

Sobre Linguagens para Modelagem de Sistemas Multiagentes em Domínio Ecológico

Míriam Blank Born¹

¹Pós-Graduação em Computação – (PPGC)
Centro de Desenvolvimento Tecnológico – (CDTec)
Universidade Federal de Pelotas – (UFPeI)

mbborn@inf.ufpel.edu.br

Abstract. *The present study aims to review the areas of Artificial Intelligence (AI) with a focus on Multiagent Systems (SMA), specific and ecological domain languages, ecosystems modeling, as well as the integration of these with SMA. Currently, several studies show satisfactorily the use of the SMA approach to studies in the field of Ecology, in this way this methodology presents significant results. Together, the research of these areas, in this work, is a study on the type system (TPO) and the descriptive language of agent society (SML), at a conceptual level and in the development phase. In this way, we seek the integration of this conceptualization, TPO and SML, and the multi-agent approach in the development of systems that help in the area of Ecology in general.*

Resumo. *O presente estudo tem por objetivo uma revisão das áreas de Inteligência Artificial (IA) com enfoque em Sistemas multiagentes (SMA), linguagens de domínio específico e ecológico, modelagem de ecossistemas, bem como a integração destas com SMA. Atualmente, diversos estudos mostram de forma crescente a utilização da abordagem de SMAs a estudos na área da Ecologia, desta forma esta metodologia apresenta resultados significativos. Juntamente às pesquisas destas áreas, encontra-se neste trabalho um estudo sobre o sistema de tipos (TPO) e a linguagem descritiva de sociedade de agentes (SML), em nível conceitual e em fase de desenvolvimento. Desta forma, busca-se a integração desta conceitualização, TPO e SML, e a abordagem multiagente no desenvolvimento de sistemas que auxiliam na área da Ecologia em geral.*

1. Introdução

A crescente demanda da utilização de sistemas multiagentes (SMA) na busca por soluções satisfatórias dos mais variados problemas, torna-se frequente. Os SMAs são vistos como uma possibilidade efetiva destas resoluções, propiciando flexibilidade e eficiência no desenvolvimento de sistemas nas diversas áreas, dentre estas, a Ecologia, Biologia, Hidrologia, Meio Ambiente, entre outras.

Embora inúmeros trabalhos vêm sendo desenvolvidos ao longo dos anos em SMA, ainda há muita pesquisa a ser desenvolvida. Na literatura, a abordagem multiagente para o desenvolvimento de aplicações é muito ampla. Entretanto alguns dos sistemas propostos até o momento são genéricos, implicando, por vezes, em especificações não detalhadas como é o caso da Ecologia. Neste contexto, a pesquisa e desenvolvimento de novas especificações torna-se essencial, com intuito de tornar os sistemas baseados em agentes o mais próximo da realidade e com resultados satisfatórios.

A revisão realizada neste trabalho, além de estudar SMA e alguns exemplos de trabalhos relacionados ao domínio ecológico e às linguagens de domínio específico, busca discutir sobre estes aspectos e, também, estudar novos conceitos que podem ser inseridos neste cenário. Um exemplo disto, é a Seção 4 deste trabalho, a qual apresenta de forma ilustrativa um sistema de tipos (TPO) e

uma linguagem descritiva de sociedade de agentes (SML), proposta por [Costa 2017b], ainda em fase de desenvolvimento.

Desta forma, o estudo e pesquisa de modelagens e linguagens existentes, neste caso de ecossistemas e domínio ecológico, auxiliarão na possível proposta deste trabalho a qual visa, integrar SMA ao contexto a ser definido.

O artigo está organizado como segue. Na Seção 2 é apresentada uma revisão bibliográfica dos principais conceitos necessários ao entendimento do contexto do trabalho. Na Seção 3 são introduzidas as linguagens de domínio específico. Na Seção 4, é apresentado um sistema de tipos e uma linguagem de modelagem para sociedades de agentes e, na Seção 5, são apresentadas as linguagens de domínio ecológico para a modelagem de ecossistemas, partes essenciais para a construção futura da proposta de tese. Por fim, na Seção 6, são apresentadas as conclusões deste trabalho.

2. Os Sistemas Multiagentes

A Inteligência Artificial (IA) surgiu na década de 50 apoiada em fundamentos de diversas outras áreas como a Filosofia, Matemática, Psicologia, Neurociência, Economia, Linguística e Engenharia de computadores [Russell and Norvig 2013].

Em 1995, surge o conceito de agentes inteligentes, sendo a Internet o ambiente propício para a utilização destes no desenvolvimento de mecanismos de pesquisa de informações, sistemas de recomendação, lançamento de produtos de determinada empresa, acompanhamento de tendências de mercado, etc. De acordo com [Russell and Norvig 2013], “um agente é algo capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por meio de atuadores”, considerando a habilidade do agente para a execução de tarefas/ações específicas e de raciocinar sobre determinado domínio.

De acordo com [Nwana 1996], os agentes inteligentes possuem três características importantes, cooperar, aprender e agir de maneira autônoma. A partir destas características o autor os divide em: agentes espertos, agentes colaborativos, agentes de aprendizado colaborativo e agentes de interface. Segundo [Coppin 2010], os agentes devem possuir estas mesmas propriedades, porém de forma diferenciada o autor os classifica em:

Agentes reativos: este tipo de agente reage a eventos no ambiente em que está inserido de acordo com regras/normas especificadas previamente. Porém, estes agentes possuem desempenho indesejado para solucionar problemas aos quais as regras não estejam definidas, visto que o mesmo somente reage a estímulos, não conseguindo adaptar-se a novas situações.

Agentes de interface: um agente de interface têm o intuito de auxiliar um usuário nas diversas aplicações, também pode ser denominado como um assistente que aprende a realizar tarefas. Este tipo de agente coopera com o usuário, pode interagir ou não com os demais agentes do ambiente, entretanto esta interação acontece com a finalidade de aprendizado.

Agentes de informação: também chamados de agentes de coleta de informações e agentes de internet, ajudam o usuário a encontrar, classificar e filtrar informações provenientes de inúmeras fontes da internet. Estes podem ser estáticos ou móveis, e também colaborativos ou independentes.

Agentes colaborativos: possuem a principal característica de cooperação, ou seja, cooperam entre si para alcançar objetivos. Contudo, estes agentes apresentam capacidade limitada de aprendizagem, sendo assim a finalidade deste tipo de agente é a combinação de muitos agentes para que um problema atinja solução satisfatória.

Contudo, as propriedades e atributos dos agentes são implementados de acordo com o problema a ser resolvido, a complexidade e com o domínio específico de cada aplicação.

Em [Russell and Norvig 2013] e [Luger 2008] diversas aplicações e algoritmos são apresentados de forma a exemplificar a busca de soluções para esta demanda crescente de problemas.

