M1 - Project Proposal

KSU CS4632 Model Simulation

Spring 2025

Important Note

This proposal template is designed to guide you through the project development process. Each section builds upon the previous ones, helping you create a comprehensive and well-structured project proposal.

Introduction

This document serves as a guideline for organizing your initial research and planning for the modeling and simulation project. It provides a structured way to document key components of your project effectively.

Expectations

The expectations for this project align with the following learning outcomes:

- Understand and describe the structure and dynamic behavior of various types of systems.
- Design conceptual models in UML for most properties of systems, ensuring a clear and comprehensive representation of system components and their interactions.

This project focuses on helping you achieve these goals by guiding your initial research and organization.

Model Selection

Key Concepts and Definitions

Before selecting your model, it's important to understand the key concepts in modeling and simulation:

- **System:** The real-world process, phenomenon, or entity you want to study (e.g., a hospital emergency room, a traffic intersection, a computer network)

- **Model:** A simplified representation of a system that captures its essential features and behaviors
  - *Conceptual Model:* The abstract description of how the system works (e.g., mathematical equations, logical relationships, flow diagrams)
  - *Computational Model:* The translation of the conceptual model into algorithms and data structures that can be implemented in code
- **Simulation:** The execution of a computational model over time to study system behavior
  - Uses the computational model to generate data about system behavior
  - Allows experimentation with different parameters and scenarios
  - Provides insights into system performance and behavior

This section will guide you through selecting an appropriate model for your project. Your goal is to identify a model that aligns with the objectives of the simulation and supports your research questions effectively.

Tool and Implementation Flexibility

This course encourages creativity and flexibility in your choice of simulation approaches:

- You may use any programming language, development environment, or simulation tool that best suits your project needs
- Options include, but are not limited to:
  - Building a custom simulation from scratch using programming languages like Python, Java, C++, C#, etc.
  - Using dedicated simulation frameworks such as SimPy, AnyLogic, or Arena
  - Leveraging specialized domain tools appropriate for your chosen simulation area
- Important considerations:
  - Your implementation must include programmatic elements for running simulations and collecting data
  - If using simulation tools, you must demonstrate automation of simulation runs and data collection through code
  - While not required, a functional user interface is encouraged to enhance usability
  - Game engines (like Unreal Engine or Unity) are acceptable development platforms, but:
    * The project must implement genuine simulation logic and data collection
    * Simply using built-in physics engines for basic demonstrations is not sufficient

    - Using pre-built games or simple modifications of existing games is not acceptable
    - The focus should be on modeling complex systems and analyzing their behavior

Purpose of Selection

Explain why this model was selected for the project. Your explanation should address the progression from system to simulation:

- **System Selection:**
  - What real-world system are you studying?
  - Why is this system interesting or important to simulate?
  - What aspects of the system do you want to understand better?
- **Model Development:**
  - How will you represent the system mathematically or logically?
  - What key components and relationships need to be captured?
  - What simplifications or assumptions are reasonable?
- **Simulation Goals:**
  - What questions do you want your simulation to answer?
  - What metrics or outcomes will you measure?
  - How will the simulation help understand the system better?
- **Implementation Considerations:**
  - How will you translate your model into code?
  - What computational approaches are most suitable?
  - How will you validate your implementation?

Model Description

Provide a detailed description of the chosen model, including:

- The primary purpose of the model
- Key components and their interactions
- Typical applications or use cases of this model
- Examples of similar models in research or industry

Domain Selection

Your simulation project can explore any domain of interest, *including but not limited to*:

- Business processes and workflows
- Healthcare systems and patient flow
- Transportation and logistics
- Manufacturing and production systems
- Environmental systems
- Social systems and behavior
- Network and communication systems
- Any other domain suitable for discrete-event simulation

Note on Implementation Depth: Regardless of the domain chosen, your project should demonstrate:

- Complex system modeling beyond simple physical simulations
- Custom simulation logic appropriate to your domain
- Meaningful data collection and analysis
- Scientific rigor in implementation and evaluation

Methodology

Outline the approach you plan to take for implementing and using this model in your simulation:

- What tools, frameworks, or programming languages will you use?
- How will the model be implemented (e.g., steps, process)?
- Are there specific methodologies (e.g., discrete-event simulation) that you will apply?
- How will you validate your model's accuracy?

Implementation Requirements

Your implementation must include:

- Source code demonstrating the simulation logic
- Automated mechanisms for:
  - Running multiple simulation scenarios
  - Collecting and analyzing simulation data
  - Generating results and statistics
- Documentation of:
  - System requirements and dependencies
  - Installation and setup procedures
  - Usage instructions and examples

Assumptions and Limitations

Discuss the assumptions and limitations associated with the chosen model:

- What assumptions underlie the model?
- Are there constraints or limitations that might impact the results?
- How will these assumptions and limitations affect your simulation?
- What steps will you take to mitigate these limitations?

