DOI 10.37882/2223-2966.2025.04.40

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ РЕАЛИЗАЦИИ ЗАДАЧИ ПРИНЯТИЯ РЕШЕНИЙ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ НЕЧЕТКОЙ ЛОГИКИ

DESIGNING AN INFORMATION SYSTEM FOR THE IMPLEMENTATION OF DECISION-MAKING PROBLEMS USING FUZZY LOGIC METHODS

N. Toichkin

Summary. The article presents a project aimed at developing an information system that facilitates work with fuzzy logic libraries, in particular Fuzzy Logic Sharp. The implemented client-server information system assumes the possibility of joint work on projects representing user decision-making models in various fields, their remote storage and access to data, as well as sending notifications to users. The design of software architecture is implemented using the MVVM pattern.

Keywords: decision making, fuzzy logic, soft computing, client-server system, MVVM pattern.

Тоичкин Николай Александрович
кандидат технических наук, доцент,
Мурманский арктический университет, г. Апатиты
toichkin@list.ru

Аннотация. В статье представлен проект, направленный на разработку информационной системы, облегчающей работу с библиотеками нечеткой логики, в частности Fuzzy Logic Sharp. Реализуемая клиент-серверная информационная система, предполагает возможность совместной работы над проектами, представляющими пользовательские модели принятия решений в различных областях, их удаленное хранение и доступ к данным, а также рассылки уведомлений пользователям. Проектирование программной архитектуры реализуется при помощи паттерна MVVM.

Ключевые слова: принятие решений, нечеткая логика, мягкие вычисления, клиент-серверная система, паттерн MVVM.

Введение

В общем смысле теория принятия оптимальных решений — это набор математических и численных методов, которые ориентированы на нахождение наилучших вариантов из множества возможных альтернатив. В идеальном сценарии принятие решений опирается на всеобъемлющую и необходимую информацию об объекте управления. Однако в реальности лица, принимающие решения, часто не имеют полных и точных данных об определенных факторах, которые могут существенно влиять на процесс выбора из множества альтернатив. Так как каждая ситуация, связанная с принятием решений, может подвергаться влиянию такого фактора, как неопределенность, которая относится к состоянию, когда точная вероятность различных параметров не может быть определена. Работа с неопределенностью является одним из самых сложных аспектов процесса принятия решений. Рассматриваемая проблема, а также потенциальные варианты выбора, как правило, неоднозначны, а имеющаяся информация по объекту неполна или полностью отсутствует. Принятие решений в условиях неопределенности, как правило, осуществляется с помощью следующих критериев: Лапласа, Вальда, Сэвиджа, Гурвица, MaxiMax (табл. 1).

Таблица 1.

Формулы нахождения оптимального решения

Критерий	Формула
Критерий Лапласа	$X = \max\left(\frac{\sum_{j=1}^{M} x_{ij}}{M}\right),\ i = 1.N$
Критерий Вальда	$X = \max(\min(x_{ij})), i = 1.N, j = 1.M$
Критерий Сэвиджа	$X = \min(\max(x_j - x_{ij})), i = 1.N, j = 1.M$
Критерий Гурвица	$X = \lambda \max(x_{ij}) + (1 - \lambda) \min(x_{ij}), i = 1.N, j = 1.M$
Критерий MaxiMax	$X = \max(\max(x_{ij})), i = 1.N, j = 1.M$

Главное различие между этими критериями определяется стратегией лица, принимающего решение [1]. Несмотря на разнообразие критериев, все они работают по одному принципу. Каждой альтернативе устанавливается соответствующая количественная оценка для каждой ситуации, а затем на основе сопоставления этих оценок происходит выбор оптимального решения. Совокупность оценок всех альтернатив по каждой из ситуации называется матрицей решений (матрица потерь, рисков, выигрышей, проигрышей).

Математическая теория нечетких множеств или, иными словами, Fuzzy Sets, а также нечеткая логика (Fuzzy Logic) — это обобщения классической теории множеств и классической формальной логики. Эти понятия были впервые предложены американским ученым Лотфи Заде (Lotfi Zadeh) [1]. Основной причиной появления этой теории стало наличие нечетких и приближенных рассуждений при объяснении человеком процессов, систем, объектов.

