



# **aiTutor**

## **um etutor de agentes inteligentes**

**José Joaquim Leitão Nunes Mascate**

Docente:  
Doutora Maria da Costa Potes Franco Barroso  
Santa-Clara Barbas

2012

Junho



Palavras-chave

Agentes Inteligentes, Moodle, Aprendizagem, Conhecimento, Ensino a distância, TROPOS, GAIA, LMS, SMA, PHP, Tutor, elearning, FIPA

## **Resumo**

Os Sistemas Multiagente são estruturas complexas que têm um conjunto próprio de objetivos e que interagem com o seu mundo dinâmico, através de um sistema organizacional, bastante complexo, que lhes permite atingir objetivos e resolver tarefas.

Com o incremento de níveis de superiores de complexidade na estrutura destes sistemas, é fundamental usar metodologias, ferramentas e notações orientadas para o desenvolvimento de sistemas orientados a agentes. Desta forma a metodologia TROPOS vem acrescentar o método para o desenvolvimento de um SMA.

Esta metodologia é apoiada na ferramenta i\* e é constituída por quatro fases de desenvolvimento: Requisitos Iniciais (Early requirements), Requisitos Finais (Late requirements), Arquitetura de Projeto (Architectural design) e Projeto Detalhado (Detailed design).

O objetivo deste relatório é o de apoiar um futuro desenvolvimento de um Sistema de Tutoria Inteligente que permita adequar as plataformas existentes, nomeadamente, o *Moodle*, a este tipo de interações.

## **Abstract**

The Multi-agent systems are complex structures that have their own set of goals and interact with their dynamic world through an organizational system, very complex, which allows them to reach goals and solve tasks.

With the increase of higher levels of complexity in their structure, it's fundamental to use methodologies, tools and oriented notations to develop systems oriented by agents. Therefore, the methodology TROPOS adds the method to the development of a SMA.

This methodology is supported by the tool i\* and it's consisted of four phases of development: Early Requirements, Late Requirements, Architectural Design and Detailed Design.

The purpose of this report is to support the future development of an Intelligent Tutoring System that allows adaptation to the existing platforms, like Moodle.

# Índice

Índice de Figuras .....	v
Índice de Tabelas .....	vi
Lista de Abreviaturas.....	vii
Introdução.....	1
Parte I - Enquadramento e Justificação .....	2
Questões e Objetivos. ....	4
Parte II - Agentes inteligentes, exploração e enquadramento de um conceito.....	5
Os agentes inteligentes autónomos .....	6
Características dos Agentes .....	6
Atributos dos Agentes .....	8
Autonomia .....	8
Sistemas Multiagente.....	8
FIPA.....	10
O <i>elearning</i> .....	12
Plataforma de elearning - O Moodle.....	13
Framework i* .....	14
Parte III - Metodologia de Investigação. ....	16
Métodos .....	16
Tropos.....	19
Projeto Detalhado .....	22
Planeamento do trabalho.....	25
Parte IV - Projeto.....	28
O aiTutor.....	28
Metodologia TROPOS .....	28
Requisitos Finais.....	33
Análise e discussão dos dados .....	38

Conclusão.....	42
Bibliografia.....	43
Anexos.....	1
Anexo 1.....	1
GAIA.....	1

# Índice de Figuras

Figura 1- Estrutura FIPA .....	11
Figura 2 - Relacionamentos de Dependência i* .....	20
Figura 3 - Faseamento do projeto .....	25
Figura 4 - Parte1: Introdução, Enquadramento e Metodologia de Investigação .....	26
Figura 5 - Parte 2 .....	26
Figura 6 - Parte 3 .....	27
Figura 7 - Parte 4 .....	27
Figura 8 - Modelo de Dependência Estratégica dos Requisitos Iniciais .....	29
Figura 9 - Modelo I* de Razão Estratégica para o ator estudante .....	30
Figura 10 - Modelo I* de Razão Estratégica para o ator Tutor .....	31
Figura 11 - Modelo I* de Razão Estratégica para o ator Plataforma .....	32
Figura 12 - Modelo de Dependência Estratégica dos Requisitos Finais .....	33
Figura 13-Moodle Manage roles .....	34
Figura 14 - Modelo de Razão Estratégica ator “aiTutor” Req. Finais.....	35
Figura 15 - Diagrama NFR baseado no estilo Integração Vertical .....	36
Figura 16 - Diagrama de Ontologias .....	37
Figura 17 - Core architecture of intelligent tutoring system(Grundspenkis, 2011).....	39
Figura 18 - Architecture of agent-based ITS (Grundspenkis, 2011).....	39
Figura 19 - Diagrama de e ligação de tabelas - Moodle.....	41
Figura 20 - Models of the Gaia methodology and their relations in the Gaia process (Zambonelli, Jennings, & Wooldrige).....	2

## **Índice de Tabelas**

Tabela 1 - Requisitos iniciais – atores e intenções .....	29
Tabela 2 - Tabela de atores e Objetivos em Requisitos finais.....	34

## **Lista de Abreviaturas**

AI – Agente inteligente

STI – Sistema Tutor Inteligente

IA – Inteligência Artificial

WWW – World Wide Web

IAC – Instrução Assistida por Computador

IIAC – Instrução Inteligente Assistida por Computador

FANN - Fast Artificial Neural Network Library

PHP - Hypertext Preprocessor

IEEE - Institute of Electrical and Electronic Engineers

FIPA - Foundation For Intelligent Physical Agents

JADE - Java Agent DEvelopment Framework

# Introdução

O desafio de uma investigação sobre um tema complexo, apesar de limitado no tempo, é sempre motivador. É baseado nesta perspectiva que surge este trabalho que tem a finalidade de compreender, apresentar e propor arquétipos de agentes inteligentes que reproduzam formas de tutoria sobre sistemas de *elearning*, em especial o *Moodle*<sup>1</sup>.

Perante tamanha tarefa, obriga-se inicialmente à definição de conceitos e apresentação da história da inteligência artificial, nomeadamente sobre os agentes inteligentes. O *Moodle* surge também, pelo facto de ser uma plataforma aberta de *software* livre muito usada nas instituições de ensino superior e secundário em Portugal, sobre o qual irá recair este projeto.

Da investigação inicial surgiu também a noção de que não existe muito trabalho desenvolvido nesta área, ou se existir também estará numa fase de desenvolvimento embrionária.

Para abarcar o conhecimento deste tema, no momento atual, é necessário compreender o funcionamento da plataforma *Moodle*, em especial a sua base de dados e também conhecer antigos e recentes desenvolvimentos de algoritmia em agentes inteligentes bem como das metodologias usadas.

É proposto neste trabalho de estágio apresentar arquétipos que permitam num futuro próximo desenvolver programas (*software*), que simulem, em determinados ambientes de aprendizagem, a ação do etutor<sup>2</sup>. Estes arquétipos, pretensiosamente, poderão mudar a forma de fazer tutoria sobre os ambientes de aprendizagem virtual.

Para a criação dos arquétipos vão ser realizados estudos sobre a base de dados Moodle da plataforma eRaizes<sup>3</sup>. Os dados irão ser extraídos e analisados conjuntamente com a estrutura das tabelas. Nesta fase deverão ser lançados os requisitos do sistema bem como os casos de uso e definidos os pressupostos de desenvolvimento aplicacional.

---

<sup>1</sup> Moodle - Modular Object-Oriented Dynamic Learning Environment – Plataforma de uso gratuito para ambientes de aprendizagem à distância.

<sup>2</sup> etutor – Tutor em ambiente de *elearning*

<sup>3</sup> eRaizes redes –Plataforma de *elearning* da Escola Superior de Educação de Santarém

Neste contexto, deverá ser possível determinar quais os tipos de agentes que deverão coexistir e que relações devem manter entre si para que seja estabelecida uma rede funcional e inteligente.

## **Parte I - Enquadramento e Justificação**

Este trabalho enquadra-se num sistema em que cada vez mais é difícil para tutores e estudantes administrarem e articularem os seus trabalhos e tarefas de acordo com as atribuições impostas pelo uso das plataformas de elearning. Os estudantes que carecem muitas vezes de acompanhamento imediato e o tutor, mesmo contando com a ajuda das ferramentas tecnológicas, têm muitas dificuldades, originadas pelo desfazamento temporal, na interação com os seus tutorados.

Sempre que se realiza um trabalho, que pretende aumentar a eficácia, resolver problemas de comunicação e aumentar a produtividade com o mínimo de intervenção humana preparando a mudança dos paradigmas da forma ensinar e aprender, responde-se aos desígnios proclamados pela União Europeia em *“The Future of Learning: Preparing for Change”* (Redecker, Leis, & Leendertse, 2011).

Neste relatório depreende-se que a aprendizagem informal, a colaboração e a personalização estão no centro da aprendizagem do futuro. Estes termos já são usados na educação e formação, mas terão de se tornar no instrumento principal de orientação para organizar a aprendizagem e o ensino no futuro. Este paradigma é, assim, caracterizado pela aprendizagem ao longo da vida, contando, sempre, com omnipresença das Tecnologias de Informação e Comunicação (TIC).

Neste quadro, os avanços na tecnologia, as mudanças dos mercados de trabalho europeus que estão implicitamente ligados à demografia, globalização e imigração, torna-se importante as competências genéricas e transversais em que os cidadãos se tornam aprendizes ao longo da vida para responder com flexibilidade e eficácia às mudanças, sendo capazes de proactivamente desenvolver as suas competências e prosperar em ambientes colaborativos de aprendizagem e de trabalho.

Muitas das mudanças apresentadas já foram previstas há algum tempo, mas só agora, é que se juntaram. Desta forma torna-se urgente e inadiável, que os decisores devem

considerá-las e implementar a mudança no paradigma de aprendizagem para este novo mundo económico e digital.

Para atingir os objetivos da aprendizagem personalizada, colaborativa e informal, são necessárias mudanças nos currículos, modelos pedagógicos, avaliação, liderança, formação de professores, etc.

Assim, este projeto torna-se muito pertinente porque as utilizações de agentes inteligentes impõem-se como a ordem natural do desenvolvimento dos sistemas de elearning, uma vez que eles irão colmatar os desfasamentos entre tutores e tutorados respondendo, solicitando e promovendo interações muito mais rápidas, certamente, muito mais eficazes nos sistemas de tutoria e desempenho dos sistemas de elearning.

Considera-se que desenvolver um Sistema Tutorial Inteligente é um trabalho bastante complexo, com um grau de dificuldade superior porque as condicionantes e variáveis envolvidas são muitas e de naturezas diferentes. Importa observar o estudante no que diz respeito aos seus comportamentos perante a plataforma e importa também observar os comportamentos dos outros, os tutores (professores), para que a escolha das estratégias a aplicar seja a mais adequada, caso a caso.

Um aspeto relevante é perceber o estudante como ser único e com características distintas. Assim surge a necessidade de investigar diferentes formas de moldar o estudante e de proporcionar interações adequadas às suas necessidades.

## Questões e Objetivos.

Considerando a temática e os objetivos propostos procura-se identificar de forma clara os problemas da investigação, tendo em conta os principais assuntos que se pretende analisar, assim o problema de investigação foca-se na criação de um arquétipo de agentes inteligentes de tutoria em sistemas de elearning, baseado na plataforma Moodle e Raizes redes da Escola Superior de Educação de Santarém, designado de “aiTutor”.

Pretende-se estudar os agentes inteligentes e a sua aplicabilidade em sistemas de elearning bem como determinar a funcionalidade e aplicabilidade no desenvolvimento dos agentes inteligentes, criar os modelos de agentes e se existir mais do que um AI, na constituição do arquétipo, instituir as relações entre esses agentes.

O estudo a realizar pretende, pois, contribuir para dar respostas à seguinte questão de investigação:

**É possível criar um sistema tutorial inteligente para plataformas de *elearning* que permita auxiliar estudantes e tutores na sua atividade *online*?**

A finalidade do presente estudo consiste no estabelecimento dos arquétipos necessários para a construção de um *software* de tutoria inteligente, Sendo assim, é tomado como corolário o seguinte objetivo geral:

- Criar um modelo de agentes inteligentes de tutoria em sistemas de elearning (aiTutor).

Tendo em conta o objetivo geral são definidos os seguintes objetivos específicos:

- Estudar os agentes inteligentes e a sua aplicabilidade em sistemas de elearning
- Criar os modelos de agentes

Para cada um dos objetivos, acima referidos, foi definido um conjunto de procedimentos metodológicos que passam a estar descritos na Parte III deste documento.

## Parte II - Agentes inteligentes, exploração e enquadramento de um conceito.

Se durante muitos anos, o homem insinuou através da ficção científica os entes artificiais que imitavam o comportamento humano, hoje nós estamos muito mais perto das máquinas inteligentes.

Já o matemático Alan Turing, nos anos 50, propôs o *teste de Turing* que considerava que uma máquina será inteligente, se porventura, um ser humano não puder distinguir o comportamento da máquina do comportamento de outro ser humano (Turing, 1950).