Implementation Plan

This section outlines the steps and resources needed to implement your chosen model. The goal is to break down the implementation into manageable tasks and provide clarity on the tools and methods you will use.

Implementation Overview

Provide a high-level summary of your implementation plan. Address the following:

- Core simulation objectives and expected outcomes
- Overall architecture and design approach
- How your implementation will facilitate:
  - Automated simulation execution

  - Data collection and analysis
  - Result visualization and reporting
- Integration strategy if using multiple tools or components

Development Environment

Describe your chosen development environment and justify your selections:

- Programming language(s) and rationale for selection
- Development tools and IDEs
- Version control system and repository structure
- If using simulation software:
  - Tool selection justification
  - Integration with custom code for automation
  - Data export and analysis mechanisms

Required Components

Your implementation must include:

- Core simulation engine or integration with simulation software
- Automated execution mechanisms
- Data collection and storage
- Analysis and reporting capabilities
- Optional but encouraged:
  - User interface for simulation control
  - Real-time visualization
  - Configuration management
  - Advanced data analytics

Development Phases

Break down your implementation into distinct phases:

- Phase 1: Environment Setup and Basic Structure
  - Development environment configuration
  - Basic model implementation
  - Initial testing framework
- Phase 2: Core Functionality
  - Simulation logic implementation
  - Data collection mechanisms
  - Basic automation features
- Phase 3: Advanced Features
  - User interface (if planned)
  - Additional automation capabilities
  - Enhanced data analysis
- Phase 4: Testing and Refinement
  - Comprehensive testing
  - Performance optimization
  - Documentation completion

Quality Assurance

Outline your approach to ensuring implementation quality:

- Testing strategy and methodologies
- Validation procedures
- Performance benchmarking
- Code review process

Documentation Requirements

Detail the required documentation for your implementation:

- Technical documentation
  - System architecture
  - Class/component descriptions
  - API documentation (if applicable)
- User documentation
  - Installation guide
  - Usage instructions
  - Configuration options
- Implementation notes
  - Design decisions and rationale
  - Known limitations
  - Future enhancement possibilities

Risk Mitigation

Identify potential risks and planned mitigation strategies:

- Technical risks
- Resource constraints
- Timeline challenges
- Tool/framework limitations

Simulation Details

This section focuses on the technical aspects of your simulation implementation, data collection, and analysis methodologies. The framework provided is adaptable to various implementation approaches while maintaining academic rigor.

Simulation Scenarios

Your simulation should support multiple scenarios to demonstrate system behavior:

- Core scenarios that demonstrate fundamental system behavior
- Edge cases that test system boundaries and assumptions
- Validation scenarios that verify system accuracy
- Performance testing scenarios that evaluate system efficiency

Data Collection Framework

Regardless of your chosen implementation approach, your simulation must include systematic data collection:

- Automated data collection mechanisms
  - Event logging and timestamping
  - State tracking and transitions
  - Performance metrics
  - Resource utilization statistics
- Data storage and organization
  - Structured data formats (CSV, JSON, etc.)
  - Database integration (if applicable)
  - Version control for data sets
- Data validation procedures
  - Input data verification
  - Output data consistency checks
  - Error detection and handling

Analysis Methods

Your analysis should demonstrate rigorous evaluation of simulation results:

- Quantitative Analysis
  - Statistical measures and tests
  - Performance metrics calculation
  - Trend analysis and pattern recognition

- Qualitative Analysis
  - Behavioral observations
  - System dynamics interpretation
  - Pattern and anomaly identification
- Comparative Analysis
  - Cross-scenario comparisons
  - Baseline performance evaluation
  - Optimization opportunities identification

Visualization Requirements

Develop appropriate visualization methods for your simulation:

- Required Elements
  - Results visualization (graphs, charts, tables)
  - Statistical analysis plots
  - Performance comparison visualizations
- Optional Enhancements
  - Real-time simulation visualization
  - Interactive data exploration tools
  - 3D visualization (if applicable)

Validation Strategy

Implement a comprehensive validation approach:

- Technical Validation
  - Code correctness verification
  - Algorithm implementation accuracy
  - Performance benchmarking
- Model Validation
  - Theoretical model consistency
  - Comparison with existing research
  - Expert review (if applicable)
- Results Validation

  - Statistical validity
  - Reproducibility verification
  - Sensitivity analysis

Documentation Requirements

Maintain comprehensive documentation throughout the simulation process:

- Technical Documentation
  - Implementation details and architecture
  - Data collection methods
  - Analysis procedures
- Experimental Documentation
  - Scenario descriptions and parameters
  - Test cases and results
  - Validation procedures
- Results Documentation
  - Analysis findings
  - Visualization explanations
  - Conclusions and implications

Deliverables

- A comprehensive report detailing your model selection, implementation approach, and planned simulation methodology
- Initial UML diagrams showing the proposed system structure
- Preliminary implementation timeline and resource requirements
- Risk assessment and mitigation strategies

Appendices

Appendices

Glossary

Simulation Model: A computational representation of a system, process, or phenomenon that enables analysis of behavior over time.

