



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

**Modelagem e Simulação Baseada em Agentes
Aplicada à Gestão de Recursos Hídricos na Bacia de
Samambaia**

Guido Dutra de Oliveira

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Orientador

Prof. Dr. José Edil Guimarães de Medeiros

Coorientadora

Prof.a Dr.a Célia Ghedini Ralha

Brasília
2019

Dedicatória

Dedico este trabalho aos colegas e professores que muito me ensinaram ao longo do curso, e a meus amigos e família, pelo seu apoio, carinho e paciência.

Agradecimentos

Agradeço às pessoas que me ajudaram ao longo da sua elaboração. Em particular, agradeço: à professora Célia, pelo constante apoio, orientação, e carinho; ao Pedro, pelo ativo envolvimento, pelas reuniões, e pelas diversas dúvidas sanadas ao longo do projeto; ao professor Queiroz, pela ajuda na elaboração da tese e na busca de um orientador; ao Cássio, pelas sugestões e indicações de plataformas e fontes de dados; à professora Conceição, por participar nas reuniões apesar da distância; ao professor Edil, pelo auxílio na orientação, matrícula e participação na banca.

Resumo

A Modelagem e Simulação Baseada em Agentes (MSBA) pode ser aplicada como um instrumento para a resolução de conflitos socioambientais, provendo maneiras de explorar cenários alternativos, fazer previsões e promover maior entendimento do relacionamento entre as partes envolvidas. Tais conflitos são cada vez mais comuns, particularmente no contexto da agricultura no Brasil. Dentre estes, têm grande importância aqueles relacionados à gestão de recursos hídricos. Este trabalho dá continuidade a um projeto anterior de mestrado, o qual aplica a MSBA para analisar a gestão e o uso dos recursos hídricos para produção agrícola na bacia do rio Samambia, no município de Cristalina, GO. Esta é uma região com alta demanda de água para irrigação e serve como um exemplo típico de um cenário de conflito no uso de água. A contribuição deste trabalho tem três objetivos principais: (i) reimplementar o modelo a partir de metodologias de engenharia de software mais rigorosas e de um estudo aprofundado da plataforma GAMA (na qual o modelo inicial foi implementado), de forma a gerar um modelo mais robusto, genérico e manutenível; (ii) atualizar partes da lógica do modelo, visando maior veracidade; (iii) determinar parâmetros de validação e sugerir os próximos passos para a aplicabilidade do modelo em cenários reais.

Palavras-chave: modelagem e simulação baseada em agentes, sistemas socioambientais, gestão de recursos hídricos, agricultura.

Abstract

Agent Based Modeling and Simulation (ABMS) can be used as an instrument in resolving socioenvironmental conflicts, providing ways to explore alternative scenarios, make predictions and promote understanding on the relations within the system. Such conflicts are ever more common, especially in the context of agriculture in Brazil. Among them, particularly important are those related to water resources management. This work is a continuation of a previous master's project, in which ABMS is applied to analyse the water management and usage by agriculture at Samambaia river's basin, in the municipality of Cristalina, in the State of Goiás. This is a region with high water demand for irrigation, and a typical example of a water usage conflict scenario. This work's contribution to the project has three main objectives: (i) reimplement the model based on more rigorous software engineering methodologies and on a deeper understanding of the GAMA platform (in which the original model was developed), aiming for a more robust, generic, and maintainable model; (ii) update certain parts of the model's logic, aiming for higher correctness; (iii) determine validation parameters and suggest next steps aiming for the applicability of the model in real life scenarios.

Keywords: Agent Based Modeling and Simulation, socioenvironmental systems, water resource management, agriculture.

Sumário

1	Introdução	1
2	Fundamentação Teórica	3
2.1	Modelagem e simulação baseada em agentes	3
2.2	Plataformas de Modelagem e simulação baseada em agentes	5
2.2.1	CORMAS	5
2.2.2	NetLogo	5
2.2.3	MAELIA	6
2.3	Plataforma GAMA	6
2.3.1	Justificativa da adoção da plataforma GAMA	6
2.3.2	A linguagem GAML	7
2.3.3	Meta-modelo da linguagem GAML	9
2.4	Documentação de MSBA	9
2.4.1	UML	9
2.4.2	ODD	10
3	Proposta	12
3.1	Modelo anterior do projeto	12
3.1.1	A bacia do Rio Samambaia	12
3.1.2	O modelo	14
3.1.3	Avaliação do modelo	17
3.2	Redefinição do modelo	18
3.2.1	Nomenclatura adotada	19
3.2.2	Modularização	19
3.3	Mudanças no modelo	20
3.3.1	Fazendeiros e escolha das culturas	20
3.3.2	Cálculo de chuva	23
3.3.3	Estimativa de lucro	24
3.3.4	Interface gráfica	24

3.3.5	Dados de entrada	24
3.4	Validação	25
4	Experimentos	27
4.1	Implementação do Novo Modelo	27
4.1.1	Condições ambientais	27
4.1.2	Lógica de plantio	27
4.1.3	Estimativa de lucro	29
4.2	Atualização dos displays	29
4.2.1	Mapa da bacia	30
4.2.2	Dados de cultivo	30
4.2.3	Demanda de água	31
4.2.4	Dados diários	31
4.2.5	Patrimônio	31
4.3	Comparação dos modelos	31
5	Conclusões	39
5.1	Visões sobre o problema e avaliação do modelo	40
5.2	Percepções sobre a plataforma GAMA	41
5.3	Possibilidades futuras	42
	Referências	44
	Apêndice	45
	A UML do modelo	46
	B ODD do modelo	47
B.1	Visão geral	47
B.1.1	Propósito	47
B.1.2	Entidades, variáveis de estado, e escalas	47
B.1.3	Processo e cronograma	49
B.2	Conceitos de Projeto	52
B.3	Detalhes	53
B.3.1	Inicialização	53
B.3.2	Dados de entrada	53
B.3.3	Submodelos	54

Lista de Figuras

2.1	Esquema da estrutura da linguagem GAML (Documentação do GAMA) . . .	10
2.2	Tabela de elementos do protocolo ODD	11
3.1	Bacia do Rio Samambaia	13
3.2	Pivôs na bacia de Samambaia	14
3.3	Produtividade de culturas	15
3.4	Disponibilidade de culturas por mês	16
3.5	Esqueleto de uma arquivo de espécie	20
3.6	Estrutura de diretórios do projeto	21
3.7	Espécie <i>Harvest</i>	21
4.1	Comparação - mapas da bacia	34
4.2	Dados de culturas - original	35
4.3	Dados de culturas - novo	35
4.4	Demanda regional de água - original	36
4.5	Demanda total de água - original	36
4.6	Demanda de água - novo	37
4.7	Dados diários - original	37
4.8	Patrimônio gerado - original	38
4.9	Patrimônio gerado - novo	38

Lista de Tabelas

3.1	Pesos dos parâmetros dos fazendeiros	22
4.1	Tempo de execução dos dois modelos (ms)	33

Capítulo 1

Introdução

Conflitos relacionados à gestão de água são recorrentes em diversos contextos socioambientais. Estes conflitos se tornam mais frequentes à medida que o mal uso sistemático deste recurso ao longo dos séculos evidencia seu caráter escasso, não somente em regiões que tradicionalmente padecem de falta de água, mas em nível global. Uma das principais atividades ligadas a estes conflitos é a agricultura, responsável por 68% [1] do uso de água no Brasil.

A Modelagem e Simulação Baseada em Agentes (MSBA) é uma técnica de modelagem computacional que permite uma abordagem mais adequada [2] ao tratamento de sistemas complexos - particularmente de sistemas que envolvem a ação antrópica, como fenômenos socioambientais - quando comparada a técnicas mais tradicionais, tais como Simulação de Eventos Discretos (SED) [3], Dinâmica de sistemas (DS) [4], Simulação em método de Monte Carlo [5], e simulação de sistemas contínuos [6].

O princípio fundamental da MSBA é o foco no *agente* enquanto entidade fundamental, cujo comportamento e interação com outros agentes e com o ambiente leva à emergência das características do sistema como um todo. É portanto uma abordagem *bottom-up*, mais “orgânica” e mais condizente com o que geralmente se observa em sistemas complexos. Por seu grande potencial no estudo de sistemas complexos e na resolução de conflitos (sociais, ambientais, econômicos, etc.), a MSBA tem sido um tópico de alto interesse acadêmico em diversos domínios de conhecimento [2].

Este trabalho aplica MSBA a um cenário típico de interação sócio-ambiental: a gestão de água em uma bacia onde há forte atividade agrícola. O contexto do modelo é o cenário da bacia do rio Samambaia, no município de Cristalina, onde há intensa atividade agrícola, com predomínio da agricultura de larga escala em pivôs centrais, e onde a água está no limiar de se tornar um recurso escasso para os agricultores, o que começa a gerar conflitos com os gestores municipais de água.

O presente trabalho está relacionado à dissertação de mestrado desenvolvida por Pedro

Phelipe Gonçalves Porto no Programa de Pós-Graduação em Tecnologia Ambiental e Recursos Hídricos (PTARH) da UnB, a qual foi orientada pela Professora Conceição de Maria Albuquerque Alves [7].

Este trabalho tem como objetivo principal viabilizar percepções sobre o comportamento de produtores em cenários agrícolas, onde o uso de recursos hídricos é fundamental para o processo produtivo. Particularmente, deseja-se investigar como se dá o processo de escolha de plantio por parte dos produtores, em função dos recursos hídricos disponíveis e como eventuais cenários de escassez de água impactam este processo e a produtividade da área como um todo.

Deseja-se que o modelo construído possa ser utilizado também por gestores e produtores locais, de maneira a trazer a eles maior conhecimento acerca da situação em que estão inseridos e do papel que nela ocupam.

Mais especificamente, será dada continuidade ao projeto nos seguintes aspectos:

- estudo e avaliação da plataforma GAMA [8], na qual foi implementado o modelo;
- análise do modelo definido da dissertação de Pedro Porto [7];
- reimplementação do código fonte, visando maior legibilidade, manutenibilidade e modularidade;
- novas funcionalidades e melhorias na lógica, visando maior fidelidade do modelo quando à sua contraparte real e maior aplicabilidade;
- determinação de parâmetros de validação do sistema, de forma a procurar avaliar sua aplicabilidade e justificar sua eventual continuação.

Este trabalho está organizado da seguinte maneira: no Capítulo 2 serão apresentados os fundamentos teóricos relacionados ao trabalho, incluindo plataformas de desenvolvimento e padrões de documentação de MSBA. No Capítulo 3 o trabalho precedente será contextualizado, e serão detalhadas as propostas de melhorias a serem implementadas. No Capítulo 4 os experimentos realizados para validar o modelo serão apresentados, com comparações com o modelo anterior. No Capítulo 5 as conclusões serão apresentadas, incluindo sugestões de trabalhos futuros.

O código do modelo está disponível na página do grupo de pesquisa *InfoKnow* no gitlab: <https://gitlab.com/InfoKnow/MASE/guidodutra-samambaiabasinabms> e na página do autor no github: <https://github.com/guidoDutra/SamambaiaBasinABMS>.

Capítulo 2

Fundamentação Teórica

2.1 Modelagem e simulação baseada em agentes

A característica fundamental da MSBA é o foco no agente como unidade fundamental da modelagem. A interação do agente com outros agentes e com o ambiente é o fenômeno fundamental a partir do qual se funda a modelagem. Esta abordagem é adequada para incluir diversos fenômenos típicos de métodos tradicionais de modelagem. No entanto, MSBA permite um leque maior de aplicações, com ênfase em fenômenos sociais e socioambientais, que tradicionalmente são de difícil manuseio por meio de tais métodos.

O surgimento da MSBA implica em dois aspectos de grande interesse: uma ferramenta de tomada de decisão mais robusta e melhor embasada para fenômenos do mundo real; um instrumento de pesquisa científica que oferece um método alternativo às tradicionais indução e dedução.

Esse dupla utilidade - social e científica - levou à popularização da MSBA, e à proliferação de comunidades de pesquisa na área. Podemos citar, entre as áreas de aplicação de tais pesquisas: administração, computação, economia, geografia, gestão ambiental, psicologia, entre outras.

Macal [2] apresenta quatro definições de MSBA, em nível crescente de completude ou complexidade, conforme segue:

1. **individual:** os agentes são representados individualmente e possuem características diversas;
2. **autônomo:** os agentes têm comportamentos internos que lhes conferem autonomia, habilidade de detectar condições no modelo, e de reagir adequadamente ao ambiente;
3. **interativo:** os agentes interagem entre si e com o ambiente onde se inserem;

4. **adaptativo:** os agentes interativos e autônomos mudam de comportamento durante a simulação, ao aprender, encontrar novas situações, ou quando populações se ajustam de forma a incluir maior proporção de indivíduos adaptados.

De acordo com a classificação acima, o modelo deste trabalho pode ser considerado um MSBA *autônomo*, pois os agentes têm comportamentos internos, e tomam decisões baseadas no estado do ambiente. Contudo, não há protocolo de interação entre os agentes, nem interferência destes no ambiente.

As definições acima são úteis para situar e comunicar um modelo em termos das noções fundamentais de individualidade, autonomia, interatividade e adaptabilidade. Nota-se que as noções fundamentadoras do modelo são todas relativas aos agentes. São citadas também algumas áreas de pesquisa de grande potencial no desenvolvimento de MSBA:

- **Modelagem de comportamento** - melhores representações de comportamento de agentes e métodos para popular modelos;
- **Análise de simulações** - métodos analíticos e estatísticos para extrair significado do resultado de simulações;
- **Modelagem híbrida** - uso de outros métodos de modelagem em conjunto com MSBA, de forma a aproveitar as vantagens de diferentes métodos e preencher lacunas na modelagem;
- **MSBA em larga escala** - modelos com milhões ou bilhões de agentes, em escala global, de maneira distribuída.

Segundo Macal, a MSBA é caracterizada por uma discrepância entre a enorme extensão de aplicação e o pouco entendimento acerca de sua própria natureza (características essenciais, métodos de aplicação, relacionamento com demais técnicas de modelagem, etc).

O autor cita pontos que considera empecilhos para a adoção generalizada de MSBA. A saber:

- **credibilidade** - devido ao caráter sistêmico e complexo de MSBA, em comparação com métodos mais tradicionais. São necessários argumentos que melhor justifiquem sua necessidade, valor e benefícios, além de métodos padronizados de validação;
- **transparência** - descrições claras e completas, e uso de protocolos estabelecidos podem ajudar a mitigar esse problema, levando a uma maior reprodutibilidade e padronização de modelos;
- **conhecimento** - deve-se construir um corpo de conhecimento bem definido e programas educacionais na área, de forma a difundir e transmitir o conhecimento existente;

- **usabilidade** - ferramentas de desenvolvimento e interfaces de usuário padronizadas devem continuar sendo desenvolvidas.

2.2 Plataformas de Modelagem e simulação baseada em agentes

Há diversas ferramentas que auxiliam o desenvolvimento de MSBA, desde plataformas completas a extensões de linguagens mais usuais e frameworks. A ferramenta utilizada neste trabalho foi a plataforma GAMA. Nesta seção faremos um sumário de algumas ferramentas alternativas.

2.2.1 CORMAS

CORMAS [9] (*Common-pool Resources and Multi-Agent Systems*) é uma plataforma focada em modelos socioambientais, particularmente modelos de sistemas de recursos renováveis. O CORMAS foi baseado no *Visual Works*, que por sua vez é uma plataforma de desenvolvimento de programas na linguagem *Smalltalk*, uma tradicional linguagem de programação orientada a objetos.