Numa resenha histórica, encontramos no Século XVII *Blaise Pascal* e *Gottfried Leibniz* que construíram “máquinas de calcular” precursoras de muitos dispositivos mecânicos que existiram para automatizar tarefas e em parte substituir o trabalho do homem. No Século XIX, *George Boole* e *Augustus De Morgan* desenvolveram a lógica booleana, que mais tarde, *Gottlob Frege* aplicou e transformou na lógica matemática moderna. Mas foi apenas com *John Von Neuman*<sup>4</sup> e *Claude Shannon* através da *The Macy Conferences*<sup>5</sup> se começou a pensar em máquinas com características similares à inteligência humana,

Desde, Alan Turing, que muitos cientistas, entusiasmados com a possibilidade de poderem criar máquinas inteligentes, se propuseram a determinar as características necessárias para se poder reproduzir a inteligência humana em máquinas (computadores). Nesse tempo, os computadores ainda estavam numa fase de desenvolvimento precoce, não conseguindo oferecer características de *hardware* que pudessem suportar a computação necessária para o desenvolvimento da inteligência artificial.

Apesar desta apetência pela pesquisa e desenvolvimento de processos conotados com a inteligência artificial, os resultados conseguidos não foram os mais frutíferos. Mas as

---

<sup>4</sup> Von Neumann foi um matemático importante do Sec. XX, que muito contribuiu para o desenvolvimento de diversas áreas da matemática e informática. Foi um dos construtores do 1º computador, conhecido, o ENIAC .

<sup>5</sup> The Macy Conferences - conjunto de 10 conferências, entre 1946 e 1953, em que participaram muitos cientistas famosos e que se considera como a origem da cibernética.

técnicas e os algoritmos desenvolvidos estão a ser aplicados noutras áreas da computação com mais interesse para a sociedade.

Foi na década de 90 que passou a estar em voga o termo *agente inteligente*, que normalmente se aplica a aplicações informáticas autónomas que utilizam algoritmos de inteligência artificial e que têm como finalidade, o de atingir, objetivos definidos pelos utilizadores. A conceção de aplicações baseadas em agentes inteligentes pode revelar-se uma tarefa difícil e complexa pois envolve processos de comunicação entre agentes, separação de tarefas, coordenação, apresentação bem como interações entre o conhecimento e os comportamentos.

Com o desenvolvimento da *internet* a partir dos anos 90 em especial a *World Wide Web*, começou a ser necessário utilizar produtos de *software* que permitissem realizar tarefas para os utilizadores aproveitando os recursos oferecidos pela rede global.

## **Os agentes inteligentes autónomos**

Devido à variedade de papéis que um agente inteligente pode cumprir, é muito difícil enunciar, um significado para agente inteligente. (Wooldridge & Jennings, 1995) Definiram agente da seguinte forma:

*“Um agente é uma peça de hardware ou (mais normalmente) um sistema computacional baseado em software que goza das seguintes propriedades: autonomia, reatividade, pró-atividade e habilidade social”*

Também podemos definir Agentes Inteligentes como entes autónomos, dotados de uma base de conhecimento e capazes de interagir com o meio em que estão, tomando assim, decisões que irão auxiliar ou até mesmo substituir o trabalho de um humano.

Mas para um melhor entendimento sobre o que é um agente inteligente deve-se olhar para as suas características.

## **Características dos Agentes**

Wooldridge e Jennings (Wooldridge & Jennings, 1995) definem um agente, como sendo um sistema informático (hardware e software) que goza das seguintes propriedades:

- **Autonomia.** Os agentes operam sem a intervenção direta de seres humanos ou outros agentes e são possuidores de controlo sobre as suas ações e funcionamento;
- **Reatividade.** Os agentes percecionam o seu ambiente e são autónomos nas respostas às alterações verificadas nos mesmos;
- **Pró-Atividade.** Os agentes são possuidores de características em que conseguem tomar a iniciativa e exhibir os comportamentos necessários para atingir os objetivos.
- **Habilidade Social.** Os agentes são capazes de interagir com outros agentes e também com seres humanos, através de uma ou várias linguagens de comunicação.

Para além das destas propriedades, Wooldridge e Jennings acrescentam ainda a noção forte de agente em que este é dotado de noções e estados mentais, usando conceitos que normalmente são usados e a aplicáveis aos seres humanos, tais como:

- **Conhecimento.** O agente é possuído de características que consistem em manter persistente uma capacidade de coleta de informação dinâmica bem como ser detentor de capacidade de análise e raciocínio sobre a informação recolhida.
- **Crenças.** As crenças são as noções que o agente possui sobre os factos. São geralmente dinâmicas, pois podem alterar o seu conceito de verdade com o tempo.
- **Intenções.** As intenções são objetivos de médio e longo prazo do agente. Resultam de padrões de comportamento que podem levar à execução de um conjunto de ações específicas, de acordo com o padronizado.
- **Obrigações.** As obrigações resultam dos compromissos assumidos. Os agentes possuem características que permitem indicar aos outros a sua disponibilidade para executar determinada tarefa e a partir do momento em que a manifestou, fica responsável por realizar as ações necessárias para essa execução.
- **Mobilidade.** É a capacidade que um agente tem por se poder movimentar de um local para outro. É uma característica que está mais relacionada, com os agentes de software e como tal a movimentação verifica-se no interior de uma rede de computadores;

- **Racionalidade.** Um agente deve agir de forma a poder atingir os seus objetivos específicos. Em cada fase, deve estar munido de conhecimento que de acordo com as suas capacidades, irá executar a ação apropriada para atingir os resultados desejados;

Outros autores, foram adicionando mais características, na noção forte, tais como: Benevolência, veracidade, continuidade temporal, robustez, aprendizagem, personalidade, credibilidade e rastreabilidade.

## **Atributos dos Agentes**

A seleção das características que devem estar presentes num agente, depende da funcionalidade que o mesmo deve ter. Um agente não necessita de todas as características, embora as suas capacidades estejam diretamente associadas a elas.

## **Autonomia**

A autonomia é uma das características mais escarpadas na definição do conceito de agente pelos investigadores da área. Segundo Nwana (Nwana, 1996) a autonomia estabelece o princípio de que os agentes vão agir baseados nas suas próprias regras de conduta, sem existir a necessidade de intervenção humana na sua execução. Os agentes possuem estados e objetivos internos, agindo de forma a atingir estes objetivos em prol dos seus utilizadores. (Wooldridge & Jennings, 1995) Defendem que a autonomia está relacionada com a pró-atividade, ou seja, a capacidade dos agentes agirem por conta própria, sem a necessidade de intervenção de um outro ente (humano ou agente). De acordo com (Yu E. ) os agentes possuem iniciativa própria e podem não ser necessariamente condescendentes com as diretivas ou desejos externos, como por exemplo, de um programador ou *designer*.

## **Sistemas Multiagente**

Os Sistemas Multiagente (SMA) é uma área bastante nova no domínio da informática. O início da investigação começou nos anos 80, só em meados dos anos 90 esta ganhou uma notoriedade digna de destaque (Wooldridge & Jennings, 1995). Ao longo dos últimos anos a investigação no campo dos Sistemas Multiagente tem sofrido um elevado crescimento, o que fez com que aparecessem no mercado, produtos para a

comunidade científica tais como revistas, livros e conferências internacionais sobre este assunto.

Os Sistemas Multiagente (SMA) são sistemas constituídos por vários agentes, que são autônomos mas ao mesmo tempo interagem com os outros agentes presentes no sistema.

Estes agentes exibem as seguintes características:

- Capacidade de agir autonomamente em que a tomada de decisões leva a execução dos seus objetivos.
- Capacidade de interagir com outros utilizando protocolos de interação social inspirados nos seres humanos incluindo as funcionalidades: coordenação, cooperação, competição e negociação.

Fazem parte também destes sistemas a pretensão de poder dar resposta às seguintes situações:

- Solucionar problemas grandes demais para serem resolvidos por um único agente
- Oferecer soluções para problemas geograficamente distribuídos;
- Oferecer clareza conceptual e simplicidade de projeto;
- Oferecer maior rapidez e confiabilidade com a distribuição de diferentes tarefas a diferentes agentes, de forma, a que a execução seja mais rápida;
- Maior robustez, pois utilizando vários agentes, não existe um ponto centralizado de falha no sistema.

Segundo (Sichman, Demazeau, & Boissier, 1992) os principais problemas encontrados na abordagem de Sistemas Multiagentes são:

- Descrição, decomposição e atribuição de tarefas: divisão de tarefas em sub-tarefas e qual vai ser a ordem de execução;
- Interação, linguagem e comunicação: que primitivas básicas de comunicação vão ser partilhadas pelos vários agentes envolvidos no trabalho cooperativo;
- Coordenação, controle e comportamento: como garantir um comportamento global coerente numa coleção de agentes, tendo cada um a sua autonomia e objetivos. Como vai ser projetado o controlo do sistema;

- Conflitos e incertezas: Qual a forma de resolver os conflitos que vão surgindo uma vez que os agentes não têm informação completa sobre o seu ambiente. Como é que os dados incompletos podem ser distribuídos de forma a garantir resultados coerentes;
- Linguagens e ambientes de programação: sob o ponto de vista técnico que linguagens de programação devem ser utilizadas e quais requisitos que um ambiente de programação deve satisfazer para possibilitar o desenvolvimento de Sistemas Multiagentes.

Essas questões têm sido levadas em consideração pela comunidade de investigadores de Sistemas Multiagentes, sendo que o foco dos estudos está no desenvolvimento de princípios e modelos computacionais para construir, descrever, implementar e analisar as formas de coordenação e interação de agentes.

## **FIPA**

FIPA (Foundation For Intelligent Physical Agents). Esta organização, sem fins lucrativos, foi fundada em 1996 e tem sede na Suíça com a finalidade de desenvolver padrões e tecnologias de sistemas multiagentes (FIPA, 2011). Foi oficialmente aceita pelo IEEE (Institute of Electrical and Electronic Engineers) como o seu 11º membro para o estabelecimento de padrões em Junho de 2005.

As especificações FIPA, não só incentivaram o aparecimento de ferramentas genéricas que garantem a interoperabilidade dos Sistemas Multiagentes, tais como o JADE (Java Agente DEvelopment Framework) (FIPA, 2012).

Além destes incentivos, foram também definidos especificações para implementação de plataformas multiagentes, consistindo estas em definir o comportamento externo dos componentes do sistema, deixando em aberto os detalhes quanto à sua estrutura e implementação.

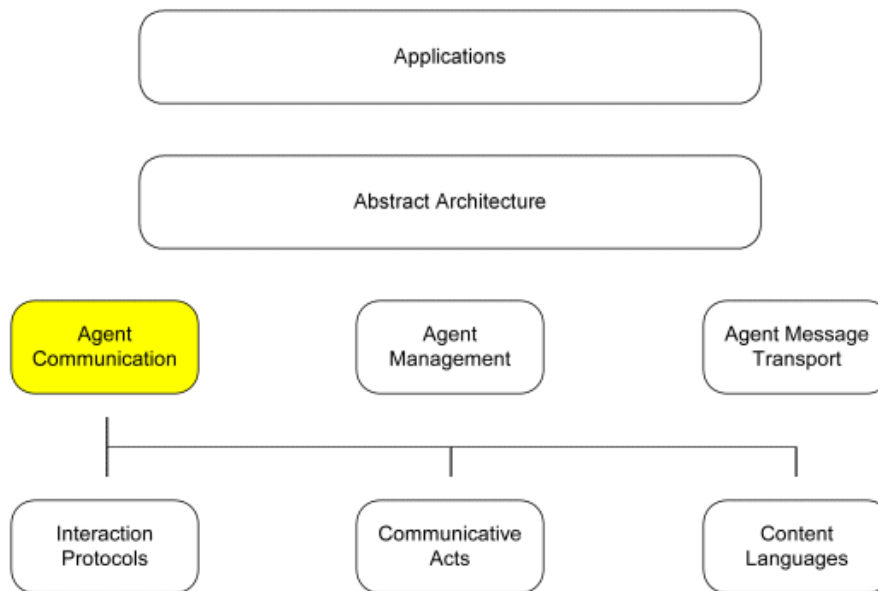
Segundo a organização FIPA (FIPA, 2012), e como se pode observar na Figura 1 as especificações são divididas em cinco categorias:

- Aplicações;
- Arquitetura Abstrata;

- Comunicação entre Agentes;
- Gestão de agentes;
- Transporte de Mensagens entre Agentes.

Os modelos de comunicação dos agentes são baseados na suposição de que dois agentes, que pretendem comunicar, compartilham uma ontologia comum no domínio do discurso. Esta ontologia vai assegurar que os agentes atribuem o mesmo significado aos símbolos usados na mensagem.

Figura 1- Estrutura FIPA



As ontologias podem ser constituintes da implementação real de software ou mesmo do próprio agente, ou ainda, podem ser especificações formais representadas por um agente, designado *Ontology Agent* (OA).

As preocupações estão centradas na produção contínua das especificações para os Sistemas Multi-Agentes, como e também nas suas melhorias e seu ciclo de vida. Uma das primeiras especificações foi a conceção de uma linguagem para o estabelecimento de padrões para a comunicação de agentes (FIPA-ACL (*Agent Communication Language*)).