Основные проблемы, решаемые в нечетком моделировании, связаны с моделированием интеллектуальных операций приближенных рассуждений человека. Согласно теореме о нечеткой аппроксимации, любая математическая система может быть аппроксимирована системой, основанной на нечеткой логике [2]. Другими словами, с помощью естественно-языковых высказываний «ЕСЛИ-ТО», с последующей их формализацией средствами теории нечетких множеств, можно сколько угодно точно описать произвольную взаимосвязь «входы-выход» без использования сложного аппарата дифференциального и интегрального исчислений, традиционно применяемого в управлении и идентификации.

Важной характеристикой нечеткой логики является то, что любая теория T может быть фаззифицирована (fuzzified) и, следовательно, обобщена путем замены понятия четкого множества в T понятием нечеткого множества. Таким способом можно прийти к нечеткой арифметике, нечеткой топологии, нечеткой теории вероятностей, нечеткому управлению, нечеткому анализу решений. Выигрышем от фаззификации является большая общность и лучшее соответствие модели действительности. Однако с нечеткими числами труднее оперировать, чем с четкими числами. Более того, значения большинства нечетких понятий зависят от контекста и/или приложения [3]. Рассмотрим нечеткий вывод на примере механизма Мамдани (Mamdani). Это наиболее распространенный способ логического вывода в нечетких системах, в котором используется минимаксная композиция нечетких множеств. Данный механизм включает в себя последовательность действий, приведенную на схеме ниже (рис. 1).

Проектирование информационной системы

Общую архитектуру информационной системы можно представить в виде схемы, приведенной на рис. 2.

Основная система состоит из трех подсистем: база данных, хранящаяся на сервере, арендованный удаленный сервер и приложение, отвечающее за бизнес-логику сервера, и клиентское desktop-приложение, позволяющее пользователю работать с алгоритмами принятия решений и нечеткой логикой, а также сохранением данных о проекте на сервере и в базе данных.

Общую структуру классов можно представить в виде UML диаграммы (рис. 3).