O CORMAS é principalmente utilizado no chamado *Companion Modeling Approach* (ComMod), que consiste no uso do modelo pelas próprias partes interessadas, como uma ferramenta de *Role Playing Game* (Jogo de encenação), visando propiciar aos envolvidos maior entendimento acerca do sistema do qual fazem parte.

Entidades do CORMAS são implementadas como classes em *smalltalk*, o que facilita o desenvolvimento para quem é familiarizado com esta linguagem, ou com outras linguagens orientadas a objetos.

2.2.2 NetLogo

A plataforma NetLogo [10] foi desenvolvida nas linguagens Scala e Java por Uri Wilensky no *Center for Connected Learning (CCL) and Computer-Based Modeling* da *Northwestern University*, Illinois, EUA. O CCL é filiado ao *Learning Sciences Department*, *Computer Science Department* e *The Northwestern Institute on Complex Systems*.

O NetLogo pode ser baixado e utilizado de forma gratuita e é uma das plataformas mais populares e mais antigas de MSBA. Tem um escopo de aplicação mais genérico, e estão incluídas na plataforma bibliotecas de modelos para uma vasta gama de domínios. Em particular, a plataforma é muito utilizada em contexto educativo e acadêmico.

2.2.3 MAELIA

O MAELIA surgiu a partir da plataforma GAMA, e tem um escopo mais reduzido, restrito a modelos de agricultura e gestão de recursos naturais. O MAELIA “provê uma arquitetura de software para abordar questões sobre a interação entre atividade agrícolas, a dinâmica das paisagens agrícolas, e a gestão de recursos naturais” [11].

2.3 Plataforma GAMA

A plataforma GAMA [8] é um Ambiente de Desenvolvimento Integrado (Integrated Development Environment - IDE) para construção de modelos baseados em agentes, especialmente focado em modelagem de sistemas sócio-ambientais. Dentre suas principais funcionalidades cita-se:

- ambiente completo de desenvolvimento e modelagem;
- linguagem especializada para modelagem baseada em agentes (GAML);
- suporte a Sistema de Informação Geográfica (Geographic Information System - GIS);
- suporte a OpenGL¹;
- interface gráfica.

2.3.1 Justificativa da adoção da plataforma GAMA

A plataforma GAMA foi escolhida para o desenvolvimento do projeto pelos seguintes motivos: (i) é um ambiente completo, podendo ser usado tanto na implementação do modelo quanto para executar simulações, gerar arquivos de resultados, entre outras funcionalidades; (ii) possui uma linguagem de alto nível específica para MSBA, de simples apreensão por usuários com pouca experiência de desenvolvimento de software; (iii) possui bom suporte à criação de interfaces gráficas, um requisito importante para o projeto; (iv) é muito popular na área de modelagem socio-ambiental, o que facilita a troca de experiências com outros pesquisadores.

Vale ressaltar que a plataforma GAMA utiliza uma abordagem de software livre, disponibilizando o código no repositório de projetos do GitHub, o que viabiliza o desenvolvimento participativo da comunidade de pesquisa de modelagem ambiental baseada em agentes.

¹Open Graphics Library (OpenGL): uma interface de programação de aplicativos (Application Programming Interface - API) que funciona em várias plataformas para renderizar gráficos vetoriais 2D e 3D, normalmente usada para interagir com uma unidade de processamento gráfico (Graphics Processing Unit - GPU) obtendo a renderização acelerada por hardware;

Apesar de algumas limitações da plataforma, que serão abordadas nos próximos capítulos, decidiu-se continuar utilizando a plataforma nesta etapa do projeto, visando um estudo mais aprofundado de suas características e aproveitamento dos esforços já realizados.

2.3.2 A linguagem GAML

GAML é a linguagem de desenvolvimento utilizada na plataforma GAMA. Trata-se de uma linguagem *orientada a agentes*, isto é, uma linguagem de alto nível criada especificamente para o desenvolvimento de modelos baseados em agentes, com foco em modelagem ambiental. A linguagem é orientada a objetos com tipagem estática (i.e., os tipos são declarados em tempo de compilação e não se alteram durante todo o processo de execução do código) [12].

A seguir descreve-se alguns conceitos e a estrutura básica da linguagem. Os termos utilizados em GAML serão traduzidos livremente e a tradução será utilizada como padrão no texto, mas o modelo definido em MSBA está na língua inglesa, de modo a facilitar o entendimento da comunidade internacional nas possíveis publicações.

Espécie (*Species*)

Espécie é um conceito fundamental da linguagem GAML. Descreve uma *espécie* de agentes, de maneira análoga a uma *classe* em linguagens orientadas a objetos. Espécies podem ser instanciadas como agentes e podem participar de cadeias de herança, tal como classes.

Uma espécie consiste em uma entidade abstrata que possui conhecimento (estado, variáveis), comportamento (ações e reflexos) e que podem ser instanciadas como agentes físicos. Neste modelo, são exemplos de espécies: *Farmer*, *Field*, *Crop*, entre outras.

Agentes (*Agents*)

Agentes são a peça básica de um modelo baseado em agentes. Em GAML um agente é uma instância de uma espécie e possui um ciclo de vida definido durante o qual interage com o ambiente.

Os agentes mais representativos deste modelo são os produtores (instâncias de *Farmer*), os quais interagem com seu ambiente para determinar qual a melhor cultura em determinada época e estado ambiental, mudando seu comportamento de acordo com fatores externos, como água disponível ou preço das culturas.

Ações (*Actions*)

Ações são análogas a *métodos* em linguagens orientadas a objetos: funções internas a uma espécie, que representam comportamentos de uma entidade. Em GAML ações podem ser invocadas em qualquer contexto, através da diretiva *ask*.

Reflexos (*Reflexes*)

Reflexos são como ações, mas em vez de serem invocados por agentes, são executados periodicamente. Um reflexo pode ser executado a cada ciclo de simulação ou quando certa condição for verdadeira, usando a diretiva *when*.

Modelo (*Model*)

O modelo é a raiz de um projeto de MSBA. Nele está definida a espécie global, que representa o mundo da simulação, e métodos auxiliares para o progresso da simulação. Um modelo pode estar associado a vários experimentos.

Espécie global

Dentro de cada modelo é definida uma espécie especial chamada *global*. Esta representa o mundo da simulação, e nela são definidas suas características (variáveis globais) e as entidades que o habitam (agentes).

Experimentos (*experiment*)

Um experimento é uma especificação de como executar a simulação. Em um experimento estão contidos parâmetros de simulação e definições de output (gráficos, animações, mapas, etc.). É possível, por exemplo, ter um experimento para simulação em interface gráfica, onde o intuito é permitir ao usuário interagir com o modelo em tempo real, e um experimento em batch, que pode ser executado várias vezes de forma a gerar dados estatísticos ou calibrar parâmetros do modelo.

Displays

Displays são utilizados em experimentos com interface gráfica, para permitir ao usuário visualizar e interagir com a simulação em tempo real. Diversos tipos de displays podem ser criados com facilidade em GAML, inclusive animações 3D baseadas em OpenGL. Usos comuns para displays são mapas, gráficos e animações de agentes.

Outras Funcionalidades

A linguagem GAML apresenta, além das funcionalidades descritas acima, outras que facilitam o desenvolvimento, tais como *tipos complexos* que implementam diversas estruturas de dados (e.g. *container*, *geometry*, *rgb*, *topology*), e uma grande quantidade de operadores especializados (e.g. *closest_to*, operador que retorna o agente mais próximo a determinado agente, dentre uma lista de agentes). Para mais informações veja as sessões correspondentes na documentação da plataforma GAMA para tipos [13] e operadores [14].

2.3.3 Meta-modelo da linguagem GAML

A Figura 2.1 representa a estrutura conceitual da linguagem GAML. Nesta figura, observa-se que o tipo primordial é a espécie abstrata *agent*. Esta não deve ser confundida com uma instância de uma espécie qualquer, mas pode ser entendida como análoga ao tipo *Object*, na Linguagem C#, por exemplo. Deste tipo derivam as espécies abstratas *model* e *experiment*, assim como qualquer espécie definida pelo usuário.

De *model* derivam as espécies que definem um modelo, e de *experiment* as que definem um experimento. As demais espécies definidas pelo usuário têm acesso à instância do modelo como *world*, e os experimentos a conhecem como uma *simulation*.

Resumindo, de acordo com a documentação do GAMA: “um modelo em GAML consiste em uma espécie herdeira de *model*, a qual está populada por entidades de espécies que herdam (direta ou indiretamente) de *agente*, e que está por sua vez abrigada em um ou mais *experimentos*, dentre os quais o usuário pode escolher qual deseja executar”.

2.4 Documentação de MSBA

O modelo foi documentado usando duas técnicas comumente aplicadas em projetos de MSBA: UML e ODD.

2.4.1 UML

UML [15] (*Universal Modeling Language*), atualmente mantida pelo *Object Management Group* é uma linguagem de modelagem utilizada para descrever e visualizar o design de um sistema genérico. Existem diversos tipos de diagrama em UML. Um destes é o diagrama de classes, o qual pode ser utilizado no contexto de MSBA para ilustrar, de maneira simples e intuitiva, quais são os agentes, como estão estruturados internamente, e quais as relações entre eles. O diagrama de classes UML deste trabalho está no Apêndice A.

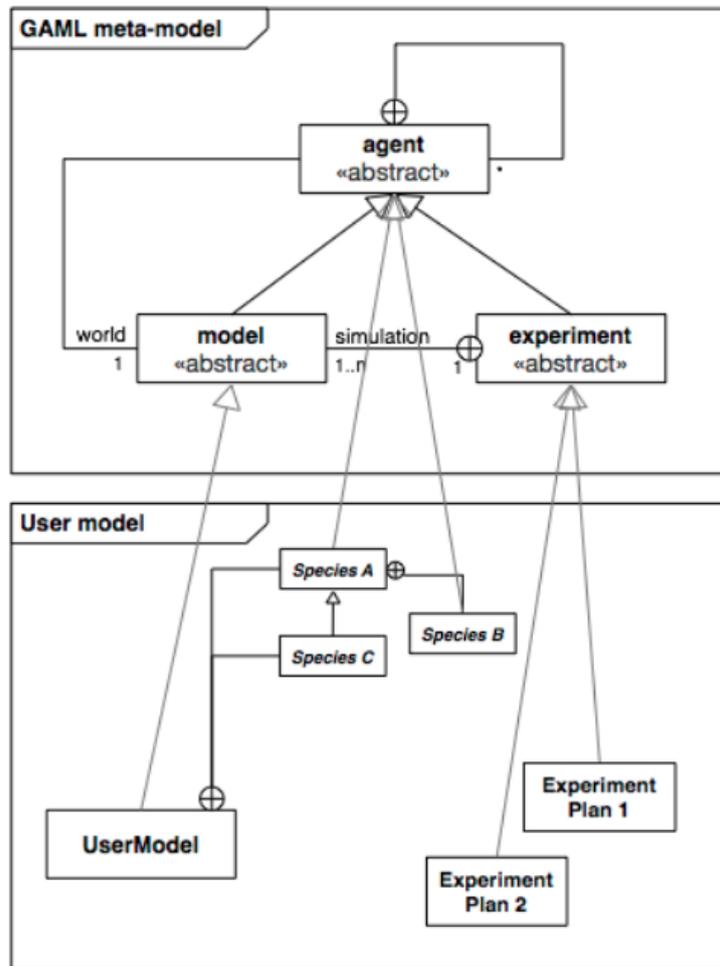


Figura 2.1: Esquema da estrutura da linguagem GAML (Documentação do GAMA)

2.4.2 ODD

O protocolo ODD (*Overview, Design concepts and Details*), definido por Grimm et al em 2006 [16], foi incorporado à literatura de MSBA com o intuito de padronizar a descrição de modelos e facilitar a comunicação e reprodução destes. Baseia-se na descrição do modelo em sete elementos, que pertencem a três categorias, como descrito na Figura 2.2 [8].

A descrição do modelo usando o protocolo ODD, de acordo com a metodologia proposta na revisão de 2010 por Grimm et al [17], pode ser vista no Apêndice B. Os nomes dos tópicos foram traduzidos livremente.

Elements of the ODD protocol	
Overview	1. Purpose
	2. Entities, state variables, and scales
	3. Process overview and scheduling
Design concepts	4. Design concepts
	- Basic principles
	- Emergence
	- Adaptation
	- Objectives
	- Learning
	- Prediction
	- Sensing
	- Interaction
	- Stochasticity
- Collectives	
- Observation	
Details	5. Initialization
	6. Input data
	7. Submodels

Figura 2.2: Tabela de elementos do protocolo ODD

Capítulo 3

Proposta

Este trabalho dá continuidade ao trabalho de dissertação de mestrado desenvolvida por Pedro Porto [7], e pretende revisá-lo, ampliar seu escopo e facilitar futuras contribuições, através dos estudos das teorias e ferramentas que o embasam, e dos conhecimentos prévios do autor. Na sequência, descreve-se o desenvolvimento do modelo anterior do projeto, com as respectivas contribuições feitas pelo autor.

3.1 Modelo anterior do projeto

Nesta seção vamos descrever o contexto inicial do projeto e resumir o que foi realizado anteriormente à contribuição do autor.

3.1.1 A bacia do Rio Samambaia

A bacia de Samambaia encontra-se parcialmente no município de Cristalina e parcialmente no Distrito federal, como mostra o mapa da Figura 3.1 [7].

A maior parte da área da bacia de Samambaia está no município de Cristalina, no Leste do estado de Goiás. Tradicionalmente, a economia deste município foi baseada na exploração e comercialização de cristal de rocha. Embora esta atividade ainda seja relevante, hoje os setores responsáveis pela maior parte da economia de Cristalina são a agropecuária e serviços, com esta primeira movimentando em torno de 895 mil dos 2,3 milhões de reais do PIB em 2016, ou seja, quase 40% deste [18].

Esta mudança na economia do município se deu em grande parte por causa do desenvolvimento de Brasília, cuja demanda crescente de alimentos favoreceu o estabelecimento da agropecuária em Cristalina [7].

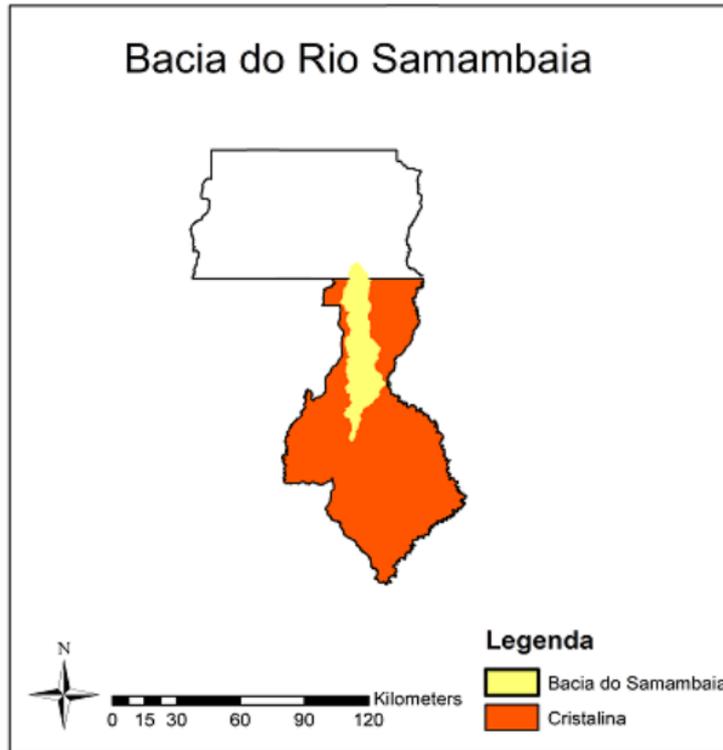


Figura 3.1: Bacia do Rio Samambaia

A bacia de Samambaia é uma região de intensa atividade agrícola no município. De acordo com relatório [19] da ANA (Agência Nacional de Águas) de 2017, a bacia ocupa uma área de 88 mil hectares, dos quais 20 mil são irrigados, e possui 228 pivôs centrais.