Os sistemas multiagentes constituem-se de diversos agentes interagindo em um ambiente. Os SMA foram introduzidos na Computação na década de 80, entretanto, somente nos anos 90 tornaram-se populares [Wooldridge 2002]. Cada agente de um sistema possui comportamento individual, porém os mesmos devem ser capazes de interagir com os demais de forma organizada, desta maneira características como: cooperação, coordenação, competição e negociação, são relevantes [Bordini et al. 2001].

Tais características tornam-se importantes pois, a maioria dos problemas a serem resolvidos buscam uma maneira distribuída de resolução. Além disso, muitos destes possuem elevada complexidade, tornando impossível encontrar a solução apenas com um agente. Em um SMA, os agentes podem cooperar em busca da solução de um objetivo geral, sendo que, cada um destes possui seu próprio objetivo, mas que juntos, alcançam uma meta maior [Alvares and Sichman 1997, Bordini et al. 2001].

Diversos são os benefícios da utilização de um sistema multiagente: (i) rapidez na resolução de problemas visto a inerência do processamento concorrente; (ii) aumento da flexibilidade e escalabilidade através da conexão de vários sistemas; (iii) aumento da capacidade de resposta à um determinado problema pelo fato de todos os recursos estarem localizados no mesmo ambiente; entre outros. Segundo [Alvares and Sichman 1997], existem as abordagens *Top-down* e *Bottom-up* para o desenvolvimento dos SMAs.

Na abordagem *Top-down* são definidos os aspectos coletivos, relacionados a organização e comunicação entre os agentes. Já na abordagem *Bottom-up* são definidos os aspectos individuais de cada agente do sistema, neste caso, a organização e a interação são configuradas do ponto de vista destes agentes.

Analisando a abordagem *Bottom-up*, o principal objetivo de um SMA é a configuração genérica de agentes, interações e organizações, apresentando as seguintes características: i) os agentes de um sistema são definidos independentemente do problema a ser resolvido; ii) não há controle centralizado do problema; iii) os protocolos de comunicação não são definidos previamente, e sim, para serem utilizados em situações genéricas; e, iv) a decomposição de tarefas, em determinado sistema, pode ser executada pelos próprios agentes.

A partir desta análise, algumas vantagens são observadas, por exemplo [Alvares and Sichman 1997]: i) viabilidade de sistemas adaptativos e evolutivos, onde novas situações são facilmente adaptadas pelos agentes, bem como a inclusão de novos agentes ao ambiente; ii) em vários casos o conhecimento, o controle e os recursos inerentes ao sistema estão distribuídos permitindo alto nível de abstração; iii) os ambientes são heterogêneos e distribuídos, ou seja, agentes diferentes podem estar divididos em plataformas distintas, ligados a uma rede de computadores, e mesmo assim podem cooperar para a solução de um problema; e, iv) permite a migração de agentes para demais sociedades, caracterizando sistemas abertos.

Dentre as principais áreas de aplicação que os SMA englobam pode-se citar [Rezende 2003]:

Indústria: na modelagem de linhas de produção, onde os agentes cooperam para alcançar um objetivo;

Gerenciamento de negócios: sistemas *workflow* onde tarefas são direcionadas à certas pessoas em determinado tempo;

Ambientes de aprendizagem: um exemplo de alguns trabalhos desenvolvidos nos ambientes de aprendizagem é os tutores inteligentes;

Controle de tráfego aéreo: uma aplicação bastante complexa, devido a quantidade de agentes envolvidos e a sincronização de várias tarefas;

Aplicações distribuídas: muito utilizados em telecomunicação, sistemas para área de saúde e logística e transportes em geral;

Simulação social: onde o objetivo principal é auxiliar cientistas sociais em seus estudos, modelando aspectos sociais e cognitivos dos indivíduos;

Aplicações para internet: problemas como, gerenciamento de informações e recomendações são muito difundidos em SMA.

No desenvolvimento de SMA a arquitetura comumente utilizada é a BDI (*beliefs, desires e intentions*), baseadas em um modelo cognitivo que representam crenças, desejos e intenções [Hübner et al. 2004].

De acordo com [Wooldridge 2002], BDI é estruturada em: crenças, representa o que o agente sabe sobre si mesmo, dos demais agentes e o ambiente ao qual está inserido; os desejos, representam os estados que o agente almeja atingir, geralmente são objetivos e, as intenções, são representadas pela sequência de ações que determinado agente executa para alcançar um objetivo.

Segundo [Hübner and Sichman 2003], a comunicação em sistemas SMA ocorre de duas formas, sendo estas:

Comunicação direta: neste tipo de comunicação, os próprios agentes do sistema tratam da comunicação sem qualquer intervenção externa. Cada agente comunica-se diretamente com os demais, devido a isto, há possibilidade de ocorrência de problemas na comunicação, como por exemplo, se todos os agentes decidirem enviar mensagens ao mesmo tempo, provavelmente ocorrerá o bloqueio do sistema.

Comunicação assistida: nesta comunicação existem agentes denominados facilitadores, ou seja, atuam na sincronização de mensagens entre todos os agentes do sistema. Esta estrutura, resolve parcialmente um eventual problema de comunicação diminuindo consideravelmente a complexidade, por outro lado, caso este agente facilitador deixe de funcionar por algum motivo, todo o sistema será bloqueado.

A comunicação entre os agentes de um SMA possui duas finalidades principais: compartilhar informação, planos ou crenças entre estes e a coordenação de tarefas entre os mesmos. Para que a comunicação eficiente seja possível é necessário que estas duas finalidades estejam alinhadas e, sendo assim, uma linguagem comum entre os agentes é indispensável; e, os seguintes elementos são caracterizados [Hübner 2003]:

Sintaxe: estrutura gramatical, símbolos da linguagem a qual contém as regras que compõem a comunicação.

Semântica: significado dos símbolos para a compreensão da linguagem.

Vocabulário: conjunto de símbolos utilizados na linguagem com breves explicações.

Regras: conjuntos de regras e símbolos pertencentes à linguagem para a interpretação.

Modelo de domínio: significado que os conjuntos de símbolos podem assumir durante a comunicação.

A comunicação entre os agentes de uma aplicação ocorre por meio de mensagens trocadas entre os mesmos. Os agentes devem ser capazes de se comunicar, assim sendo, podem enviar e receber mensagens e por isso estes agentes são divididos em quatro tipos, de acordo com a sua capacidade de comunicação: agente básico, este apenas aceita informação exterior; agente passivo, capaz de dialogar, aceita perguntas e responde à elas; agente ativo, capaz de dialogar ativamente, aceitando e também fazendo perguntas; e, agente interlocutor, assume o papel de interlocutor entre os demais agentes.

Para que a comunicação entre os agentes de um SMA seja possível, protocolos e níveis de comunicação são pertinentes ao processo. O nível inferior define o método de interligação entre os agentes; o intermediário é responsável pela sintaxe da informação a ser transmitida; e, no superior, as especificações semânticas da informação são definidas.

