УДК 004.8, 658.512.2

СИНТЕЗ ЭЛЕКТРОННОЙ СТРУКТУРЫ ИЗДЕЛИЯ НА ОСНОВЕ ЯЗЫКОВЫХ МОДЕЛЕЙ, ПРОГРАММИРОВАНИЯ В ОГРАНИЧЕНИЯХ И ТРАНСЛЯТОРОВ ПРОМЕЖУТОЧНЫХ ПРЕДСТАВЛЕНИЙ

Габриелян Г.А.

МИРЭА – Российский технологический университет, e-mail: gabrielyan@mirea.ru

В статье предложен подход к автоматизированному синтезу электронной структуры изделия (ЭСИ) на основе больших языковых моделей (LLM), программирования в ограничениях и трансляторов промежуточных представлений. Рассмотрена проблема преобразования неструктурированных естественно-языковых требований экспертов в формализованные машинно-читаемые представления, соответствующие стандартам ГОСТ Р 2.053-2023 и ГОСТ 2.054-2013. Ключевой особенностью подхода является использование LLM в качестве семантического преобразователя, генерирующего промежуточное представление (ПП) в соответствии с формальной контекстно-свободной грамматикой на основе JSON. Представлена архитектура системы, включающая модули семантического синтеза, валидации промежуточного представления и трансляторов в целевые форматы. В работе представлена модульная архитектура системы, включающая три основных уровня: взаимодействия, обработки и синтеза, а также данных и интеграции. Детально описаны компоненты синтеза, валидации промежуточного представления и трансляторов в форматы SQL и UML. Разработаны и экспериментально апробированы два алгоритма: базовый синтез ЭСИ и расширенный с применением подхода «человек в цикле» (human-in-the-loop) для итеративного уточнения требований. Практическая значимость работы заключается в возможности интеграции разработанных алгоритмов в системы автоматизации проектирования, технологической подготовки производства и управления жизненным циклом изделий на производственных предприятиях.

Ключевые слова: электронная структура изделия, большие языковые модели, трансляторы, промежуточные представления, системы управления базами данных, структурированный язык запросов, автоматизация проектирования

ELECTRONIC PRODUCT STRUCTURE SYNTHESIS BASED ON LANGUAGE MODELS, CONSTRAINT PROGRAMMING, AND INTERMEDIATE REPRESENTATION TRANSLATORS

Gabrielyan G.A.

MIREA – Russian Technological University, e-mail: gabrielyan@mirea.ru

The article proposes an approach to the automated synthesis of the electronic structure of a product based on large language models (LLM), constraint programming and intermediate representation translators. The problem of converting unstructured natural language requirements of experts into formalized machine-readable representations conforming to GOST R 2.053-2023 and GOST 2.054-2013 standards is considered. A key feature of the approach is the use of LLM as a semantic converter that generates an intermediate representationin accordance with a formal context-free grammar based on JSON. The architecture of the system is presented, which includes modules for semantic synthesis, validation of intermediate representation and translators into target formats. The paper presents a modular system architecture that includes three main levels: interaction, processing and synthesis, as well as data and integration. The components of synthesis, validation of intermediate representation, and translators into SQL and UML formats are described in detail. Two algorithms have been developed and experimentally tested: the basic synthesis of ESI and an extended one using the human-in-the-loop methodology for iterative refinement of requirements. The practical significance of the work lies in the possibility of integrating the developed algorithms into automation systems for design, technological preparation of production and product lifecycle management in industrial enterprises.

Keywords: electronic product structure, large language models, translators, intermediate representations, database management systems, structured query language, design automation

Введение

Современные подходы к интеграции конструкторских данных в рамках единой информационной среды производственного предприятия требуют перехода от ручного, текстового или графического описания структуры изделия к строго формализованному, машинно-читаемому представлению [1, 2]. Так, в работе [1] предложена

методика идентификации объектов структуры электронной модели изделия по формализованным параметрам для обеспечения возможности переноса части задач конструкторской и технологической подготовки производства с инженера на машину, в работе [2] рассмотрена методика автоматизированного проектирования технологической оснастки на основе формализованной электронной модели изделия.

