

10 - 3 | 2022

Modelo Teórico de Teste para Agentes Cognitivos de Software

Theoretical Model of Software Cognitive Agent Testing

Luís Roberto da Silva Olumene

Versão eletrónica

URL: <https://revistas.rcaap.pt/uiips/> ISSN: 2182-9608

Data de publicação: 31-12-2022 Páginas: 9

Editor

Revista UI_IPSantarém

Referência eletrónica

Olumene, L. (2022). Modelo Teórico de Teste para Agentes Cognitivos de Software. *Revista da UI_IPSantarém. Edição Temática: Ciências Naturais e do Ambiente, Ciências Exatas e da Engenharia e Ciências da Vida e da Saúde. Número Especial: III Simpósio de Economia e Gestão da Lusofonia. 10(3), 44-52.* <https://doi.org/10.25746/ruiips.v10.i3.29121>

MODELO TEÓRICO DE TESTE PARA AGENTES COGNITIVOS DE SOFTWARE

Theoretical Testing Model for Software Cognitive Agents

Luís Roberto da Silva Olumene

Universidade Politécnica A POLITÉCNICA, Moçambique

2015RSilva@gmail.com

RESUMO

A tecnologia de Agentes de Software vem sofrendo significativo desenvolvimento no âmbito da Inteligência Artificial (IA), com progressos relevantes nas áreas de fundamentação teórica e de experimentação prática. Todavia, uma área que tem sido negligenciada é a área de “teste” apesar de inúmeros pesquisadores reportarem, em seus estudos, a necessidade de avanços nesta área. O presente trabalho tem como objectivo contribuir no preenchimento desta lacuna, apresentando uma abordagem de “teste baseado em modelos”. Para tal desenvolveu-se um modelo teórico designado ANCD composto por 4 fases (Ambiente, Necessidade, Comportamento e Desempenho). São contribuições específicas da pesquisa a apresentação de um modelo geral de teste para um software inteligente a partir do qual são despoletadas e propostas diversas teorias (teoria de técnicas de teste, teoria de tabelas de decisão, teorias de geração de casos de teste).

Palavras-chaves: Inteligência Artificial, Agentes de software, Modelo de Testes.

ABSTRACT

The software technology agents are suffering significant development in the scope of the Artificial Intelligence (AI) with a relevant progress on theoretical reasons area and practical experimentation. Nevertheless, an area that has become neglected is a “test” area; despite many researchers report in their studies the need of advances on this area. This paper has with objective contribute in the filling of this gap, presenting an approach “based on models tests”. For such, a Theoretical model designated ANCD was developed composed by 4 phases (environment, necessity, behavior and performance). A specific contribution of this research is a general test model presentation for an intelligent software from which are triggered and several theories were proposed (test techniques theory, decision table theory, test case generation theory).

Keywords: Artificial intelligence, Software agents, Tests Model.

1. INTRODUÇÃO

Actualmente, os computadores são utilizados para resolver problemas complexos, em ambientes de constante mudança que requerem cada vez menos a presença ou acção humana. Naturalmente, há que dotar o software do computador de características – da psicologia ou filosofia - reconhecidas aos humanos, tais como, mas não se limitando: Reactividade; Aprendizagem; Mobilidade;

Autonomia; Cooperação; Competição; Benevolência; Veracidade; Racionalidade; Imprevisibilidade; Intencionalidade; Personalidade; Capacidade inferencial; Receptividade; Agência entre outras.

Desta feita, segundo Costa and Simões (2008), aos sistemas computacionais que possuem algumas destas características, acima, dá-se o nome genérico de agentes. A ideia de agentes foi originada por John McCarthy no ano de 1950 e, o termo foi cunhado por Oliver G. Selfridge alguns anos mais tarde quando ambos estavam no MIT (Bradshaw, 1997). No entanto, embora na década 50, se tenha cunhado o termo agente, o olhar da inteligência artificial, numa perspectiva de agente, foi adoptada por Russell e Norvig (Russell & Norvig, 2021).

Nesta direcção, segundo Russell and Norvig (2021) IA é o estudo de agentes que recebem percepções do ambiente e realizam ações sendo o agente, Russell and Norvig, qualquer coisa que possa ser vista como percebendo seu ambiente por meio de sensores e agindo sobre esse ambiente através de atuadores.