Os protocolos normalmente possuem a seguinte estrutura de dados: emissor, receptor, linguagem, funções de codificação e decodificação e ações realizadas pelo receptor [Hübner 2003].

As linguagens de comunicação começaram a ser difundidas nos anos 90 com a fundação, nos EUA, do *Knowledge Sharing Effort* – KSE [Hübner 2003]. Este órgão tinha como objetivo principal o desenvolvimento de protocolos para troca de informações. Originou-se do KSE as linguagens de comunicação KQML, KIF e FIPA ACL.

A linguagem KIF (do inglês, *Knowledge Interchange Format*) foi desenvolvida para a representação de propriedades de um domínio específico, sendo baseada em lógica de primeira ordem. Esta linguagem permite a descrição de propriedades e relações de determinado domínio de forma simples, utilizando operadores comuns de lógica (*and*, *or*, *not*, entre outros) e estruturas de dados como números, *strings* e caracteres. *KIF* também aceita descrição de sequência de ações, entidades, fatos e conhecimento dos agentes [INTERCHANGE 1998].

A linguagem KQML (do inglês, *Knowledge and Query Manipulation Language*) é mais específica para a troca de mensagens, caracteriza-se pela especificação de todo o conteúdo necessário para a comunicação e com número de parâmetros determinados; bastante utilizada na descrição de Ontologias [Finin et al. 1994, Labrou and Finin 1997].

A linguagem FIPA ACL (do inglês, *Agent Communication Language*) assemelha-se a *KQML*, sendo que esta define um número específico de performativas e sua estrutura define o exterior das mensagens. A *FIPA ACL* possui performativas mais específicas quanto a execução de processos de negociação [Fipa 2002].

3. As Linguagens de Domínio Específico

As Linguagens de Domínio Específico (DSL – do inglês, *Domain specific languages*) caracterizam-se por serem uma linguagem de programação de computadores a qual apresenta limitações e é focada em um domínio específico [Fowler 2010, Mernik 2012]. Segundo [Deursen et al. 2000] estas linguagens podem apresentar benefícios, desde que bem projetadas, entre estes destacam-se:

- + As soluções são expressadas de acordo com o domínio do problema, desta forma, os especialistas conseguem entender, validar e modificar o desenvolvimento;
- + As DSLs geralmente são concisas, auto-documentadas e utilizadas para fins diferentes;
- + Aumentam a produtividade, confiabilidade, portabilidade e facilidade de manutenção;
- + Permitem a reutilização pois incorporam conhecimento de domínio e a validação.

Por outro lado, de acordo com [Deursen et al. 2000], as DSLs trazem algumas desvantagens:

- Custos de projeto, implementação e manutenção destas linguagens;
- Dificuldades quanto a escopo e disponibilidade limitada;
- Dificuldades entre a especificidade do domínio do problema e no desenvolvimento de funções e métodos da linguagem;
- Perda de eficiência em comparação de *software* com codificação manual.

Atualmente existem diversos exemplos de DSLs como: HTML, SQL, VHDL, C#, R e S, entre muitas outras. Estes tipos de linguagens podem ser agrupadas por áreas, tais como:

Engenharia de software: as linguagens de banco de dados, produtos financeiros, arquiteturas de software, controle e coordenação.

Sistemas de software: como especificações de *driver* de vídeo, estrutura de dados em C, especificações de sistema operacional, etc.

Computação multimídia: para manipulação de imagens, desenhos, animações.

Telecomunicações: no desenvolvimento de protocolos de comunicação e também em variados tipos de simulações, tais como agentes, SMA, robôs de controle [Deursen et al. 2000].

As linguagens de domínio específico podem ser classificadas como internas e externas. A DSL interna é projetada a partir de regras sintáticas e semânticas de uma linguagem definida; realizando as atividades de verificação léxica, sintática, semântica da linguagem [Mernik 2012].

A DSL externa é uma linguagem com sua própria sintaxe, sendo separada da linguagem principal; e, sua infraestrutura é separada para análise léxica, sintática, semântica, interpretação, compilação, otimização e geração de código.

De forma geral, quanto ao processo de desenvolvimento, as DSLs são divididas nas etapas de: decisão, análise, projeto, implementação, validação, instalação e manutenção [Mernik 2012]. Estas fases serão brevemente descritas a seguir:

Decisão: nesta etapa são levantadas informações importantes como o domínio, comunidades de usuários, ferramentas que serão utilizadas no desenvolvimento e custos desta implementação.

Análise: as entradas que constituem esta etapa dizem respeito às fontes de conhecimento, como pesquisas, manuais, entrevistas com especialistas, documentos, etc.

Projeto: descreve a natureza do projeto e a relação com linguagens existentes.

Implementação: na implementação de uma DSL geralmente são utilizadas duas abordagens: compilada e interpretada. Porém, podem ser inseridas outras abordagens, pré-processada, extensão de compiladores/interpretadores, interna e híbrida.

Validação: avalia a DSL em relação aos requisitos implementados nesta, esta avaliação pode ser quantitativa ou qualitativa.

Instalação: conjuntos de atividades que se inter-relacionam, garantindo um método para identificações de versões e configurações da linguagem.

Manutenção: atualização de novos requisitos da linguagem.

Esta seção visou descrever brevemente sobre os principais aspectos relacionados as DSLs, visto que alguns trabalhos mencionados no decorrer deste artigo, mencionam a utilização destas linguagens para a implementação de domínio específico, por exemplo as áreas de Ecologia e Hidrologia.

4. O Sistema de Tipos e A Modelagem de Sociedade de Agentes

Nesta Seção, será abordado o trabalho de [Costa 2017b], como um exemplo de pesquisa da conceitualização de um sistema de tipo (TPO) e uma linguagem de modelagem para sociedades de agentes (SML). Este estudo apresenta uma abordagem baseada em tipos buscando uma caracterização formal aplicada a uma sociedade de agentes, ou seja, um sistema multiagente.