Согласно ГОСТ Р 2.053-2023, электронная структура изделия (ЭСИ) представляет собой компьютерную модель изделия, описывающую изделие с точки зрения его функционирования, изготовления, эксплуатации, технического обслуживания, ремонта и других процессов жизненного цикла, в виде совокупности элементов структуры изделия и связей между ними. ЭСИ является электронным конструкторским документом, предназначенным для использования в вычислительной среде, содержащим описание изделия, иерархические отношения между его составными частями и другие данные в зависимости от назначения ЭСИ, предназначенным для организации информационного взаимодействия между автоматизированными системами [3, 4].

В рамках реализации единого информационного пространства производственного предприятия [6, 7] ЭСИ будет рассматриваться в качестве структурной основы представления электронного описания изделия (ЭОИ), которое представляет собой логически полную совокупность информационных объектов, содержащих всю необходимую информацию, описывающую конструкцию изделия и свойства (характеристики) изделия на конкретной стадии разработки. ЭОИ консолидирует техническую (конструкторскую) информацию об изделии и обеспечивает взаимосвязь данных, полученных в ходе выполнения стадий жизненного цикла изделия (ЖЦИ): проектирование (разработка) и изготовление, эксплуатация, ремонт и утилизация [5].

ЭОИ, как правило, физически реализуют в виде базы данных, содержащей все информационные объекты, из которых следует формировать совокупность электронных конструкторских документов, однозначно определяющей конструкцию изделия и его свойства (характеристики) – форму, размеры, материалы, а также способы и средства поддержки жизненного цикла изделия [5].

Процесс разработки нового изделия является основным процессом конструкторско-технологической подготовки производства. Жизненный цикл начинается со сбора требований, которые могут быть представлены в различной форме. После того как требования согласованы и утверждены, начинается этап эскизного проектирования. На данном этапе начинается проработка будущего изделия, чтобы получить общее представление о конструкции будущего изделия и о необходимых ключевых технических решениях. За эскизным проектированием следует этап технического проектирования, итогом которого выступает полная конструкция изделия и весь объем применяемых технических решений. Перед началом серийного производства изделия наступает этап разработки технической документации на изделие, включая необходимые чертежи узлов и деталей, материальные спецификации, технологические карты производственного процесса, описание процессов контроля и испытаний и прочее. Совокупность этих описаний формирует электронную структуру изделия, которая создается конструкторами и технологами производств с использованием информационных систем [8].

Последние достижения в области искусственных нейронных сетей и, в частности, больших языковых моделей (англ. Large Language Models, LLM), позволили автоматизировать решение ряда задач, связанных с обработкой естественного языка, ранее решавшихся вручную. Такие задачи включают задачи синтеза программ [9, 10], формальных спецификаций [11], документации к коду, рекомендаций по улучшению характеристик программных систем [12].

В этой связи перспективным направлением исследований является изучение возможности применения LLM для автоматизации прототипирования структур баз данных для хранения модели ЭСИ на основе требований экспертов, сформулированных в свободной форме на естественном языке. При этом, как показано в работе [11], для ускорения прототипирования, а также с целью исключения ошибок целесообразно использовать LLM для синтеза высокоуровневой спецификации с дальнейшей трансляцией этой спецификации в целевые форматы.

В данной работе предложен подход, основанный на применении LLM для автоматического преобразования неструктурированного естественно-языкового описания изделия, сформулированного рядом экспертов, в структурированное промежуточное представление (ПП) [11] с его последующим преобразованием в целевые форматы – от схем реляционных баз данных [13], используемых в вычислительных системах, до диаграмм архитектурного моделирования [14], используемых в документации.