Por outro lado, teste de software é uma actividade associada a qualquer processo, cujo objectivo seja produzir um produto. Segundo a IEEE et al. (2010), teste é uma actividade na qual um sistema ou componente é executado sobre certas condições, o resultado é observado ou gravado, e uma avaliação é feita para alguns aspectos do sistema ou componente.

Existem duas técnicas relevantes de teste: Teste de caixa branca e Teste de caixa preta. Esta pesquisa vai abordar o teste de caixa preta (teste comportamental, funcional ou baseados na responsabilidade) cujo objectivo não é a estrutura interna do programa, em vez disso, concentra-se em encontrar circunstâncias nas quais o programa não se comporta de acordo com as especificações.

Em outra perspectiva, um modelo de teste, segundo Binder (2000), é uma representação com intenção de explicar um comportamento, de algum aspecto, de um artefacto de teste ou actividade de teste. Outrossim, esta pesquisa abarca duas áreas do conhecimento: Engenharia de Software (ES) e Inteligência Artificial (IA) e a partir desta perspectiva avançar para os Testes baseados em Modelos.

1.1 Problema

As conferências RAISE (Realizing Artificial Intelligence Synergies in Software Engineering), hoje na 10ª edição, iniciaram em 2012 como uma plataforma, workshops, de discussão dos problemas da não aplicabilidade das teorias e práticas da ES na IA e vice-versa. Segundo RAISE (2021) a pergunta que motiva e impulsiona este workshop é: os pesquisadores de ES e IA estão ignorando insights importantes de IA e ES?.

De facto, Jennings and Wooldridge (1998) lamentam o facto da tecnologia de agentes da IA estar a ser largamente usada quando, todavia, o teste de software da ES continua a ser uma das áreas mais negligenciadas no processo de desenvolvimento de software. Estes concluem que nós precisamos, urgentemente, de ferramentas de teste de software, que permitam observar o comportamento do agente.

No entanto, o problema reside pelo facto de não existirem, teorias (técnicas, métodos) de teste de agentes devidamente padronizadas e, ou consensuais na comunidade científica da computação, que suportem o desenvolvimento deste tipo de software o que justifica as conferências RAISE. No mesmo diapasão, Dankel and Gonzales (1993), nenhuma técnica singular desenvolvida ganhou aceitação universal mas muitas abordagens existem que, em combinações com outras podem servir para alcançar.

1.2 Hipótese

(1) Testes psicológicos baseados em características podem ajudar no desenho de testes baseados em modelo para Agentes Cognitivos de software pelo facto da característica ser determinante para o funcionamento e objectivo do Agente Cognitivo no ambiente; (2) Testes psicológicos baseados em características não podem ajudar no desenho de testes baseados em modelo para Agentes

Cognitivos de software pelo facto da característica não ser determinante para o funcionamento e objectivo do Agente Cognitivo no ambiente;

1.3 Objectivos da pesquisa

O objectivo geral da pesquisa é Desenvolver um modelo para suportar o desenho de teste de Agentes Cognitivos de Software. Por esta via, como objectivo específico vamos: 1) Construir um Modelo de Teste de Software; 2) Montar a tabela de decisão do Modelo; 3) Elaborar uma fórmula matemática para geração de casos de teste no agente, com base no modelo construído.

2. MODELO DE TESTE

Na figura.1, abaixo, a parte tracejada, será objecto de estudo (desenho de teste baseado em modelos). Os pontos (2) e (3) são a parte inicial e fundamental de todo processo de teste e o ponto (5), que surge como resultado da tabela de decisão é a etapa final.