A Figura 3.2 indica a localização dos pivôs na bacia, conforme mapeamento por satélite realizado em 2017 pela ANA [7].

O uso de água na bacia de Samambaia está consolidado, sendo o recurso utilizado apenas para irrigação e não há perspectivas de diversificação. No entanto, há pressão por parte dos produtores para o aumento da área irrigada, o que é complicado pelo fato de que a outorga de água na bacia está próxima do limite outorgável, em grande parte devido à outorga de uso da usina hidrelétrica de Batalha [7]. Este fato leva a eventuais conflitos na região.

A bacia do rio Samambaia foi escolhida para modelagem pelos seguintes motivos:

- já existem iniciativas de organização dos usuários (gestores e produtores) juntamente com a ANA para implantar a alocação coletiva da água;
- há uma situação potencial de agravamento de conflitos, dada a tendência crescente de expansão da área irrigada;
- o caráter de sua atividade econômica e os trabalhos prévios realizados simplificam a modelagem.

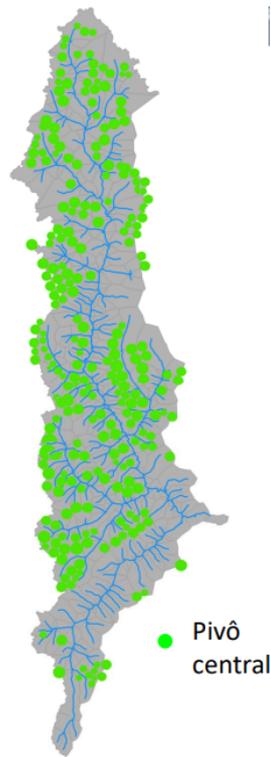


Figura 3.2: Pivôs na bacia de Samambaia

A modelagem, portanto, teve como objetivo principal o estudo da aplicabilidade de MSBA a cenários como o descrito, sendo a bacia de Samambaia um exemplo simples.

3.1.2 O modelo

Para uma descrição detalhada do modelo, ver o ODD do modelo original, que está disponível junto com o código original no CoMSES¹ [20]. Vamos agora resumir sua estrutura básica, de forma a contextualizar as discussões seguintes.

Tempo de simulação

Cada simulação se passa em um período de dez anos, de 2008 a 2017, período ao qual estão associados os dados históricos utilizados na simulação. A simulação tem granularidade diária, isto é, o *step* ou iteração da simulação acontece a cada dia no mundo da simulação.

Dados de entrada

São utilizados os seguintes dados de entrada:

¹CoMSES é uma comunidade aberta dedicada à pesquisa de desenvolvimento de modelos baseados em agentes no contexto de sistemas socioambientais. Funciona também como um repositório online onde é possível armazenar e compartilhar código de modelos.

- shapefiles (arquivos de informação geográfica) da bacia, dos rios e dos pivôs, provenientes da ANA;
- Vazão de água mensal em cada sub-bacia;
- preço mensal das culturas;
- chuva e temperatura mensal.

Além destes, foram utilizados dados de produtividade e tempo de plantio para cada cultura. Estes foram retirados de diversas fontes, sendo a principal os dados do IFAG/-SENAR [21], conforme descrito no ODD do modelo original [20], e foram implementados diretamente no código do modelo. Os dados podem ser vistos nas Figuras 3.3 e 3.4, retiradas da descrição do modelo original [7].

Na figura 3.4, os números representam cada cultura, de acordo com a primeira coluna da figura 3.3. 0 representa nenhuma cultura plantada.

Cultura	Produtividade Máxima (Kg/m ²)	Custo de Produção (R\$/m ²)	Perda de produtividade por falta de água (%)
Soja - farmercrop1	0,42	0,4384	26
Milho - farmercrop2	1,26	0,5119	26
Algodão - farmercrop3	0,45	1,0564	62
Feijão - farmercrop4	0,30	0,4090	20
Batata - farmercrop5	6,00	1,1309	50
Alho - farmercrop6	1,60	5,7652	75
Cebola - farmercrop7	6,00	1,4655	17
Tomate - farmercrop8	9,50	2,2045	88
Trigo - farmercrop9	0,60	0,3505	53

Figura 3.3: Produtividade de culturas

Espécies

Existem seis espécies no modelo, que podem ser divididas entre *ativas* e *passivas*, sendo estas últimas responsáveis apenas por prover informação necessária para a atividade das outras.

Janeiro	fevereiro	Março	Abril	Maio	Junho
0,3,7	0,2,3,4,6,7,8	0,2,4,5,6,8	0,4,8,5,9	0,4,8,9	0,4,8,9
Julho	Agosto	Setembro	Outubro	Novembro	Dezembro
0	0	0,2	0,1,2,3,4	0,1,2,3,4	0,1,3,4

Figura 3.4: Disponibilidade de culturas por mês

As espécies “ativas” são *Farmer* e *CropField*; as “passivas” são *Water*, *Weather*, *Market* e *Calendar*. Segue uma breve descrição de cada uma.

- **Farmer:** Representa um produtor. Sua principal atividade é a escolha da cultura a ser plantado a cada ciclo, visando a maximização do lucro. Possui uma lista de pivôs ou parcelas em uma região contígua do mapa.
- **CropField:** Representa um pivô ou parcela, onde é plantada uma determinada cultura. Utiliza água do rio mais próximo à sua localização na bacia, com consumo determinado por seu tamanho e pela cultura atual.
- **Water:** Representa uma fonte de água dentro da bacia. São consideradas três fontes de água na bacia, cada uma representada como uma espécie filha de *Water*: Samambaia norte, samambaia sul, e córrego do rato. Cada uma possui uma quantidade total de água e determinada vazão.
- **Weather:** Calcula, atualiza e disponibiliza dados relacionados ao clima e chuva ao longo do tempo.
- **Market:** Mantém dados de preço das culturas ao longo do tempo.
- **Calendar:** Dados de disponibilidade mensal das culturas.

Experimentos

Há dois experimentos no modelo:

- um experimento em batch, em que a simulação é executada 100 vezes e os resultados são armazenados em um arquivo csv; e
- um experimento em GUI (interface gráfica), em que o usuário pode ver a simulação em tempo real.

A interface gráfica consiste nos displays listados a seguir. No Capítulo 4 serão apresentadas figuras da simulação em execução e comparações com a versão final do modelo.

- Mapa da bacia e dos pivôs, com indicação de cultura atual em cada pivô.
- Gráficos de plantio: total produzido, área por cultura total; área por cultura imediata.
- Gráficos de demanda regional instantânea de água para as três sub-bacias.
- Gráficos de demanda total de água.
- Gráficos de riqueza total e distribuição entre produtores.

3.1.3 Avaliação do modelo

O modelo, embora funcional, apresenta problemas estruturais. Serão apresentados alguns deles de forma breve antes de propormos as mudanças a serem realizadas.

Design

O paradigma da MSBA é comprometido quando se faz a escolha de representar como espécies aspectos passivos do mundo da simulação. Visto que o fundamento de um agente é justamente sua capacidade de interação e tomada de decisão, é contraintuitivo que existam agentes passivos. Em inspeção mais aprofundada, todas as funções empenhadas pelos agentes passivos do modelo podem ser cumpridas utilizando estruturas de dados mais simples e variáveis globais. Este foi o caminho escolhido para a revisão do modelo.

Deve-se notar, contudo, que a própria estrutura da linguagem GAML desfavorece a implementação de tais estruturas. A linguagem GAML dá um enfoque grande na representação de qualquer estrutura como um agente. Pode-se dizer que a máxima “tudo é um objeto”, muito utilizada em linguagens orientadas a objetos mais tradicionais, se traduz em GAML como “tudo é um agente”. O desenvolvedor deve então decidir até que ponto está disposto a separar objetos tradicionais de agentes. Mais detalhamento sobre esta questão será apresentado na Seção 3.2.

Organização de código

Muitas questões de engenharia de software foram negligenciadas, o que leva a um código de baixa legibilidade e manutenibilidade. Um ponto importante é a falta de modularidade: todo o código do programa está em um único arquivo, o que dificulta o entendimento das relações entre as diferentes partes do programa e torna muito custosa a navegação no código.

Novamente, este é um ponto fraco da plataforma GAMA: todos os exemplos da documentação oficial são projetos de um único arquivo, e não parece haver uma maneira

intuitiva de dividir o código em módulos. Maior detalhamento será apresentado na Seção 3.2.

Outro ponto é a nomenclatura. Nomes de variáveis e funções são geralmente pouco intuitivos, de modo que é necessário recorrer constantemente à documentação ou comentários no código para entender o que representam.

O ponto anterior se torna mais problemático devido à redundância que permeia o código. Valores de variáveis são constantemente copiados de um contexto para outro. Uma porção significativa do código é dedicada a esse tipo de operação, que é computacionalmente custosa e em última análise desnecessária. Esse problema pode ser atribuído a uma confusão sobre o funcionamento do escopo em programas complexos, e pode ser remediado de maneira simples (embora laboriosa) através de refatoração.

Otimização

Algumas escolhas de lógica tem sérios impactos de custo computacional. Embora não haja sinais perceptíveis de baixa performance ao executar a simulação, estes provavelmente se tornariam um empecilho para a escalabilidade do modelo. Optou-se por abordar estes problemas de otimização na revisão do modelo e fazer um teste de performance comparando os dois, para melhor compreender seu impacto. Um ponto de alto custo computacional será abordado na próxima seção.

3.2 Redefinição do modelo

A escolha de redefinir e reimplementar o do modelo foi tomada pelos seguintes motivos:

- Como visto na Seção 3.1.3, o modelo anterior é inadequado do ponto de vista de manutenibilidade, apresentando baixa modularidade, baixa legibilidade, nomenclatura inconsistente, e processos de alta redundância e custo computacional. Por esse motivo, uma refatoração pontual seria extremamente custosa e pouco eficaz.
- O processo de reimplementação foi considerado uma boa oportunidade para realizar um estudo mais profundo da plataforma GAMA, e para estipular boas práticas de organização de código no contexto da linguagem GAML.
- Este processo facilitou também a compreensão, revisão e depuração do código. Várias partes do código foram corrigidas ou otimizadas em decorrência deste processo.

Feita a escolha da reimplementação, foram estipuladas convenções de organização e nomenclatura que serão expostas na sequência.

3.2.1 Nomenclatura adotada

A documentação do GAMA não provê sugestões de nomenclatura. O modelo anterior utilizava uma nomenclatura inconsistente e por vezes incompreensível sem consultar a documentação. Renomeamos a maioria das espécies, funções e variáveis utilizando uma convenção comum em programação orientada a objetos:

- espécies (classes): Camel case começando com maiúscula;
- agentes (objetos): Camel case começando com minúscula;
- ações e Reflexos (Métodos): snake case;
- variáveis: snake case.

Abreviações foram evitadas, preferindo-se uma nomenclatura mais verbosa porém mais compreensível.

3.2.2 Modularização

A documentação do GAMA tampouco aborda modularização. Na realidade a linguagem GAML nos pareceu limitada em relação ao suporte a modularização, o que é corroborado pelo fato de que nenhum dos exemplos de projetos na documentação possui mais de um arquivo fonte. Essa não parece ter sido até o momento uma preocupação para os designers da linguagem.

Todo arquivo .GAML deve obrigatoriamente declarar um modelo (model), o que implica que, por design, não há suporte a arquivos de espécies, que seria o equivalente a arquivos de classes em linguagens orientadas a objetos, o padrão universal em tais linguagens. É possível, porém, declarar arquivos de modelo com apenas uma espécie (ou com espécies que se relacionam por herança) e importá-los em outros arquivos de modelo (outras espécies ou modelo principal). Esta foi a abordagem utilizada. Um exemplo de esqueleto de arquivo de espécie pode ser visto na Figura 3.5

Mecanismos de escopo e namespaces também não parecem ser contemplados pela linguagem. É possível apenas importar arquivos, o que dá acesso a todo o conteúdo do arquivo em escopo global.

Considerando as limitações da linguagem, utilizamos a seguinte estrutura de organização de código:

- diretórios separados para modelos, espécies, experimentos, e dados;
- um arquivo para o modelo principal;
- variáveis globais declaradas no arquivo do modelo principal

```

1 model ExampleSpecies
2
3 import './AnotherSpecies.gaml'
4
5 species ExampleSpecies {
6     float basic_variable;
7     AnotherSpecies speciesReference;
8     // other variable declarations
9
10
11 action some_action {
12     // action body
13 }
14
15 reflex some_reflex when: some_condition {
16     // reflex body
17 }
18 }
19

```

Figura 3.5: Esqueleto de uma arquivo de espécie

- é preciso importar o modelo principal em todos os arquivos de espécies que acessam variáveis globais;
- uma espécie por arquivo
 - espécies herdeiras foram colocadas no mesmo arquivo que seus pais.

As Figuras 3.6 e 3.7 mostram a estrutura de diretórios e um exemplo de arquivo de espécie.

3.3 Mudanças no modelo

Nesta seção serão discutidos alguns aspectos do modelo original e serão expostas as mudanças propostas.

3.3.1 Fazendeiros e escolha das culturas

A organização dos fazendeiros e a lógica de escolha das culturas foram significativamente alterados. No modelo original, os fazendeiros eram divididos em três grupos, de acordo com a quantidade de pivôs em sua posse e com sua localização na bacia. Estes grupos divergiam em relação ao peso dado a três parâmetros, que são critérios utilizados na lógica de decisão de escolha da cultura, após a colheita. São estes: preço de venda das culturas; custo de manutenção das culturas; custo de irrigação. Assumi-se, no modelo antigo, que:

- fazendeiros de grande porte dão o mesmo peso aos três fatores;

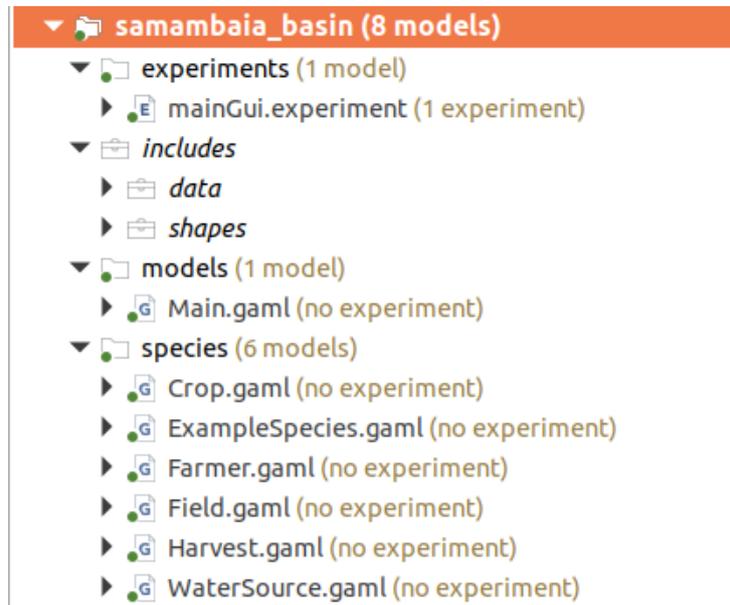


Figura 3.6: Estrutura de diretórios do projeto

```

1 model Harvest
2
3 import '../models/Main.gaml'
4 import './Crop.gaml'
5
6 species Harvest {
7   Crop crop;
8   int current_days <- 0 update: current_days + 1;
9   int current_stage <- 0 update: calculate_current_stage();
10  bool has_changed_stage update: false;
11  float current_water_demand update: evapotranspiration * crop.coefficient[current_stage];
12
13  action calculate_current_stage type: int {
14    int previous_stage <- current_stage;
15    if current_days < crop.stage_change_days[0] {
16      return 0;
17    } else if current_days < crop.stage_change_days[1] {
18      return 1;
19    } else if current_days < crop.stage_change_days[2] {
20      return 2;
21    } else {
22      return 3;
23    }
24    if (previous_stage != current_stage) {
25      has_changed_stage <- true;
26    }
27  }
28 }
29

```

Figura 3.7: Espécie *Harvest*

- fazendeiros de médio porte dão maior peso ao preço de venda, e menor aos outros dois critérios;
- fazendeiros de pequeno porte dão maior peso ao custo de manutenção e irrigação, e menor ao preço de venda.