O sistema de tipo deste estudo denomina-se observacional, ou seja, a estrutura do modelo a suas operações são visualizadas externamente e a linguagem de modelagem social busca fornecer uma sintaxe concreta para objetos de TPO. Tanto TPO quanto SML foram definidas em trabalhos anteriores [Costa and Dimuro 2009, Costa and Demazeau 1996, Demazeau and Costa 1996, Costa 2016]. A Figura 1 representa os componentes da arquitetura da sociedade de agentes proposta pelo autor, como segue:

População (Pop): representa os agentes e suas redes;

Sociedade (Soc): a sociedade é dividida em regras de sociabilidade e redes de sociabilidade;

Organização (Org): este componente da arquitetura é representado por regras organizacionais, unidades organizacionais e redes de unidades organizacionais;

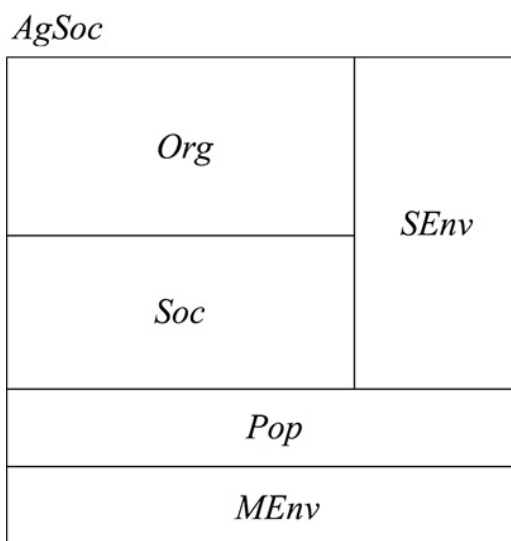


Figura 1. Arquitetura dos componentes da sociedade de agentes. Fonte: [Costa 2017b]

Ambiente material (MEnv): representação dos objetos materiais; e,
Ambiente simbólico (SEnv): representação dos objetos simbólicos.

No modelo em estudo as sociedades de agentes interagem entre si através de canais de importação e exportação. A partir destes canais, uma sociedade por exemplo, pode interagir com uma ou mais sociedades da rede de forma aberta, sendo que estas são reconfiguradas de maneira dinâmica por meio de mudanças ambientais, culturais e organizacionais.

Neste cenário, surge o conceito de caracterização formal que, segundo [Costa 2017b], torna este tipo de modelagem mais flexível no sentido que uma sociedade evolui ao longo do tempo assim como o sistema de agentes e, desta forma, o mesmo está propenso a alterações de acordo com sua evolução.

No sistema TPO, considerando que o mesmo é observacional, existem objetos animados e inanimados. O primeiro corresponde ao objeto em que pelo menos uma sequência de eventos foi causada por este, caso não ocorra uma associação ao objeto, este é denominado inanimado.

Ainda considerando as principais características do sistema TPO e de acordo com a arquitetura deste, tem-se os objetos materiais e simbólicos. Os objetos, como mencionado anteriormente, correspondem a objetos do ambiente material da sociedade de agentes, e os objetos simbólicos são objetos do ambiente simbólico do sistema.

Os ambientes simbólicos são responsáveis pela modelagem dos aspectos culturais das sociedades estudadas, sendo assim são chamados ambientes efetivos os quais estas sociedades possuem permissão para operar. Estes ambientes detêm as noções ideológicas que regulam a forma como os agentes, as unidades organizacionais e a sociedade comportam-se e interagem entre si.

Quando fala-se em objeto simbólico as normas são importantes pois regulam o comportamento e as interações dos agentes, das sociedades e das unidades organizacionais, estas ditam como todas as entidades de um sistema realizam suas operações. Desta forma, no sistema TPO as normas são um subtipo do tipo de objetos simbólicos.

Outros pontos significativos a serem considerados no sistema de tipos são: externalismo, internalismo e intencionalidade. Os agentes são classificados como internalistas, visto que os conceitos mentais destes fornecem papel central no contexto do sistema, por outro lado, as sociedades de agentes, os sistemas de agentes intersociais e as unidades organizacionais são classificadas como

externalistas, visto que estes conceitos mentais representam papel complementar.

Contudo, o externalismo permite a atribuição de intencionalidade a entidade, considerando o que estas podem ou não ter. Sob esta forma externalista as entidades, agentes, sociedades de agentes e sistemas de agentes intersociais não são necessariamente obrigadas a possuir os conceitos como desejo, crença e intenção.

Neste estudo, o sistema TPO e a linguagem SML são apresentados como um sistema de tipo e uma linguagem universal para as entidades de sociedade de agentes e sistemas intersociais. TPO também pode ser estendido e aprimorado para atender a outras aplicações específicas [Costa 2017b]. Ainda, de acordo com [Costa 2017b], SML é a linguagem descritiva dos conceitos do sistema TPO, sendo que os estágios de desenvolvimento deste tipo de linguagem descritiva compreendem:

Modelo formal: neste caso, sociedade de agentes, representa um sistema definido;

Tipo de sistema: base para a definição de linguagem descritiva abstrata, especificação sintática das relações entre os diversos tipos do sistema de tipos;

Sintaxe abstrata: base para o desenvolvimento de um modelo semântico formal para a linguagem descritiva e uma ou mais sintaxe para a representação concreta.

O estudo da sociedade de agentes, do sistema TPO e da linguagem SML encontra-se em fase de desenvolvimento segundo o autor, entretanto menciona-se neste estudo, alguns dos principais conceitos definidos até o presente momento. A Tabela 1 representa uma das estruturas relacionada ao sistema de tipos.

Tabela 1. Sintaxe abstrata de SML: Estrutura da população e representação no sistema TPO.
Fonte: [Costa 2017b]

Estrutura da População	TPO
AG := [AGENT]*	AG: $\varphi(\text{Agent})$
AGProp := [AGENT PROP]*	AGProp: $\varphi(\text{Agent} \times \text{Prop})$
AGBeh := [AGENT Proc]*	AGProp: $\varphi(\text{Agent} \times \text{Proc})$
AGInter := [AGENT AGENT ExchProc]*	AGInter: $\varphi(\text{Agent} \times \text{Agent} \times \text{ExchProc})$
AGRel := [AGENT AGENT REL]*	AGRel: $\varphi(\text{Agent} \times \text{Agent} \times \text{Rel})$
Pop := AG, AGProp, AGBeh, AGInter, AGRel	Pop : AG \times AGProp \times AGBeh \times AGInter \times AGRel

Na Tabela 1 a representação do “*” significa que, a expressão entre “[]” pode ser repetida zero ou mais vezes. A expressão contida entre “[]” indica ocorrência opcional desta; e, o sequenciamento dos elementos é disposto lado a lado e separado por “,” e as regras da gramática são representadas por “cabeçalho := corpo”. Esta representação é detalhada no material do autor para cada uma das estruturas que compõe o estudo.

Neste trabalho, apresentou-se uma visão geral sobre os conceitos e características abordadas por [Costa 2017b], sendo que as formas de representação destes conceitos, detalhamento do sistema de tipos e da linguagem SML e características podem ser visualizados no material disponibilizado¹.

Outro trabalho recentemente apresentado por [Costa 2018], descreve um estudo de caso, no qual este utiliza os conceitos propostos em [Costa 2017b], para a caracterização formal de sistemas econômicos elementares.

Nestes sistemas, foi observado o comportamento, a troca e os processos envolvidos. O conjunto de tais processos caracterizam o equilíbrio do sistema e são considerados os objetos energéticos para a sociedade de agentes materiais.