Рис. 1. Общая схема логического нечеткого вывода

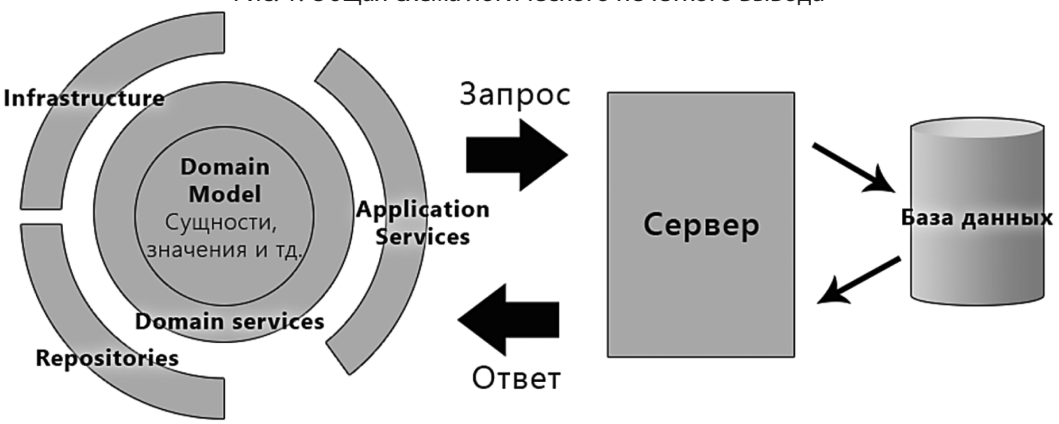

Рис. 2. Общая архитектура информационной системы

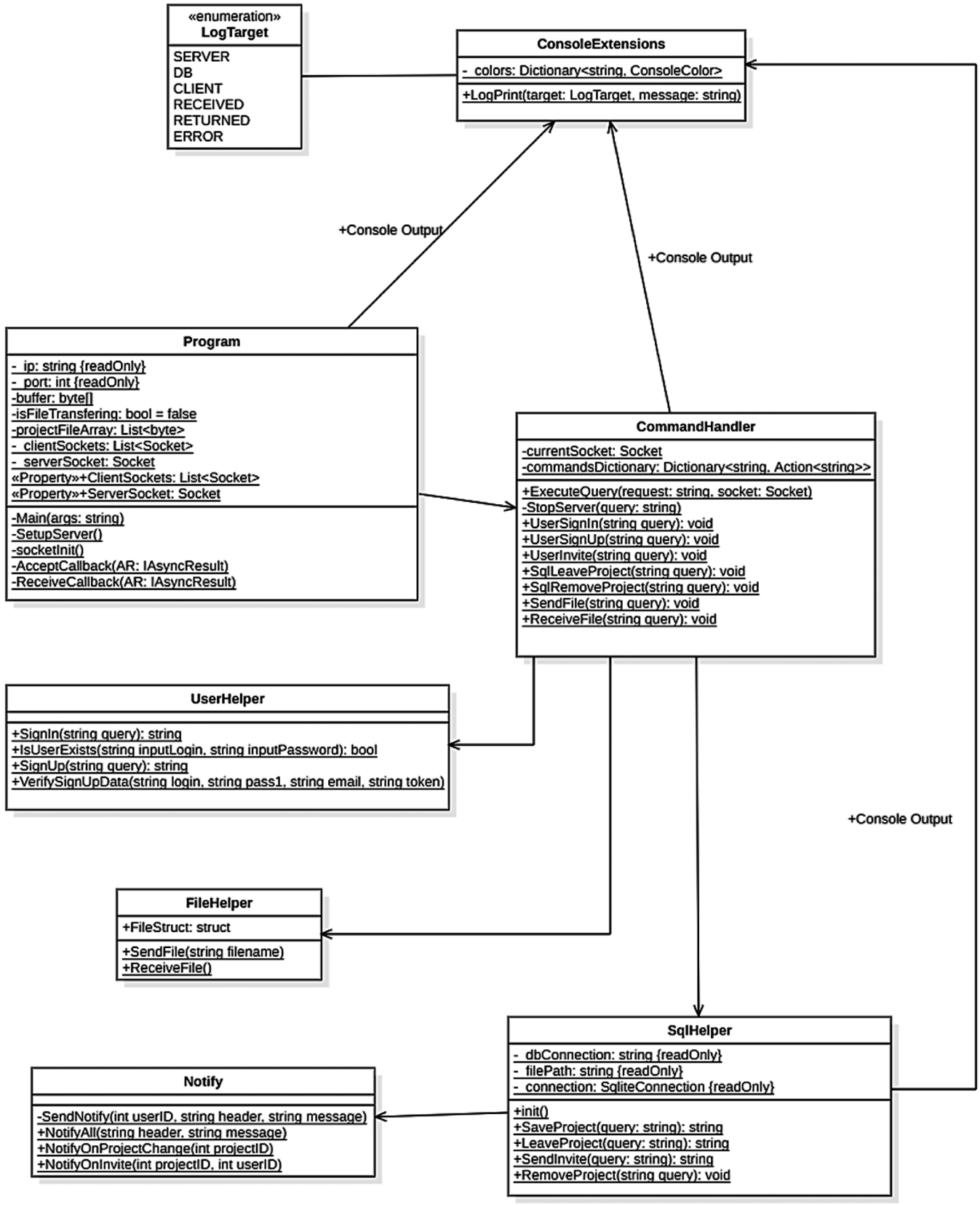

Рис. 3. Диаграмма классов серверного приложения

Сервер осуществляет логику взаимодействия с базой данных, авторизации/регистрации пользователей, рассылки уведомлений и выдачи клиенту необходимой информации в зависимости от запроса. На диаграмме помимо основного класса Program объявлено еще несколько классов, каждый из которых отрабатывает определенную логику работы серверной программы. Например, CommandHandler обрабатывает все входящие запросы и в зависимости от запроса, будь то команда серверу или SQL запрос, вызывает соответствующий команде метод в том или ином классе. А класс UserHelper отвечает за работу с авторизацией и регистрацией пользователей в системе.

База данных системы хранит информацию о пользователях, и их настройках, а также пути к пользовательским файлам, содержащим информацию о моделях и проектах, используемым при решении задач принятия решений.

Сама база хранится локально на сервере. В базе данных представлено 4 таблицы: пользователи, проекты, сводная таблица пользователь-проект и таблица шаблонов проектов (рис. 4).

Задачей клиентского приложения является удобное интерактивное взаимодействие с моделями принятия решений, в том числе и принятия решений на основе нечеткой логики. Расположение этого аппарата на клиентской стороне позволяет работать пользователю в режиме offline и не быть постоянно зависимым от подключения к серверу. Также, задачами клиента являются: сохранение и загрузка проектов — как локальных, так и проектов с сервера, формирование запросов серверу и обработка полученных ответов.

Клиентское приложение представляет собой desktop-приложение, реализующее архитектурный патерн MVVM [4] (рис. 5).

Пользовательский интерфейс посредством технологии связывания данных (binding) использует данные из класса ViewModel.