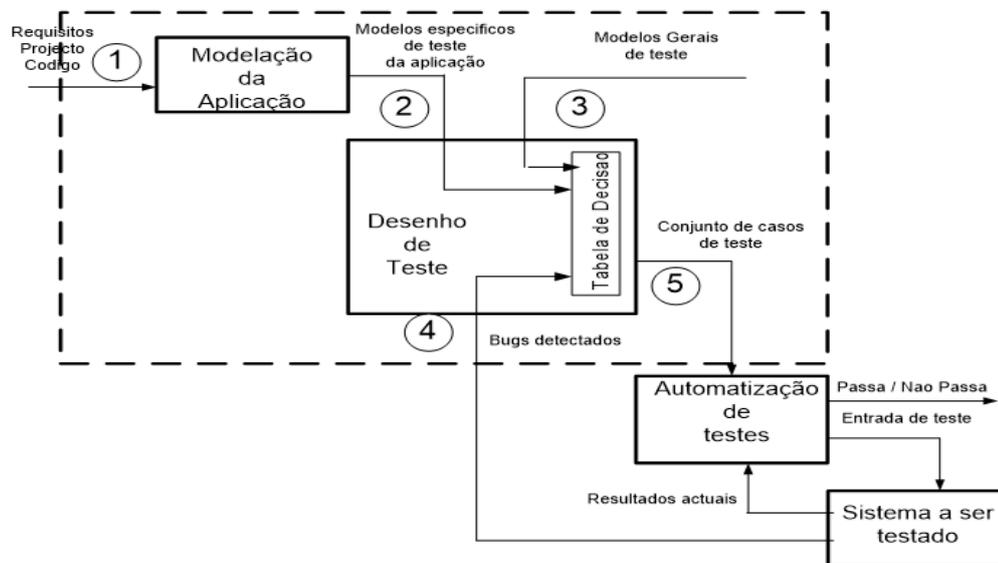


Figura.1 Vista de teste em engenharia de sistemas (Binder, 2000)

Neste contexto, Binder (2000) afirma que a complexidade do software requer o desenvolvimento de modelos para suportar o desenho de teste, sem os quais, esta actividade seria uma tarefa em vão. Um teste baseado em Modelo é uma técnica de teste de caixa preta que utiliza as informações contidas no modelo de requisitos como base para a geração de casos de teste (Pressman, 2010).

2.1. Tabela de decisão

Conforme a figura.1 após o desenho dos modelos segue-se a montagem da tabela de decisão.

Binder (2000), afirma que as tabelas de decisão são ideais, para representação de modelos de teste, pelo facto de poderem suportar desenhos de teste em qualquer dimensão e, também, serem usados para implementação ou desenho de testes baseados na responsabilidade (avaliar se o comportamento observado está conforme a representação). Por outro lado, segundo a IEEE et al. (2010), a tabela de decisão é uma tabela usada para mostrar, um conjunto de condições e as acções resultantes delas.

Nesta direcção, as tabelas de decisão tem duas sessões (Binder, 2000):

- 1) **Secção de condição:** Possui variáveis estáticas, independentes e não sequenciais. Esta secção apresenta-se na vertical e tem as características que se pretende testar ou observar. 2) **Secção de acção:** Possui variáveis dinâmicas, dependentes e sequenciais. Esta secção apresenta-se na horizontal, e tem os itens que suportam ou permitem a execução de testes ou observações. Vide tabela.1.

2.2. Geração de casos de teste

Conforme a figura.1 após a montagem da tabela de decisão segue-se a fase de geração de casos de teste. Neste contexto, segundo o SWEBOOK (2014) o teste baseado em modelo é usado para validar requisitos, verificar sua consistência e gerar casos de teste focados nos aspectos comportamentais do Programas.

Desta feita, casos de teste são um conjunto de entradas de teste, condições de execução e resultados esperados desenvolvidos para um objetivo específico, como para exercer um determinado caminho do programa ou para verificar o cumprimento de um requisito específico (IEEE et al., 2010).

3. MÉTODOS

A presente pesquisa é resultado de uma dissertação de mestrado em Engenharia de Software submetida a prova pela Universidade Eduardo Mondlane de Moçambique, Maputo. Desta feita, a forma de abordagem do problema levantado, nesta pesquisa, conduz ao uso de uma pesquisa qualitativa, na medida em que não se usará métodos ou técnicas estatísticas. Neste trabalho, quanto a natureza da pesquisa, adotar-se-á uma pesquisa aplicada que vai gerar conhecimento para aplicações práticas dirigidas à soluções de problemas específicos.

Outrossim, quanto a finalidade nossa pesquisa é pura. Isto é, visa dar avanços a aspectos de conhecimento na área da IA e engenharia de software (testes de agentes inteligentes) envolvendo verdades e interesses universais para a comunidade científica da computação.

Por outro lado, a construção do Modelo será com base os seguintes: Modelo Agente, tarefa, ambiente e comportamento (Costa & Simões, 2008); Modelo de teste BDI - Beliefs, Desires, Intentions (Wooldridge, 2009); Modelo PEAS - Desempenho, Ambiente, Actuadores e Sensores (Russell & Norvig, 2021); Modelo em cascata de desenvolvimento de software (Sommerville, 2003; Pressman, 2010) e testes psicológicos baseados nas características (Gleitman et al., 1999).