Ключевым элементом предлагаемого в работе алгоритма синтеза модели ЭСИ является определение ПП как упорядоченного множества отношений, где каждое отношение (таблица) характеризуется набором атрибутов (столбцов) с заданными типами данных и, при необходимости, ссылками на другие отношения, что формально соответствует реляционной модели данных, широко использующейся на практике в реляционных системах управления базами данных (РСУБД) [13, 14]. Отличительной особенностью предлагаемого подхода является применение LLM не для генерации текста на естественном языке, а в качестве сложного функционального

преобразователя, способного интерпретировать семантику экспертных высказываний и представить её в виде структурных связей на языке, определяемом формальной грамматикой.

1. Алгоритм синтеза модели ЭСИ на основе требований экспертов на естественном языке

Формализация предметной области в предлагаемом подходе начинается с представления экспертных знаний в виде текстового описания на естественном языке, содержащего как декларативные утверждения (например, «корпус литой»), так и структурные зависимости (например, «радиатор является частью корпуса»). Для корректного преобразования таких высказываний в ПП вводится формальная грамматика [15], определяющая допустимые конструкции описания сущностей и связей.

В рамках данной работы для ПП используется упрощённая контекстно-свободная грамматика (КСГ) [16] на основе формата JSON (JavaScript Object Notation), определяемого в RFC 8259 [17], в которой терминальными символами выступают типы данных («int», «str», «bool», «float») и имена сущностей, а нетерминальными – конструкции вида «атрибут ссылается на поле другой сущности»:

$$\text{ПП} \rightarrow \text{"\{" E \{ "," E \} "\}"}, \quad (1)$$

$$\text{E} \rightarrow \text{N ": " "\{" S "\}"}, \quad (2)$$

$$\text{S} \rightarrow \text{C \{", " C\}}, \quad (3)$$

$$\text{C} \rightarrow \text{N ": " Y}, \quad (4)$$

$$\text{Y} \rightarrow \text{""" T [R] """}, \quad (5)$$

$$\text{T} \rightarrow \text{int} \mid \text{str} \mid \text{bool} \mid \text{float}, \quad (6)$$

$$\text{R} \rightarrow \text{I " " I}, \quad (7)$$

$$\text{N} \rightarrow \text{""" I """}, \quad (8)$$

$$\text{I} \rightarrow \text{(L | "\_") \{L | D | "\_"\}}, \quad (9)$$

где правило L соответствует букве русского или английского алфавита, а правило D – цифре от 0 до 9, в кавычки заключены терминальные символы языка, определяемого КСГ, а символы { и } без кавычек используются для обозначения повторений.

Правило ПП (1) определяет корневую структуру ПП, состоящую из одной или нескольких сущностей, структуру которых определяет правило E (2). Каждая сущность (2) характеризуется именем N (8) и набором атрибутов S (3). Тип Y (5) каждого атрибута C (4) может являться как простым типом данных T (6), так и ссылкой на поле другой сущности R (7).

Представленная грамматика позволяет однозначно интерпретировать промежуточное представление, сгенерированное LLM, как элемент языка, порождаемого этой грамматикой, что позволяет обеспечить детерминированность последующей трансляции ПП в целевые форматы, такие как SQL [18] или UML [13]. Такой подход, в отличие от подхода, основанного на использовании LLM не только для генерации ПП, но и для преобразования форматов данных, позволяет гарантировать корректность результатов генерации и соответствие целевых форматов ПП.

ПП в предлагаемом подходе представляет собой конечное множество:

$$\mathcal{R} = \{R_1, R_2, \dots, R_n\}, \text{ где } R_i = (A_i, F_i), \quad (10)$$

каждая сущность R_i $(i = \overline{1,n})$ представлена парой (A_i, F_i), где $A_i = \{a_{i1}, a_{i2}, \dots, a_{ik}\}$ – множество атрибутов сущности с заданными скалярными типами из множества $\mathcal{T} = \{\text{int}, \text{str}, \text{bool}, \text{float}\}$, а $F_i \subseteq A_i \times \mathcal{R} \times A_j$ – множество внешних ключей, задающих бинарные связи «многие-к-одному» между атрибутами текущей сущности и первичными ключами других сущностей, таким образом, ПП является частным случаем реляционной схемы данных.