¹Disponível em: <https://github.com/miriamborn/tpo-sml>

Neste contexto, conforme o estudo em [Costa 2017b], a energia de uma sociedade de agentes materiais é obtida na forma de objetos de energia. sendo esta produzida e distribuída a estes agentes componentes do sistema. Um agente material possui um corpo ao qual necessita de energia para o funcionamento [Costa 2018].

Desta forma, o sistema econômico elementar da sociedade é composto pelo seu conjunto de processos e para que seja possível a formalização em uma sociedade de agentes materiais, estes agentes devem produzir outros tipos de objetos para que ocorra a troca entre os mesmos.

Sendo assim, o autor descreve no trabalho de que forma ocorrem os comportamentos, as trocas e os processos neste sistema econômico elementar tanto as regras comportamentais individuais como em grupos. As sociedades de agentes materiais e os sistemas de energia que os compõe, os sistemas econômicos elementares e suas definições formais e por fim, os ecossistemas como um estudo de caso para esta análise específica [Costa 2018].

Entretanto, [Costa 2018] ressalta que, para a definição de sistemas econômicos plenos será necessário o levantamento de outras restrições: a) que o condicionamento operante seja direto (isto é, devido ao parceiro que interage), permitindo condicionamentos operantes indiretos, através de redes de parceiros indiretos; b) a ausência de ciclos nos processos econômicos, levando a complexas redes de processos econômicos interligados; c) admissão de apenas dois tipos de objetos em trocas econômicas.

5. As Linguagens de Domínio Ecológico e A Modelagem de Ecossistemas

Nesta Seção, serão abordados trabalhos sobre ecossistemas e seus principais conceitos e, também, a modelagem destes em sistemas multiagentes. As pesquisas mencionadas a seguir, mostram um panorama geral de modelagens e simulações desenvolvidas nos últimos anos, com intuito de buscar soluções satisfatórias a esta demanda de problemas.

O trabalho de [Costa 2017a] trata do estudo de ecossistemas, descrevendo os principais aspectos organizacionais e funcionais destas paisagens, em uma abordagem de sociedade de agentes. Tais estruturas ecológicas representam os agentes desta sociedade e a paisagem seria o sistema de sociedade de agentes.

Para o mapeamento dos conceitos básicos sobre Ecologia em um sistema de sociedade de agentes, o entendimento dos mesmos são necessários. Segundo [Odum and Odum 2003], os ecossistemas são caracterizados por quatro dimensões:

Populacional: refere-se a população do ecossistema, ou seja, tipo de indivíduo ou conjuntos destes que compõem o sistema;

Organizacional: são as interações ocorridas neste sistema, como os indivíduos ou grupos se relacionam;

Funcional: execução das funções de cada indivíduo ou conjuntos destes, as formas como ocorrem as interações entre organismos e grupos e a coordenação dos indivíduos dentro do sistema; e,

Geográfica: abrange as áreas ocupadas pelos ecossistemas e a constituição destes.

Outro aspecto importante diz respeito a organização e funcionamento dos ecossistemas. Por exemplo, aspectos indispensáveis a vida de qualquer espécie devem ser considerados, como por exemplo: o sol (principal fonte de energia), a atmosfera (com fornecimento de O₂, CO₂ e água), o solo (para o abastecimento de água e nutrientes) e os organismos (são produtores, consumidores ou decompositores).

Para a modelagem deste ecossistema em sociedade de agentes foi utilizado o termo *agente sociedade* para designar um tipo particular de sistema multiagente. Conforme apresentado

em [Costa 2017a], um sistema multiagente é considerado: i) aberto, no sentido de que os agentes podem entrar e sair livremente; ii) organizado, onde os processos de sistemas podem ser identificados, sendo realizados por um indivíduo (agente) ou um grupo; iii) persistente, significando que a organização dos processos de sistemas persiste no tempo; e, iv) situado, os agentes que constituem o sistema operam em um ambiente determinado (físico ou simulado), cujos objetos os agentes podem fazer uso em seus processos individuais e sociais.

A formalização utilizada por [Costa 2017a] é idêntica a apresentada anteriormente em [Costa 2017b], a partir dos conceitos de sociedade de agentes já mencionados neste trabalho, este estudo mostra primeiramente o mapeamento informal do ecossistema em questão. Neste caso, foi utilizado o conceito de importação e exportação, mencionado no estudo de [Costa 2017b], supondo:

Nível populacional: as populações são os agentes dos ecossistemas;

Nível micro-organizacional: as populações implementam os papéis organizacionais das sociedades;

Nível meso-organizacional: as comunidades são as unidades organizacionais

Nível macro-organizacional: os ecossistemas são os agentes de importação e exportação;

Nível ambiental: é o ambiente material, sendo ocupado pelas espécies e comunidades; e,

Ambiente simbólico: irá capturar, no modelo formal baseado em agente, os aspectos culturais do grupo humano responsável por essas ações.

A formalização do modelo de ecossistema mapeado para o sistema de sociedade de agentes é mostrado em detalhes no trabalho [Costa 2017a]. Desta forma, foi considerado que esta é uma abordagem viável para a modelagem e simulação de ferramentas computacionais neste contexto ecológico, ecossistemas e paisagens.

Os SMAs, introduzidos conceitualmente neste contexto, são comparados a modelagens baseadas em indivíduos. Este trabalho menciona diversos propósitos para a utilização desta abordagem, tanto na teorização de conceitos quanto na tomada de decisão.

Para um gerenciamento satisfatório de um ecossistema, os problemas de acesso e utilização de recursos naturais e renováveis são questões fundamentais. As ferramentas de desenvolvimento destes sistemas podem ser divididas em três tipos, considerando que, a maioria delas é implementada com linguagens de orientação a objetos e algumas utilizam plataformas organizadas como segue:

Plataformas genéricas: estas plataformas trabalham com conceitos mais amplos, não especificamente um domínio, os autores citam a ferramenta Swarm [Bousquet and Le Page 2004] como uma das mais utilizadas e, também, o NetLogo visto como um ambiente de fácil utilização e com foco nos aspectos cognitivos dos agentes.

Plataformas de simulação social e ecológica: as ferramentas de desenvolvimento neste tipo de plataforma, fornecem representação espacial e simulação de utilitários para métodos Monte-Carlo, por exemplo, e *links* para outros *softwares*. Diversos algoritmos e estruturas de dados implementam a conexão entre os agentes e seu ambiente, assim como elementos para a organização destes. Exemplos de ferramentas, a Ecosim, orientada a ecologia e Repast ou Cormas que implementam a dinâmica social com interação a dinâmica de recursos naturais.

Plataformas dedicadas: são ferramentas com aplicações específicas de determinada área, exemplos: Manta [Drogoul 1994], que trabalha com a atribuição de ações em uma sociedade de insetos; Arborscape [Bousquet et al. 2002b] que modela a dinâmica da floresta com destaque na diversidade; a BacSim, ferramenta para modelagem dinâmica microbiológica [Kreft et al. 1998]; e, a Mobydic [Ginot et al. 2002] que modela a dinâmica das populações de peixes, dentre outras.