Neste contexto, a partir da análise e correlação dos modelos acima referenciados vai se derivar as seguintes variáveis na forma lógica sequenciada: Ambiente, Necessidades, Comportamento e Desempenho.

4. RESULTADOS

Nesta sessão, vamos apresentar o modelo de teste construído ao qual designamos Modelo ANCD. Este modelo possui quatro etapas ou fases descritas a seguir:

a) Fase do Ambiente (A): O ambiente é a primeira classe de abstracção que garante as condições de existência do agente e medeia, tanto as interacções com outros agentes, como o acesso a recursos (Odell et al., 2007);

b) Fase da Necessidade (N): Tal como nos humanos, a necessidade do agente vai obedecer à teoria das necessidades, por vezes chamada, teoria do conteúdo da motivação. Segundo Davidoff (2001), o motivo (ou motivação) refere-se a um estado interno que pode resultar de uma necessidade. A necessidade activa o motivo, o motivo acciona um comportamento voltado para o retorno do equilíbrio;

c) Fase do Comportamento (C): Segundo Weiten (2006), motivos são necessidades e desejos que impulsionam as pessoas, em certas direcções. A motivação envolve comportamento direccionado a um objectivo. Em sentido geral, e neste contexto, o comportamento designa a mudança, o movimento ou a reacção do agente de software, em relação à sua necessidade;

d) Fase do Desempenho (D): Russell and Norvig (2021), afirmam que a medida de desempenho incorpora o critério para o sucesso do comportamento do agente. Quando o agente é mergulhado num ambiente, ele gera uma sequência de acções, de acordo com o que ele percebe. Esta sequência de acções causa no ambiente uma sequência de estados. Se a sequência é desejável, então o agente tem bom desempenho.

4.1. Elementos fundamentais do modelo de teste construído

Binder (2000) afirma que no geral um modelo tem 4 elementos fundamentais: Objectivo, Representação, Técnica e Ponto de vista/teoria. Sendo assim, para o Modelo ANCD teremos:

a) Objectivo do modelo ANCD: Facultar maior informação possível para apurar se o agente exhibe determinada característica sobre o teste;

b) Representação do modelo ANCD: O modelo no eixo vertical e no sentido ascendente representa o teste, na horizontal e no sentido decrescente representa a observação. Por outro lado, as zonas de intercepção (zonas 4,5,6), são de teste e as de não intercepção (zonas 1,2,3), são zonas de observação.

c) Ponto de vista do Modelo ANCD. O modelo ANCD tem dois pontos de vista:

- Ponto de vista de observação: É uma fase de recolha de dados relevantes à fase de teste, onde se estuda e se investiga o agente. O objectivo é; *Analisar e apurar se o agente exhibe ou não determinada característica sobre teste; A observação deve ser feita apenas para as primeiras três fases do modelo ANCD, isto é, ANC.*
- Ponto de vista de teste: É a fase de descoberta de eventuais erros comportamentais de agente. O objectivo desta fase é; *Analisar (documentar) caminhos de pontos de falha; Analisar as medidas de desempenho (critérios de sucesso) e, Concluir sobre o estado de desempenho.*

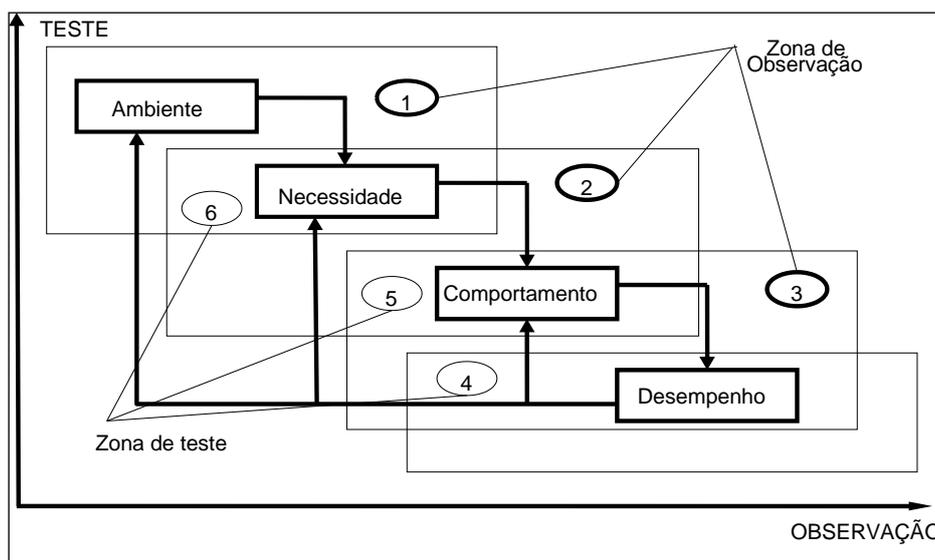


Figura.2 - Modelo de teste ANCD para um agente cognitivo de software. Fonte: Autor (2010)

