2-D Fluid Dynamics Simulator Application

Software Architecture Document

Version 1.0

Tech Geeks

Presented by:

Nagasindhu Kannekanti
Ayman Almusalam
Michael Chase Bonifant
Sai Krishna Kalakonda
Srikanth Reddy Bitla

Prepared for:

Principles of Software Engineering
FALL 2015

Document history

Date	**Version**	**Description**	**Author**
10/20/2015	Section 1	Introduction	Nagasindhu Kannekanti
10/20/2015	Section 2	Architectural requirements	Ayman Almusalam
10/20/2015	Section 3	Logical View	Michael Chase Bonifant
10/20/2015	Section 4	Implementation View	Sai Krishna
10/20/2015	Section 5	Deployment View	Srikanth

Distribution

Name	**0.01**													

Document approval

<Company>

[Provide signature and name of the signer]

…………………

<Supplier> *[provide the name of the supplying company]*

[Provide signature and name of the signer]

………………….

Contents

1. Introduction 4
    1.1 Purpose of this document 4
    1.2 References 4
    1.3 Document Overview 4

2. Architectural requirements 6
    2.1 Non-functional requirements 6
    2.2 Use Case View (functional requirements) 6

3. Logical View 7
    3.1 Tiers 7
    3.2 Subsystems 7
    3.3 Use Case Realizations 7

4. Implementation View 7
    4.1 Structure of the packages 7
    4.2 Realization of tiers 7
    4.3 (Re)use of components and frameworks 8

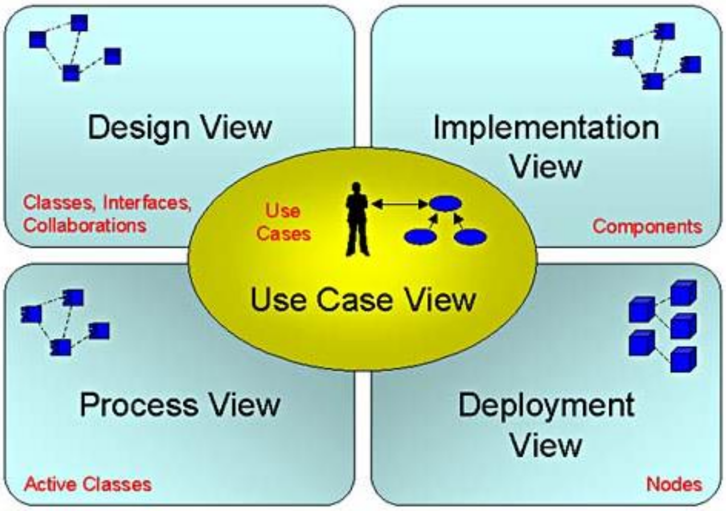
5. Deployment View 8

1. Introduction

1.1 Purpose of this document

The software Architecture Document can report the architectural overview of 2-D fluid dynamics simulator application. The main purpose of this document is to explain the different architectural views and features and Interfaces of the application.

The description makes use of the well-known 4+1 view model.



The 4+1 view model enables various stakeholders to establish the impact of the chosen architecture from their own perspective.

1.2 Scope

The single software product to be produced is the 2D Fluid Dynamics Simulator system. The system will provide simulations for single fluids in a configurable simulation/virtual chamber. It will show in 2D the flow density of the fluid being simulated and allow the user to specify basic environmental factors about the simulated environment like obstructions in the chamber and the viscosity of the fluid being simulated.

This tool will be beneficial as a learning tool for introducing students to fluid dynamics. It will not be useful in simulating large scale fluid systems (for instance, the flow pattern of run off from rain over a textured surface, the human circulatory system, or a city sewer system).

Once the Testing application has been deployed and is fully operational, any support or maintenance is out of the scope for this project.

1.3 References

https://www.ibiblio.org/e-notes/webgl/gpu/fluid.htm
http://www.kevinbeason.com/scs/fluid/
http://http.developer.nvidia.com/GPUGems/gpugems_ch38.html
http://physics.weber.edu/schroeder/fluids/
http://www.cims.nyu.edu/~billbao/report930.pdf
https://en.wikipedia.org/wiki/Lattice_Boltzmann_methods
"Fluid dynamics in Group T-3 Los Alamos National Laboratory:(LA-UR-03-3852)".