Dito de uma forma simples, esta linguagem é baseada em primitivas associadas às teorias dos atos de fala em que o conteúdo da mensagem vem encapsulado como sendo também um dos seus atributos. Para esta representação foi desenvolvida a FIPA-SL

(*Semantic Language*), em que os agentes podem exprimir as suas propriedades e as relações entre os objetos pertencentes ao seu domínio, que é definido por uma ontologia da linguagem externa.

A organização FIPA define ainda um conjunto de protocolos de interação que servem para estabelecer padrões de comunicação entre os agentes. Estas especificações podem ser encontradas em (FIPA, 2012)

## ***O elearning***

O *eLearning* está cada vez mais disseminado nos diversos níveis de ensino, quer no apoio ao ensino presencial, quer na distribuição de cursos a distância. As plataformas de *eLearning* atuais, designadas genericamente de ambientes virtuais de aprendizagem, disponibilizam diversas funcionalidades que permitem a gestão dos cursos, a comunicação e a distribuição de conteúdos. A grande maioria das plataformas é baseada no paradigma da sala de aula, em que os conhecimentos são transmitidos da mesma forma para todos os estudantes. Este paradigma usa os conteúdos como o único meio de transferência do conhecimento. Mas hoje, existem novas ideias de que o paradigma tem de mudar refletindo as exigências das sociedades baseadas no conhecimento, o que implica um ensino mais personalizado e baseado na aquisição de competências.

Muitas empresas e instituições de ensino superior e secundário usam a *World Wide Web* (WWW) para disponibilizar o acesso a cursos à distância dos mais variados tipos e para disponibilizar conteúdos, também dos mais variados tipos. Com a oferta e disponibilidade de plataformas de eLearning gratuitas (freeware), tais como o Moodle, têm vindo a massificar o uso destes sistemas.

É uma realidade que nesta data, quase todas as universidades e escolas secundárias dispõem de uma plataforma deste género. Mas mesmo usando estes sistemas, existe ainda um rácio de estudantes por professor bastante elevado, não existindo a possibilidade de ter um tutor individualizado.

No sentido de dar resposta a este problema da personalização da aprendizagem, é necessário adaptar o ambiente de aprendizagem às necessidades e características específicas do estudante em que os conteúdos não devem ser vistos como o único meio de transmissão do conhecimento. Este para ser assimilado de uma forma sólida deve ser

adquirido através da experiência, com base na cooperação entre os estudantes e o professor.

Para uma melhor integração do estudante no ambiente de aprendizagem, tendo como ponto de partida o percurso do estudante. A proposta de adoção de teorias de inteligência artificial, nomeadamente na educação, baseadas na experiência dos estudantes, para que os conteúdos e os contextos de aprendizagem possam ser reutilizados e adaptados a novas situações.

Numa breve análise, dá-se conta de que nem todas as experiências de aprendizagem são síncronas é possível que estas possam contribuir para o melhoramento de outras experiências de aprendizagem. Isto só é possível porque são registadas as dificuldades registadas durante o percurso de aprendizagem. Nas últimas décadas, a inteligência artificial tem vindo a ser utilizada no ensino de formas diversas. As primeiras experiências de utilização da inteligência artificial no ensino remontam a 1984, entretanto outras abordagens se sucederam tendo sempre em vista a utilização da IA na educação.

## **Plataforma de elearning - O Moodle**

O Moodle de uma forma simplista é uma aplicação informática que permite estabelecer relações de aprendizagem entre estudantes e tutores. Possuindo as seguintes características:

**Tutoria ativa** - dinamiza a aprendizagem de cada formando e do grupo, promovendo a participação de todos e motivando-os para o percurso formativo;

**Interação** - entre tutor e formando e entre os próprios formandos: ocorre através do chat, do e-mail, do fórum e de audioconferência. O formando não está sozinho num espaço virtual: aprende em colaboração com os restantes colegas do grupo e com o tutor;

**Percurso de aprendizagem individual ou em grupo** - o formando trabalha autonomamente ou realiza tarefas com os outros colegas;

**Durações das ações de formação** - as ações de formação a distância têm uma duração variada que, geralmente, oscila entre uma e seis semanas no caso dos cursos de curta

duração ou vários meses no caso de cursos aproximados do modelo de uma pós-graduação;

**Conteúdos** - disponíveis em vários formatos e disponibilizados, cada vez mais, através da Internet, propondo várias situações de interação;

**Calendarização das tarefas e atividades** - constitui um fator de motivação adicional, ao criar balizas para o processo de aprendizagem, apesar do eLearning permitir uma gestão do tempo mais flexível;

**Momentos de avaliação diversos** - permitem ao formando verificar se está a cumprir os objetivos fixados e fornecem "feedback" ao tutor sobre a progressão de cada participante e a adequação da orientação pedagógica adotada.

Neste projeto a plataforma *Moodle* surge apenas como repositório de dados, resultantes das interações entre os estudantes e os tutores. Num desenvolvimento posterior a este trabalho, pode surgir, como peça fundamental para a aplicação e utilização do sistema multiagente, *aiTutor*.

## **Framework i\***

A ferramenta i\* é uma linguagem que permite modelar graficamente, num sistema organizacional as relações entre os atores e o sistema que os provê (Yu, Mylopoulos, Maiden, & Giorgini, 2011).

Nesta *framework*, os atores são definidos como entidades em que podem ser atribuídas dependências funcionais, que são classificadas como: papéis, agentes e posições:

- Um papel representa a caracterização em modo abstrato do comportamento de um ou vários atores em determinados contextos sociais, exprimindo as funções que podem ser exercidas por um agente.
- Um agente representa uma entidade real tal como uma pessoa ou um *hardware* ou um *software*.
- Uma posição representa as relações entre um agente e um papel e é definida como um conjunto de papéis tipicamente ocupados por esse agente.

(Yu E. ) Afirma ainda que um agente executa (plays) um determinado papel e ocupa (occupies) uma determinada posição, a qual cobre (covers) um papel.

Segundo Yu a ferramenta é composta por dois modelos: o Modelo de Dependência Estratégica (*Strategic Dependency - SD*) e o Modelo de Razão Estratégica (*Strategic Rationale - SR*).

Ainda segundo (Yu E. ) o Modelo de Dependência Estratégica representa-se a um alto nível do sistema processual porque captura apenas os aspetos importantes esquecendo os detalhes com menos importância. O Modelo de Razão Estratégica por sua vez relata os entendimentos que estão por trás das configurações dos processos nomeadamente no que diz respeito aos seus elementos e das relações entre eles.

## **Parte III - Metodologia de Investigação.**

A metodologia é o processo em que o investigador equaciona e estrutura as atividades a desenvolver no seu projeto. Este conjunto de métodos tem a designação de metodologias. Existem várias metodologias, mecanismos de recolha de dados e técnicas de análise reconhecidas que procuram responder à prossecução dos objetivos do projeto.

### **Métodos**

Como este processo está inserido na área científica das tecnologias da informação e comunicação e nesta área estão identificados os métodos de investigação qualitativos, os métodos quantitativos e os métodos mistos (combinado dos dois anteriores).

Os métodos quantitativos que se baseiam na experimentação e nos modelos matemáticos são muito usados nas áreas dos sistemas de informação. Os métodos qualitativos apareceram para dar resposta à necessidade de estudos que envolvem diversos intervenientes, nomeadamente seres humanos e máquinas e sistemas de software, em que a quantificação é difícil ou impossível de alcançar. Myers (Myers, 1997) menciona que os métodos qualitativos surgiram nas ciências sociais para potenciar o estudo dos seres humanos e a sua interligação com o ambiente que as rodeia.

Apesar da controvérsia existente sobre qual dos métodos é mais eficaz, a investigação deve reger-se por uma abordagem definida e orientada pela metodologia, epistemologia e ontologia. (Orlikowski, Robe, D., & Research, 1991)

A investigação qualitativa em oposição à quantitativa visa compreender o ser humano nos contextos sociais e culturais em que estão inseridos. A investigação qualitativa tem por base epistemológica a investigação positivista, interpretativa e crítica (Positivist, Interpretative and Critical Science).

A investigação positivista é baseada no princípio de que a realidade pode ser descrita por um conjunto de propriedades mensuráveis, que são independentes do observador e dos instrumentos de recolha e análise. (Orlikowski, Robe, D., & Research, 1991)

A investigação interpretativa posiciona-se para que o acesso à realidade tem de ser realizado através da construção social, tomando como exemplos os conceitos repartidos e a linguagem por exemplo, a língua, a consciência e os conceitos partilhados.

A investigação crítica baseia-se nos conflitos, oposições e contradições de forma a poder encontrar a justificação para os problemas mas assume que a verdade social é historicamente continuada, pois é produzida e reproduzida por seres humanos.

Este tipo de investigação têm por base a investigação crítica, interpretativa mas sempre positivista. O positivismo é baseado no princípio de que se pode medir a realidade através de propriedades mensuráveis, independentemente do observador e dos mecanismos de recolha e análise.

Depois de descritos os principais tipos de metodologias, importa olhar para as principais metodologias que se têm utilizado ao longo das últimas décadas as principais metodologias de investigação usadas, são: (Orlikowski, Robe, D., & Research, 1991)

### ***Experiências de Campo (Experiments)***

O método experimental é muito usado em situações reais quer seja no tecido empresarial ou em sectores onde exista trabalho direto com o objeto de estudo. Tem a vantagem de se apresentar colado à realidade mas torna-se muito difícil determinar todas as variáveis, não garantindo sempre que a repetição da experiência produza os mesmos resultados.

### ***Estudo de Mercado (Survey)***

Este é um método qualitativo em a informação provém dos dados recolhidos através de inquéritos, questionários e entrevistas permitindo através da análise (quantitativa) dos dados explicar os assuntos que levaram à produção desses inquéritos.

### ***Estudo de Caso (Case Study)***

É um método apoiado na investigação de acontecimentos no contexto de uma organização. É muito usado nos sistemas de informação, porque objetivo destes são o estudo de sistemas de gestão de informação nas organizações, onde a importância dos fenómenos técnicos tem-se virado mais, para uma questão, de problemas organizacionais.

Aqui os investigadores partilham um objetivo comum, o de resolver e explicar as questões que afetam as organizações, em especial, o seu sistema de informações. Tem como desvantagem implícita a dificuldade de generalizar os resultados.

### ***Simulação (Simulation)***

Este método tenta simular o comportamento do sistema em estudo, possibilitando a criação de vários cenários para estudo de um determinado fenómeno.

### ***Investigação-Ação (Action Research)***

Neste método, o cientista/investigador é um elemento muito ativo na criação e concretização. Tem também como ponto de partida a análise dos resultados obtidos por alterações e mudanças induzidas no objeto em estudo.

### ***Etnografia (Ethnograph)***

Provém das ciências sociais e humanas, nomeadamente a antropologia social e compreende o estudo de situações e objetos através dos contactos e vivências com o objeto de estudo e a realidade onde está inserido.

A escolha de uma metodologia de investigação tem de estar relacionada com o objeto que se vai estudar. A investigação que obtém resultados através de meios não estatísticos ou quantificadores, deve combinar metodologias qualitativas como, por exemplo, *Case Study e Action Research*, recorre a técnicas de recolha adequadas a estes métodos, criando investigação com abordagem qualitativa. Por outro lado quem recorrer a técnicas como Experiments ou Survey, irá obter dados estatísticos ou quantificáveis que seguem metodologias quantitativas.

De acordo com o problema e objeto de estudo, as questões de investigação e os objetivos gerais e específicos deste trabalho, pode considerar-se que o estudo a desenvolver tem por base uma estratégia de investigação assente em métodos e técnicas predominantemente qualitativos, assumindo-se, naturalmente, o *case Study* como o aconselhado a este desenvolvimento

Apesar destas metodologias serem bastantes abrangentes, não satisfazem na totalidade o desenvolvimento de um produto deste tipo. Após intensas pesquisas de trabalhos similares, deu-se a conhecer várias metodologias de investigação, associadas à Engenharia de Software, nomeadamente e particularmente as associadas a sistemas baseados em agentes inteligentes.

As principais metodologias orientadas a agentes são GAIA (Anexo 1) e TROPOS. Estas metodologias permitem uma maior abstração na análise e desenvolvimento, uma vez que estão incorporadas de meios que permitem um nível de pensamento orientado para

as características e comportamentos dos agentes. Como resultado da evolução das metodologias de desenvolvimento, vamos usar TROPOS neste projeto.