Discrete-Event Simulation: A modeling paradigm where system state changes occur at distinct points in time, triggered by events.

Simulation Framework: Software tools, libraries, or platforms that provide foundational capabilities for building simulations.

Custom Implementation: A simulation built from scratch using programming languages and custom code rather than pre-built simulation tools.

Automation Script: Code that enables automatic execution of simulation runs, data collection, and analysis without manual intervention.

Data Collection Framework: Systematic approach to gathering, storing, and managing simulation data for analysis.

Validation: Process of verifying that a simulation accurately represents the intended system or behavior.

Verification: Process of confirming that the simulation implementation correctly follows its design specifications.

Sensitivity Analysis: Study of how changes in input parameters affect simulation outcomes.

Statistical Analysis: Application of statistical methods to analyze and interpret simulation results.

Frequently Asked Questions (FAQ)

Implementation Choices

1. **Can I use any programming language?**
    - Yes, you may use any programming language that can meet the project requirements
    - Consider language features, available libraries, and your expertise
    - Ensure your choice supports required data collection and analysis capabilities
2. **Can I use simulation software instead of coding everything?**
    - Yes, but you must implement automation through custom code
    - Your implementation must include programmatic data collection
    - Document how your solution meets all project requirements
3. **What if I want to use multiple tools or languages?**
    - This is acceptable if well-justified
    - Document integration approaches clearly
    - Ensure consistent data handling across tools

Project Requirements

4. **How complex should my simulation be?**
    - Focus on clear demonstration of fundamental concepts
    - Prioritize quality over complexity
    - Ensure thorough documentation and analysis
5. **Do I need a user interface?**
    - A UI is encouraged but not required
    - Focus on core simulation functionality first
    - Consider UI benefits for result visualization
6. **What are the minimum technical requirements?**
    - Automated simulation execution
    - Systematic data collection
    - Result analysis and visualization
    - Comprehensive documentation

Development Process

7. **How should I handle technical challenges?**
    - Document issues and attempted solutions
    - Consult course resources and instructor
    - Consider alternative approaches when stuck
8. **What if my chosen tool has limitations?**
    - Document limitations clearly
    - Implement workarounds where possible
    - Consider hybrid approaches if necessary
9. **How detailed should documentation be?**
    - Include setup and usage instructions
    - Document key design decisions
    - Provide analysis of results
    - Enable reproducibility of findings

Evaluation Criteria

10. **How will my project be evaluated?**
    - Technical implementation quality
    - Meeting project requirements
    - Documentation comprehensiveness
    - Analysis depth and rigor
    - Overall project presentation

Frequently Asked Questions (FAQ)

This section addresses common questions and concerns about the project. Refer to this FAQ for guidance before seeking additional assistance.

1. **What if I don't know how to create a UML diagram?**
    - Resources such as online tutorials and tools like Lucidchart, draw.io, or Visual Paradigm can help. Examples will also be provided in class materials.
2. **Can I use a programming language other than Python?**
    - Yes, you can use any suitable programming language, provided it supports the tools or frameworks required for your model and simulation.
3. **What tools are recommended for simulation?**
    - Popular tools include SimPy (Python), MATLAB, AnyLogic, and Arena. Choose one based on your familiarity and project needs.
4. **How detailed does my simulation model need to be?**
    - Focus on capturing the essential features and dynamics of the system. Overly complex models can be difficult to manage and may not provide additional value.
5. **What should I do if I encounter technical difficulties?**
    - First, consult the course resources and FAQs. If the issue persists, seek help from peers, instructors, or online communities relevant to the tool you are using.
6. **What if I don't have access to required software?**
    - Many tools offer free or educational licenses. Explore alternatives or consult your instructor for potential solutions.
7. **Can I modify my model after initial submission?**

- Yes, as long as changes align with project milestones and you document the revisions in your deliverables.

8. **How should I handle unexpected simulation results?**
   - Investigate the cause by reviewing your model, assumptions, and input data. Unexpected results can provide valuable insights and should be documented.
9. **What are the submission requirements?**
   - Follow the guidelines provided in the "Deliverables" section of this document. Ensure your files are well-organized and named correctly.
10. **How will I receive feedback?**
    - Feedback will be provided after each milestone submission, highlighting strengths and areas for improvement.