4.2. Tabela de decisão

A tabela de decisão ANCD, vai basear-se numa matriz de correlacionamento, que tem como pressuposto o princípio de que todas as características dos agentes, vistas na introdução, e outras, podem estar directamente relacionadas com as variáveis ANCD.

4.2.1. Modo de uso da tabela

O número da característica (variável estática) depende de quantas foram embutidas no agente ou sistema. Podemos ter um software com duas características embutidas (ex: Reactividade, mobilidade) neste caso são características binárias. Por outro lado, podemos ter um software com apenas uma característica (ex: Autonomia) neste caso é uma característica unária.

Tabela.1

Tabela de Decisão ANCD

Agente de Software				
Variáveis Estáticas ANCD	Variáveis Dinâmicas ANCD			
	Ambiente	Necessidade	Comportamento	Desempenho
Reactividade				
Mobilidade			X	X
Autonomia		X	X	X
Entre outras				

Fonte: Autor (2010)

4.3. Fórmula de geração de casos de teste do Modelo ANCD

A fórmula será elaborada tendo como base os pontos de vista do Modelo ANCD, ponto 4.1 c). Desta feita, dado que pretendemos testar um agente como um todo, teste de integração, independentemente das características serem uma, duas ou mais, o nº da característica será constante e igual a um (1). Ex: um humano tem várias características mas o teste de humanos é feito de forma integral com um todo. Outrossim, a observação é sempre obrigatória, tanto para características juntas ou individuais, daí que é sempre igual a 3. Nesta direcção, propõe-se a seguinte fórmula de geração de casos de teste com base neste modelo:

$$\text{Casos de teste} = \text{n}^\circ \text{ de características} \times (\text{n}^\circ \text{ de zonas de observação} + \text{n}^\circ \text{ de zonas de teste})$$

Figura.3 – Fórmula de geração de casos de teste.

Fonte: Autor (2010)

5. DISCUSSÃO DE RESULTADOS

A hipótese das teorias de teste psicológicas, poderem ser usadas no paradigma orientado a agentes, permitindo o desenvolvimento de novas teorias de teste, foi objecto de investigação nesta pesquisa. Essa questão, chamou logo a atenção, a partir da exposição das inúmeras características humanas no software inteligente e das perspectivas de pesquisa referidas e anunciadas por pesquisadores conceituados da área de agentes de software.

Enquanto alguns autores, se referem a necessidade de abordar a questão de teste, baseado em regras, procurou-se abordar a questão por um ângulo bem diverso: *Teste baseado em modelos*.

Desta forma, foi possível montar a tabela de decisão com base nas características o que torna a hipótese.1 verdadeira.

Uma das principais ideias que resultam desta pesquisa, é que a transposição para sistemas computacionais das características psicológicas não passa por uma mera extensão ou reformulação dos testes tradicionais, como tem sido proposto pelas principais correntes de investigação na área, mas implica antes, um entendimento profundo de teorias psicológicas. O modelo de teste de agentes proposto, constitui uma das direcções em que essa mudança pode ocorrer.

Por outro lado, tomemos como exemplo, a determinação de caso de teste, para uma característica Autonomia e Mobilidade na tabela.1.

[Autonomia]: Casos de teste = nº da característica x (nº de zonas de observação + nº de zonas de teste) = $1 \times (3 + 3) = 6$ casos de teste;

Interpretação:

(1) Sentido decrescente, sentido de observação, para frente e da esquerda para direita começando do ambiente: caso ambiente, caso necessidade e caso comportamento;

(2) Sentido ascendente, sentido de teste, para trás e da direita para a esquerda começando do desempenho e aos pares: caso desempenho-comportamento, caso comportamento-necessidade e caso necessidade-ambiente.