1.4 Document Overview

This document descibes the four views of the system. First one is the design view which describe the logical structure of the system and functionalities and interfaces that provides by the system to the users. Second, Implementation view, which illustrate the implementation of the logical view by describing the system component from the programmer perspective. Next, precess view, which descibes the analysis & Design discipline of the appilication. Finally, Deployment view, which demonstrate how the system is going to be deployed and which hardware is needed, and whether the system need other software in order to be deployed.

Chapter	Reader	Objective
0. **Error! Not a valid bookmark self-reference.**	Software Architect	Overview of architecturally relevant requirements.
3. Logical View	Developer	Knowledge of the application's conceptual structure, as a basis for technical designs.
4. Implementation View	Developer	Knowledge of the application's technical structure.
5. Deployment View	System Administrator roles	Knowledge of the way in which the application is deployed and (internal and external) communication takes place.

2 Architectural requirements

2.1 Non-functional requirements

Source	Name	Architectural relevance	Addressed in:
SRS Document	Performance	The system will give an accurate result.	Section 3.5.1
SRS Document	Reliability	There are no reliability specifications.	Section 3.5.2
SRS Document	Availability	The system is to be available as a self-contained desktop application.	Section 3.5.3
SRS Document	Security	There are no security specifications.	Section 3.5.4
SRS Document	Maintainability	The system will be easy to use and the developers can support the users with any changes.	Section 3.5.5
SRS Document	Portability	The system can work on either Windows or Linux operating system with high performance.	Section 3.5.6

2.2 Use Case View (functional requirements)

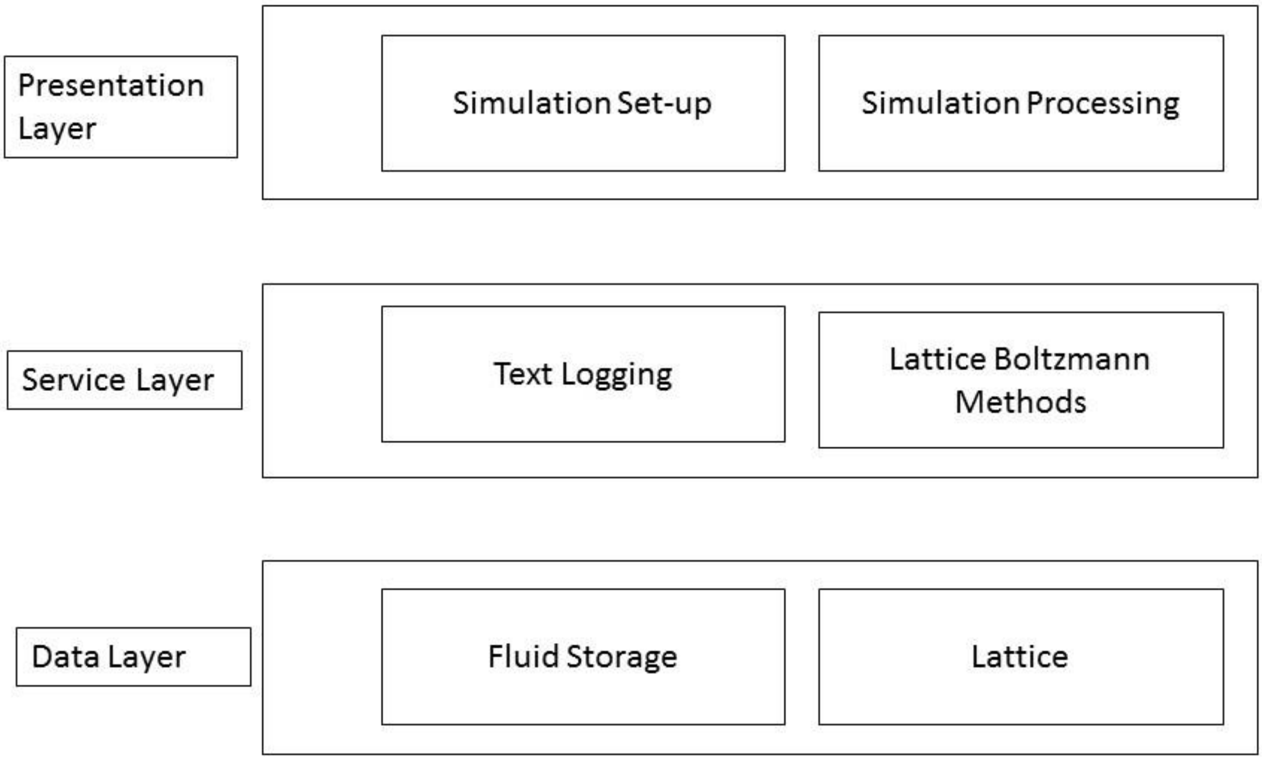
The overview below refers to architecturally relevant Use Cases from the Use Case Model (see references).