## **Tropos**

Tropos é uma Metodologia Orientada a Agentes (MAS) em que se destacam duas ideias chave, sendo que a primeira é a noção de agente e todas as relações mentais que se estabelecem, como por exemplo, os objetivos e os planos. Estes conceitos vão ser usados em todas as fases de desenvolvimento do software, desde a análise até à sua execução. A segunda ideia chave corresponde à análise de requisitos que deve corresponder a uma melhor e mais profunda compreensão do ambiente onde os agentes irão operar e do tipo de interligações com outros agentes e as interações com os “agentes humanos”

Esta é uma metodologia que abrange as seguintes fases:

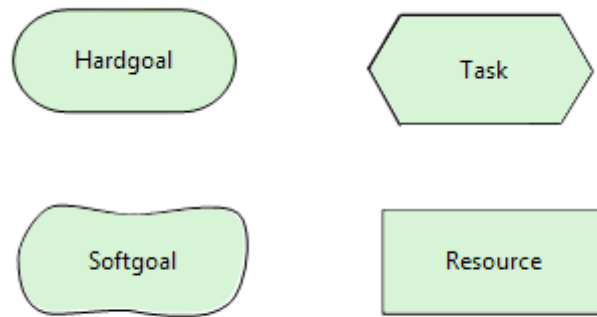
- Requisitos Iniciais (Early requirements)
- Requisitos Finais (Late requirements)
- Arquitetura de Projeto (Architectural design)
- Projeto Detalhado (Detailed design)

### ***Requisitos iniciais***

Os Requisitos iniciais prendem-se com a compreensão de um problema, estuda-se a configuração organizacional e tem como output o modelo organizacional que inclui os Atores, os seus objetivos e as suas interdependências. Os autores (Castro, Kolp, & Mylopoulos, 2002) definem que a fase de requisitos iniciais se preocupa com a compreensão do problema através do estudo da configuração organizacional.

De acordo com a *framework* i\* cada relacionamento é intencional e depende de um acordo entre dois atores. Os relacionamentos de dependência usados em i\* descrevem a essência do acordo e podem ser de quatro tipos: Task, Hardgoal, Softgoal, Resource.

Figura 2 - Relacionamentos de Dependência i\*



### ***Requisitos Finais***

Os Requisitos Finais definem o ambiente operacional bem como as funções relevantes e a qualidade dos mesmos. Este sistema é representado com as relações de dependência do ator com o sistema e os outros atores do organismo. Estas dependências definem os requisitos funcionais e os não funcionais. Também, segundo (Mouratidis, Giorgini, & H., 2004) estes modelos conceptuais são revistos e devem ser estendidos para incluir um novo ator, que representa o sistema.

### ***Arquitetura de Projeto***

A fase de Arquitetura de Projeto define todo o *Design* do sistema bem como os subsistemas e os seus fluxos de controlo. (Castro, Kolp, & Mylopoulos, 2002) definiram um conjunto de estilos que auxiliam na definição da arquitetura do sistema.

Estes estilos são:

#### **Estrutura em 5 (Structure in 5))**

Este estilo está associado aos sistemas que normalmente se encontram em diversas organizações onde confluem alguns componentes tipificados tais como os estratégicos e os logísticos. O núcleo do sistema regista as tarefas básicas, operações de *output*, *input* e procedimentais associados à execução do sistema. Estas tarefas são a base do sistema ao passo que o topo é composto pelos atores executivos estratégicos.

### **Estrutura Plana (Flat Structure)**

Neste estilo não existe uma estrutura fixa e este assume que nenhum ator tem controlo sobre outro ator. Uma arquitetura deste tipo apresenta as vantagens relacionadas com a autonomia, distribuição e evolução contínua de uma arquitetura. A desvantagem deste estilo é que requer uma maior quantidade de argumentação e comunicação por parte de cada ator participante.

### **Pirâmide (Pyramid)**

Este estilo é representado com uma estrutura similar aquela que lhe dá o nome, em que os atores na base ou níveis mais baixos dependem dos atores que estão no topo. É uma estrutura de autoridade e que atua dentro dos limites da organização em que a supervisão é o mecanismo crucial. Este estilo adapta-se muito bem a sistemas distribuídos simples.

### **União Estratégica (Joint Venture)**

Junta acordos entre dois ou mais parceiros com a finalidade de obter benefícios a uma escala maior. Existem atores responsáveis por gerir tarefas, operações e a partilha de conhecimentos e também recursos que permitem atingir objetivos e propósitos comuns. Os parceiros têm capacidade de se poderem gerir e controlar a si, mesmo e de interagir com os seus semelhantes. Nestas interações são realizadas trocas de serviços, dados e conhecimentos mas se existirem atividades globais comuns, estas são coordenadas pelo ator que realiza a gestão global.

### **Tomada de Controlo (Takeover)**

Neste estilo existe uma delegação de autoridade e gestão de dois ou mais parceiros para um ator (takeover). É semelhante ao *joint venture* com a diferença de que não permitidas preservação das autonomias e identidades dos subordinados, porque o *takeover* as absorve

### **Comprimento de Braço (Arm's Length)**

Aqui existem acordos entre atores independentes e competitivos, mas que também são parceiros. Estes parceiros conseguem manter sua autonomia e independência, mas tem que colocar os seus recursos e conhecimentos

disponíveis para a realização de objetivos comuns. As autoridades são delegadas ou perdidas de um parceiro para o outro.

### **Oferta (Bidding)**

Neste estilo estão envolvidos mecanismos que oferecem competitividade e os atores agem como se estivessem a participar num leilão. O ator leiloeiro comanda as operações: anuncia o leilão, recebe propostas de outros atores e assegura a comunicação e *feedback* com o emissor do leilão.

### **Contratação Hierárquica (Hierarchical Contracting)**

Combina características do *Arm's Length* com a *Pyramid* em que são combinados os mecanismos de coordenação com os aspetos de autoridade da *Pyramid*

### **Integração Vertical (Vertical Integration)**

Consiste em comprometer atores no atingir os objetivos ou realizar tarefas relacionadas, mesmo que estas estejam em estágios diferentes de um processo de produção.

### **Apropriação (Co-optation)**

Neste estilo estão envolvidos os processos de incorporação de agentes externos na estrutura ou no comportamento de quem tem de tomar a responsabilidade de decisão.

Estes estilos são estruturas genéricas que instanciam os projetos de arquitetura de uma aplicação. Atualmente a abordagem, está diretamente relacionada e centrada nos requisitos sendo uma metodologia que adota um modelo baseado no “i Yu Eric \*”.

## **Projeto Detalhado**

Esta fase tem como objetivo especificar os agentes a um nível bastante detalhado. (Mouratidis, Giorgini, & H., 2004) Os aspetos dos agentes estão relacionados com a comunicação e comportamentos (objetivos dos agentes, crenças e capacidades) através

de extensões da UML para suportar a especificação de sistemas multi-agentes, tais como a AUML (Agent Unified Modeling Language).

O projeto detalhado divide-se em três partes:

- Conceção do diagrama NFR;
- Seleção do estilo de arquitetura;
- Refinamento dos modelos SD e SR.

A seleção do estilo de arquitetura é realizada através do *diagrama* NFR (*Non-Functional Requirements*) e da correspondência ente os estilos de arquitetura selecionados. Estes estilos têm sido avaliados e comparados. São usados atributos de qualidade reconhecidos por arquiteturas que envolvem componentes coordenados autônomos, tais como integridade, segurança, previsibilidade, adaptabilidade, coordenação, cooperação, disponibilidade, modularidade e agregação.

Nesta fase devem ser desenhados os diagramas de classe, sequência, comunicação e de planos:

- O diagrama de classes define a estrutura das classes utilizadas pelo sistema, estabelecendo os atributos e métodos que cada classe possui, além de estabelecer como as classes se relacionam e trocam informações entre si.
- O diagrama de sequência gera a ordem temporal dos eventos e mostra como é que as mensagens são trocadas entre os objetos envolvidos em um determinado processo. Este diagrama identifica o evento gerador do processo modelado, bem como o ator responsável por este evento, e determina como o processo deve se desenrolar e ser concluído por meio da chamada de métodos disparados por mensagens enviadas entre os objetos.
- O Diagrama de Comunicação está amplamente associado ao Diagrama de Sequência fazendo com que um complemente o outro. As informações mostradas no Diagrama de Comunicação com frequência são, quase sempre, as mesmas apresentadas no de Sequência, porém com uma abordagem distinta, uma vez que este diagrama não se preocupa com o tempo do processo centrando-se mais na forma em como os objetos estão ligados e quais

mensagens trocam entre si durante o processo. Nestes diagrama também se pode visualizar os cenários identificados pelos casos de uso e podem ser traçados em conjunto com o diagrama de classes.

- Os diagramas de plano são usados para especificar o processamento de atores individuais. Estes diagramas não têm apenas as descrições do comportamento do sistema, contem também as prescrições da execução dos comportamentos dos agentes envolvidos no processo em questão. (Castro, Kolp, & Mylopoulos, 2002), definiram um conjunto de estereótipos, etiquetas e restrições para acomodar os conceitos de Tropos dentro da UML, tais como <<i\* actor>> e <<i\* dependency>>, que podem ser usados na *design* de diagramas de classes e outros.

De acordo com (Mouratidis, Giorgini, & H., 2004), essa fase envolve o uso de plataformas de desenvolvimento específicas e depende das características da linguagem de programação adotada. Assim, esta etapa está diretamente relacionada com as escolhas da implementação

## Planeamento do trabalho

Para alcançar os resultados pretendidos foi elaborado o seguinte plano de trabalhos.

Figura 3 - Faseamento do projeto

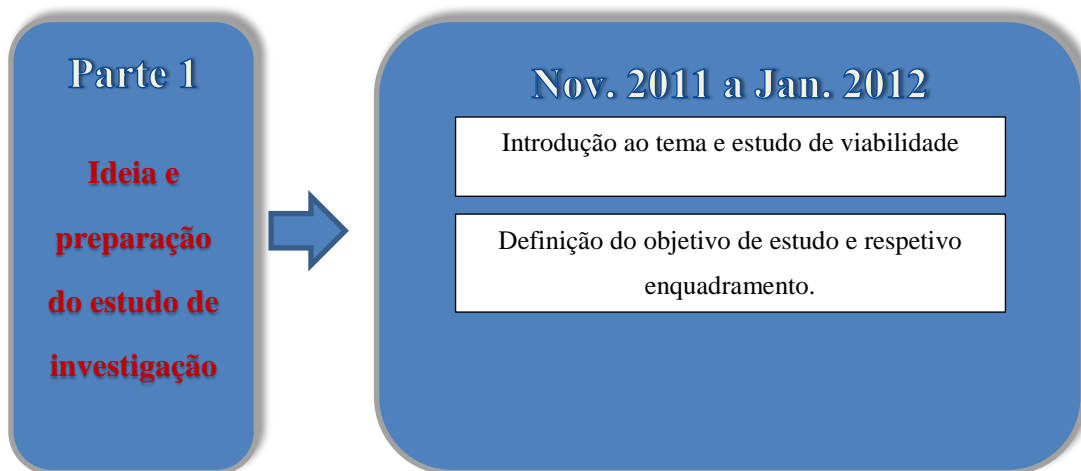


O cronograma apresentado na Figura 3, apresenta um período bastante curto para o desenvolvimento do projeto, mas será o suficiente para apresentar um modelo de um AI.

O relatório está estruturado em quatro Partes principais. Cada Parte terá o seu próprio tempo e modelo de desenvolvimento que se apresenta relatado desta forma:

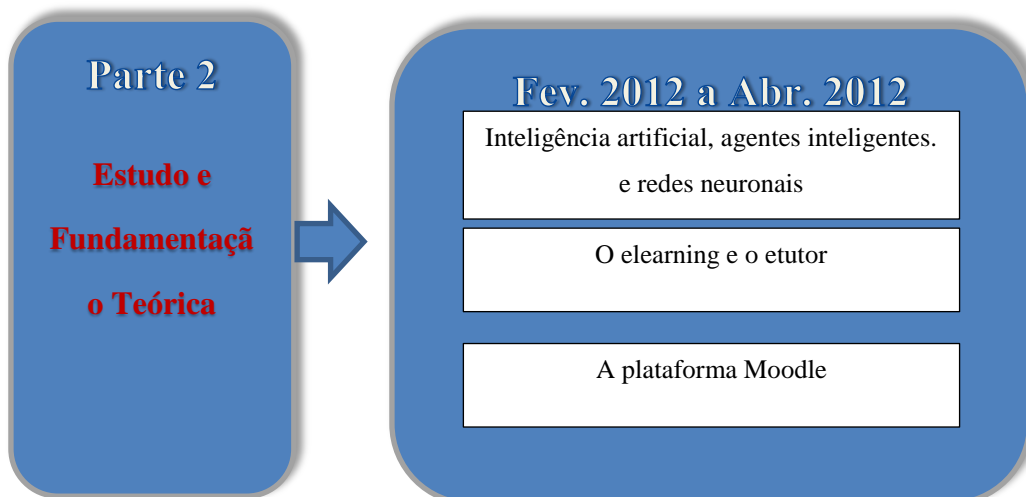
Na 1ª Parte apresenta-se a introdução genérica, o enquadramento do trabalho, os objetivos gerais e específicos, a metodologia e a estrutura do relatório.