Предлагаемый подход к генерации ПП на основе LLM основан на методе решения задач программирования в ограничениях «сгенерировать и проверить» [19, 20]: модель получает инструкцию, требующую выдать ответ в виде ПП, после чего сгенерированный ПП проверяется на соответствие формальной грамматике и другим ограничениям. В случае несоответствия сгенерированного ПП ограничениям процесс генерации повторяется до получения корректного ПП. LLM рассматривается не как «чёрный ящик», а как параметризованный оператор:

$$\mathcal{L}: \Sigma^* \rightarrow \mathcal{IR}, \quad (11)$$

где Σ^* – множество допустимых текстовых описаний предметной области, а $\mathcal{IR} \subseteq \mathcal{P}(\mathcal{E} \times \mathcal{A} \times \mathcal{T}^*)$ – пространство корректных ПП, соответствующих грамматике (1-9), где $\mathcal{E}$ – множество имён сущностей, $\mathcal{A}$ – множество имён атрибутов, $\mathcal{T} = \{\text{int}, \text{str}, \text{bool}, \text{float}\}$ – множество допустимых типов данных, $\mathcal{T}^*$ – расширенное множество типов данных, включающие ссылки вида $(t, e, a) \in \mathcal{T} \times \mathcal{E} \times \mathcal{A}$.

Тогда LLM может быть рассмотрено как условное распределение:

$$\mathcal{L}_\theta(S) = p_\theta(P|S), \quad (12)$$

где θ – параметры LLM; $p_\theta(P|S)$ – вероятность генерации ПП P при заданном запросе S; $S \in \Sigma^*$, $P \in \mathcal{IR}$.

В предложенном подходе используется ограниченная генерация, при которой LLM (11) вынуждена синтезировать ПП, придерживаясь синтаксиса, который соответствует КСГ подмножества JSON. Это достигается посредством дополнения экспертных требований запросом, который содержит следующую инструкцию q:

«Сгенерируй описание БД по описанию предметной области заказчика. В ответе приведи только описание таблиц в виде JSON-объекта типа dict[str, dict[str, str]] для машинного разбора, в котором ключами являются имена таблиц на русском языке без пробелов, а значениями — словари, в которых ключами являются имена колонок на русском языке, а значениями — стандартные типы данных языка Python в строковой форме ('int', 'str', 'bool', ...). Если поле одной таблицы ссылается на поле другой, то значение колонки помимо типа данных также содержит через пробелы имя таблицы и имя поля, на которое нужно сослаться, например: "int таблица поле". Ниже приведено описание предметной области по мнению ряда экспертов, в ответе необходимо учесть мнение каждого из экспертов».

Несмотря на стохастическую природу LLM (12), наличие ограничивающей инструкции q и проверка сгенерированного ПП P на соответствие ограничениям позволяет обеспечить корректность генерируемого ПП. Реализация таких ограничений эквивалентна применению отображения:

$$G: \Sigma^* \to \mathcal{IR}, \tag{13}$$

$$G(S) = \begin{cases} P, \ P = \mathcal{L}_\theta(\mathrm{S}) \wedge \text{ПП} \to^* P \\ \perp, \ \text{иначе} \end{cases}, \tag{14}$$

где Σ^* – множество допустимых текстовых описаний предметной области, $S \in \Sigma^*$, описание S включает ограничивающую инструкцию q, $\mathcal{IR}$ – множество ПП, соответствующих КСГ, $\perp$ – ошибка разбора, $\mathcal{L}_\theta$ – LLM с обученными параметрами θ, выражение ПП $\to^* P$ обозначает, что текст P выводится из правила ПП (1) в результате последовательного применения одного или нескольких правил вывода, в случае получения значения $\perp$, соответствующего ошибке запроса, отображение (14) предлагается выполнять повторно до получения корректного результата.