В статическом классе ViewModel есть экземпляр класса MainWindowViewModel, который представляет сущность проекта, над которым работает пользователь. Также, в классе ViewModel находится экземпляр класса UserData, в котором хранятся данные о пользователе, в том числе списки последних открытых проектов и проектов, хранящихся на сервере, представленных в виде коллекции объектов ProjAttributes.

В классе MainWindowViewModel определены поля и свойства, отражающие данные о проекте, а также экземпляр класса FuzzyHelper, отвечающий за работу с библиотекой нечеткой логики Fuzzy Logic Sharp.

Классы MainWindowViewModel и UserData наследуются от универсального класса json, для реализации возможности сериализации проекта и данных пользователя в cache-файл.

Класс Alternatives представляет коллекцию сущностей Alternative, для создания матриц решений и реализует стандартные методы управления коллекцией.

Класс Situations представляет сущность ситуации, необходимой для формирования матрицы решений и представляет собой коллекцию объектов Situation, и также реализует методы по управлению коллекцией.

Класс DecisionHelper осуществляет прогон матрицы решений по нескольким критериям и в качестве результата выводит пользователю самую часто встречающуюся в результатах вычислений по критериям альтернативу.

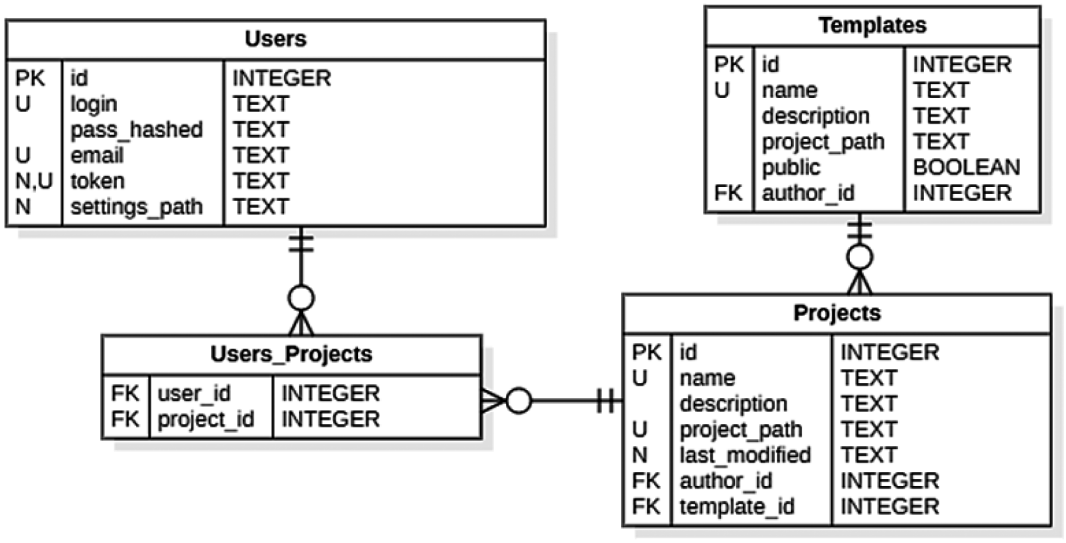

Рис. 4. Структура и связи таблиц базы данных

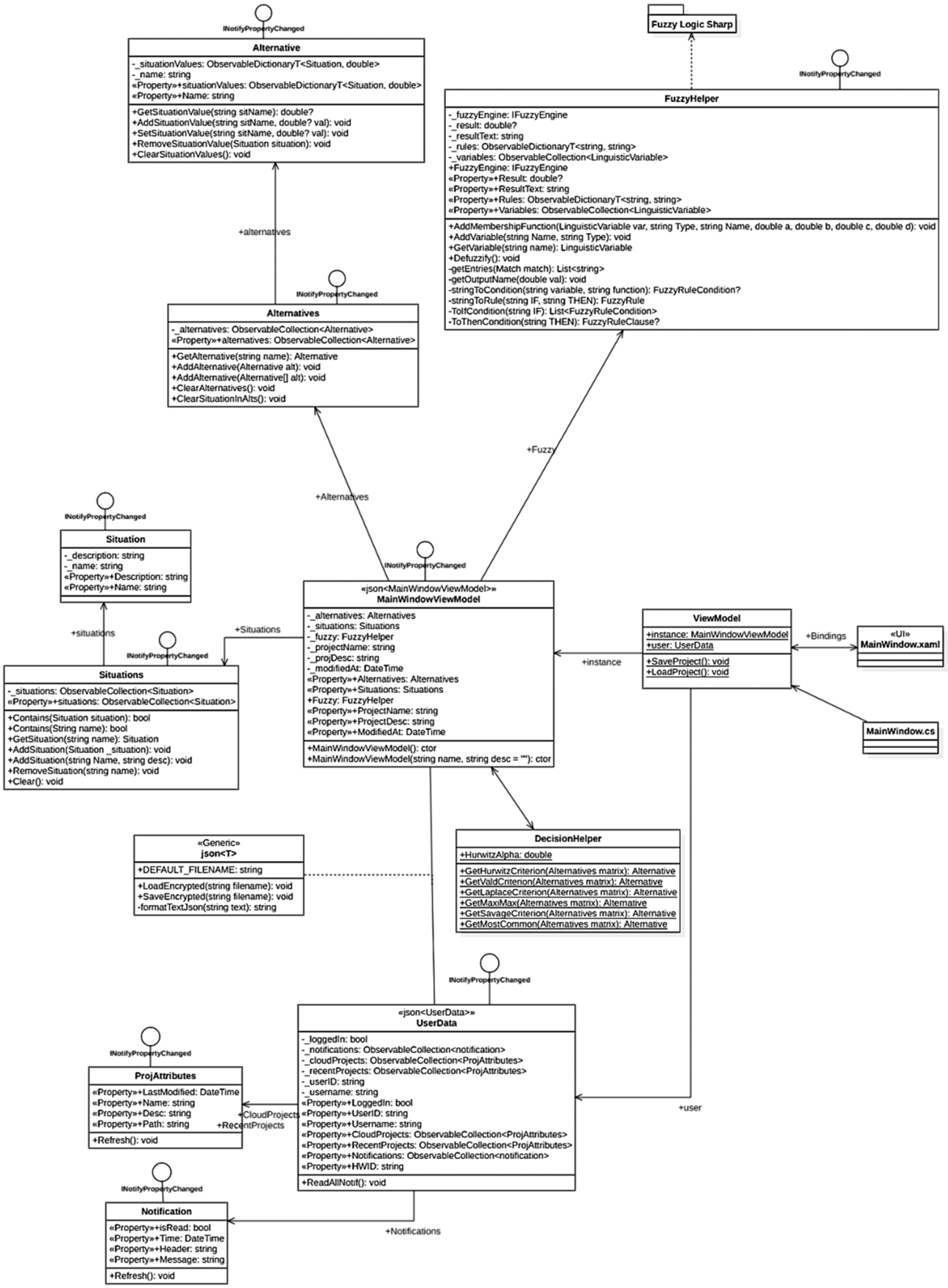

Рис. 5. Диаграмма классов клиентского приложения