A pesquisa de [Bousquet and Le Page 2004] apresenta uma revisão sobre simulações multiagentes para o gerenciamento de ecossistemas. As ferramentas existentes que utilizam a metodologia

SMA auxiliam nesta área, desta forma o comportamento e as interações são de extrema importância para a modelagem da organização de ecossistemas.

Na literatura, o trabalho de [Holst and Belete 2015] apresenta uma revisão sobre as linguagens de domínio específico para a modelagem Ecológica. A Tabela 2 sumariza as principais linguagens ou ferramentas de modelagem para este domínio.

Tabela 2. Representação de Linguagens de Domínio Específico em Modelagem Ecológica.
Fonte: [Holst and Belete 2015]

DSL	Autores	Aplicação	Código	Interface
MOSES	[Wenzel 1992]	Ecology	Fortran	Não
OOMP	[Holst and Belete 2015]	Population dynamics	C++	Não
SML	[Maxwell and Costanza 1997]	Landscape	any	STELLA
MickL	[Reed et al. 1999]	Hidrology	MickL	Canvas
IMT	[Villa 2001]	Ecology	any	Não
SELES	[Fall and Fall 2001]	Landscape	SELES	Canvas
FarmSim	[Good 2005]	Agronomy	C#	End-user
JAMS	[Kralisch and Krause 2006]	Hidrology	Java	Não
CMP	[Moore et al. 2007]	Agro-environment	any	Não
Ocelet	[Degenne et al. 2009]	Landscape	Ocelet	Eclipse
OMS3	[David et al. 2012]	Hidrology	Java	Não
FlexSem	[Larsen et al. 2013]	Estuaries	XML	Não
UniSim	[Holst 2013]	Ecology	C++	Não

A partir desta revisão constata-se que diversas pesquisas são realizadas nesta área de modelagem de linguagens e/ou ferramentas que utilizem os conceitos da Ecologia e Hidrologia em geral. Três *softwares* desenvolvidos com aplicabilidade em Ecologia são mencionados na Tabela 2, o MOSES, o IMT (*Integrated Modelling Toolkit*) e o UniSim.

No MOSES [Wenzel 1992], baseou-se em 20 processos ecológicos diferentes, onde seus *scripts* eram lidos por um sistema desenvolvido em Fortran e sem interface gráfica. De acordo com o autor esta linguagem formal permitiu “uma estruturação hierárquica de modelos a partir de modelos parciais autônomos que estão armazenados em um banco modelo”. MOSES continha blocos de construção denominados objetos genéricos, porém este não era orientado a objetos totalmente.

O IMT [Villa 2001] era composto por uma hierarquia de módulos de paisagem de origem diversa e seus *scripts* são escritos em XML. Este, é considerado semelhante ao SML visto que os módulos são definidos como variáveis de estado com fluxos de entrada e saída especificando um modelo de equação diferencial. O autor também destaca que módulos existentes podem interagir com os demais por meio de sua definição e permitindo a cooperação entre estes.

O UniSim [Holst 2013] aplica a metodologia de *design* de *software* ágil nos modelos ecológicos visando uma ferramenta universal de utilização. O código do *Universal Simulator*² é aberto e foi desenvolvido em C++ e XML. A instalação do UniSim³ bem como a descrição detalhada da estrutura deste simulador encontra-se disponível para acesso.

Neste estudo, destacam-se também os modelos que baseiam-se na área Hidrológica, conforme a Tabela 2 citam-se: MickL, JAMS e OMS3. A linguagem MickL [Reed et al. 1999] é semelhante a C, orientada a objetos e utilizada para a codificação de modelos OME (*Open Modelling Engine*).

²<http://http://www.ecolmod.org/index.html>

³http://www.ecolmod.org/unisim_explained-2013-11-25.pdf

Estes modelos são escritos nesta linguagem a partir de equações com parâmetros de entrada e produzem uma saída. Alguns tipos de variáveis (as que começam com letra maiúscula) podem ser lidas e utilizadas por outros modelos, porém outros tipos são usadas somente localmente. Uma simulação, neste contexto, possuem possibilidade de composição com uma hierarquia de ambos os modelos e equações. Uma primeira versão efetuou a leitura e interpretação do código em MickL, entretanto nas posteriores foi utilizado um compilador visto a agilidade deste.

JAMS [Kralisch and Krause 2006] possui uma modelagem de natureza genérica, os blocos são componentes configurados a partir de *scripts* XML fornecendo conceitos para o contexto espacial e temporal. Os componentes são desenvolvidos em JAVA e possuem interface simples composta por três métodos: *init*, *run* e *cleanup*, sendo o método *run* utilizado para a atualização do componente conforme seu contexto.

O OMS3 foi criado por [David et al. 2012] e funciona como um sistema de execução capaz de interpretar modelos configurados em OMS3 e implementado na linguagem Groovy. Um *script* DSL informa ao OMS3 quais são os componentes do modelo e os *links* existentes entre a entrada e a saída do mesmo. No componente do código fonte, as variáveis podem ser declaradas como entradas ou saídas usando @In e anotações @Out, permitindo que estas sejam vinculadas ao código DSL.

Outros trabalhos relacionados a utilização de sistemas multiagentes para a modelagem de processos coletivos, utilizando aprendizagem para o gerenciamento de ecossistemas, através de jogos e papéis foi desenvolvido por [Bousquet et al. 2002a]. Neste trabalho, alguns pontos foram considerados visando a utilização dos SMAs, nos quais destacam-se:

- Simulação da variabilidade dos esquemas de irrigação de terras no Senegal, sendo desenvolvido um jogo com papéis específicos dos agentes interessados no domínio da aplicação;
- No auxílio da biodiversidade em processos de gestão em Madagascar, com um modelo espacial de dinâmica de recursos com aplicabilidade em negociações;
- Para a prevenção de incêndios, no âmbito florestal de uma região francesa, foi desenvolvido um modelo de negociação e estratégias na gestão da terra, entre silvicultores e criadores de gado, buscando desenvolver capacidades de gerenciamento nestes usuários;
- Simulação de cenários de adaptação em ecossistemas naturais, com intuito de modelar as ações individuais de pastores conservacionistas na utilização dos recursos de terras;
- Teste de hipóteses sobre a diferenciação de domicílios no final do período cooperativo, simulando um modelo de organização e coleta de informações; e,
- Facilitar discussões entre a população e os representantes locais, em relação a alocação de terras no Senegal, auxiliando no desenvolvimento de um modelo comum às partes interessadas.