Алгоритм синтеза модели ЭСИ на основе требований экспертов на естественном языке показан на рис. 1.

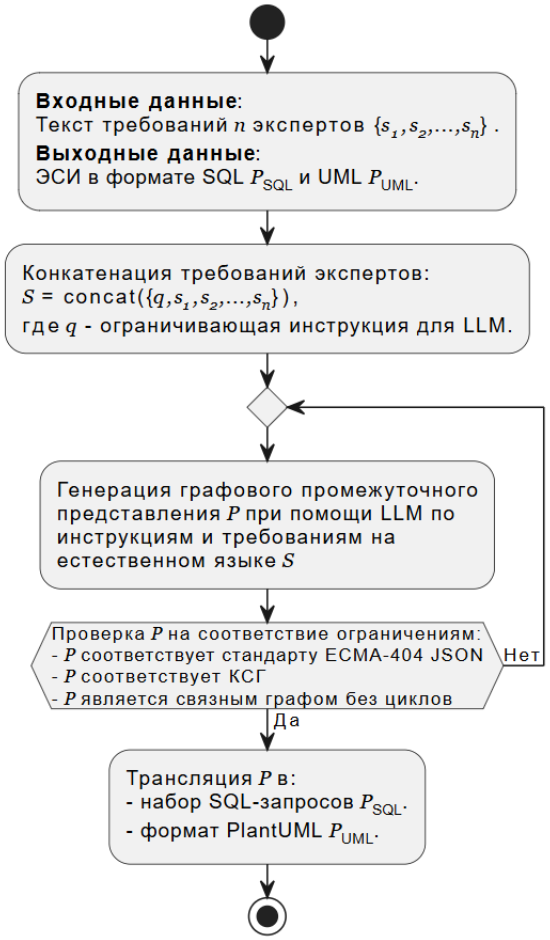

Рисунок 1 – Алгоритм синтеза модели ЭСИ на основе требований экспертов на естественном языке

В предложенном алгоритме синтеза модели ЭСИ поддерживается обработка требований к изделию нескольких экспертов на естественном языке. Применение этого алгоритма целесообразно в случае, если экспертные описания требований на естественном языке формулируются в рамках единого внутреннего стандарта предприятия. Сформулированные n экспертами требования к изделию $\{s_1, s_2, \dots, s_n\}$ и текст с инструкциями для LLM q объединяются в единый текстовый запрос $S = \{q, s_1, s_2, \dots, s_n\}$, что позволяет LLM выполнить семантическую агрегацию знаний: выявить пересекающиеся сущности (например, «корпус» у нескольких экспертов), разрешить потенциальные противоречия и интегрировать комплементарные требования (например, «герметичность» и «ребра охлаждения»). Выразительная мощность ПП ограничена реляционной моделью с целью обеспечения совместимости ПП с промышленными РСУБД и для упрощения интеграции предлагаемого подхода в существующие ИТ-ландшафты [12].

В разработанном алгоритме (см. рис. 1) задача интеграции данных декомпозирована на два независимых уровня: семантический, включающий интерпретацию текста на естественном языке и построение ПП, и синтаксический, включающий трансляцию ПП в целевые форматы SQL и UML. Это позволяет заменять LLM без переработки трансляторов ПП в набор SQL-запросов или в формат UML. Кроме того, набор целевых форматов также может быть расширен, не затрагивая ядро семантического анализа.

Транслятор ПП в SQL-схему реализует отображение:

$$\tau_{SQL}: \mathcal{IR} \to \Sigma^*_{SQL}, \tag{15}$$

где Σ^*_{SQL} – множество корректных SQL-выражений, $\mathcal{IR}$ – множество ПП из отображения (13).