[Mobilidade]: Casos de teste = nº da característica x (nº de zonas de observação + nº de zonas de teste) = $1 \times (3 + 2) = 5$ casos de teste;

Interpretação:

(1) sentido decrescente, para frente e da esquerda para direita começando do ambiente: caso ambiente, caso necessidade, comportamento. O desempenho não faz parte da observação, como já se referiu, existem apenas três zonas de observação (ambiente, necessidade e comportamento) e três zonas de teste (desempenho, comportamento, necessidade), esta última em forma de pares;

(2) sentido ascendente para trás e da direita para a esquerda começando do desempenho e aos pares: caso da desempenho-comportamento, caso comportamento-necessidade.

6. CONCLUSÃO

Na literatura da Inteligência Artificial existe um “consenso” dos pesquisadores de que se um programa de computador se comportar como os seres humanos (incluindo cometer os mesmos erros), então ele é um bom modelo, de como pensamos. Este aparente “consenso dos pesquisadores” levantou, ao longo da pesquisa, aspectos como a possibilidade do agente cognitivo mentir, ocultar informação ou até mesmo errar. O agente ao se comportar desta forma estaremos perante um erro ou bug de software? e, ademais, quem seria o responsável pelos danos causados pela acção de um software inteligente?.

Desta forma, em última análise, concluímos que testar um software da Inteligência Artificial (Agentes Cognitivos) vai para além de um simples teste de software tradicional por incorporar, também, os aspectos humanos. Notamos, para os testes de agentes cognitivos de software, existir uma relação entre Engenharia de Software, Psicologia e Inteligência Artificial.

Todavia, acreditamos ser, este, um dos grandes obstáculos para avanços de testes de software, por exemplo, nas conferências RAISE. A nosso ver, as conferências RAISE não estão a produzir os resultados esperados volvidos 10 anos de existência. O SWEBOK é bom indicativo nesse direcção pois, mesmo na sua última edição (versão.3) continua a abordar aspectos de teste do software tradicional e raramente encontramos – de forma explicita – testes de software inteligente ou da Inteligência Artificial (agentes de software).

7. REFERÊNCIAS

Binder, R.V. (2000). Testing Object-Oriented Systems. Addison Wesley.

Bradshaw, J.M. (1997). Software agents :An introduction to software agents, AAAI Press/The MIT Press.
<http://agents.umbc.edu/introduction/01-Bradshaw.pdf>

Costa, E., & Simões, A. (2008). Inteligência Artificial: Fundamentos e Aplicações (2nd ed). FCA.

Dankel, D.D., & Gonzalez, A.J. (1993). The engineering of Knowledge-Based Systems: Theory and Practice. Prentice Hall.

- Davidoff, L.L. (2001). *Introdução a psicologia* (3ª ed). Makron Books.
- Gleitman, H., Fridlund, A.J., & Reisberg, D. (1999). *Psicologia* (8ª ed). Fundação Calouste Gulbenkian.
- IEEE, ISO, IEC, (2010). *Systems and software engineering — Vocabulary* (1th ed). Michigan State University. <https://www.cse.msu.edu/~cse435/Handouts/Standards/IEEE24765.pdf>
- Jennings, N.R, Wooldridge M. (1998). *Application of intelligent agents*. <http://agents.umbc.edu/introduction/jennings98.pdf>
- Olumene, L.R.S. (2010). *Modelo de Teste para Agentes cognitivos de software*. [Dissertação de Mestrado não publicada].
- Odell, J. & Weyns, D. & Omicini, A. (2007). *Environment as a First-Class Abstraction in Multiagent Systems: Agents and Multiagent Systems*, Kluwer Academic Publishers. <http://www.jamesodell.com/publications.html>
- Pressman, R.S. (2010). *Software Engineering: A Practitioner’s Approach* (7th ed). McGraw-Hill.
- RAISE, (2021). *Realizing Artificial Intelligence Synergies in Software Engineering*. <https://raise-workshop.gitlab.io/2021/>
- Russell, S. & Norvig, P. (2021). *Artificial Intelligence: A modern approach* (4th ed). Pearson Education.
- Sommerville I. (2003). *Software Engineering* (6ª ed). Addison Wesley.
- SWEBOK. (2014). *Guide to the Software Engineering Body of Knowledge* (3th version). IEEE computer society. <https://cs.fit.edu/~kgallagher/Schtick/Serious/SWEBOKv3.pdf>
- Wooldridge, M. J. (2009). *An introduction to MultiAgent Systems* (2nd ed). Wiley.
- Weiten, W. (2006). *Introdução a psicologia* (4ª ed). Thonson learning.