How to Make This Proposal

This section explains how to structure your research proposal. The document you create will evolve throughout the semester, serving as a living document for your project. By following this structure, you will also build the foundation for your final deliverable.

Organizing the Proposal

Your proposal should follow a logical order, ensuring clarity and consistency. Below is the recommended structure:

1. **Abstract**
   - Summarize the purpose, methods, and expected outcomes of your project in 150–250 words.
   - Write this last, after completing the other sections, to reflect the full scope of your proposal.
2. **Introduction**
   - Provide background information on the problem or system you are addressing.
   - Explain the importance of your research questions or hypotheses.
   - Clearly state the objectives of your project.
3. **Methodology**
   - Detail the approach you will take to model the system.
   - Include tools, frameworks, or programming languages you plan to use.

- Describe your plan for testing and validating the model.

4. **Implementation Details**
   - Outline your planned steps for implementation.
   - Include early deliverables, such as UML diagrams and initial model design.
   - Discuss challenges and proposed solutions.
5. **Experimentation and Simulation Runs** (Add Later in the Semester)
   - Document simulation runs, including scenarios, variables, and data collection methods.
   - Present your results and analyze them using appropriate techniques.
6. **Validation and Verification** (Add Later in the Semester)
   - Describe how you verified the accuracy and reliability of your model.
   - Include any comparison to real-world data or theoretical predictions.
7. **Conclusion**
   - Summarize the findings of your project.
   - Discuss implications and potential future work.

Deliverables

For this initial proposal (Milestone 1), you will submit:

- **Research Proposal (LaTeX Required):**
  - Problem/system identification and justification
  - Initial model selection and rationale
  - Proposed methodology and approach
  - Preliminary implementation plan
  - Expected outcomes and significance
- **Project Planning:**
  - Initial timeline for all project milestones
  - Preliminary resource requirements
  - Initial risk assessment

Future Milestones Overview

For context, your project will progress through these phases:

- **Setup Phase:**
  - M1: Project Proposal (Current)
  - M2: Literature Review and Model Documentation
  - M3: UML Design and Architecture
- **Implementation Phase:**
  - M4: Core Implementation (Major Milestone)
  - M5: Simulation Runs and Data Collection
  - M6: Analysis and Optimization
  - M7: Performance Evaluation
- **Completion Phase:**
  - M8: Final Documentation and Presentation

Note: All research paper components throughout the project must be prepared using LaTeX and submitted as PDFs. See the Documentation Format Requirements section for detailed guidelines.

Deliverables for the Proposal

For this initial proposal, you will need to submit:

- **A well-structured document:** Follow the guidelines in this section to create a clear and comprehensive proposal.
- **Preliminary research and ideas:** Provide sufficient background and initial thoughts to validate the project as viable.
- **Planned tools and methodologies:** Indicate which tools, languages, and frameworks you plan to use.

Note: Creating a public GitHub (or other source control tool) repository will be required by Milestone 4 (Model Implementation). While not mandatory at this stage, it is encouraged to start thinking about how you will organize and share your project for future development and showcasing.

Living Document

Throughout the semester, you will:

- Start with the initial sections (Abstract, Introduction, Methodology, Implementation Details).
- Gradually expand your proposal by adding the Experimentation, Validation, and Conclusion sections as the project progresses.
- Regularly update your document based on feedback and new insights.

Writing Tips

- Use clear, concise language and organize ideas logically.
- Include figures, diagrams, or tables where appropriate to enhance understanding.
- Cite all sources properly to maintain academic integrity.

This structure will not only help you succeed in this project but also provide a framework for future research proposals and technical documents.

Documentation Format Requirements

LaTeX Requirements

All research paper deliverables must be prepared using LaTeX:

- **Required Documents:**
  - Project proposal
  - Literature review
  - Model documentation
  - Progress reports
  - Final research paper
- **Format Flexibility:**
  - Single or double column format is acceptable
  - No specific requirements for headers/footers
  - No mandatory title page style
  - Bibliography/references must be included using BibTeX
- **Submission Requirements:**

  - Submit final PDF documents only
  - Source .tex files are not required for submission
  - Ensure all figures and tables are properly rendered in the PDF
  - Include all references in the final PDF

- **Content Guidelines:**
  - Use appropriate sectioning for document organization
  - Include figures and tables where relevant
  - Properly cite all external sources
  - Maintain consistent formatting throughout

Additional Documentation

Other project documentation may be submitted in appropriate formats:

- Code documentation in standard format for the chosen language
- README files in markdown format
- API documentation in language-appropriate format
- User guides in PDF format (LaTeX not required)

Note: While LaTeX source files are not required for submission, maintaining version control of your LaTeX documents is strongly recommended for managing updates and revisions throughout the semester. Using tools like Overleaf is recommended.