Source	Name	Architectural relevance	Addressed in:

Source	Name	Architectural relevance	Addressed in:
SRS Document	2-D Fluid Simulation	*The user has to input the visualize flow density dynamically*.	Section 3.2.1
SRS Document	Physical Parameters	The user can control in varying the viscosity, temperature, initial steady state flow speed, and perturbation force of the simulation.	Section 3.2.2
SRS Document	Fluid Selections	The user can select which liquid is going to use (water or glycerin).	Section 3.2.3
SRS Document	Spatial Configuration	The user will be able to select the objects to add to the simulation, via the GUI.	Section 3.2.4
SRS Document	On Demand Feedback	The user can select any point in the simulation chamber and immediately get feedback as to the flow values of that location in the simulation immediately.	Section 3.2.5
SRS Document	Monitoring Points	The user shall be able to select monitoring points in the simulation (flow meters) to show information during execution without affecting the flow of the fluid.	Section 3.2.6
SRS Document	Clear Start and End of Simulation	The user selects when to start the simulation and when to terminate it. There is no specification as to how this information is to be selected, beyond once it starts the user can terminate the program at any moment.	Section 3.2.7
SRS Document	Textual Logging	There is no specification as to inputs, (but presumably a name for each simulation to attribute to the generated text file is required).	Section 3.2.8
SRS Document	Instant Replay	The user can replay any simulation by reloading the simulation from its generated textual log file.	Section 3.2.9
SRS Document	Generic Error Handling	The Results of the simulation can be trusted and depends on the degree of uncertainty and on various errors	Section 3.2.10

3 Logical View

3.1 Tiers



The diagram shows different tiers of the system which consists of three tiers. The first tier is the GUI which enables the user to interact with the system. Second tier is service tier which is concerned with logging the simulation and providing the Lattice Boltzmann Methods. The third tier is the contains data storage, which holds the actual lattice the LBM is applied to and storage for defining the possible liquids that the simulator can process.

3.1.1 Presentation

The presentation layer defines the user interface, which is split between set-up for a simulation and the running and display of a simulation.