Каждая сущность $R_i \in \mathcal{IR}$ транслируется в оператор CREATE TABLE, а её атрибуты – в столбцы с типами, определяемыми посредством фиксированного отображения $\phi: \mathcal{T} \to \mathcal{T}_{SQL}$ (например, int → INTEGER, bool → INTEGER). Если атрибут $a \in A_i$ содержит ссылку на поле другой таблицы, алгоритм генерирует ограничение FOREIGN KEY. При этом гарантируется, что все ссылочные сущности существуют в ПП, что обеспечивается на этапе проверки удовлетворения ПП указанным ограничениям (см. рис. 1). В результате формируется реляционная схема, готовая к развёртыванию в промышленной РСУБД без дополнительной корректировки.

Транслятор ПП в предметно-ориентированный язык описания UML-диаграмм PlantUML реализует отображение:

$$\tau_{UML}: \mathcal{IR} \to \Sigma^*_{PUML}. \tag{16}$$

Здесь каждая сущность представляется как элемент диаграммы классов с типом «entity», а атрибуты отображаются в виде строк с указанием типа данных. Связи «многие-к-одному» визуализируются с использованием нотации «||..o{» PlantUML.

Таким образом, два целевых формата взаимно дополняют друг друга: SQL обеспечивает возможность сохранения и обработки модели ЭСИ в вычислительных системах, PlantUML позволяет представить синтезированную модель в виде, понятном для эксперта, для последующей возможной корректировки или документирования результатов синтеза. Трансляторы обладают линейной вычислительной сложностью относительно числа атрибутов и связей в ПП, что делает их применимыми даже для сложных изделий с сотнями компонентов. Для поддержки других языков описания данных (JSON Schema, GraphQL, XML Schema) достаточно реализовать новое отображение τ_*, не затрагивая синтезатор ПП.

2. Оценка качества алгоритма синтеза модели ЭСИ на основе требований экспертов

Предложенный алгоритм был реализован на языке Python, причём для организации программного доступа к LLM использовался сервис OpenRouter [21]. В рамках экспериментального исследования предложенного алгоритма, показанного на рис. 2, на вход LLM DeepSeek Chat v3.1 [22] алгоритму подавались следующие требования:

Эксперт 1, параметр s_1: «*В основе изделия должен быть герметичный экранированный радиочастотный блок, внутри которого компактно размещён приёмопередатчик, усилитель мощности и дуплексер, чтобы минимизировать длину ВЧ-тракта. Этот усилитель мощности должен быть жёстко прикреплён к массивному алюминиевому радиатору, который является частью внешней стенки корпуса, для эффективного отвода тепла. Все внешние ВЧ-разъёмы, а это должны быть разъёмы типа N, должны быть напрямую, без переходов, установлены на стенке этого RF-блока. И обязательно нужен отдельный, хорошо экранированный отсек для блока цифровой обработки, чтобы его шумы не влияли на чувствительный приёмный тракт*».

Эксперт 2, параметр s_2: «*Конструктивно мы реализуем изделие в виде прочного литого алюминиевого корпуса с ребрами охлаждения по бокам, на него крепится радиатор от RF-блока. Внутри корпус разделён на три независимых отсека: первый – для RF-компонентов, второй – для плат контроллера и обработки данных, и третий – для блока питания и аккумуляторов. На задней стенке необходимо предусмотреть штатные точки*

крепления под кронштейн мачты, а снизу – кабельные вводы с сальниками для питания и данных. Вся конструкция должна собираться на основе несущей рамы, которая обеспечит жёсткость при вибрациях».

В результате выполнения алгоритма синтеза модели ЭСИ (см. рис. 1) с приведенными выше параметрами при помощи LLM было синтезировано ПП, которое затем было преобразовано в диаграмму «сущность-связь» (Entity Relationship, ER-диаграмма) [16] в результате трансляции ПП в формат PlantUML (см. рис. 2).