Os pesos foram fixados de acordo com a Tabela 3.1:

Tabela 3.1: Pesos dos parâmetros dos fazendeiros

	venda	manutenção	irrigação
pequeno	0.5 a 0.8	1.2 a 1.5	1.2 a 1.5
médio	1.2 a 1.5	0.5 a 0.8	0.5 a 0.8
grande	0.8 a 1.2	0.8 a 1.2	0.8 a 1.2

O racional por trás desta escolha é que os produtores de pequeno porte tendem a preocupar-se mais com os custos do que com o retorno dos plantios, os produtores de médio porte têm comportamento oposto e os produtores de grande porte tendem a ser mais equilibrados em suas decisões.

Durante reuniões posteriores em que o modelo foi discutido, a decisão de definir estes parâmetros e atribuir-lhes diferentes pesos para cada grupo de fazendeiros foi levantada, e foi considerada inadequada. Trata-se de uma maneira um pouco artificial de gerar variação de comportamento entre os fazendeiros, que não necessariamente condiz com a realidade. Resolvemos optar por uma abordagem em que não há distinção entre os grupos de fazendeiros, e a variação de comportamento entre estes deriva diretamente das diferenças entre os recursos disponíveis a cada um, de maneira que:

- todos os fazendeiros possuem a mesma estratégia de escolha de culturas;
- esta consiste no cálculo de maximização da renda dentro de um limite de segurança financeira;
- fazendeiros de menor porte têm menos espaço de manobra, em termos de risco financeiro, e naturalmente tendem a procurar estabilidade (diversificar culturas, escolha de culturas de preço estável);
- fazendeiros de grande porte podem utilizar-se de estratégias de maior risco, pois tem mais recursos para gastar.

Com o objetivo de possibilitar a lógica descrita acima e de tornar o modelo mais realista, devem ser adicionados aos dados de preço das culturas os seguintes aspectos:

- variabilidade de preços, obtido a partir de análise de dados históricos;

- escalabilidade (porcentagem do gasto diminuído por hectare plantado), estimada para cada cultura.

Também foi observado nas reuniões e considerado no modelo o fato, analisado durante pesquisas de campo realizadas em momento anterior do projeto, de que fazendeiros de maior porte muitas vezes têm uma demanda mais estável de produtos, por vezes vendendo quantidades fixas a clientes específicos de maneira contínua. Este ponto foi modelado como uma diminuição na variabilidade dos produtos proporcional ao porte do produtor. Considerando variabilidade e escalabilidade de preços, podemos explicar o comportamento observado em campo sem recorrer à segmentação dos fazendeiros em diferentes grupos com comportamentos diferentes.

3.3.2 Cálculo de chuva

Outro ponto levantado foi a influência da chuva no custo de irrigação. Na implementação original possuíamos apenas dados mensais de chuva. Porém, a regularidade da chuva é tão relevante para esta relação quanto a média mensal, pois as culturas necessitam de uma quantidade diária de água, e qualquer quantidade de chuva que exceda este valor é desperdiçada. Ou seja: chuvas bem distribuídas são muito mais benéficas para a cultura, e chuvas muito espaçadas, mesmo que em grande quantidade, têm pouco impacto.

A maneira com que esse problema foi contornado, no modelo original, foi simular uma distribuição de chuvas artificial, da seguinte maneira:

- a cada dia, existe 50% de chance de um evento de chuva;
- a quantidade de chuva pode ir de 0 ao máximo mensal (dado histórico para o mês em questão);
- uma vez atingido o máximo mensal, não há mais chuvas durante aquele mês.

Esta implementação, durante a revisão, foi considerada inapropriada pelos seguintes motivos:

- não são considerados padrões de distribuições naturais de cada época do ano sendo que a única influência do mês na simulação é o total de chuva;
- a escolha de parâmetros e o mecanismo de decisão são arbitrários, e levam a resultados igualmente arbitrários e pouco úteis;
- quando combinado à implementação original da lógica de escolha de cultura, que envolve uma estimativa da chuva nos meses seguintes ao momento do plantio (a computação da chuva é feita para cada dia deste período), causa um grande overhead de processamento, sendo uma abordagem computacionalmente custosa.

Implementamos os seguintes passos para a correção deste problema:

- buscar dados de chuva com maior granularidade, mesmo que seja preciso utilizar dados de uma região maior (ou de uma localidade próxima);
- utilizar uma abordagem analítica em vez de simulada: a partir dos dados armazenar a média e a distribuição mensal, e calcular a influência da chuva considerando que quanto maior a distribuição maior seu impacto.

3.3.3 Estimativa de lucro

Para calcular a estimativa de lucro de cada cultura, durante a escolha de plantio dos produtores, é necessário estimar o uso de água, que por sua vez depende da quantidade de chuva durante o plantio. Para determiná-la, foi utilizada uma abordagem de simulação, i. e. para cada cultura executa-se um processo iterativo que determina o total de chuva, utilizando a mesma abordagem descrita na Seção 3.3.2. Isto implica em um processo extremamente ineficiente, em que uma “mini-simulação” é executada para cada cultivo por cada agricultor. Deve-se substituir esta abordagem por outra em que a chuva durante o período de plantio possa ser estimada matematicamente.

3.3.4 Interface gráfica

É de extrema importância que os gráficos e displays utilizados na simulação interativa sejam claros e informativos, uma vez que são a principal maneira com que os usuários finais (fazendeiros e gestores da região) poderão interagir com o modelo. Implementamos as seguintes melhorias neste sentido:

- melhorar as legendas e títulos dos gráficos;
- remover gráficos redundantes ou pouco informativos;
- melhorar o estilo dos gráficos (cores, layout, etc);
- indicar no mapa a área de cada produtor;
- adicionar controles de parâmetros chave à interface gráfica.

3.3.5 Dados de entrada

Os dados de preço de culturas originais apresentavam lacunas que tiveram que ser estimadas ou buscadas de outras fontes, o que ocasionou inconsistências nos preços, podendo ter alterado significativamente os resultados da simulação.

Os dados de chuva apresentavam baixa granularidade (mensal), o que teve que ser contemplado adotando uma estratégia computacionalmente ineficiente e um pouco arbitrária, como descrito na Seção 3.3.2.

Neste sentido, foram pesquisadas novas fontes de dados nestas duas frentes, objetivando maior consistência, granularidade e padronização. Para tanto, contudo, tivemos que abrir mão da precisão de localização e do tamanho da série histórica, como descrito a seguir.

Para preços de culturas, utilizamos os dados disponíveis no portal da Conab [22]. Por meio da interface de busca do portal, baixamos os dados de cada cultura em formato csv, e então tratamos os dados. Deve-se observar que foi necessário adotar um tempo menor de simulação, pois para datas anteriores a 2016 os dados apresentaram muitas lacunas e baixa precisão. Adotamos, então, um período de 3 anos e 8 meses, de 01/2016 a 09/2019. Como estes dados têm precisão estatal, foi necessário escolher entre os dados do DF ou do estado de Goiás.

Mesmo com a restrição de tempo, as duas séries apresentaram lacunas para alguns dos produtos. Portanto, foi necessário utilizar ambas. Utilizamos dados do DF para batata inglesa e cebola, e dados de GO para as demais culturas. Em análise informal, considerou-se que as séries são suficientemente parecidas para que esta substituição não altere consideravelmente o resultado. Foi necessário também retirar o trigo das culturas contempladas no modelo, uma vez que os dados para esta cultura, em ambas as séries, apresentaram inúmeras lacunas.

Para distribuição de chuva, usamos dados do INMET [23], em procedimento similar: coletamos e tratamos manualmente os dados a partir do portal. A estação de captura mais próxima da bacia é estação de Roncador. Porém, esta apresentava lacunas nos dados. Por isso foram utilizados dados da estação de Brasília. Os dados são capturados diariamente, o que nos dá uma excelente granularidade.

3.4 Validação

Um grande problema do modelo era a falta de parâmetros claros de validação. O caráter explorativo e abrangente do modelo dificultam o levantamento de critérios precisos de validação, mas considerando as demandas dos usuários, discussões em grupo e consultas a pesquisadores parceiros, sugerimos os seguintes critérios:

- consistência qualitativa do comportamento dos produtores na simulação com o comportamento observado no mundo real;

- influência da disponibilidade de água na bacia nas estratégias dos produtores, e diminuição da renda total da região em comparação com cenários de abundância.

Estes são critérios genéricos, mas servem como uma base a partir da qual pode-se estimar se o modelo está condizente com a realidade.

Outro método de analisar a veracidade do modelo é gerar métricas a partir da comparação entre os resultados obtidos na simulação e seus equivalentes coletados no mundo real. Esta análise pode ser realizada em múltiplos períodos da simulação, de modo que podemos analisar o grau de correlação entre o modelo e a realidade ao longo do tempo. Dois parâmetros para os quais tal metodologia seria bem empregada são:

- produção total de cada cultivo e renda total na região; e
- nível de água observado na bacia.

Para que este procedimento gere métricas adequadas é necessário, contudo, que existam dados confiáveis de produção, renda e nível de água na região no período da simulação. Também deve ser considerado que os dados de entrada não são exatamente da região, e podem comprometer a precisão na análise.

Capítulo 4

Experimentos

Neste capítulo, primeiro será detalhada a implementação das principais propostas de mudança no modelo, descritas na Seção 3.3. Depois, será realizada uma análise comparativa do modelo original com o modelo novo, ilustrando as diferenças com imagens da interface da plataforma GAMA durante a execução dos dois modelos. Concluiremos o capítulo com uma discussão sobre as conclusões obtidas a partir desta análise.

4.1 Implementação do Novo Modelo

Segue uma descrição da implementação das mudanças sugeridas na Seção 3.3. Maior detalhamento pode ser visto no ODD do modelo no Apêndice B.

4.1.1 Condições ambientais

Utilizando os dados diários de chuva e temperatura disponíveis no portal da INMET [23], foi possível melhorar a precisão da modelagem do uso de água no modelo. A modificação mais importante foi a remoção do módulo de geração de chuva descrito na Seção 3.3.2, que se tornou obsoleto em decorrência do uso de dados diários de chuva. O valor da evapotranspiração, cujo cálculo envolve temperatura máxima e mínima, também passou a ser atualizado diariamente em vez de mensalmente, o que por sua vez aumentou a precisão do cálculo de demanda de água diária por plantio.

4.1.2 Lógica de plantio

Para tornar a tomada de decisão dos produtores mais verossímil e permitir o plantio simultâneo de múltiplas culturas, como discutido na Seção 3.3.1, foi remodelado o módulo de escolha de plantio conforme descrito a seguir.

Em vez de escolher a cultura cuja estimativa de lucro seja a maior dentre as disponíveis no momento, cada produtor agora considera também a flutuação de preço das culturas, de modo que se esta foi maior que sua tolerância à flutuação, a segunda cultura mais lucrativa será plantada também, até que o produto dos coeficientes de flutuação se torne menor que a tolerância. Estipulou-se um máximo de três culturas simultâneas.

Seguem algumas observações que detalham o funcionamento desta modelagem.

Coeficiente de flutuação

O coeficiente de flutuação é obtido aplicando-se à serie histórica de cada cultura a seguinte fórmula

$$CV = \frac{\sqrt{\sum(x_i - \bar{x})^2/n}}{\bar{x}} \quad (4.1)$$

onde:

CV = coeficiente de variação

x_i = valor individual

\bar{x} = média dos valores

n = número de valores

Este coeficiente não é ideal, pois é caracterizado mais pelas variações anuais de preço do que por flutuações de menor período (diárias e mensais). Contudo, considerou-se satisfatório dado que somente obtivemos acesso a dados mensais.

Tolerância à flutuação

A tolerância à flutuação de preços é a soma de um valor base e de um valor por pivô, de modo que quanto mais pivôs o produtor possuir, maior sua tolerância. Embora um pouco artificial, esta métrica considera a influência da escala de produção nas características da produção, sem que seja preciso lidar com dados de escalabilidade (aumento de produtividade e diminuição de custo por área plantada), aos quais não obtivemos acesso.

Valor da tolerância

Os valores de tolerância base e tolerância por pivô foram estipulados empiricamente e adicionados à interface gráfica como parâmetros, de modo que o usuário pode modificá-los de acordo com suas necessidades. Os valores padrão são tais que agricultores de grande porte costumam plantar uma só cultura, e fazendeiros de médio e pequeno porte costumam plantar duas ou três, dependendo da flutuação dos plantios em questão.

Restrições

Estipulou-se, para simplificar a simulação, que quando há múltiplas culturas simultâneas: (i) o total de pivôs é dividido igualmente entre elas; (ii) só é realizado um novo plantio após a colheita de todas as culturas; (iii) nenhuma das culturas plantadas no último ciclo pode ser plantada novamente.

Estes pontos podem ser revisados futuramente mediante maior análise e entrevistas com os produtores.

4.1.3 Estimativa de lucro

O cálculo da estimativa de lucro depende da estimativa do uso de água, a qual foi originalmente implementada de maneira ineficiente (vide Seção 3.3.3). Com uso dos dados mais precisos de precipitação, pudemos substituir a implementação atual por uma abordagem matemática, como apresentado na Equação 4.2.

A quantidade de água da bacia necessária para a produção depende da chuva durante o período. Esta influência se dá tanto pelo total de chuva quanto pela sua regularidade, como visto na Seção 3.3.2. Para modelar matematicamente este fenômeno, utilizamos um coeficiente de distribuição

$$k = D/2P, k \in [0, 1[\quad (4.2)$$

onde D é o desvio absoluto e P o total da série histórica de precipitação. O coeficiente foi pensado de modo a representar, em um valor de 0 a 1, a variabilidade de chuva, com 0 representando mesma quantidade de chuva todos os dias e 1 representando toda chuva em um único instante. Com esse coeficiente estimamos a influência da chuva no uso de água como

$$P * (1 - k)$$

Nota-se que quanto maior o desvio da série, maior será o coeficiente e menor a influência da chuva. Isto equivale a um cenário em que a maior parte da chuva está concentrada em poucos dias, muito comum nos períodos de seca.

4.2 Atualização dos displays

Foram feitas alterações significativas nos displays da interface de simulação, de modo a permitir maior legibilidade e torná-los mais informativos, além de adaptá-los às mudanças no modelo. Na sequência vamos descrever estas mudanças e ilustrá-las com imagens dos displays antigos e novos. As imagens utilizam o seguinte padrão: (i) à esquerda está o

display antigo, e à direita a versão nova; (ii) todos os prints foram realizados no ciclo (dia) 1320, ou seja, no último dia de simulação de acordo com a implementação nova.

4.2.1 Mapa da bacia

A Figura 4.1 mostra os displays do mapa da bacia. No display novo, as cores de fundo foram alteradas para maior visibilidade do mapa, incluindo a divisão entre sub-bacias. Cada pivô, que no display antigo era preenchido com uma imagem da cultura atual, agora é preenchido com uma cor que representa esta cultura (o índice de cores foi adicionado no canto superior direito) e um número que representa o produtor a quem este pivô pertence, de modo a tornar esta informação disponível ao usuário.