Класс MainWindow.cs отвечает за логику управления окном.

Все классы, за исключением MainWindow.cs, а также коллекции и словари реализуют интерфейс INotifyPropertyChanged, используемый для уведомления клиентов об изменениях значений свойств. За счет чего достигается наиболее стабильная и удобная реализация паттерна MVVM.

Заключение

Достоинством подхода к принятию решений с помощью нечеткой логики является его близость к естественному языку, что позволяет пользователю формализовать свои нечеткие представления, приведя их в язык количественных оценок. Разрабатываемая в рамках данного проекта информационная система позволит разрешать неопределённость выбора, в различных предметных областях.

ЛИТЕРАТУРА

1. Машунин К.Ю. Моделирование технических систем в условиях неопределенности и принятие оптимального решения / К.Ю. Машунин, Ю.К. Машунин // Известия Российской академии наук. Теория и системы управления. — 2013. — № 4. — С. 19.
2. Борисов А.Н., Крумберг О.А., Федоров И.П. и др. Принятие решений на основе нечетких моделей. Рига: / «Зинатне». — 1990. — 184 С.
3. Тоичкин Н.А. Создание программного интерфейса для поддержки решения исследовательских задач в области мягких вычислений с использованием библиотеки Phyton Scikit-fuzzy / Н.А. Тоичкин, В.Н. Богатиков // Мягкие измерения и вычисления. — 2021. — Т. 43. — № 6. — С. 61–80.
4. MVVM: полное понимание (+WPF) Часть 1 // Habr URL: https://habr.com/ru/articles/338518/ (дата обращения: 30.11.2024).

Журнал «Современная наука: актуальные проблемы теории и практики»