O grupo de pesquisa agrícola e cooperação internacional Cirad⁴, na França, desenvolve pesquisas para o desenvolvimento sustentável das regiões tropicais e mediterrâneas. Diversos trabalhos foram publicados, de acordo com as abordagens mencionadas anteriormente, pelo grupo os quais utilizam técnicas de jogos de papéis RPG (do inglês, *Role-playing game*) e SMA.

Nas simulações desenvolvidas pelo grupo, além da utilização em estudo de casos reais, pode ser aplicada em laboratórios virtuais, caso a ausência destes seja inerente ao problema. Como por exemplo no trabalho de [Barreteau et al. 2004] onde buscou-se explorar uma modelagem baseada em agentes por meio de simulação de cenários.

Outro trabalho destacado pelo grupo é de [Le et al. 2008], onde os sistemas multiagentes auxiliam também na simulação de dinâmicas espaço-temporais de um sistema humano-paisagem, sendo desenvolvido um simulador dinâmico de uso terrestre chamado LUDAS. O *framework* LUDAS foi

⁴<https://www.cirad.fr/en>

aplicado em uma bacia hidrográfica da montanha no centro do Vietnã, buscando auxiliar neste processo de políticas para uso de terras. O objetivo principal do estudo é avaliar os impactos relativos as intervenções políticas, considerando a utilização das terras, nesta sociedade [Le et al. 2008].

A metodologia ComMod (*Companion Modeling*) baseia-se em uma abordagem de gestão participativa por meio de jogos de papéis e SMA. Esta abordagem é utilizada nos trabalhos do grupo, assim como em [Barnaud et al. 2010], visando estimular a negociação e o aprendizado coletivo sobre a utilização da água entre os moradores da região. O principal objetivo da pesquisa é a construção de modelos de simulação integrando as diferentes visões dos atores envolvidos no processo.

Esta metodologia mostra-se eficaz para este tipo de simulação de cenário visto que o grupo Cirad é uma das referências mundiais na busca por solução de problemas que envolvem recursos naturais e seres humanos. A integração das duas técnicas RPG e SMA apresenta resultados satisfatórios no âmbito destas pesquisas.

Neste contexto, inúmeras pesquisas estão sendo desenvolvidas integrando as áreas e metodologias mencionadas. O foco deste estudo, o qual culminará na proposta de tese de doutorado, vislumbra focar na área Ecológica e Hidrológica, agregando assim contribuição científica neste contexto.

6. Considerações Finais

O presente trabalho apresentou uma pesquisa na área de IA, sistemas multiagentes, especificação do sistema de tipos (TPO) e da linguagem (SML) e trabalhos relacionados a linguagem de domínio ecológico.

O estudo sobre um novo sistema de tipos e uma linguagem descritiva para sociedade de agentes torna-se importante neste contexto, visto que é notável a consolidação de diversas pesquisas que utilizam o desenvolvimento de sistemas multiagentes. Com a utilização de SMA, muitos trabalhos encontraram soluções satisfatórias, inclusive no âmbito do domínio ecológico, porém há um longo caminho que pode ser explorado sob esta perspectiva.

Atualmente, a utilização de sistemas multiagentes no contexto ecológico possui grande destaque, visto que os agentes e, por consequência, os SMAs possuem grande adaptabilidade à qualquer área do conhecimento. Sendo assim, é possível modelar e simular estes cenários nesta abordagem. Bem como, buscar soluções de determinado domínio aplicáveis a estudo de caso específicos. Os SMA também são vistos como sistemas inteligentes e possuem uma variedade de técnicas que auxiliam na aquisição do conhecimento. Dentre as técnicas comumente utilizadas em sistemas inteligentes ou especialistas destacam-se: aprendizado de máquina, algoritmos genéticos, redes neurais, mineração de dados, aprendizado a partir de jogos, etc.

Para a continuidade deste trabalho vislumbra-se algumas possibilidades de direcionamento desta pesquisa, dentre elas: a integração destas abordagens a um cenário específico para a modelagem de sistemas ecológicos, por exemplo, na parte de recursos hídricos, para simulação e modelagem de sistemas de apoio às pesquisas em bacias hidrográficas. Neste caso, a sociedade de agentes (SMA) deve ser conhecida, então uma modelagem detalhada dos atores do sistema torna-se necessária.

O estudo de caso envolve a bacia hidrográfica Mirim-São Gonçalo localizada no sudeste do Rio Grande do Sul a qual abrange divisa com o Uruguai e fazem parte desta 20 municípios da região Sul. O principal objetivo é a utilização do recurso hídrico nesta bacia por seus atores.

Outro fator que pode ser explorado, a partir desta modelagem mencionada, é inserir a conceitualização formal de sociedade de agentes com TPO e SML, ao menos em parte da modelagem ou em regiões da bacia previamente estabelecidas. Além disso, pode-se utilizar o formalismo de sociedade de agentes em algumas estruturas dos cenários ou alguns atores, como os órgãos de defesa do meio

ambiente e suas regras, ou em outros grupos de atores os quais que podem ser definidos no decorrer deste estudo.

Referências

- Alvares, L. O. and Sichman, J. S. (1997). Introdução aos sistemas multiagentes. In *XVII Congresso da SBC-Anais JAI'97*.
- Barnaud, C., Van Paassen, A., Trebuil, G., Promburom, T., and Bousquet, F. (2010). Dealing with power games in a companion modelling process: lessons from community water management in thailand highlands. *Journal of agricultural education and extension*, 16(1):55–74.
- Barreteau, O., Bousquet, F., Millier, C., and Weber, J. (2004). Suitability of multi-agent simulations to study irrigated system viability: application to case studies in the senegal river valley. *Agricultural Systems*, 80(3):255–275.
- Bordini, R. H., Vieira, R., and Moreira, A. F. (2001). Fundamentos de sistemas multiagentes. In *Anais do XXI Congresso da Sociedade Brasileira de Computação (SBC2001)*, volume 2, pages 3–41.
- Bousquet, F., Barreteau, O., d'Aquino, P., Etienne, M., Boissau, S., Aubert, S., Le Page, C., Babin, D., and Castella, J.-C. (2002a). Multi-agent systems and role games: collective learning processes for ecosystem management. *Complexity and ecosystem management: The theory and practice of multi-agent systems*, pages 248–285.
- Bousquet, F. and Le Page, C. (2004). Multi-agent simulations and ecosystem management: a review. *Ecological modelling*, 176(3-4):313–332.
- Bousquet, F., Le Page, C., and Müller, J.-P. (2002b). Modélisation et simulation multi-agent. *deuxiemes assises du GDRI3*.
- Coppin, B. (2010). *Inteligência Artificial*. Rio de Janeiro: LTC, 3a edition.
- Costa, A. C. d. R. and Demazeau, Y. (1996). Toward a formal model of multi-agent systems with dynamic organizations. In *Proceedings of 2nd. International Conference on Multi-agent Systems, ICMAS*, volume 96, page 431.
- Costa, A. C. R. (2016). Symbolic environments and the cultural aspects of augmented worlds. In *Workshop on Agents Living in Augmented Worlds - ALAW@ECAI 2016*. www.ResearchGate.net.
- Costa, A. C. R. (2017a). Ecosystems as agent societies, landscapes as multi-societal agent systems. In *Multi-Agent-Based Simulations Applied to Biological and Environmental Systems*, pages 25–43. IGI Global.
- Costa, A. C. R. (2017b). Sml - a society modeling language. In *Anais do XI Workshop-Escola de Sistemas de Agentes, seus Ambientes e aplicações (WESAAC 2017)*, São Paulo.
- Costa, A. C. R. (2018). Elementary economic systems in material agent societies. In *Anais do XII Workshop-Escola de Sistemas de Agentes, seus Ambientes e aplicações (WESAAC 2018)*, Fortaleza/CE.
- Costa, A. C. R. and Dimuro, G. P. (2009). A minimal dynamical mas organization model. In *Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models*, pages 419–445. IGI Global.
- David, O., Lloyd, W., Ascough II, J. C., Green, T. R., Olson, K., Leavesley, G., and Carlson, J. R. (2012). *Domain specific languages for modeling and simulation: use case OMS3*. PhD thesis, International Environmental Modelling and Software Society (iEMSs).