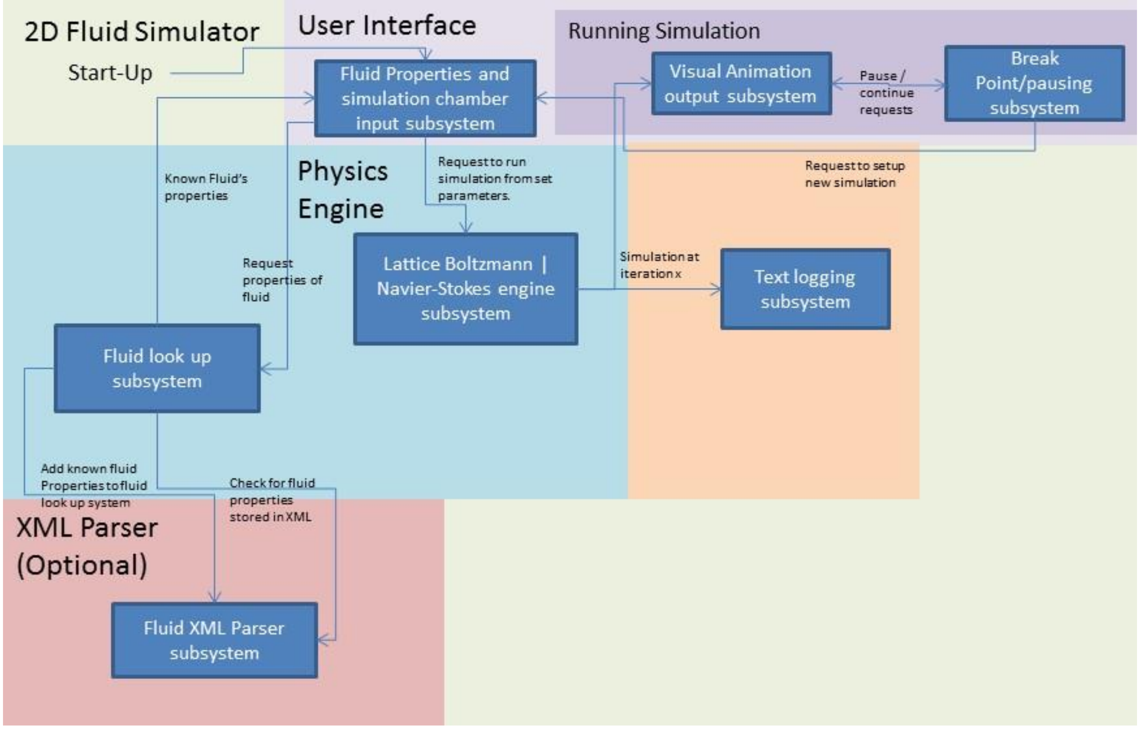
3.1.2 Service

The service layer contains the LBM and text logging the simulation provides. The LBM is the actual process by which the simulation is created and run. From the GUI the user can run this and pause it as they desire. The text logging records the Lattice at each iteration of the simulation.

3.1.3 Data

Below the services are the foundational data structures. There’s the actual Lattice structure that defines the simulation, and additionally a storage system for defining liquids, which the GUI uses to validate that liquids the users are try to simulate are valid.

3.2 Subsystems



There are three main subsystems to the system. The User Interface, the Physics Engine, and the Text Logger. Additionally an XML parser. The User Interface serves as a means for the user of the system to interact with the underlying physics engine, giving them a way to supply it inputs and start and stop a simulation. The physics engine takes the inputs and processes them in accordance with the Lattice Boltzmann Methods which define the how a simulation should run once inputs are given. The Text Logger saves the state of the simulation at each iteration of the LBM. The XML parser if implemented will add the additional functionality of giving the users the ability of filling out an XML form that defines optional additional fluids beyond the required water and glycerin that the system can simulate.

3.2.1 User Interface

The User Interface is split between two different modes or views. The first is a view that allows the user to define their liquid, the simulation chamber, and additional parameters before launching the simulation. The second view displays the simulation chamber as it runs, with features for pausing, restarting, and replaying the simulation.

3.2.2 Physics Engine/Fluid Simulator

The Physics Engine primarily defines the Lattice structure LBM methods that provide the fluid simulation. The physics engine additionally validates user given simulation parameters which confirms things like if the user is actually trying to simulate a liquid ie: it stops the user from simulating water above 100C or below 0C at 1atm since water isn't fluid under those conditions, its either solid or gaseous.