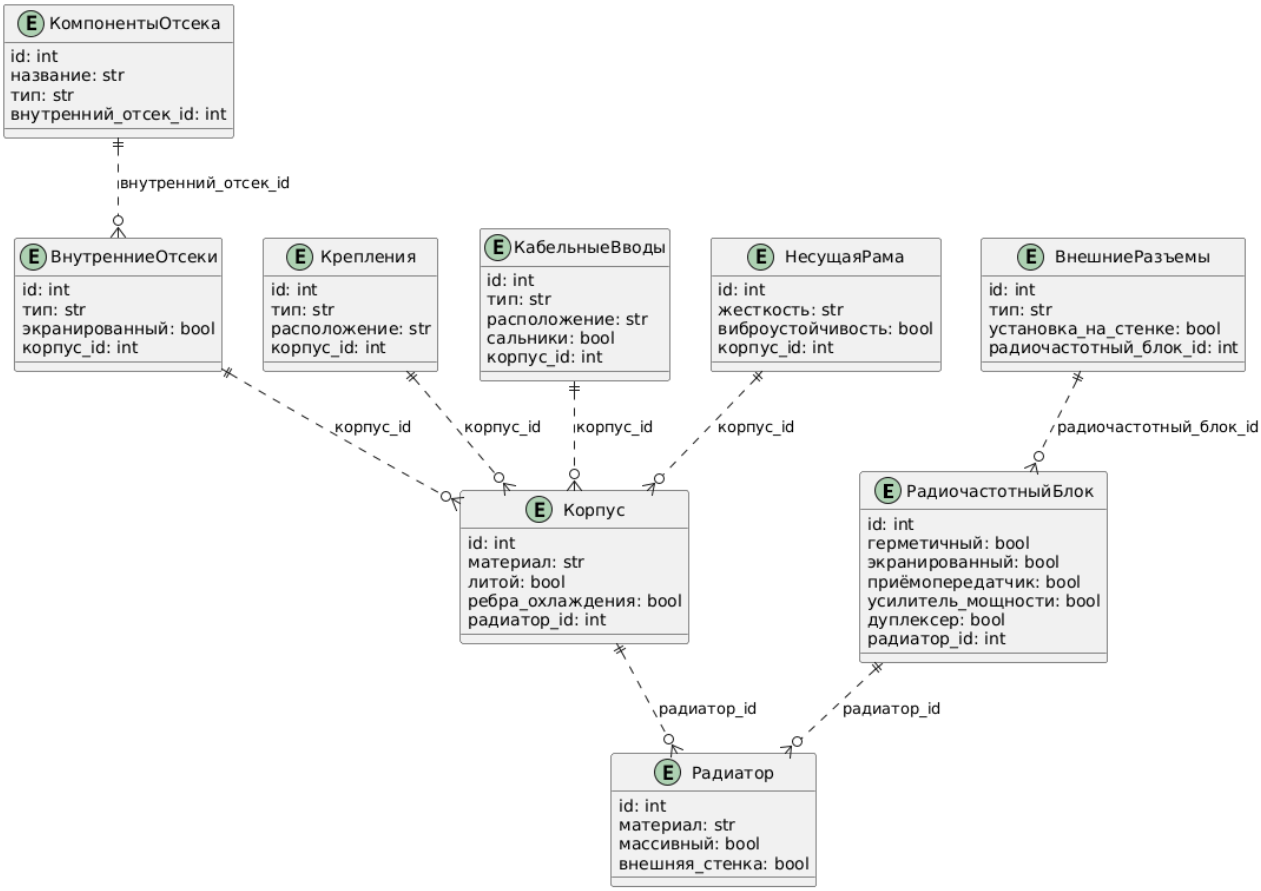

Рисунок 2 – ER-диаграмма, построенная по алгоритму синтеза модели ЭСИ на основе требований экспертов на естественном языке, в формате PlantUML