O uso de numeração para representar o produtor não é ideal, mas foi a solução encontrada para contornar o fato de que a área de cada produtor é determinada dinamicamente no começo da simulação, i. e. não é gerada a partir de um arquivo shapefile, como a área dos pivôs (vide a seção de “inicialização” do ODD do modelo no Anexo B). É possível tornar a interface menos poluída e mais funcional por meio do zoom.

Outro ponto foi a alteração do método de renderização de OpenGL para Java2D. OpenGL é recomendado somente para renderizar objetos em 3D, sendo que para figuras bidimensionais deve-se utilizar Java2D para aumento de performance. Pode-se notar que o eixo de três dimensões visto na primeira figura não aparece na segunda.

4.2.2 Dados de cultivo

As Figuras 4.2 e 4.3 apresentam o display com informação sobre culturas. No display original, há três gráficos: (i) área total por plantio; (ii) culturas sendo produzidas no momento (no dia atual da simulação); e (iii) produção total por cultura em quilogramas.

No display novo, os dois primeiros itens foram retirados. (i) foi considerado pouco informativo, uma vez que a produtividade e o tempo de plantio variam para cada colheita, o que torna a área total pouco informativa. (ii) foi movido para um display dedicado a dados diários. Foi adicionado também um gráfico de faturamento total por cultura, o que permite ver quais culturas têm maior impacto econômico na região.

Para aumentar a legibilidade, as métricas foram alteradas, os títulos modificados para português, e legendas foram adicionadas. A ordem dos plantios não foi alterada, de modo que “Crop1” equivale a “Soja” e assim sucessivamente. Porém, “Crop9”, que equivale ao trigo, foi retirado do modelo (vide Seção 3.3.5), e portanto do gráfico.

4.2.3 Demanda de água

No modelo original havia dois displays com gráficos de demanda de água: um deles apresentava dados de disponibilidade de água no dia para cada região e o outro apresentava a demanda em toda a bacia no dia e ao longo do tempo, como visto nas Figuras 4.4 e 4.5, respectivamente. Este esquema foi considerado redundante, por isso optou-se pelo display visto na Figura 4.6, em que há gráficos de demanda de água ao longo do tempo para cada região e para a totalidade da bacia. Um gráfico com disponibilidade diária de água também foi adicionado ao display dedicado a dados diários, como visto na Seção 4.2.4. As métricas e legendas também foram alteradas para maior legibilidade.

4.2.4 Dados diários

Adicionou-se um display dedicado a dados do dia. Nele estão gráficos de parâmetros instantâneos da simulação, úteis somente para análise em tempo real, i. e. durante a execução da simulação. Há dois gráficos neste display: (i) Área utilizada por cultura no dia; e (ii) demanda de água total no dia. Estes gráficos estavam originalmente em displays dedicados a dados de culturas e dados de demanda de água, como visto nas seções 4.2.2 e 4.2.3. Este display pode ser visto na Figura 4.7.

4.2.5 Patrimônio

Neste display pode-se visualizar dados de patrimônio gerado na bacia ao longo do tempo. No display original (Figura 4.8, pode-se ver um gráfico para riqueza total ao longo do tempo, e um para distribuição de riqueza por produtor. No display novo foi adicionado um gráfico para patrimônio gerado por categoria (pequeno, médio ou grande produtor, de acordo com o número de pivôs), e patrimônio médio por produtor em cada categoria. As categorias foram determinadas da seguinte maneira: até 10 pivôs - pequeno; 10 a 30 pivôs - médio; mais de 30 pivôs - grande. Legendas e métricas também foram atualizadas.

4.3 Comparação dos modelos

Nesta seção serão apresentados os resultados, comparando-se o modelo anterior com o modificado.

Demanda de água

Observando as Figuras 4.5 e 4.6, podemos afirmar que a oferta e demanda de água não se alterou significativamente. A disponibilidade de água na bacia permanece muito maior que a demanda, mesmo nos períodos de seca.

Esta característica do modelo mostra-se, portanto, inadequada para capturar a situação real observada em campo, na qual o uso de água pela agricultura está próximo do limite outorgável. Na Seção 5.1 veremos a provável causa desta discrepância, e passos a serem tomados para aliviá-la.

Culturas

Pode-se notar, comparando as Figuras 4.2 e 4.3, que o modelo novo apresenta maior diversidade na escolha de culturas. Isto se deve à mudança da lógica de escolha de plantio descrita na Seção 4.1.2. Porém, percebe-se que a proporção de riqueza gerada por cada cultura é similar em ambos os modelos.

Apesar de haver algumas diferenças significativas entre os dados originais e os dados novos de preço de culturas, esta diferença é minimizada pelo fato de que os dados de produtividade (peso por área) e tempo de plantio, expostos na Seção 3.1.2, foram mantidos iguais.

Observa-se que o tomate têm uma rentabilidade discrepante em relação às outras culturas. Isto pode estar relacionado a um erro nos dados de produtividade para esta cultura, o que é corroborado pelo fato de que, comparando estes dados com os dados de produtividade sem irrigação para esta cultura, obtemos uma diferença de produtividade de 80%, muito maior do que o valor médio obtido para as demais culturas (41%).

Riqueza

A geração de riqueza em ambos os modelos é muito similar, como observado nas Figuras 4.8 e 4.9. No modelo novo a riqueza gerada é um pouco menor, o que pode ser explicado pela necessidade, em alguns casos, de plantar mais de uma cultura, e pelas restrições observadas na Seção 4.1.2.

Análise de desempenho

Para avaliar a performance do modelo, idealmente gostaríamos de obter estatísticas de uso de recursos, principalmente memória e CPU. A documentação da plataforma GAMA não menciona maneiras de obter estes dados, porém é possível fazer uma análise de tempo de execução de qualquer parte da simulação através da diretiva *benchmark*. Em análise informal observamos que a medição de trechos de código apresenta alta variação, de modo

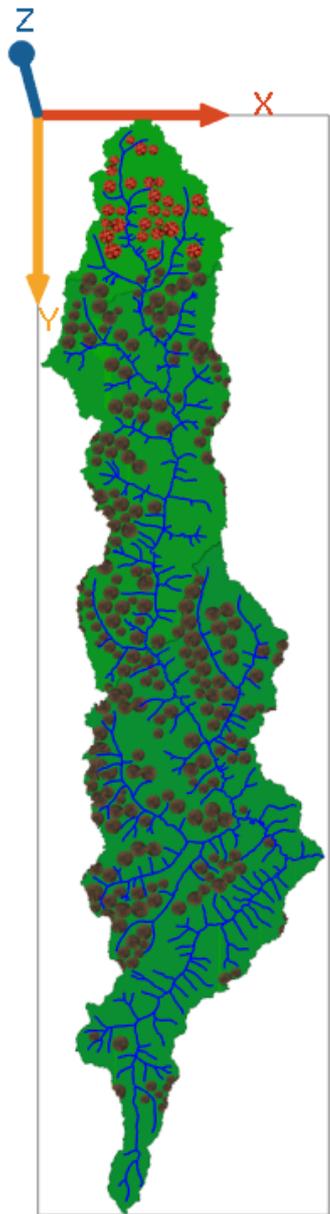
que optamos por fazer um teste de tempo de execução da simulação toda. Primeiramente igualamos o número de ciclos do modelo original ao do novo modelo (1320 ciclos). A média dos resultados obtidos está na Tabela 4.1:

Tabela 4.1: Tempo de execução dos dois modelos (ms)

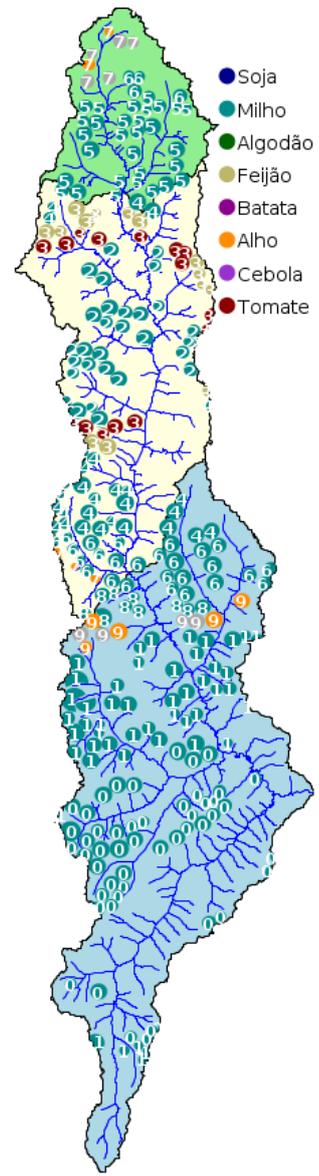
	média do ciclo	duração total
modelo original	54	71465
modelo novo	7	10323

Mesmo com poucos parâmetros de avaliação, podemos observar que a performance do modelo melhorou significativamente, com tempo de execução em torno de sete vezes menor. Testes foram realizados para verificar o impacto na performance da adoção de Java2D em vez de OpenGL para a renderização do mapa, mas esta não foi significativa na máquina do autor. Portanto, acredita-se que essa melhora de desempenho possa ser atribuída à maior eficiência e menor redundância do código do modelo.

Nesse capítulo foram apresentados os resultados experimentais do modelo modificado comparando-o ao modelo anteriormente definido. No Capítulo 5 serão apresentadas algumas conclusões do trabalho realizado.