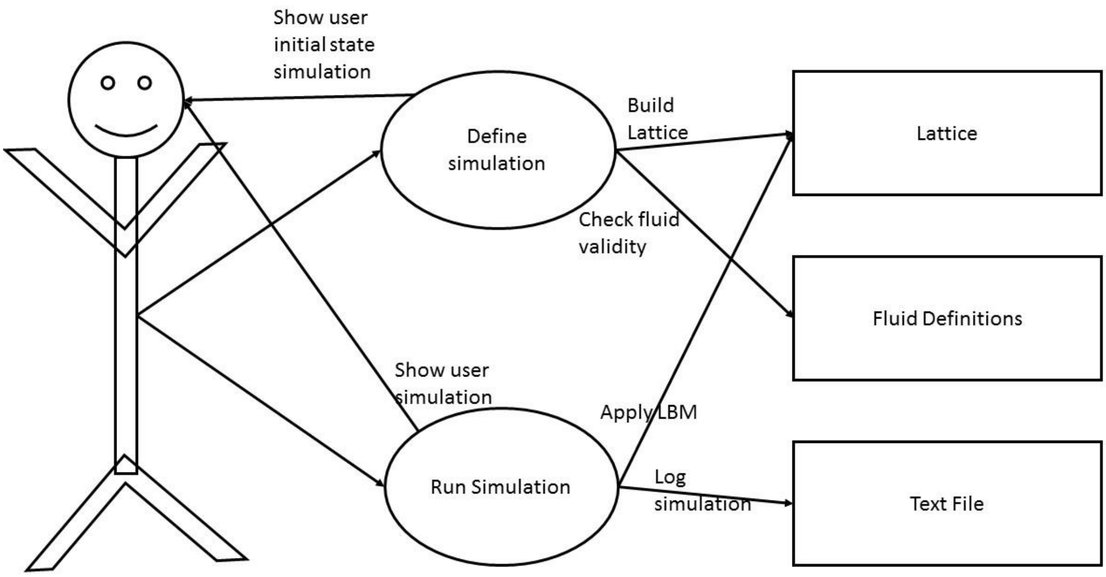
3.2.3 Text Logger

The Text Logger takes the Lattice structure from the physics engine and uses it to record the state of simulation at any given iteration across the simulation. It additionally is able to read the text files it's written and feed that back to the Physics Engine allowing the user to restore the simulation from any moment in the saved simulation.

3.2.4 XML Parser (optional)

The optional XML Parser would define a format for properly defining liquids beyond water and glycerin. Such files would be saved to a folder in the system and it would then be able to open them up at start up read them and allow the user to use these liquids in addition to those predefined for the system.

3.3 Use Case Realizations



The Liquid, the 2-D Fluid Dynamics Simulation is software that enables users to intrdouce themselves to fluid dynamics.

First users are able to define the initial state of a simulation. They can set up parameters selecting a fluid (water or glycerin) and can vary its temperature. They can also define the simulation chamber, initially just a blank cube, they can add cyldinrical and rectangular columns to the chamber to create obstructions to the flow of the fluid. They can additionally provide a starting velocity of the fluid, a sink, or a source for the fluid. They need not define these parameters if they've run the simulation before, they could then alternatively load a prior simulation from a text file.

Provided they've selected a valid fluid configuration, they can then run the simulation, at which point the simulation is displayed to the user, and the logging system starts recording the simulation at each iteration.

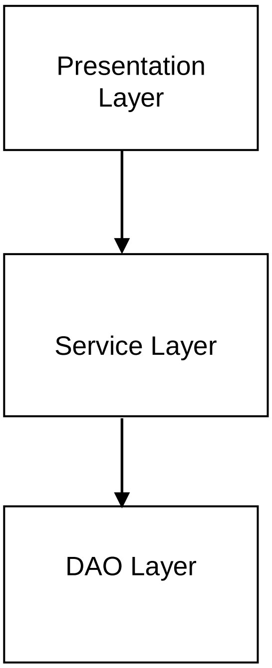
As the simulation runs the user can pause it at any given moment. While the simuation is running they can mouse over any spot on the simulation and view data about the fluid at the given point they've selected.

4 Implementation View

4.1 Structure of the packages

- com.simulator.gui
- com.simulator.physics
- com.simulator.logger

4.2 Realization of tiers



4.3 (Re)use of components and frameworks

The frame works which are used in this application are

- Hyper Text Markup Language(HTML)
- Cascade Style Sheet(CSS)
- JavaScript / JQuery
- Electron.atom.io

5 Deployment View

The simulator software would be deployed on any computer machine and no dependency with any web servers and web browsers.

Name	Type	Description
Computer	Node	Any Electric device that process data.