Figura 4 - Parte1: Introdução, Enquadramento e Metodologia de Investigação



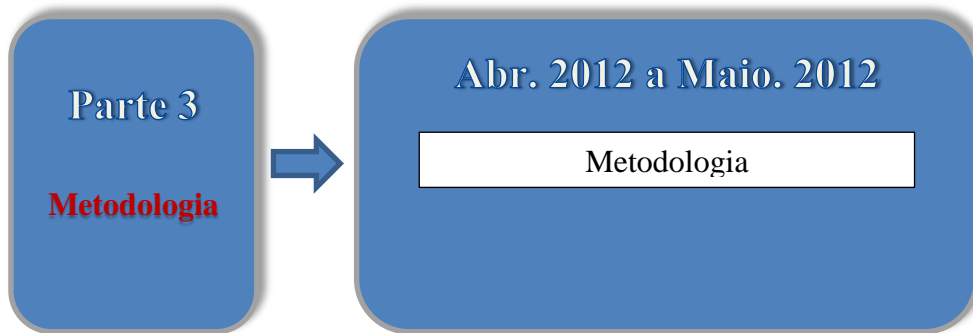
No Parte nº 2 apresenta-se a fundamentação teórica, focalizada nos agentes inteligentes, tipos de agentes, na plataforma de elearning Moodle.

Figura 5 - Parte 2



Na 3ª Parte é apresentada a metodologia que irá ser usada no desenvolvimento com destaque para a TROPOS.

Figura 6 - Parte 3



Na Parte 4 apresenta-se a arquitetura de projeto a análise do trabalho realizado, bem como as principais conclusões. Também são perspetivadas as melhorias bem como os trabalhos de continuidade.

Figura 7 - Parte 4



Este relatório apresenta um conjunto de documentos que fazem parte deste projeto de investigação e que devem ser consultados de acordo com a sua numeração e identificação proposta.

## Parte IV - Projeto

Esta parte apresenta um estudo de caso desenvolvido para identificar os agentes que necessitam de existir para a implementação de um sistema de tutoria “semi-inteligente”. É o propósito desta parte, criar uma proposta detalhada de um projeto de desenvolvimento de uma sociedade de agentes tendo como base a metodologia de desenvolvimento TROPOS.

### O aiTutor

Como já foi referido, a abordagem ao aiTutor foi realizada através da metodologia TROPOS, através das suas quatro primeiras fases desta “ferramenta” (Requisitos Iniciais, Requisitos finais, Arquitetura de Projeto e Projeto Detalhado) através dos *standards* FIPA (Foundation For Intelligent Physical Agents)

### Metodologia TROPOS

#### *Requisitos Iniciais*

A modulação dos requisitos iniciais é realizada através dos modelos de dependência estratégica (SD) e razão estratégica (SR).

#### **R**    **Requisitos**

---

#### **Atores do Sistema**

Face às observações, constata-se que existe sempre uma grande dificuldade em estabelecer relações de igualdade entre os diferentes atores de um sistema. As plataformas (nome pelo qual são conhecidas as aplicações informáticas que tratam de estabelecer comunicações entre vários atores) vêm de, algum modo, fazer este papel, porque além de permitir a comunicação e interação entre esses atores, também regulamentam e disciplinam essas comunicações.

Começa-se por definir os atores do sistema que são: Professores (tutores), Estudantes, Autores e Plataforma de elearning (Moodle).

#### **Atores**

**Ator Autor.** O ator autor aqui representado é o autor que representa a autoria dos materiais que vão ser selecionados e usados pelos outros atores do sistema.

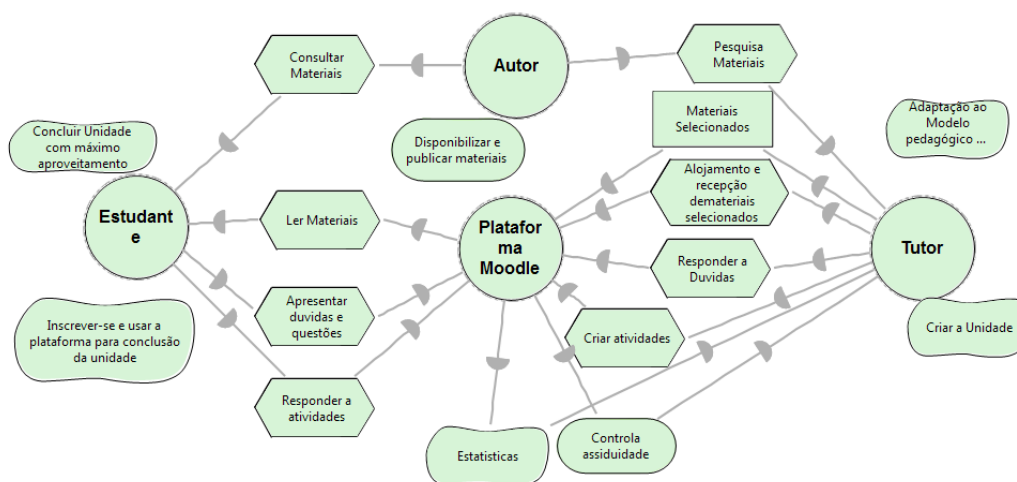
**Ator Estudante.** O ator estudante aqui representado é o estudante da instituição, que está matriculado num ou mais anos letivos em um curso.

**Ator Tutor:** O ator tutor é o docente, pertencente no ano letivo aos quadros da instituição que leciona unidades curriculares em um ou mais cursos

**Ator Plataforma:** O ator plataforma é a aplicação informática que possibilita a utilização do elearning. Neste projeto a escolhida é o Moodle.

Depois de identificados os atores principais que participam nas atividades da plataforma Moodle, é apresentado na Figura 8 o diagrama de Dependência Estratégica dos Requisitos iniciais.

Figura 8 - Modelo de Dependência Estratégica dos Requisitos Iniciais



Após a visualização da, conseguimos identificar claramente os “*stakeholders*”<sup>6</sup>, os seus relacionamentos e as suas intenções (goals).

Para cada ator irá existir um conjunto de tarefas e objetivos, que podem ser característicos do desse ator ou transversal a vários.

Tabela 1 - Requisitos iniciais – atores e intenções

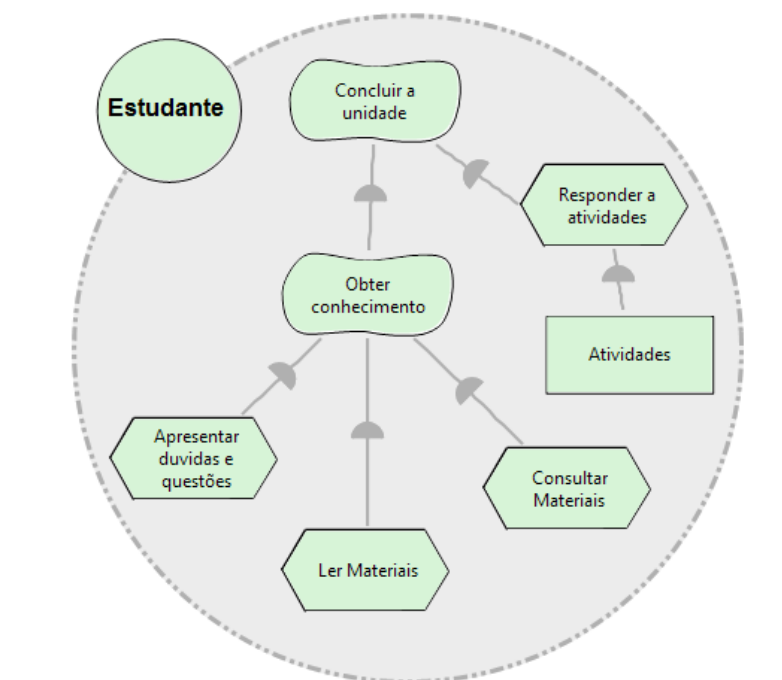
Ator	Goals
------	-------

<sup>6</sup> Stakeholders são os elementos essenciais e fundamentais para a realização de um projeto

<b>Estudante</b>	Concluir Unidade Obter aproveitamento Obter conhecimento
<b>Plataforma</b>	Publicar Materiais Estabelecer comunicação entre Estudante e Tutor
<b>Tutor</b>	Promover a aprendizagem Manter a unidade com informação e atividade
<b>Autor</b>	Produzir Materiais

Como se pode constatar na Tabela 1, o ator Estudante têm como intenção principal o de concluir a unidade com aproveitamento, enquanto os outros atores têm como intenção, o de criar as condições necessárias a uma obtenção de conhecimento por parte do estudante e respetivo reconhecimento publico com a atribuição de uma classificação.

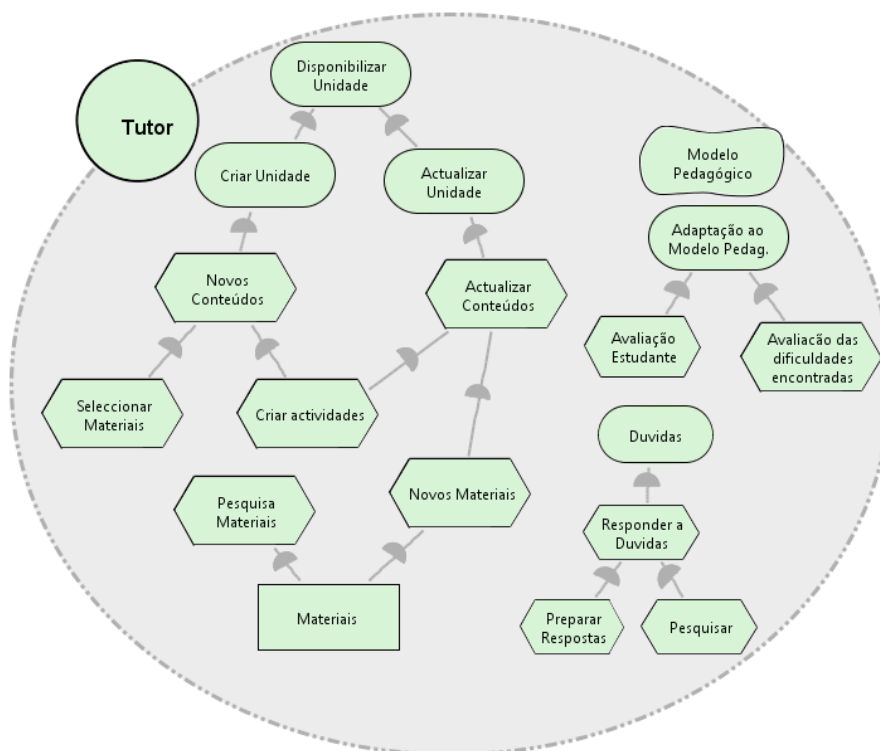
Figura 9 - Modelo I\* de Razão Estratégica para o ator estudante



A Figura 9 apresenta o ator estudante de uma forma expandida. Aqui regista-se a atividade principal deste ator. O principal objetivo é o conseguir aceder à informação das unidades curriculares para uma conclusão das mesmas com sucesso. Os meios utilizados são a leitura da informação, consulta dos materiais, publicados, trocas de

informações com a apresentação de dúvidas e questões a tutores, participação em atividades e troca de informação com os outros atores.

Figura 10 - Modelo I\* de Razão Estratégica para o ator Tutor



A Figura 11 Figura 9 representa esquematicamente o ator Tutor, onde se constata, que a sua função está decomposta nos objetivos, disponibilizar a unidade, adaptação ao modelo pedagógico, atualizar unidades, esclarecimentos de dúvidas.

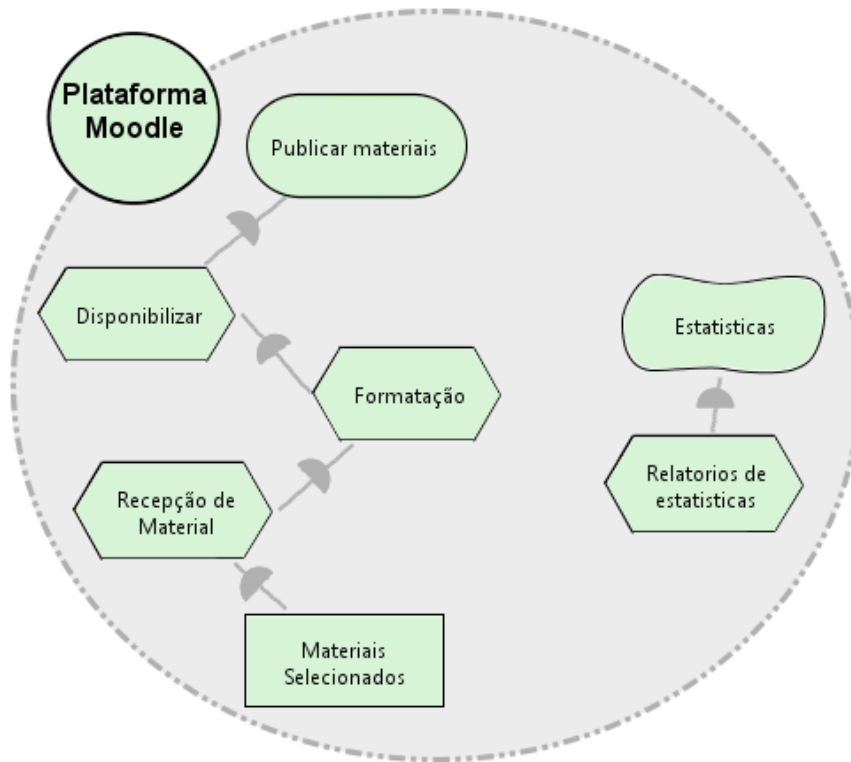
O objetivo “dúvidas” representa a intenção do Tutor em responder às dúvidas submetidas pelos estudantes através dos meios “pesquisar” e “preparar respostas”.