(a) Mapa da bacia - original



(b) Mapa da bacia - novo

Figura 4.1: Comparação - mapas da bacia

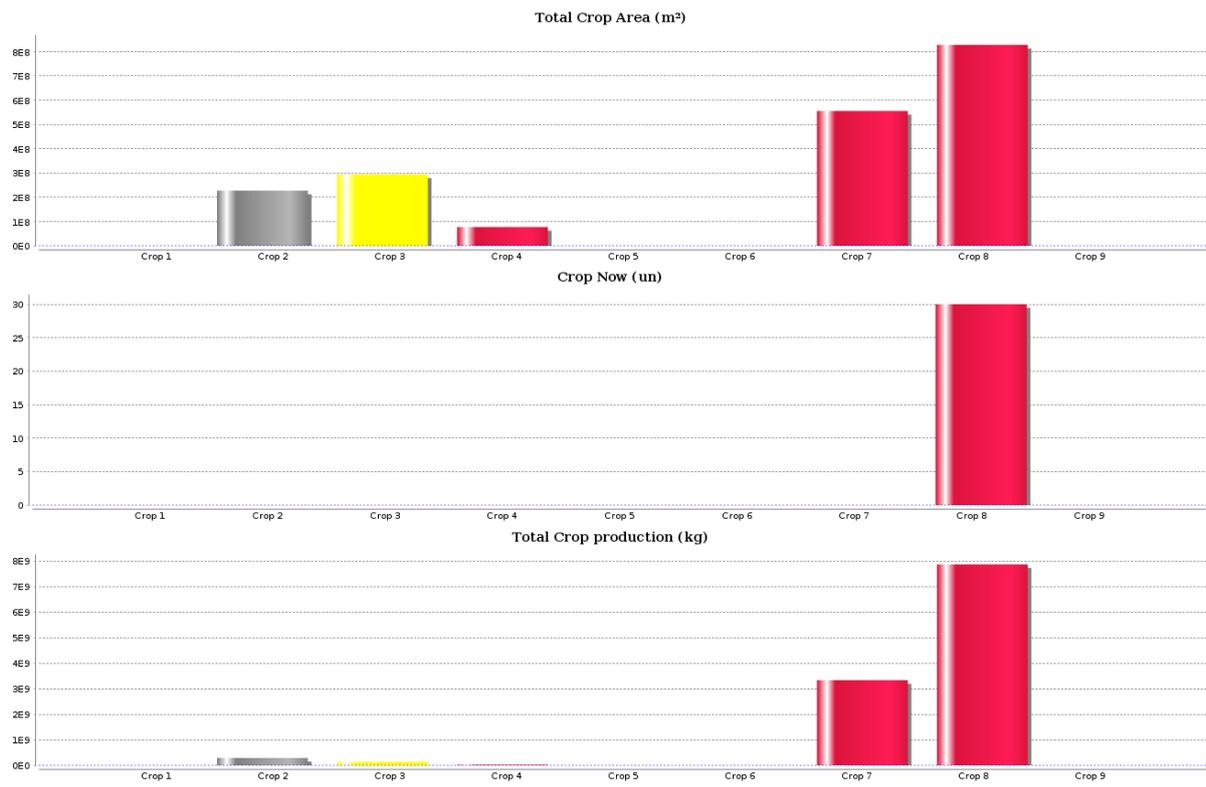


Figura 4.2: Dados de culturas - original

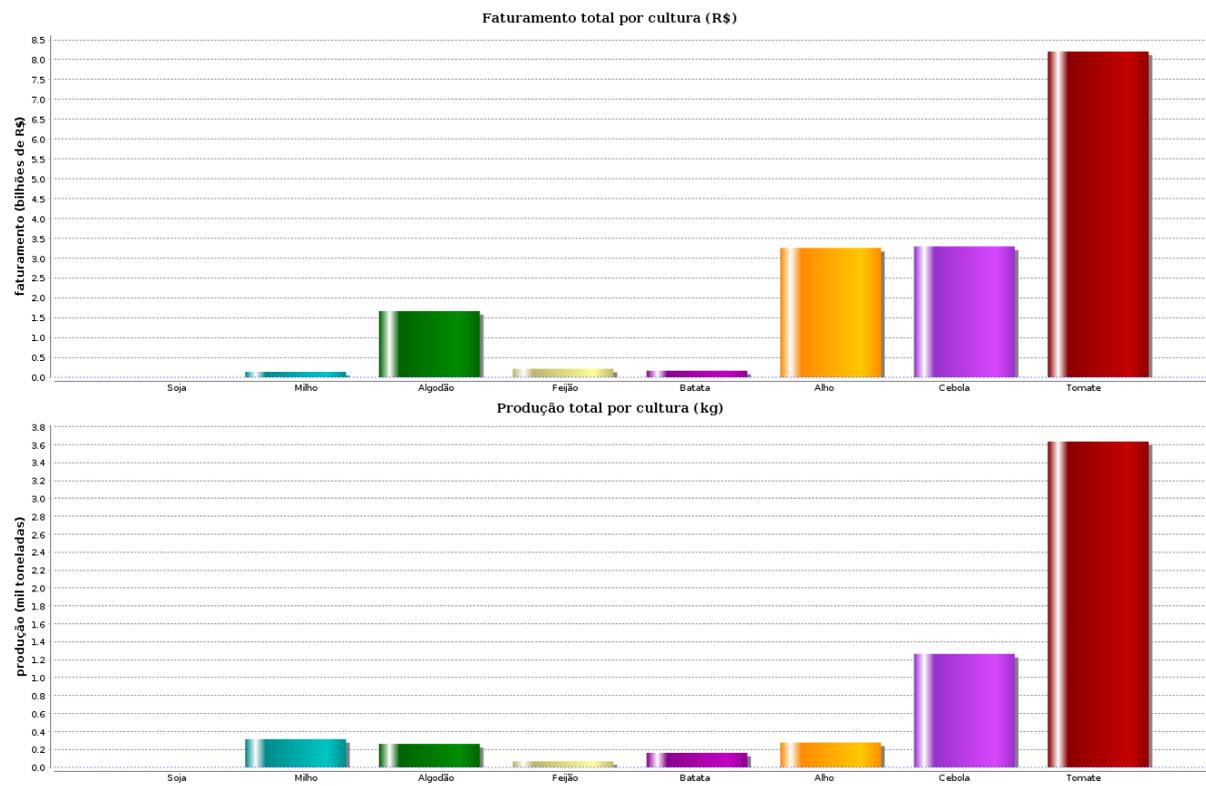


Figura 4.3: Dados de culturas - novo



Figura 4.4: Demanda regional de água - original

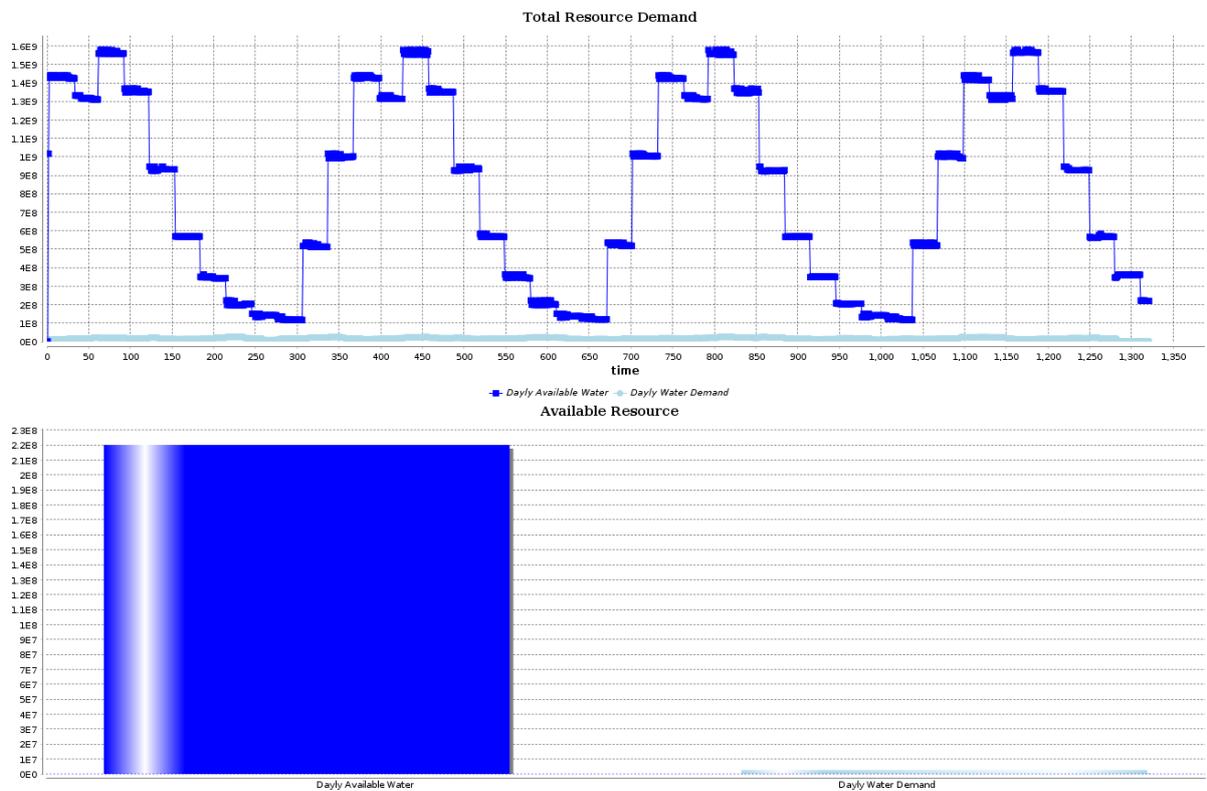


Figura 4.5: Demanda total de água - original

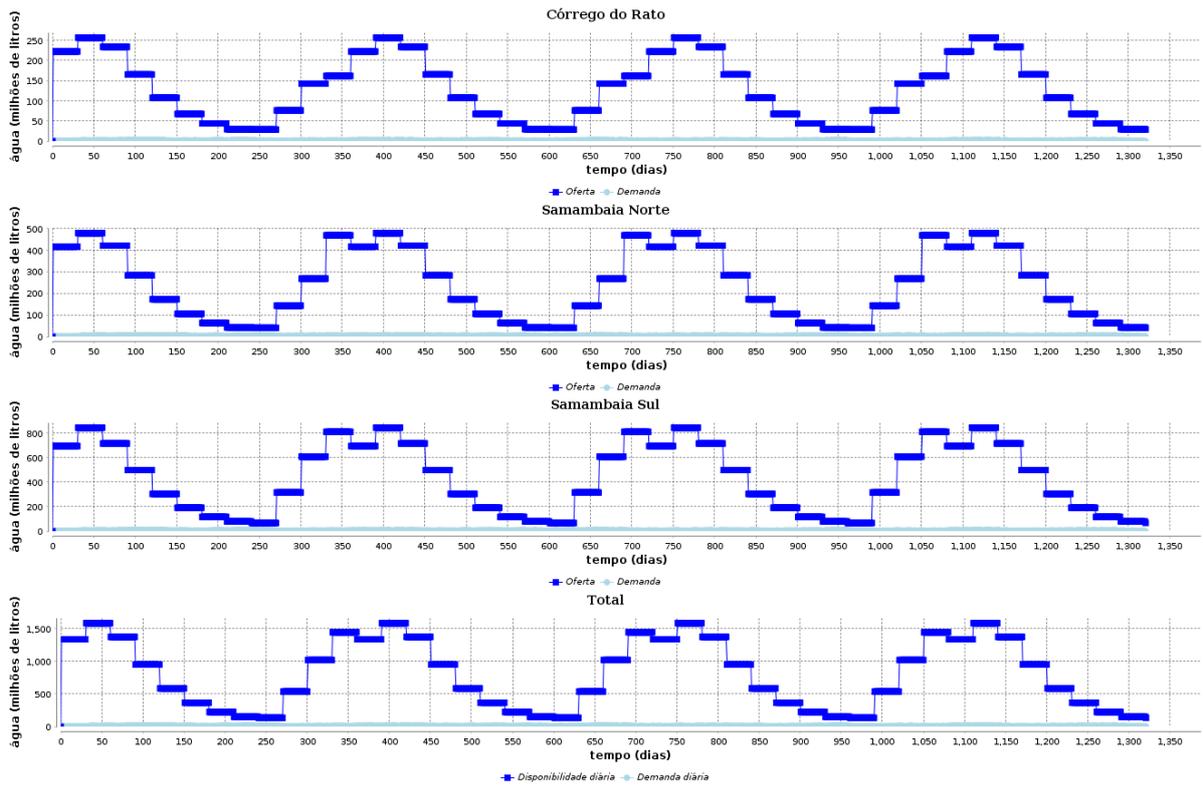


Figura 4.6: Demanda de água - novo

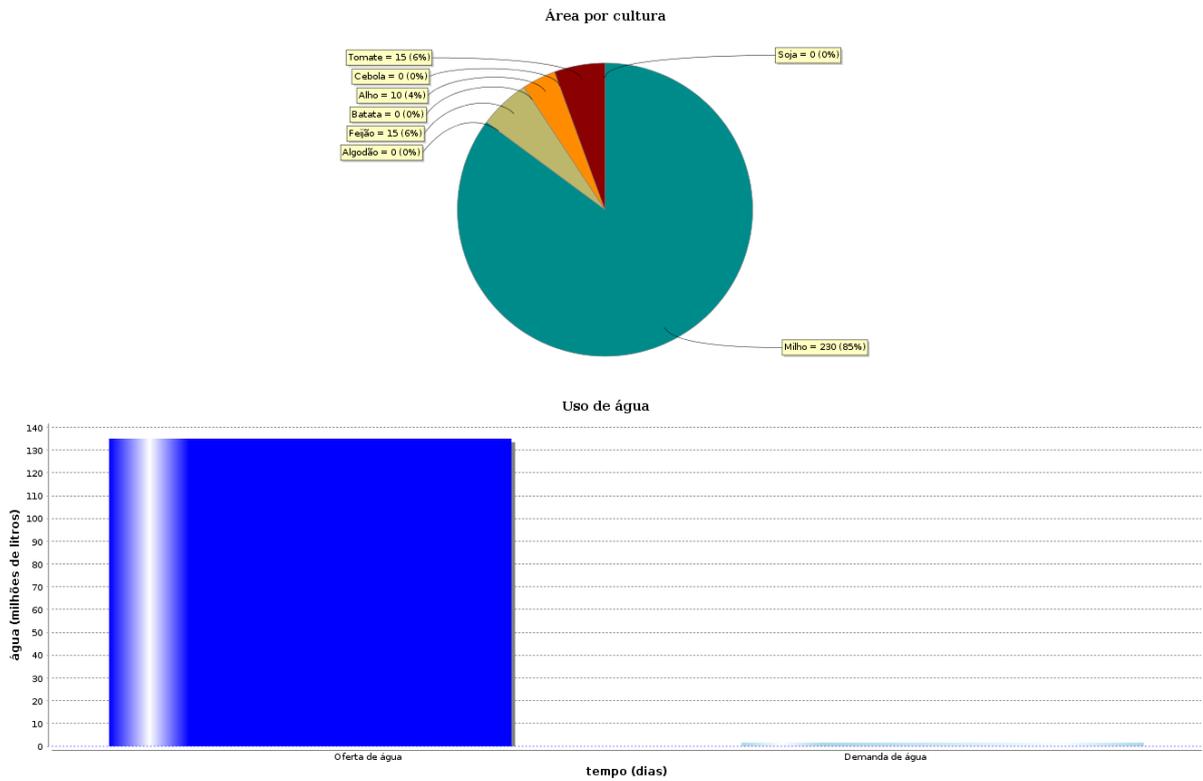


Figura 4.7: Dados diários - original

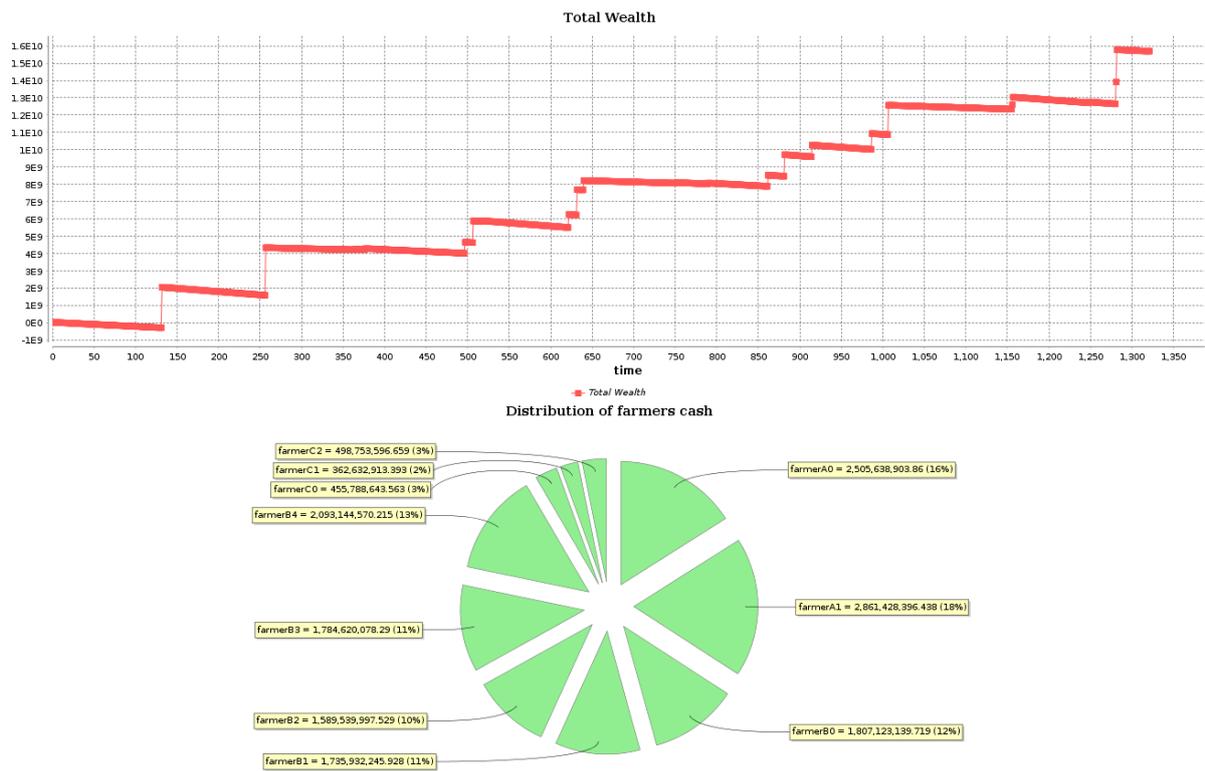


Figura 4.8: Patrimônio gerado - original

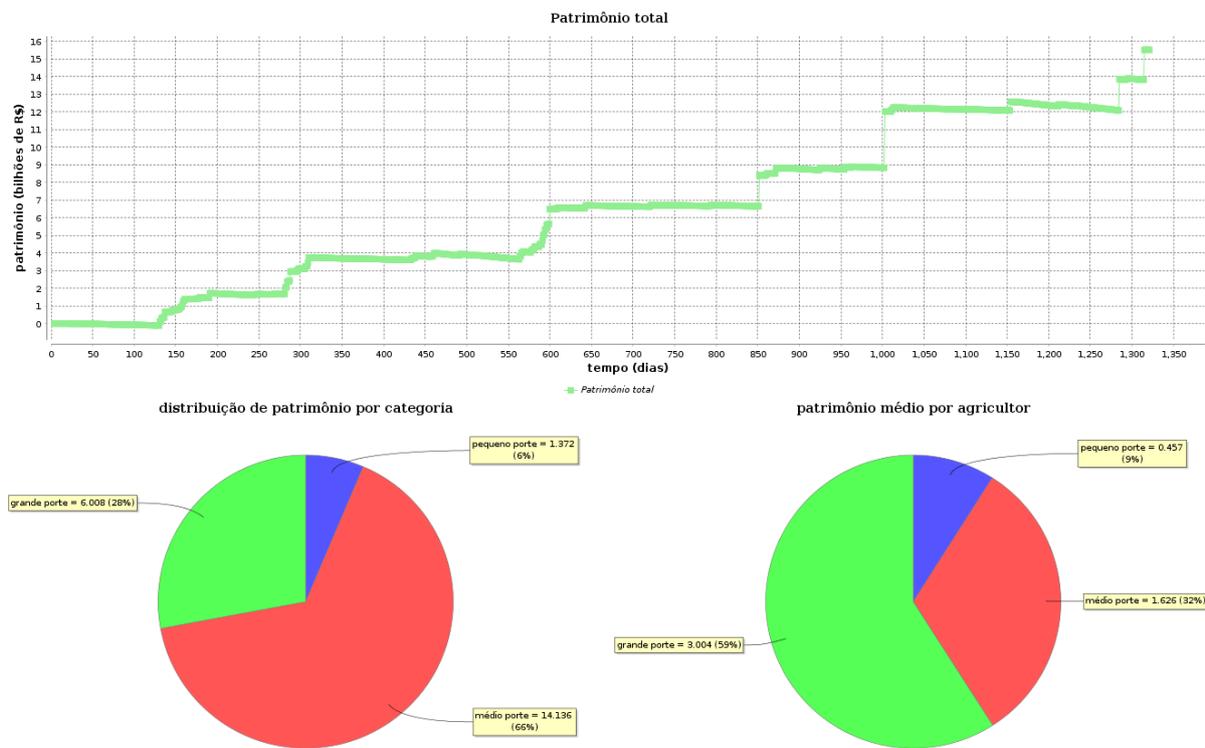


Figura 4.9: Patrimônio gerado - novo

Capítulo 5

Conclusões

A realização deste trabalho proporcionou um aprofundamento das percepções do autor em muitos níveis. Do ponto de vista técnico (diga-se, relativo à ciência da computação e à engenharia de software), podemos citar: o estudo da teoria de modelagem computacional; o estudo de sistemas complexos em geral e sistemas socioambientais em particular; modelagem e simulação baseada em agentes; estudo da plataforma GAMA e outras plataformas de desenvolvimento de MSBA.