- Degenne, P., Seen, D. L., Parigot, D., Forax, R., Tran, A., Lahcen, A. A., Curé, O., and Jeansoulin, R. (2009). Design of a domain specific language for modelling processes in landscapes. *Ecological Modelling*, 220(24):3527–3535.
- Demazeau, Y. and Costa, A. C. R. (1996). Populations and organizations in open multi-agent systems. In *Proceedings of the 1st National Symposium on Parallel and Distributed AI*, pages 1–13.
- Deursen, A. V., Klint, P., and Visser, J. (2000). Domain-specific languages: An annotated bibliography. *ACM SIGPLAN NOTICES*, 35:26–36.
- Drogoul, A. (1994). La mémoire collective dans les sma réactifs: l'exemple de manta. *Mémoire Collective, Dialogue Entre les sma Artificiels et les Sciences Humaines et Biologiques. Centre de Recherche de Royallieu, Compiègne*.
- Fall, A. and Fall, J. (2001). A domain-specific language for models of landscape dynamics. *Ecological modelling*, 141(1-3):1–18.
- Finin, T., Fritzson, R., McKay, D., and McEntire, R. (1994). Kqml as an agent communication language. In *Proceedings of the third international conference on Information and knowledge management*, pages 456–463. ACM.
- Fipa, A. (2002). Fipa acl message structure specification. *Foundation for Intelligent Physical Agents*, <http://www.fipa.org/specs/fipa00061/SC00061G.html> (30.6. 2004).
- Fowler, M. (2010). *Domain-Specific Languages (Addison-Wesley Signature Series (Fowler))*. Addison-Wesley Professional.
- Ginot, V., Le Page, C., and Souissi, S. (2002). A multi-agents architecture to enhance end-user individual-based modelling. *Ecological modelling*, 157(1):23–41.
- Good, J. (2005). The benefits and practicalities of using extensible markup language (xml) for the interfacing and control of object-oriented simulations.
- Holst, N. (2013). A universal simulator for ecological models. *Ecological informatics*, 13:70–76.
- Holst, N. and Belete, G. F. (2015). Domain-specific languages for ecological modelling. *Ecological informatics*, 27:26–38.
- Hübner, J. F. (2003). *Um modelo de reorganização de sistemas multiagentes*. PhD thesis, Universidade de São Paulo.
- Hübner, J. F., Bordini, R. H., and Vieira, R. (2004). Introdução ao desenvolvimento de sistemas multiagentes com jason. *XII Escola de Informática da SBC*, 2:51–89.
- Hübner, J. F. and Sichman, J. S. (2003). Organização de sistemas multiagentes. *III Jornada de Mini-Cursos de Inteligência Artificial (JAIA'03)*, 8:247–296.
- INTERCHANGE, K. (1998). The darpa knowledge sharing effort: Progress report. *Readings in agents*, page 243.
- Kralisch, S. and Krause, P. (2006). Jams—a framework for natural resource model development and application.
- Kreft, J.-U., Booth, G., and Wimpenny, J. W. (1998). Bacsim, a simulator for individual-based modelling of bacterial colony growth. *Microbiology*, 144(12):3275–3287.
- Labrou, Y. and Finin, T. (1997). A proposal for a new kqml specification. Technical report, Technical Report Technical Report TR-CS-97-03, University of Maryland Baltimore County.
- Larsen, J., Mohn, C., and Timmermann, K. (2013). A novel model approach to bridge the gap between box models and classic 3d models in estuarine systems. *Ecological modelling*, 266:19–29.

- Le, Q. B., Park, S. J., Vlek, P. L., and Cremers, A. B. (2008). Land-use dynamic simulator (ludas): A multi-agent system model for simulating spatio-temporal dynamics of coupled human–landscape system. i. structure and theoretical specification. *Ecological Informatics*, 3(2):135–153.
- Luger, G. F. (2008). *Artificial intelligence : structures and strategies for complex problem solving*. Pearson Education India, 6a edition.
- Maxwell, T. and Costanza, R. (1997). A language for modular spatio-temporal simulation. *Ecological modelling*, 103(2-3):105–113.
- Mernik, M. (2012). *Formal and Practical Aspects of Domain-Specific Languages: Recent Developments: Recent Developments*. IGI Global.
- Moore, A., Holzworth, D., Herrmann, N., Huth, N., and Robertson, M. (2007). The common modelling protocol: A hierarchical framework for simulation of agricultural and environmental systems. *Agricultural Systems*, 95(1-3):37–48.
- Nwana, H. S. (1996). Software agents: An overview. *The knowledge engineering review*, 11(3):205–244.
- Odum, H. T. and Odum, B. (2003). Concepts and methods of ecological engineering. *Ecological Engineering*, 20(5):339–361.
- Reed, M., Cuddy, S. M., and Rizzoli, A. E. (1999). A framework for modelling multiple resource management issues—an open modelling approach. *Environmental Modelling & Software*, 14(6):503–509.
- Rezende, S. O. (2003). *Sistemas inteligentes: fundamentos e aplicações*. Editora Manole Ltda.
- Russell, S. and Norvig, P. (2013). *Inteligência Artificial*. Campus, 3a edition.
- Villa, F. (2001). Integrating modelling architecture: a declarative framework for multi-paradigm, multi-scale ecological modelling. *Ecological modelling*, 137(1):23–42.
- Wenzel, V. (1992). Semantics and syntax elements of a unique calculus for modelling of complex ecological systems. *Ecological Modelling*, 63(1-4):113–131.
- Wooldridge, M. (2002). An introduction to multi agent systems, department of computer science, university of liverpool, uk.