Este ator, tem como função principal coordenar todos os estudantes inscritos na sua unidade, e para isso, têm objetivos que têm de ser atingidos numa fase anterior à da execução, tais como “disponibilizar unidade”. Os materiais também já devem estar definidos para consulta do ator “estudante”, apesar de poderem ser introduzidos posteriormente através do *goal* “atualizar unidade”.

Já no que diz respeito ao ator “plataforma”( Figura 11) o seu objetivo principal é o de definir e estabelecer a comunicações síncronas e assíncronas entre o ator “estudante” e o ator “tutor” de forma a que o primeiro possa atingir os objetivos propostos pelo segundo. Serve também como plataforma onde os materiais se encontram publicados

bem como é possuidora de ferramentas de comunicação síncronas(ex: chat) e assíncronas (ex: fóruns e mensagens).

Figura 11 - Modelo I\* de Razão Estratégica para o ator Plataforma

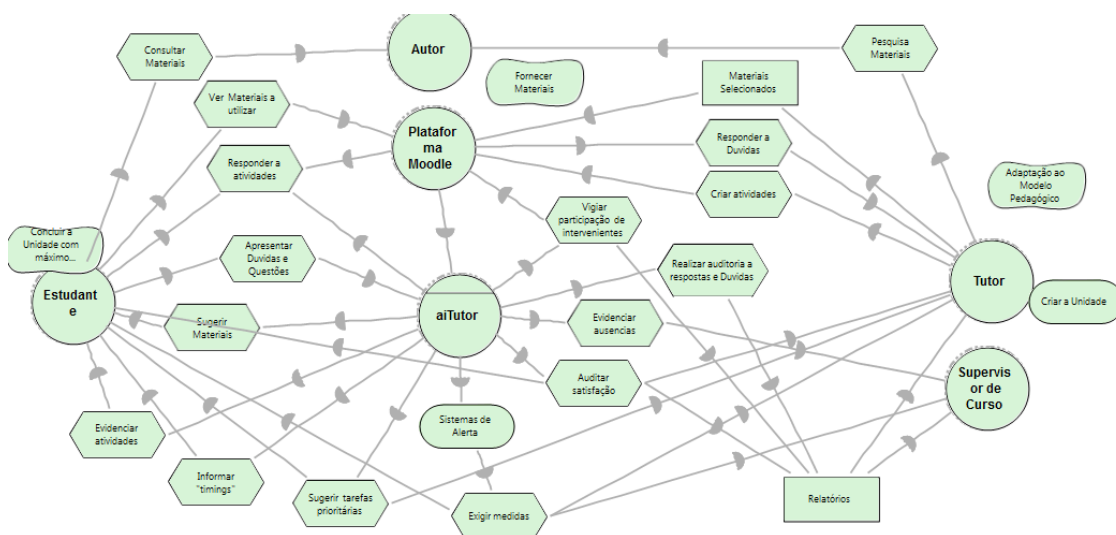


## Requisitos Finais

Tal como é registado na metodologia esta fase serve para melhorar e alterar os modelos produzidos na fase de Requisitos Iniciais. Aqui devem ser realizados as alterações à fase anterior para que possam ser incluídos, os novos atores, que irão representar os novos sistemas bem como os seus dependentes.

Foram adicionados dois novos atores nesta fase, o aiTutor e o Supervisor de Curso, como se pode constatar na observação da Figura 12.

Figura 12 - Modelo de Dependência Estratégica dos Requisitos Finais



Estes novos atores, vêm adicionar ao sistema vigente, um conjunto de procedimentos que irão potenciar o incremento de ações entre os atores. Destaca-se a formalização do ator “Supervisor de Curso” que existe na prática mas que não têm especial relevo, até como se observa na Figura 13. A justificação, para esta formalização, vem da necessidade do sistema controlar com rigor a participação e assiduidade de todos os elementos na plataforma. Quando acontecem desvios os alertas têm de chegar a alguém com responsabilidades superiores em todo o sistema, de forma a poder tomar medidas imediatas, para garantir o sucesso do mesmo.

<a href="#">Manage roles</a> <a href="#">Allow role assignments</a> <a href="#">Allow role overrides</a> <a href="#">Allow role switches</a>			
Role ?	Description	Short name	Edit
Manager	Managers can access course and modify them, they usually do not participate in courses.	manager	+2 ✕
Course creator	Course creators can create new courses.	coursecreator	↑ +2 ✕
Teacher	Teachers can do anything within a course, including changing the activities and grading students.	editingteacher	↑ +2 ✕
Non-editing teacher	Non-editing teachers can teach in courses and grade students, but may not alter activities.	teacher	↑ +2 ✕

Figura 13-Moodle Manage roles

Assim podemos observar na Tabela 2 os novos atores e as suas funcionalidades

Tabela 2 - Tabela de atores e Objetivos em Requisitos finais

Ator	Goals
<b>Estudante</b>	Concluir unidade Obter aproveitamento Obter conhecimento
<b>Plataforma</b>	Publicar materiais Estabelecer comunicação entre Estudante e Tutor
<b>Tutor</b>	Promover a aprendizagem Manter a unidade com informação e atividade
<b>Autor</b>	Produzir materiais
<b>aiTutor</b>	Vigiar comportamentos Auditar a utilização da plataforma Produzir alertas Sugerir tarefas Priorizar tarefas Produzir perfil Automatismos na interação
<b>Supervisor de curso</b>	Toma medidas Analisa relatórios Afere o sistema Responsabilidade

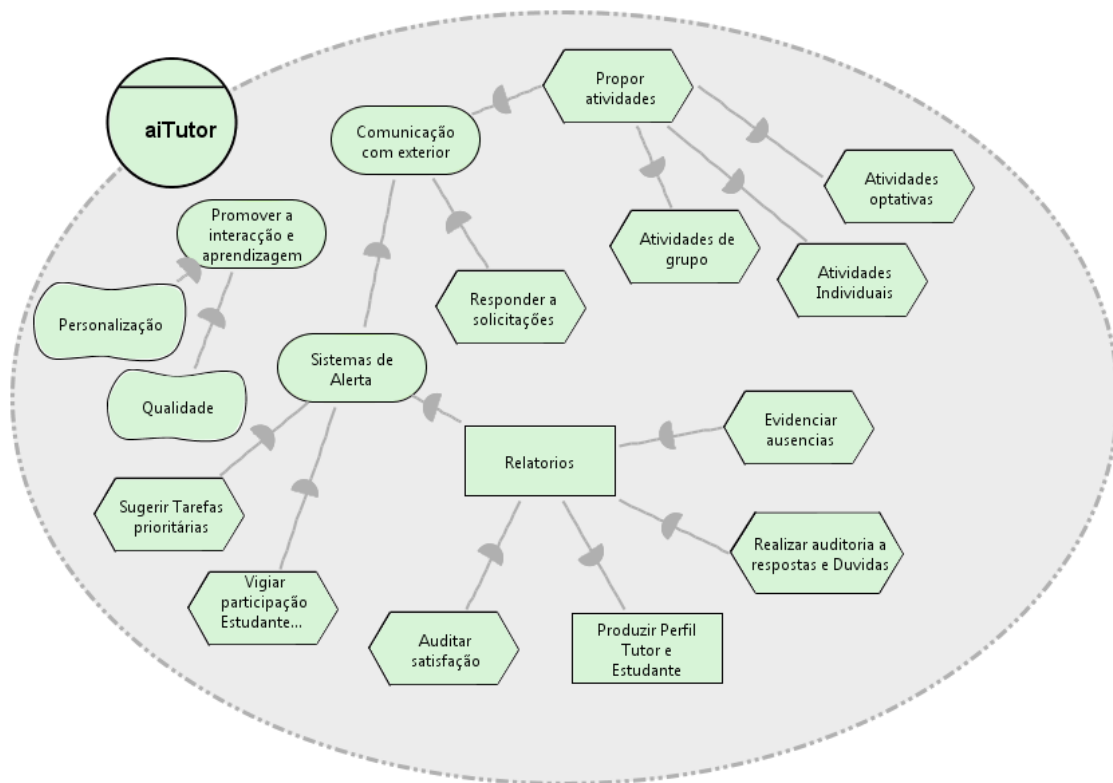


Figura 14 - Modelo de Razão Estratégica ator “aiTutor” Req. Finais

O ator “aiTutor” está representado graficamente na Figura 14, em que são visíveis os objetivos e tarefas atribuídas a este novo elemento. Podemos visualizar que os objetivos a atingir são a aprendizagem através de sistemas de qualidade e de uma personalização e que devem ser manifestadas com as tarefas de monitorização e execução que lhe são atribuídas, tais como: Geração de perfil, auditoria, vigilância, *reporting*, controlo e comunicação.

## Arquitetura de Projeto

Esta fase permite escolher qual o estilo mais adequado ao desenvolvimento do sistema. São definidos como elementos fundamentais os atributos descritos na fase de Requisitos Finais. Para executar esta análise, a metodologia TROPOS indica a utilização da ferramenta NFR.

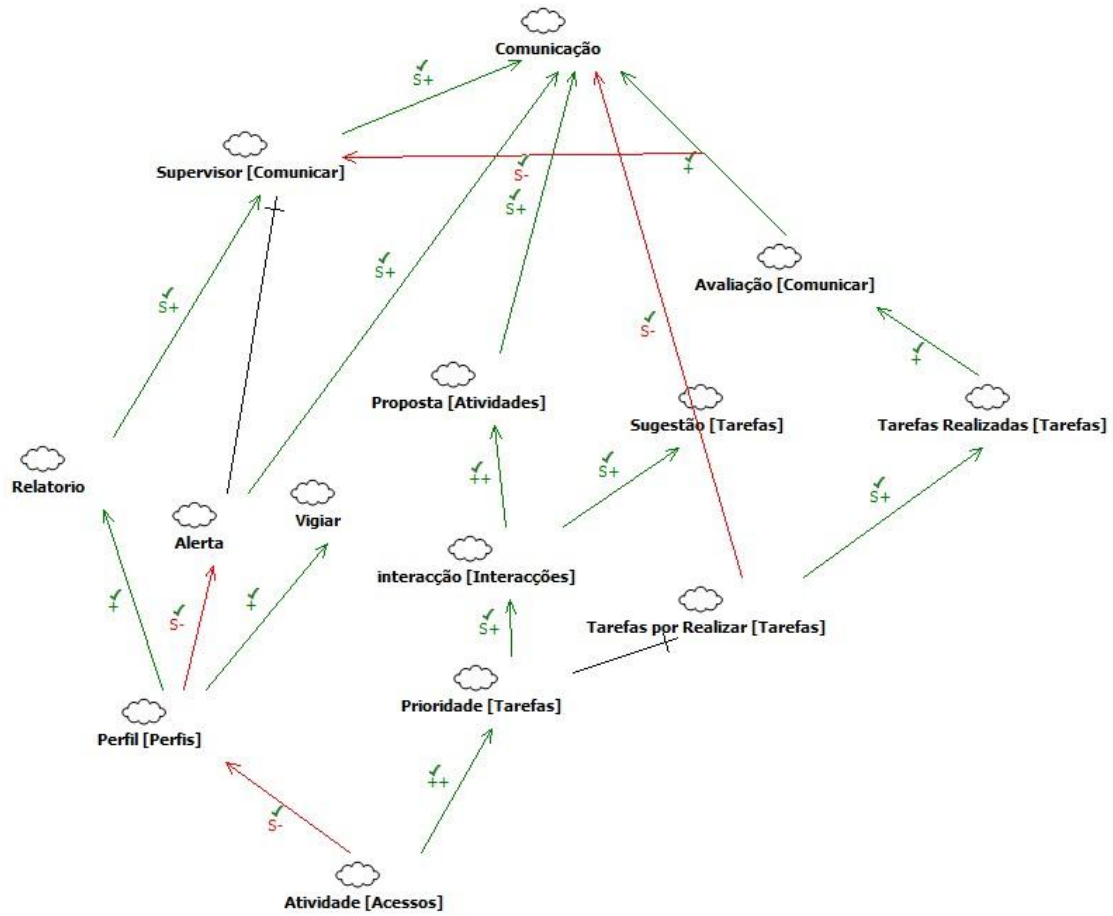
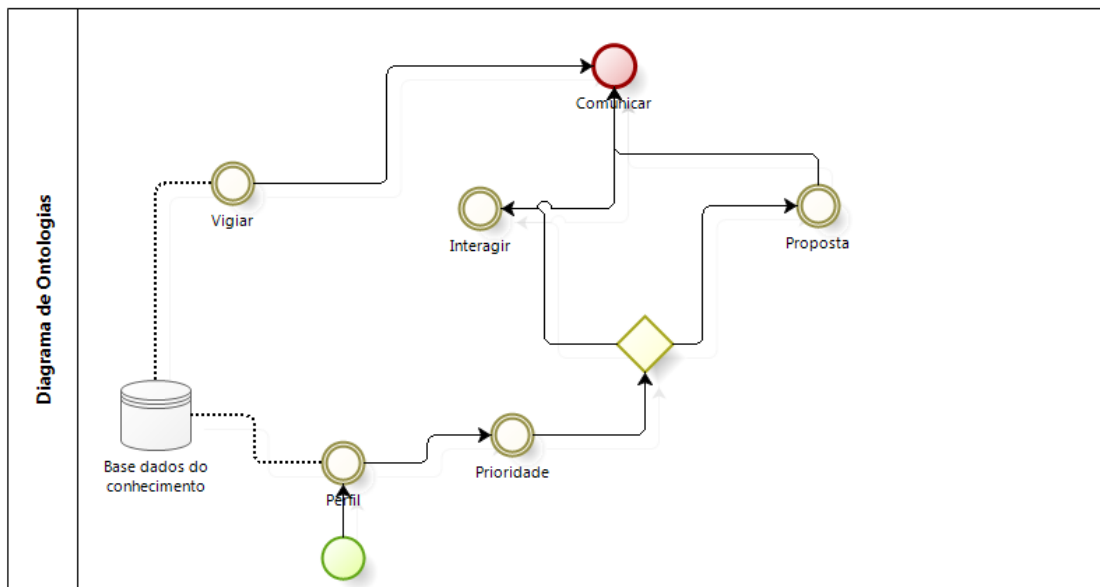