Para o autor, cujo campo de maior familiaridade é a engenharia de software no contexto do desenvolvimento de sistemas web, foi desafiador trabalhar com uma plataforma cujo paradigma é tão diferente do habitual naquele contexto.

A realização do projeto não se restringe, no entanto, ao aspecto técnico e teórico, nem ao escopo da teoria da computação. Foi necessária a familiarização do autor com aspectos da ecologia, particularmente da relação existente entre a agricultura, fenômenos climáticos, e recursos hídricos. Além disso, foi considerado o aspecto social do problema: a relação dos produtores e gestores entre si e com a terra.

Um ponto que foi particularmente desafiador for lidar com dados reais. Os dados coletados na etapa anterior do projeto apresentavam algumas inconsistências e baixa granularidade, como discutido na Seção 3.3.5. A busca por dados mais consistentes foi muito custosa, e o seu resultado não foi plenamente satisfatório. A importância e dificuldade da coleta de dados fez-se muito clara para o autor, cuja experiência prévia com esta etapa de projetos era pequena.

De maneira resumida, chegou-se à conclusão de que para que um projeto desse escopo seja bem sucedido, muito mais fatores estão em jogo do que apenas um modelo bem feito e um código bem desenvolvido.

5.1 Visões sobre o problema e avaliação do modelo

Mesmo no cenário relativamente simples da bacia do rio Samambaia, existem muitos fatores em jogo para que se tenha pleno entendimento da situação econômica, social e ambiental da região.

Um dos objetivos deste projeto era construir um modelo que pudesse ser utilizado pelos produtores e gestores locais para entender melhor o papel que ocupam na região e possíveis tendências ambientais desta. Acreditamos que o modelo atual ainda não está adequado para cumprir este objetivo. Com os dados e a modelagem atual, não foi possível observar situações de grave escassez de água, e portanto não foi possível observar como os produtores reagiriam a tais situações no longo prazo.

Para contornar este problema foi adicionado um parâmetro que multiplica o total de água na bacia a cada dia por um fração. Com isso foi possível perceber que a escassez de água realmente influencia as decisões e a performance dos produtores. Esta é, contudo, apenas uma prova de conceito e não substitui uma modelagem apropriada.

A provável causa desta discrepância entre o resultado esperado e o observado na simulação é a maneira como foi implementada a disponibilidade de água e um entendimento errôneo do real cenário na bacia. Em discussões posteriores à modelagem, foi observado que embora a outorga de água esteja próxima do limite, gerando uma situação de risco de conflito, este fato não necessariamente implica que a água disponível na bacia está escassa ou que haja falta de água para os produtores. Isto porque grande parte da água disponível está outorgada para uso pela usina hidrelétrica de Batalha. Ou seja: a outorga de água para os produtores não corresponde ao total de água disponível na bacia, devendo sempre haver uma quantidade considerável de água disponível.

Uma alternativa de modelagem, portanto, seria utilizar os dados reais de outorga de água para atividade agrícola, em vez do total de água disponível. Isto tornaria a modelagem mais simples, precisa e informativa.

Além da validação descrita acima, acreditamos que para cumprir os objetivos do projeto seria necessário adotar uma abordagem de modelagem mais versátil, em que a quantidade de pivôs na bacia pudesse ser aumentada, por exemplo, ou que parâmetros como o nível de água na bacia, a quantidade de chuva, e a vazão dos rios pudesse ser manipulada facilmente pelos usuários.

Em relação ao comportamento dos fazendeiros (escolha de culturas), consideramos que os resultados obtidos são melhores. Embora não tenha sido realizada uma validação formal destes, observamos que o comportamento dos agentes na simulação, detalhado na Seção 4.3, é similar àquele descrito durante as reuniões com os produtores realizadas em etapa anterior do projeto [7]: as culturas plantadas com maior frequência foram as mesmas em ambos os casos, e as mudanças de comportamento em função do porte do

produtor foram similares. Resta fazer uma validação formal e analisar se esta relação se mantém para cenários de escassez de água.

5.2 Percepções sobre a plataforma GAMA

A plataforma GAMA tem diversas vantagens em comparação com outras plataformas de desenvolvimento de MSBA. É uma plataforma moderna, de código aberto, com bom suporte gráfico, muitas funcionalidades embutidas, e que goza de popularidade entre pesquisadores da área de ciências ambientais.

Particularmente, o ambiente de desenvolvimento (IDE) é muito satisfatório, principalmente durante a execução da simulação em GUI (interface gráfica), sendo muito fácil inspecionar detalhes da simulação e alterar parâmetros a qualquer momento.

No entanto, do ponto de vista do autor, a plataforma GAMA padece de problemas em seu ponto mais fundamental: sua linguagem de desenvolvimento, o GAML. Em seguida exporemos alguns deles. Deve-se notar que muitos destes problemas são opiniões pessoais do autor, e outros são comuns à maioria das plataformas existentes. Ainda assim deve-se considerá-los para maior compreensão do cenário atual de pesquisas em MSBA.

Modularidade

O ponto mais claro a ser notado é a falta de uma estrutura de modularização adequada. Isto prejudica muito a escalabilidade do projeto, pois quanto mais entidades e modelos existirem no projeto mais difícil será manter a organização e o encapsulamento do código.

Flexibilidade (níveis de linguagem)

Um fenômeno muito comum em frameworks de desenvolvimento de alto nível é o sacrifício da flexibilidade - do *poder* da linguagem - em prol da facilidade de escrita. É muito fácil implementar certos procedimentos (aqueles que estão contemplados pelos métodos embutidos da linguagem), mas muito custoso fazer qualquer operação menos usual. A linguagem GAML padece deste mal: embora haja diversas estruturas de dados, operadores e constantes embutidas na linguagem, é muito difícil manipular a linguagem em nível mais baixo e gerar código customizado. Esse ponto certamente é exacerbado pelo baixo conhecimento da linguagem por parte do autor. Porém, a documentação do GAMA corrobora com esta dificuldade por focar em exemplos de alto nível em detrimento de manipulações mais básicas da linguagem.

Paralelismo

A linguagem GAML não tem suporte a *threads*, eventos, ou métodos assíncronos. O fluxo de execução observa um sincronismo fixado pela iteração da simulação, o que impossibilita o uso de paralelismo. i. e. todos os comandos são executados sequencialmente no começo de cada iteração. É possível controlar a ordem de execução dos agentes através do comando *schedules*. Porém, esta abordagem é estática (não pode ser alterada durante a simulação) e não é muito versátil para um controle preciso do fluxo de execução. No contexto de um sistema complexo, caracterizado por interações complexas e laços de *feedback*, esta limitação se mostra muito restritiva.

5.3 Possibilidades futuras

A principal etapa a ser cumprida futuramente é a reimplementação do uso de água, de acordo com a discussão da Seção 5.1, e a validação do modelo. Possibilidades de validação foram apresentadas na Seção 3.4. Em seguida serão elaborados alguns pontos a seu respeito.

Os principais pontos que se deseja validar no modelo são o uso de água e o comportamento dos produtores. Para o primeiro ponto, desejamos saber se o uso de água no modelo está próximo do uso de água observado na realidade. Para tal, pode-se seguir os seguintes passos:

- obter os dados históricos de outorga de água para agricultura na bacia;
- reimplementar a lógica de uso de água considerando os dados de outorga obtidos, e não apenas o total de água na bacia;
- verificar se o uso de água observado no modelo está próximo à outorga, como observado na realidade;

Deve-se observar que a validação do uso de água na bacia se tornará muito mais precisa se for possível obter dados históricos de uso de água na bacia pela atividade agrícola. Com estes dados o modelo poderia ser comparado diretamente com a realidade, e poderia-se obter métricas de precisão do modelo, e trabalhar para aumentar sua precisão.

Caso não seja possível obter tais dados, no entanto, uma comparação entre o uso de água no modelo e os dados de outorga seria um bom primeiro passo para avaliar a veracidade do modelo no quesito de uso de água.

Em relação ao comportamento dos produtores, considera-se que a melhor opção para validação seja coletar dados de produção, mediante pesquisa de campo, para produtores de diferentes portes. Cada produtor organiza suas finanças de maneira diferente, portanto

não deve pretender-se obter dados precisos para longos períodos de tempo. Porém, quando maior a quantidade e precisão dos dados obtidos, melhor será a validação. Os dados desejados dizem respeito a: (i) culturas plantadas em cada época do ano ao longo do tempo; (ii) custo e preço de culturas em diferentes épocas do ano ao longo do tempo; (iii) renda obtida ao longo do tempo.

Com estes dados poderia ser feita uma comparação precisa, que permitira ajustes nos dados de entrada e na modelagem de forma a aumentar a precisão do modelo.

Excluída a validação, e em termos mais generalistas, consideramos as seguintes etapas fundamentais para a continuação do projeto:

- Definir com maior precisão qual será o foco do modelo a partir deste momento, e assegurar que existem critérios de validação claros para guiar sua elaboração.
- Assegurar fontes de dados confiáveis, completas, e precisas, de forma a evitar ambiguidade na avaliação dos resultados. Pode-se procurar estes dados em outras regiões.
- Focar na ampliação da flexibilidade do modelo, de forma que se possa aplicá-lo a diferentes áreas e diferentes cenários ambientais e econômicos sem muitas dificuldades. Com isto o modelo poderia ser testado em diferentes cenários e portanto melhor avaliado, além de aumentar suas possibilidades de uso por usuários locais (produtores, gestores, etc).

Se as etapas acima puderem ser contempladas, acreditamos que este projeto tenha boas possibilidades de continuidade, podendo ser útil enquanto instrumento de pesquisa e como ferramenta de resolução de conflitos. Considera-se que gestores e produtores poderiam, através do uso do modelo, criar um entendimento comum da situação, o qual embasaria o diálogo entre eles.

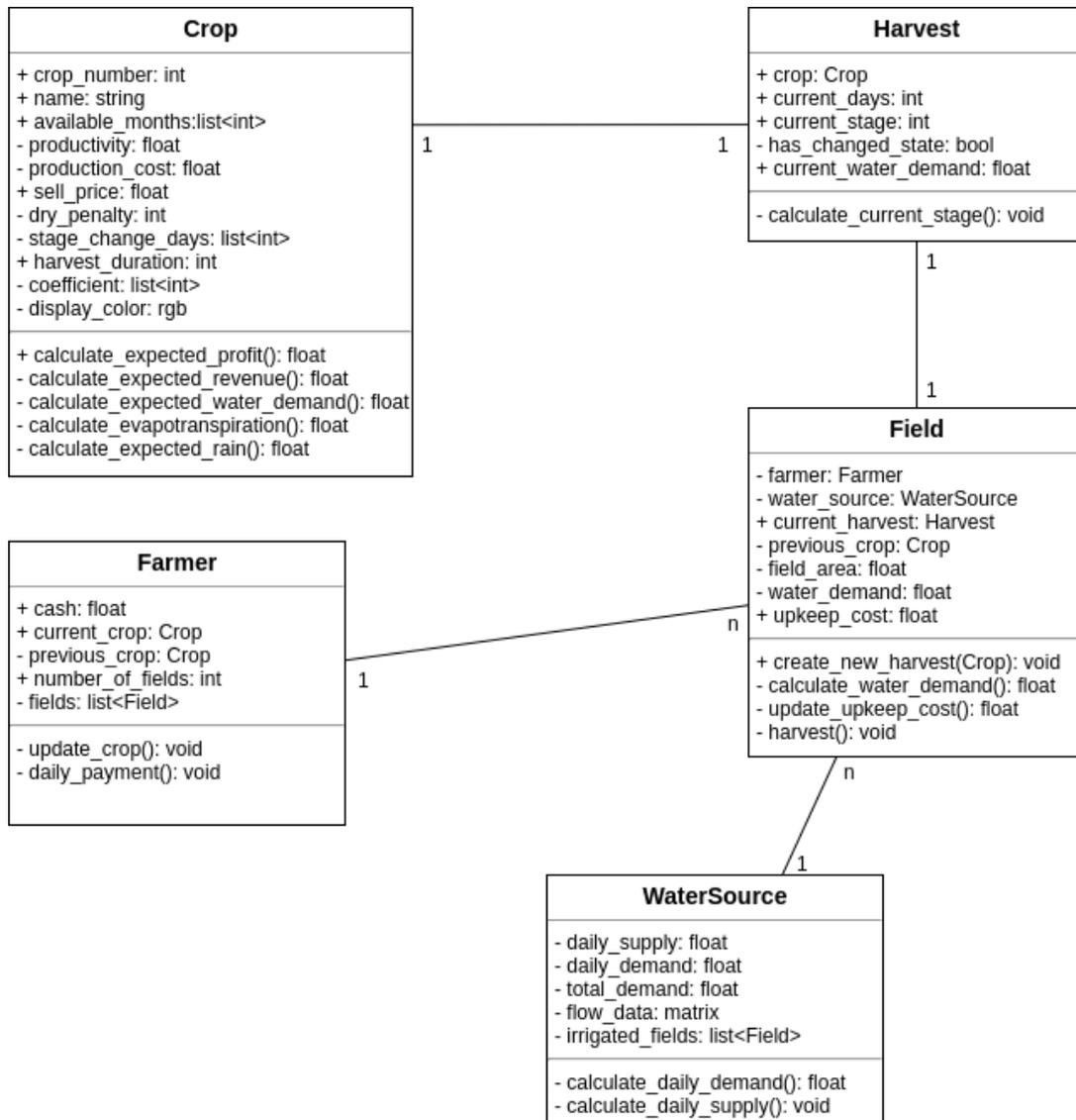
Referências

- [1] ANA: *Conjuntura recursos hídricos brasil 2018*. 1
- [2] Macal, C. M.: *Everything you need to know about agent-based modelling and simulation*. *Journal of Simulation*, 10(2):144–156, 2016. 1, 3
- [3] Law, Averill M. e David M. Kelton: *Simulation Modeling and Analysis*. McGraw-Hill Higher Education, 5rd edição, 2015. 1
- [4] Richardson, George P.: *System Dynamics, The Basic Elements of*. Springer New York, New York, NY, 2011, ISBN 978-1-4419-7701-4. https://doi.org/10.1007/978-1-4419-7701-4_48. 1
- [5] Rubinstein, Reuven Y. e Dirk P. Kroese: *Simulation and the Monte Carlo Method*. Wiley Publishing, 3rd edição, 2016, ISBN 1118632168, 9781118632161. 1
- [6] E., Cellier François e Enrensto Kofman: *Continuous system simulation*. Springer, 2011. 1
- [7] Porto, Pedro Phelipe Gonçalves: *Construção de modelo multiagente para avaliação de cenários de manejo de recursos hídricos*. Dissertação (Mestrado em Tecnologia Ambiental e Recursos Hídricos)—Universidade de Brasília, Brasília, 2019. 2, 12, 13, 15, 40, 50
- [8] *Platform · gama*, nov 2019. <https://gama-platform.github.io/wiki/Home>. 2, 6, 10
- [9] Bousquet, François, Innocent Bakam, Hubert Proton e Christophe Le Page: *Cor-mas: Common-pool resources and multi-agent systems*. Em Pobil, Angel Pasqual del, José Mira e Moonis Ali (editores): *Tasks and Methods in Applied Artificial Intelligence*, páginas 826–837, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg, ISBN 978-3-540-69350-5. 5
- [10] *Site do netlogo*, oct 2019. <https://ccl.northwestern.edu/netlogo/>. 5
- [11] *Maelia*, jun 2019. <http://maelia-platform.inra.fr/>. 6
- [12] Tucker, Allen B.: *Programming Languages: Principles and Paradigms*. McGraw-Hill, Inc., New York, NY, USA, 2001, ISBN 007112280X. 7
- [13] *Gama documentation section on types*, nov 2019. <https://gama-platform.github.io/wiki/DataTypes>. 9

