia - vice-versa não é verdadeiro

3 Projeto - Parte 1

3.1 Arquitetura de software

A arquitetura de software aborda a complexidade inerente ao desenvolvimento de software por meio de uma série de vertentes que estão interligadas.

3.1.1 Métricas

As métricas são medidas de quantificação da arquitectura de um software indicadoras da qualidade dessa arquitectura;

O acoplamento é uma métrica inter-modular que mede o grau de interdependência entre os módulos de um sistema. Pode ser medido através da:

- **Direção:** Unidirecional vs Bidirecional (uni representa menos acoplamento);
- **Visibilidade:** Quando menor for a visibilidade de um módulo, menor é o seu acoplamento;
- **Ordem:** (de menos acoplamento para mais) Herança → Composição → Agregação → Associação → Dependência.

A coesão é uma métrica intra-modular que determina o nível de coerência funcional de um subsistema/módulo, seja pela sua organização (i.e., cada modulo está organizado por conteúdo) ou pela sua funcionalidade (e.g., *single responsibility principle* - cada modulo tem uma única responsabilidade).

3.1.2 Princípios

Os princípios no contexto da arquitectura de software são um conjunto de convenções que orientam a sua definição, garantindo a qualidade de produção da mesma. Alguns exemplos são:

- **Abstração:** Define a forma como os componentes de um sistema são representados, permitindo a ocultação de detalhes de implementação;
- **Modularização:** Ao qual está associado a decomposição (e.g, divisão do sistema em sub-módulos) e o encapsulamento (i.e., ocultação de detalhes de implementação e/ou manutenção de estado privado e interno);
- **Factorização:** Onde a arquitectura é dividida em camadas, cada uma com um conjunto de responsabilidades bem definidas. Pode ser estrutural (e.g, Herança) e Funcional (e.g, Delegação);

3.2 Processo de Desenvolvimento de Software

O processo de desenvolvimento de software consiste na criação da organização de um sistema de forma progressiva, através de diferentes níveis de abstracção:

- **Modelo (Conceptual):** Representação abstrata do sistema, que define o que o sistema deve fazer, sem especificar como;
- **Arquitetura (Modelo Concreto):** Representação concreta do sistema, que define como o sistema deve ser implementado;
- **Implementação:** Código fonte que implementa o sistema definindo como o sistema deve ser executado.

Consiste num processo iterativo, em que as diferentes actividades de desenvolvimento são alternadas ao longo do tempo em função do conhecimento e do nível de detalhe envolvido. Essa alternância poderá ser circular (i.e., implementação → arquitectura → modelo → imple-

Tabela 1: Tipos de Implementação

Tipo	Modelo Associado	Designação
Estrutural	<i>UML</i>	Define a estrutura de um sistema, ou seja, a forma como os componentes se relacionam entre si.
Comportamental	<i>Sequence Diagram</i>	Define o comportamento de um sistema, ou seja, a forma como os componentes interagem e comunicam entre si.

mentação).

Os diagramas de sequência ou atividade representam o fluxo de controlo de um sistema, ou seja, a sequência de atividades que um sistema executa e a sua ordem. Definem-se como modelos de interação com uma organização bidirecional (i.e., horizontal \rightarrow tempo e vertical \rightarrow estrutura) e são compostos por diferentes elementos de modelação (e.g., mensagens, operadores, linha de vida).

A linguagem de modelação unificada (UML) representa um modelo de comportamento com interação como perspetiva principal de modelação. Este tipo de modelação descreve a forma como as partes de um sistema interagem entre si e com o exterior para produzir o comportamento do sistema.

Referências

- [1] FreeCodeCamp contributors. A thorough introduction to distributed systems. <https://www.freecodecamp.org/news/a-thorough-introduction-to-distributed-systems-3b91562c9b3c>, 2024. [Online; accessed 5-March-2024].