Figura 15 - Diagrama NFR baseado no estilo Integração Vertical

Verifica-se neste processo, analisando a Figura 15 que existe um comprometimento dos atores no atingir dos objetivos ou da realização de tarefas relacionadas, estando estas em estágios diferentes. Assim respondendo ao estilo “Integração Vertical” porque verifica-se o comprometimento dos atores na execução de tarefas e o de poderem atingir os objetivos pretendidos.

Figura 16 - Diagrama de Ontologias



No diagrama existente na Figura 16, constata-se que a ontologia “perfil” retém a informação e atualiza a mesma, numa base de dados do conhecimento. Esta ontologia aplica-se a estudantes e tutores e pretende constituir o elemento fundamental para a obtenção de um sistema personalizado ao ator.

A ontologia “prioridade” tem a responsabilidade de priorizar as atividades e demais tarefas, para que os atores possam responder às solicitações dos outros atores do sistema. Desenvolve um sistema de comunicação para duas ontologias de atividade, que são a ontologia “proposta” e a “interagir”. A diferença entre estas ontologias está nas tarefas e acontecimentos que no caso de a “interagir” são ações criadas pelos agentes internos do aiTutor, e na “proposta” ações produzidas pelos outros atores do sistema.

No sistema podemos visualizar que a ontologia “vigiar” corresponde a um sistema autónomo que é residente e está sempre ativo. A cada alteração ao seu estado, tem a responsabilidade de ativar ou não o sistema de comunicações através da ontologia “comunicar”. Usa a base de dados do conhecimento para registar e atualizar a informação referente a todos os elementos do sistema. Poderá ser necessário, englobar esta ontologia num único sistema de “hardware/software” externo, devido à sua complexidade e natureza das operações

## **Análise e discussão dos dados**

Os cursos de educação à distância, já existem há muitos anos e em formatos bastantes diferentes. Com o desenvolvimento da *World Wide Web (Web)*, houve uma profusão de cursos de educação à distância no formato *online*. A vantagem deste tipo de cursos é o de permitir que o estudante tenha liberdade para escolher o local e o horário de estudo.

A importância deste tipo de cursos é tanta, que levou a que a União Europeia através do relatório “*The Future of Learning: Preparing for Change*” (Redecker, Leis, & Leendertse, 2011) a manifestar-se perante os Estados Membros que a aprendizagem informal, a colaboração e a personalização estão no centro da aprendizagem do futuro. Este paradigma é, assim, caracterizado por uma aprendizagem constante ao longo da vida, contando, sempre, com omnipresença das Tecnologias de Informação e Comunicação (TIC).

Um aspeto relevante é perceber o estudante como ser único e com características distintas. Assim surge a necessidade de investigar diferentes formas de moldar o estudante e o de proporcionar interações adequadas às suas necessidades.

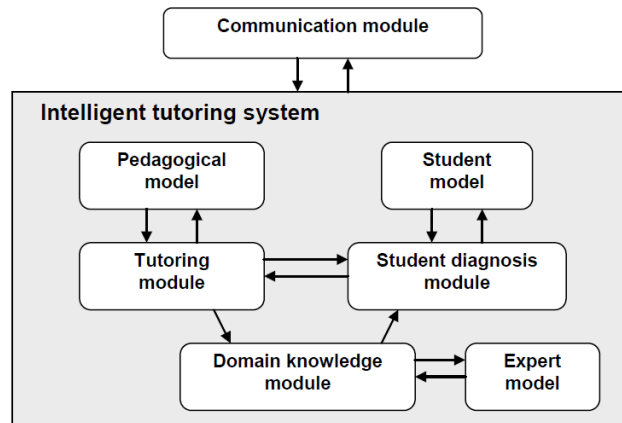
As plataformas atuais, nomeadamente, o *moodle* não têm, na sua génese ferramentas que executem uma gestão de processos inteligente. Autonomizando as plataformas no que concerne a muitas tarefas existentes no trabalho de tutoria, podemos responder melhor às situações de aprendizagem, pois as interações serão mais rápidas, menos onerosas e com melhores resultados na aprendizagem informal e personalizada.

Para que isto aconteça, é necessário desenvolver um Sistema de Tutoria Inteligente, que permita adequar as plataformas existentes, nomeadamente, o *Moodle*, a este tipo de interações.

Na Conferência Internacional sobre o eLearning e a Sociedade do Conhecimento (International Conference on E-Learning and the knowledge Society) é apresentado um trabalho desenvolvido no Departamento de Teoria de Sistemas e Design da Universidade Técnica de Riga em que é descrita a implementação de um sistema de tutoria baseada em agentes inteligentes. (Grundspenkis, 2011)

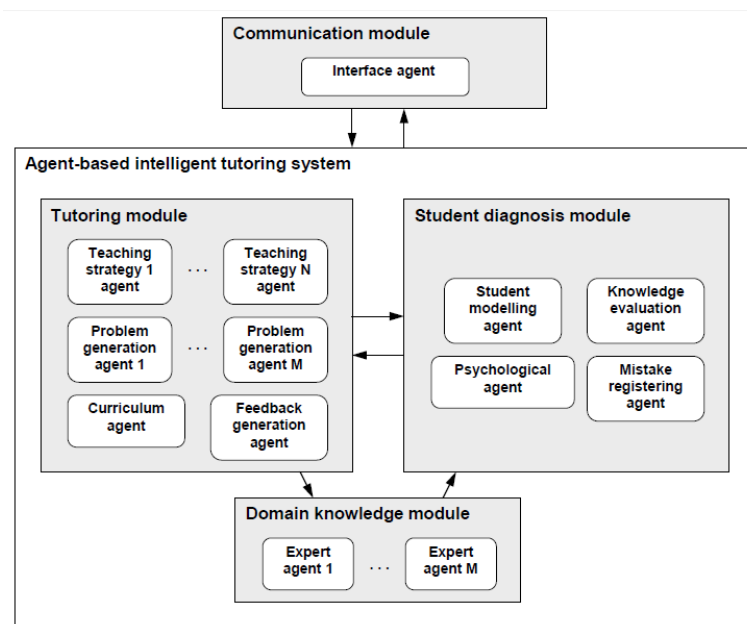
O aproveitamento deste estudo é uma mais-valia para o desenvolvimento do “aiTutor”, assim sendo é apresentado um modelo adaptado à realidade do deste estudo.

Figura 17 - Core architecture of intelligent tutoring system(Grundspenkis, 2011)



O que leva a este detalhe

Figura 18 - Architecture of agent-based ITS (Grundspenkis, 2011)



Com base nestes pressupostos o sistema deve seguir o modelo com a abordagem FIPA porque este fornece a infraestrutura de conhecimento necessária para a adaptação do ensino e aprendizagem às características e necessidades de cada estudante. É possuidor de atividades adequadas à aprendizagem bem como a geração de problemas, estando sempre apto a receber por parte do estudante as soluções e elaborar o *feedback* necessário (explicação e ajuda). O módulo de conhecimento do domínio (expert) possui algoritmos para geração de problemas e respetivas soluções. O módulo de diagnóstico

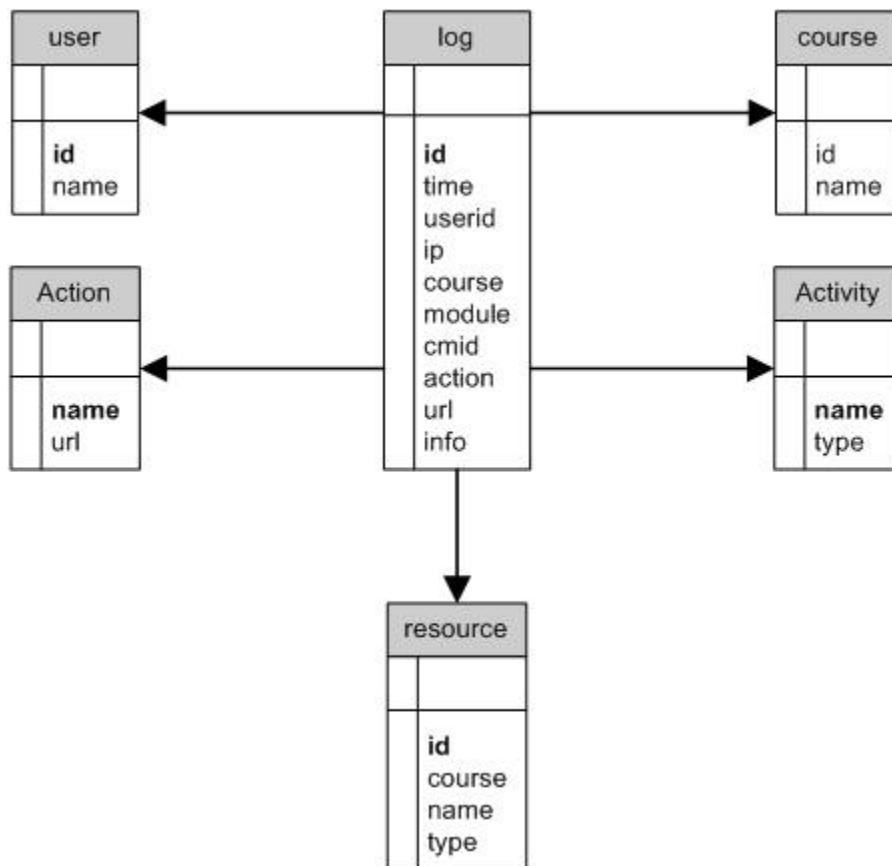
processa as informações sobre os estudantes para a estimativa de seu estado atual do conhecimento, realiza avaliação de conhecimentos, analisa o percurso e constrói/reconstrói um modelo de estudante.

Estes desenvolvimentos, estão quase sempre associados à linguagem de programação JAVA, que é adotada por grande parte da comunidade académica, para o desenvolvimento de sistemas baseados em agentes inteligentes. Desta forma pode-se encontrar várias “frameworks”, já disponíveis para a realização destas tarefas, tais como o JADE(Java Agent DEvelopment Framework). (SpA, 2012)

O Jade é um ambiente gráfico que oferece um suporte para linguagens de conteúdo e ontologias. Entre outros, consegue realizar as conversões e a verificação das linguagens de conteúdos. Assim é permitido aos programadores e analistas manipularem as informações dos agentes como objetos JAVA com pouco trabalho extraordinário.

Também nesta fase, foram realizados estudos sobre a base de dados do Moodle, pelo facto de ser necessário para resolução deste projeto. Na análise realizada à base de dados Moodle foram identificadas as tabelas e as ligações que estão representadas na Figura 19.

Figura 19 - Diagrama de e ligação de tabelas - Moodle



É verificável que consegue-se com estes dados registados em base de dados aceder à informação necessária para que o sistema aiTutor consiga funcionar com o sistema do Moodle. As interações entre o sistema de logs e as suas ligações a tabelas fundamentais tais como “*action*”, “*activity*” e “*resource*”, vão atribuir grande potencial ao aiTutor na medida em que os serviços baseados nas ontologias “vigiar” e “perfil” têm aqui a informação que necessita para o seu funcionamento.

## Conclusão

O objetivo inicial deste trabalho de estágio consistia na definição e construção de um modelo que consiste no estudo agentes inteligentes bem como a sua aplicabilidade em sistemas de elearning e criar os modelos de agentes através de procedimentos metodológicos adequados.

Estes objetivos foram atingidos e foi usada a metodologia TROPOS para o desenvolvimento de sistemas multiagentes, que se mostrou eficaz para a construção e apresentação de um modelo. No estudo levado a cabo, ficou patente o levantamento dos pormenores e a sua clareza na visualização, através das demonstrações gráficas.