- [14] *Gama documentation section on operators*, nov 2019. <https://gama-platform.github.io/wiki/OperatorsAA>. 9
- [15] Booch, Grady, James Rumbaugh e Ivar Jacobson: *Unified Modeling Language User Guide, The (2Nd Edition) (Addison-Wesley Object Technology Series)*. Addison-Wesley Professional, 2005, ISBN 0321267974. 9
- [16] Volker Grimm, Uta Berger, Finn Bastiansen et al.: *A standard protocol for describing individual-based and agent-based models*, Jun 2006. <https://www.sciencedirect.com/science/article/pii/S0304380006002043>. 10
- [17] Volker Grimm, Uta Berger, et al.: *The odd protocol: A review and first update*. Ecological Modelling, 221:2760–2768, novembro 2010. 10
- [18] *Ibge cidades*, nov 2019. <https://cidades.ibge.gov.br/brasil/go/cristalina/pesquisa/38/47001?tipo=ranking>. 12
- [19] *Relatório do projeto samambaia*, Nov 2017. <https://www.ana.gov.br/aceso-a-informacao/institucional/apresentacoes-1/apresentacao-sobre-o-projeto-samambaia-jose-zoby.pdf>. 13
- [20] *Computational model library*, nov 2019. <https://www.comses.net/codebases/47ded80a-75a1-4cca-b939-b7c7a4b58dc7/releases/1.0.1/>. 14, 15
- [21] *Ifag - insituto para o fortalecimento da agropecuária de goiás*, jun 2019. <http://ifag.org.br/>. 15
- [22] *Conab*, nov 2019. <http://sisdep.conab.gov.br/precosiagroweb/>. 25, 49, 54
- [23] *Inmet*, nov 2019. <http://www.inmet.gov.br/portal/index.php?r=bdmep/bdmep>. 25, 27, 54
- [24] Monteplan: *Relatório técnico - análise de resolução ana 562/2010 - marco regulatório rio são marcos*, novembro, 2011. 49

Apêndice A

UML do modelo



Apêndice B

ODD do modelo

B.1 Visão geral

B.1.1 Propósito

Este é um modelo do uso de água e da atividade agrícola na bacia do rio Samambaia. Com este modelo pretende-se investigar como se dá a escolha de culturas por parte dos produtores, de acordo com os recursos disponíveis a eles. Em particular, queremos também explorar como cenários de escassez de água influenciam o comportamento dos produtores e a produtividade da região como um todo.

B.1.2 Entidades, variáveis de estado, e escalas

Na sequência são descritas as entidades do modelo.

Global

A espécie global representa o mundo (ambiente) da simulação. Espacialmente, o mundo é caracterizado pela bacia do rio Samambaia, sendo esta dividida em três sub-regiões: samambaia sul, samambaia norte, e córrego do rato. Temporalmente, a simulação ocorre num período de 3 anos e 8 meses - de janeiro de 2016 a agosto de 2019, período cujos dados históricos são utilizados. Utilizamos a aproximação de que cada mês dura exatamente 30 dias. A resolução da simulação é diária, i. e. cada iteração da simulação equivale a um dia no ambiente. Dados de temperatura e precipitação também estão associados ao mundo da simulação, e são atualizados diariamente.

Farmer

Representa um produtor ou fazendeiro. Há dez produtores no ambiente, e estes permanecem durante todo o decorrer da simulação. Cada produtor está associado a uma área, dentro da qual estão localizados n pivôs centrais. O fazendeiro escolhe o que será plantado em sua região a cada colheita, visando maximização do lucro e estabilidade econômica.

O fazendeiro possui capital, com despesas e ganhos. Suas despesas são a soma do custo de manutenção de suas plantações, e sua renda é o retorno das colheitas. O objetivo do fazendeiro, portanto, é maximizar seu capital e evitar a falência.

Dentre os dez produtores instanciados, há três categorias nas quais o número de pivôs e a localização na bacia são diferentes:

- pequeno porte: 10 pivôs, próximo ao córrego do rato;
- médio porte: 30 pivôs, próximo a samambaia norte;
- grande porte: 50 pivôs, próximo a samambaia sul.

A localização exata de cada fazendeiro é aleatória dentro de uma circunferência cujo centro está próximo ao centro de cada sub-região.

Field

Representa um pivô central, que pertence a um produtor (Farmer) e onde pode estar plantada uma cultura (Crop). Cada pivô utiliza água proveniente de uma fonte de água (WaterSource), dependendo da sua localização dentro da bacia. A quantidade de água utilizada depende do tamanho do pivô e da cultura atual. O tamanho do pivô também influencia seu custo de manutenção. A mesma cultura não pode ser plantada no mesmo pivô duas vezes seguidas. As entidades são geradas a partir do *shapefile* mapeado pela ANA, e se mantém ativas durante toda a simulação.

Harvest

Representa um ciclo de colheita, desde o momento onde as sementes são plantadas até o momento da colheita. Cada entidade está associada a um pivô (Field), é gerada no momento do plantio, e deixa de existir após a colheita. Um Field terá, portanto, vários Harvests ao longo da simulação.

Crop

Representa uma cultura dentre as opções disponíveis na simulação. As culturas disponíveis foram escolhidas mediante pesquisa de campo, em que se avaliou quais as culturas mais

comuns na região. Dados de preço de venda foram obtidos do portal da CONAB [22], e dados de produtividade e custo e tempo de produção foram obtidos de diversas fontes, sendo principal destas o Instituto para Fortalecimento da Agricultura no Goiás - IFAG.

Cada cultura tem um *coeficiente de cultura*, que, associado à evapotranspiração no ambiente, determina sua demanda de água. Este coeficiente muda durante o ciclo de vida da cultura, portanto estipulou-se, para cada cultura, quatro estágios, cada um com um coeficiente diferente. Os dados para determinação de estágios e coeficientes foram obtidos do relatório técnico da Monteplan [24].

WaterSource

Representa uma fonte de água dentro da bacia. Foram consideradas três fontes de água, cada uma associada a uma sub-bacia (Samambaia sul, Samambaia norte e Córrego do rato). Cada fonte de água tem vazão e total de água específicos, e fornece água aos pivôs dentro da sua região.

Cada sub-bacia é implementada como uma classe que herda de WaterSource, a dizer: *CorregoRato*, *SamambaiaSul* e *SamambaiaNorte*. Cada uma destas é instanciada uma única vez, no começo da simulação, e se mantém ativos durante toda sua duração.

B.1.3 Processo e cronograma

A simulação tem duração de 3 anos e 8 meses - de janeiro de 2016 a agosto de 2019 - e cada iteração da simulação equivale a um dia. No GAMA, todos os processos são síncronos e ocorrem uma vez por iteração, não havendo sistema de eventos ou processos paralelos.

Variáveis de estado que são atualizadas diariamente incluem: demanda de água; custo de manutenção de culturas; dinheiro dos produtores; precipitação; temperaturas mínima, média e máxima; evapotranspiração.

Variáveis de estado que são atualizadas mensalmente incluem: preço de venda dos cultivos; vazão dos rios.

A principal atividade que ocorre esporadicamente na simulação é a colheita e o plantio, que ocorrem da seguinte maneira: cada cultivo tem determinado tempo produção. Quando chega o momento da colheita, o produtor recebe o retorno financeiro, de acordo com o preço de venda daquele plantio no mês em questão. Então, o produtor determina qual o próximo cultivo a ser plantado. Para isto, ele faz uma estimativa do retorno esperado para cada tipo de cultivo, baseando-se nos dados do ano anterior (utiliza-se os dados do mesmo ano durante o primeiro ano).

O produtor escolhe o plantio que lhe dará maior retorno, e procura manter a margem de variação de preço abaixo de um limite determinado pelos seus recursos (terra sob sua

posse). Caso necessário para diminuir a insegurança, o produtor pode plantar mais de um cultivo simultaneamente. Nota-se que produtores de menor porte tendem a plantar mais de um plantio, para evitar prejuízos que poderiam levá-los à falência.

Para calcular o retorno financeiro esperado para cada cultura, é necessário estimar o custo de manutenção, custo de irrigação, e preço de venda esperado. Nesta seção será detalhada cada uma destas estimativas.

Custo de manutenção

O custo de manutenção é constante para cada cultura, e é determinado por dados de entrada (vide B.3.2. Para pivôs sem nenhuma cultura, foi estimado um custo de 0.01 reais por dia [7].

Custo de irrigação

Para calcular o custo de irrigação, é necessário estimar a demanda de água e influência da chuva durante o ciclo de plantio. O dia e mês de referência utilizados nos cálculos abaixo são o dia e mês em que será realizado o plantio. Os dados históricos utilizados se referem ao ano anterior quando disponível, ou ao ano atual caso seja o primeiro ano de simulação.

A demanda de água diária é dada por $CC * evp$, onde CC é o coeficiente de cultura para o estágio atual de plantio e evp é a evapotranspiração diária.

Considerando que cada cultura têm quatro estágios em que coeficiente de cultura CC muda, deve-se estimar a demanda de água para cada estágio do plantio. Para tornar menos custosa a computação, considera-se a evapotranspiração evp constante durante o estágio, e seu valor igual à evapotranspiração durante o primeiro dia do estágio ($evap(dia1)$). As Equações B.1 e B.2 representam a demanda total e por estágio, respectivamente.

$$D_{total} = \sum_{n=1}^4 D_{estagio} \quad (B.1)$$

$$D_{estagio} = evp(dia1) * CC \quad (B.2)$$

Em seguida estima-se a influência da chuva (C) na demanda de água. Esta foi considerada como o total de chuva durante o ciclo de plantio (C_{total}) multiplicado pelo fator K , que determina a regularidade de chuva, de acordo com as Equações B.3, B.4 e B.5.

$$C = C_{total} * K \quad (B.3)$$

Nas Equações B.4 e B.5, $Cmensal$ é o total de chuva durante o mês de referência (dados históricos), T é a duração do plantio em dias e $Dabs$ é o desvio absoluto da série histórica durante o mês de referência.

$$Ctotal = Cmensal * T \quad (B.4)$$

$$K = Dabs / (2 * Cmensal), K \subset [0, 1[\quad (B.5)$$

Escolha do plantio

O lucro para cada cultura é estimado da seguinte forma:

$$L = P - (Cm + Ci), \quad (B.6)$$

onde P é o preço de venda médio para o mês de referência (mês de colheita), Cm é o custo de manutenção e Ci o custo de irrigação.

Além do lucro estimado L , é considerada a variabilidade de preços das culturas. Para representar esta variabilidade utiliza-se o coeficiente de variação (CV) da série histórica de preços mensais das culturas:

$$CV = \frac{\sqrt{\sum(x_i - \bar{x})^2/n}}{\bar{x}}, \quad (B.7)$$

onde:

CV = coeficiente de variação

x_i = valor individual

\bar{x} = média dos valores

n = número de valores

Caso o CV da cultura mais lucrativa seja maior que a tolerância do agricultor, a segunda cultura mais lucrativa é adicionada, e assim sucessivamente.

Tolerância à variabilidade de preços

A tolerância T consiste em um valor base Tb e um valor por pivô Tp :

$$T = Tb + Npivos * Tp, \quad (B.8)$$

de forma que quanto mais pivôs o produtor possuir ($Npivos$) maior será sua tolerância à variabilidade de preços e menos culturas simultâneas ele tende a plantar.

Os valores Tb e Tp foram determinados empiricamente de maneira que produtores de grande porte tendem a plantar apenas uma cultura, enquanto produtores de médio e pequeno porte costumam plantar duas ou três. Estes valores podem ser alterados pelo usuário na interface gráfica preliminarmente à execução da simulação.

Restrições

Determina-se as seguintes restrições, de forma a simplificar a modelagem:

- o total de pivôs é dividido igualmente entre as culturas determinadas;
- só é realizado um novo plantio após a colheita de todas as culturas;
- nenhuma das culturas plantadas no último ciclo pode ser plantada novamente.

B.2 Conceitos de Projeto

- **Princípios básicos:** assume-se que produtores buscam maximizar sua renda de maneira economicamente sustentável, por meio da escolha de culturas, e que a disponibilidade de água e outros fatores climáticos influenciam esta decisão, de forma que a dinâmica do uso de água na bacia deriva da interação entre fatores econômicos e hidro-climáticos predominantes na bacia.
- **Emergência:** a estratégia dos fazendeiros muda dependendo da disponibilidade de água, o que afeta o tipo de produção e a renda da bacia como um todo.
- **Adaptação:** a escolha do cultivo depende das condições ambientais e das condições econômicas dos produtores. Contudo, não há dinamicidade neste comportamento, mantendo-se constante ao longo da simulação.
- **Objetivos:** o principal objetivo dos produtores é o de maximizar o lucro e evitar a falência. Nota-se que cenários de escassez de água podem dificultar esse objetivo em nível global.
- **Aprendizado:** não há exemplos de aprendizado significativo no modelo.
- **Predição:** um exemplo de predição é a escolha de plantio pelos produtores, que envolve uma estimativa do lucro obtido para cada cultura.
- **Sensoriamento:** assume-se que os fazendeiros e plantios têm acesso a dados de temperatura, precipitação, e disponibilidade de água no ambiente.
- **Interação:** não há interação significativa entre os agentes, fora casos simples como o sensoriamento de condições ambientais e a interação entre o produtor e seus plantios.

- **Estocasticidade:** na escolha de plantio pelos produtores são utilizadas predições que possuem variabilidade.
- **Coletivos:** como há pouca interação entre agentes da mesma espécie, coletivos não têm um papel relevante no modelo.
- **Observação:** coletamos dados de produção (tipo de cultura e renda de cada fazendeiro) e dados de uso e disponibilidade de água. Como isso podemos investigar como se dá a relação entre estes, e se os resultados estão próximos das observações reais.

B.3 Detalhes

B.3.1 Inicialização

A simulação começa no dia 01/01/2015. Neste instante, é gerado o ambiente da bacia, as sub-bacias, os fazendeiros e os pivôs. Os pivôs estão inicialmente sem nenhum plantio. Para cada fazendeiro, um dia do primeiro mês é escolhido aleatoriamente para realizar o plantio da primeira cultura.

A localização dos fazendeiros na bacia é determinada dinamicamente no começo da simulação. Fazendeiros de pequeno porte (10 pivôs) são instanciados próximo ao córrego do rato, fazendeiros de médio porte (30 pivôs) próximo a samambaia norte, e fazendeiros de grande porte (50 pivôs) próximo a samambaia sul. A cada fazendeiro são atribuídos os pivôs disponíveis mais próximos, até que todos tenham sido atribuídos.

B.3.2 Dados de entrada

Fazemos uso dos seguintes *shapefiles*, obtidos da ANA:

- bacia do rio samambaia
- córrego do rato
- samambaia sul
- samambaia norte
- mapeamento dos pivôs
- mapeamento dos rios

Além dos *shapefiles*, fazemos uso dos seguintes dados de entrada:

- vazão de água em cada sub-bacia (ANA)
- preço de mercado das culturas (CONAB [22])
- produtividade de culturas (múltiplas fontes)
- tempo e custo de produção (múltiplas fontes)
- temperatura na região (INMET [23])
- precipitação (INMET)

B.3.3 Submodelos

Não há submodelos neste modelo. Deve-se observar, contudo, que devido à estrutura da linguagem GAML, cada espécie está contida em um arquivo que declara um modelo *.gaml*, como descrito na Seção 3.2.2. Este arquivo não representa, contudo, um modelo do ponto de vista da teoria de MSBA.