Neste trabalho não foram apresentados os diagramas de classes e diagramas de sequência, porque de acordo com (Mouratidis, Giorgini, & H., 2004), esta fase envolve o uso de plataformas de desenvolvimento específicas que dependem das características da linguagem de programação adotada. Assim, esta etapa está diretamente relacionada com as escolhas da implementação e da linguagem de programação a ser usada.

Como o projeto também foi idealizado para utilizar a abordagem FIPA e neste caso, se o desenvolvimento fosse orientado para ser realizado na linguagem de programação JAVA, podia-se usar, por exemplo a “*framework Jade*” e aí ao criar e utilizar esses diagramas nessa *framework* específica eles poderiam facilmente converter-se em objetos da linguagem JAVA. Mas como este trabalho foi articulado na linha da plataforma Moodle e com esta está desenvolvida em PHP e não é conhecido neste momento nenhuma “*framework*” orientada a esta linguagem de programação, foi decidido não desenhar, de momento, esses diagramas até poder ser tomada uma decisão quanto a este facto.

Como trabalhos futuros devem ser considerados neste tema, formar uma equipe de desenvolvimento, pois o tema é bastante complexo para ser realizado, por apenas um estudante. Deve ser considerada a hipótese de desenvolver uma *framework* FIPA para outras linguagens, onde esteja incluído a PHP. Além destes desenvolvimentos também devem ser considerados, desenvolver em AUML os diagramas de comunicação, sequência e de classes, para suporte à programação propriamente dita.

## Bibliografia

- Andrade, A. M. (2005). *Ensino a Distância e elearning*. Lisboa: Instituto de Educação - Universidade Católica Portuguesa.
- Barbas, M. d.-C. (2007). *b-learning como espaço integrador de mudanças de formatos do papel ao ecrã*. Santarém: Escola Superior de Educação de Santarém.
- Castro, J., Kolp, M., & Mylopoulos, J. (Set de 2002). Towards requirements-driven information systems engineering: The tropos project. Information Systems. Reino Unido.
- Costa, E., & Simões, A. (2008). *Inteligência Artificial - Fundamentos e Aplicações*. Lisboa: FCA - Editora de Informática, Lda.
- FIPA. (2012). *The Foundation of Intelligent Physical Agents*. Obtido de FIPA: <http://fipa.org/index.html>
- Grundspenkis, J. (2011). Mipits and Ikas – Two Steps towards Truly Intelligent Tutoring System Based on Integration of Knowledge Management and Multiagent Technique. *International Conference On E-Learning and the Knowledge Society* (pp. 21-39). “JUMI” Riga, Latvia: Alla Anohina-Naumeca; Janis Grundspenkis; Angel Smrikarov.
- Hewitt, C. (04 de 10 de 2004). *Viewing Control Structures as Patterns of Passing Messages*. Obtido de DSpace@MIT: <http://hdl.handle.net/1721.1/6272>
- Horst, S. (2011). *The Computational Theory of Mind*. (E. N. Zalta, Ed.) Obtido em 12 de 2012, de The Stanford Encyclopedia of Philosophy: <http://plato.stanford.edu/archives/spr2011/entries/computational-mind/>
- Machado, J. (2001). *Agentes inteligentes como objectos dum sistema distribuído de realidade virtual*. Obtido em 28 de 12 de 2012, de <http://repositorium.sdum.uminho.pt>: <http://hdl.handle.net/1822/30>
- Machado, J. (2001). *E-Learning em Portugal – Como a formação online pode mudar a sua vida*. Lisboa: FCA - Editora de Informática.

- Mouratidis, P. B., Giorgini, P., & H., G. M. (2004). *Multi-Agent Systems and Security Requirements Analysis*.
- Myers, M. D. (Jun de 1997). *Qualitative Research in Information Systems*. Obtido em 01 de 02 de 2012, de MISQ Discovery: <http://www.misq.org/supplements>
- Nwana, H. S. (9 de 1996). *Software Agents: An Overview*. (C. U. Press, Ed.) Obtido de UMBC AgentWeb: <http://agents.umbc.edu/introduction/ao/>
- Orlikowski, Robe, W. a., D. a., & Research, S. S. (1991). *Information technology and the structuring of organizations*. Center for Information Systems Research, Sloan School of Management, Massachusetts Institute of Technology.
- Pardal, L., & Correia, E. (1995). *Métodos e Técnicas de Investigação Social*. Porto: Areal Editores.
- Redecker, C., Leis, M., & Leendertse, M. (2011). *The Future of Learning*:. Obtido em 11 de 2011, de European Commission: <http://ftp.jrc.es/EURdoc/JRC66836.pdf>
- Ridge, M. (2011). *Reasons for Action: Agent-Neutral vs. Agent-Relative*. (E. N. Zalta, Ed.) Obtido em 12 de 2011, de The Stanford Encyclopedia of Philosophy: <http://plato.stanford.edu/archives/win2011/entries/reasons-agent/>
- Rodrigues, M. F. (24 de Julho de 2007). *Proposta de uma framework para desenvolvimento de sistemas tutores inteligentes*. Obtido em 20 de 12 de 2011, de Repositorio da Universidade do Minho: <http://hdl.handle.net/1822/7656>
- Sichman, J. S., Demazeau, Y., & Boissier. (1992). When Can Knowledge-Based Systems Be Called Agents? *In Proceedings of the Ninth Brazilian Symposium on*, (pp. 172-185). Rio de Janeiro.
- Sousa, A. A., & Oliveira, E. (2010). 2010 Doctoral Symposium in Informatics Engineering - DSIE'10. Porto: Faculdade de Engenharia da Universidade do Porto.
- SpA, T. I. (12 de 04 de 2012). *Java Agent DEvelopment Framework*). Obtido de JADE: <http://jade.tilab.com/>
- Turing, A. M. (01 de 10 de 1950). *Computing Machinery and Intelligence*. Obtido de Oxford Journals: <http://mind.oxfordjournals.org/content/LIX/236/433.short>

- Wooldridge, M., & Jennings, N. R. (1995). *Towards a Theory of Cooperative Problem Solving*. Obtido de University of Liverpool - Mike Wooldridge - Publications: <http://www.csc.liv.ac.uk/~mjw/pubs/maamaw94.pdf>
- Yu, E. (s.d.). *Agent-Oriented Modelling: Software Versus the World*. (U. o. Toronto, Ed.) Obtido de <http://www.cs.toronto.edu/pub/eric/istar-bk-Ch1.pdf>
- Yu, E., Mylopoulos, J., Maiden, N., & Giorgini, P. (2011). *Social Modeling for Requirements Engineering*. (MIT, Ed.) Massachusetts, USA.
- Zambonelli, F., Jennings, N. R., & Wooldridge, M. (s.d.). *Developing Multiagent Systems: The Gaia Methodology*. Liverpool, Liverpool, Inglaterra.

# **Anexos**

## **Anexo 1**

### **GAIA**

A metodologia GAIA foi uma das primeiras Metodologias Orientada a Agentes (MAS) que estabelece um sistema de agentes como uma organização computacional, onde cada papel é interpretado por um agente e estes cooperam com outros agentes, de forma a poderem atingir o objetivo principal de um aplicativo. Como esta metodologia, inicialmente, só se podia aplicar a sistemas multiagentes fechados e com a evolução do conhecimento nesta área, chegou-se à conclusão que os agentes, devem ser sempre cooperativos. Com naturalidade, surgiu a Versão 2 desta metodologia, combinada com uma representação UML (Unified Modeling Language) dedicada a agentes aUml (UML Agente).

Com a metodologia GAIA começa-se com a abstração ao nível dos conceitos e após cada etapa de construção os conceitos vão sendo refinados até se transformarem em conceitos concretos. Esta metodologia divide-se em duas partes: Análise e Design (Sousa & Oliveira, 2010).

Na aplicação de Gaia, começamos com conceitos abstratos e, em cada etapa, os modelos são fabricados até conceitos concretos que podem ser implementados para atender aos requisitos.

A metodologia está dividida em duas partes: Análise e Design.

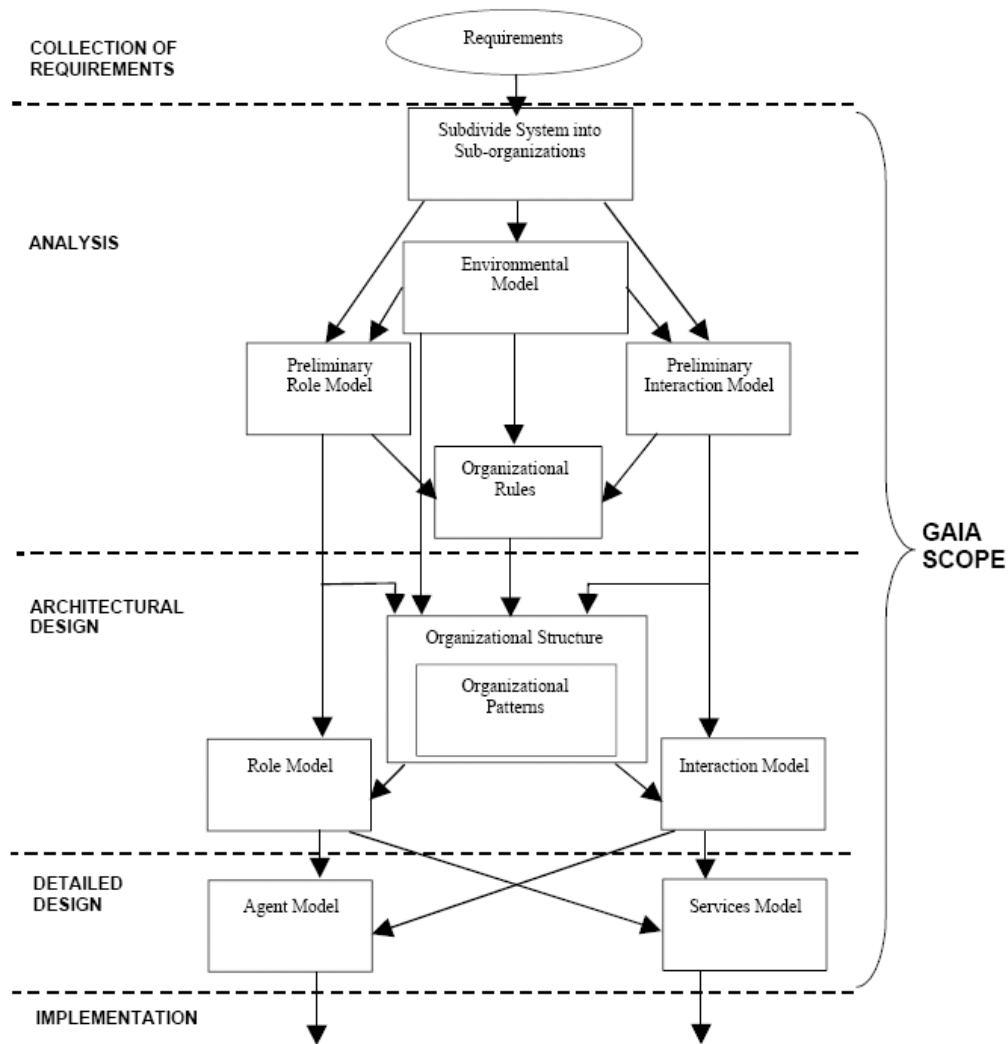
Alguns autores identificam a fase de elicitação/captura de requisitos como uma fase independente da análise. Esta fase tem como objetivo descrever a estrutura do modelo do sistema bem como identificar as peças da organização para que haja uma melhor compreensão deste modelo. Na análise tem de se transformar os resultados em bruto em unidades de baixo nível para que a abstração possa descer a um nível inferior de modo a que seja possível visualizar a implementação dos agentes.

Na análise em GAIA tem que se definir os papéis e determinar as funcionalidades do sistema. Quando isto acontece, existe um protótipo em que os papéis estão definidos, existem responsabilidades atividades, permissões e protocolos atribuídos produzindo assim uma função bem delimitada.

A seguir a esta fase deve ser desenvolvido um modelo de interação onde será demonstrada a comunicação entre os papéis. Os *outputs* desta fase, contêm todos os aspectos importantes, incluindo regras de organização, protocolos e atividades.

Pode visualizar-se na Figura 20 as fases da GAIA

Figura 20 - Models of the Gaia methodology and their relations in the Gaia process (Zambonelli, Jennings, & Wooldrige).



Apesar das melhorias, apresentadas pela Gaia V. 2, ainda existem outros aspectos que têm de ser observados e talvez até melhorados (Sousa & Oliveira, 2010). Uma das primeiras é a limitação de não poder capturar adequadamente os aspectos dinâmicos do meio ambiente, que são essenciais para sistemas MAS complexos. Além disso, as estruturas organizacionais são modeladas usando uma notação simples o que atribui imensas dificuldades à construção e conceção de sistemas de grandes dimensões. A

limitação final refere-se à engenharia de requisitos, que não é tratada em Gaia da mesma forma como em outras metodologias.