Кроме того, также была выполнена автоматическая трансляция синтезированного ПП в SQL-запрос, обеспечивающий создание таблиц SQLite [24] по ER-модели, показанной на рис. 2 и полученной из того же ПП:

```
CREATE TABLE "РадиочастотныйБлок" (
  "id" INTEGER,
  "герметичный" INTEGER,
  "экранированный" INTEGER,
  "приёмопередатчик" INTEGER,
  "усилитель_мощности" INTEGER,
  "дуплексер" INTEGER,
  "радиатор_id" INTEGER,
  FOREIGN KEY (радиатор_id) REFERENCES Радиатор(id));
CREATE TABLE "ВнешниеРазъемы" (
  "id" INTEGER,
  "тип" VARCHAR(255),
  "установка_на_стенке" INTEGER,
  "радиочастотный_блок_id" INTEGER,
  FOREIGN KEY (радиочастотный_блок_id) REFERENCES РадиочастотныйБлок(id));
...
```

Предложенный подход к синтезу модели ЭСИ на основе совместного использования LLM и трансляторов ПП в целевые форматы сравнивался с наивным подходом, основанным на применении LLM как для генерации ПП, так и для его трансляции в целевые форматы.

Было выполнено 10 попыток генерации SQL-кода и PlantUML-диаграмм по алгоритму, показанному на рис. 1, а также при помощи наивного подхода, в котором LLM использовалась как для генерации ПП, так и для его трансляции в целевые форматы. Была вычислена доля корректно сгенерированных PlantUML-диаграмм, доля

SQL-запросов, соответствующих сгенерированному ПП, время генерации SQL-кода и PlantUML-диаграмм.

Полученные результаты приведены в табл. 1.

Таблица 1 – Оценки качества подходов к синтезу модели ЭСИ на основе требований экспертов

Метрика	**Наивный подход на основе LLM**	**Алгоритм на рис. 1**
Доля корректных SQL-запросов	90%	100%
Доля корректных PlantUML-диаграмм	70%	100%
Время генерации SQL- и PlantUML-кода, с.	28,02 ± 13,79	15,60 ± 3,6

Как показано в табл. 1, при использовании наивного подхода, в котором LLM выполняет не только генерацию ПП, но и его трансляцию в целевой формат, доля корректно сгенерированных SQL-запросов и PlantUML-диаграмм отличается от 100% – в 1 случае из 10 LLM был синтезирован SQL-код и PlantUML-код не полностью, а в 2 случаях из 10 в синтезированном PlantUML-коде был допущен ряд ошибок, препятствующих корректной визуализации модели.

Алгоритм синтеза модели ЭСИ, показанный на рис. 1, в котором LLM выполняет только генерацию спецификации в виде ПП, удовлетворяющего ограничениям, позволяет всегда получать корректный SQL-код и PlantUML-код. Это связано с тем, что целевые форматы были сформированы алгоритмическим, а не вероятностным, методом. Предложенный алгоритм синтеза модели ЭСИ (см. рис. 1) также отличается от наивного подхода на основе LLM ускоренной генерацией результата.

Важно отметить, что эффективность предложенного подхода напрямую зависит от качества инструкций для LLM и способности той или иной LLM корректно интерпретировать инженерную терминологию и структурные зависимости. В рамках экспериментальной апробации алгоритма использовались экспертные описания механических и радиочастотных сборок, однако предложенный подход может использоваться и для других предметных областей.

3. Алгоритм синтеза верифицированной модели ЭСИ на основе требований экспертов и подхода human-in-the-loop

Алгоритм синтеза модели ЭСИ на основе требований экспертов на естественном языке, показанный на рис. 1, применим на практике в том случае, если экспертные описания требований формулируются в рамках единого внутреннего стандарта предприятия. Экспертам необходимо придерживаться единой терминологии (например, всегда использовать термин «корпус» вместо синонимов), в противном случае распределение случайной величины в (12) становится мультимодальным и требуется постобработка результатов генерации в виде SQL-запросов или UML-диаграмм с привлечением как экспертов, так и ИТ-специалистов.

Для решения обозначенной проблемы в работе предлагается алгоритм синтеза верифицированной модели ЭСИ на основе требований экспертов и подхода «человек в цикле» (human-in-the-loop, HITL) [25, 26]. В методе HITL человек активно участвует в процессе работы интеллектуальной системы, обеспечивая контроль, корректировку и совершенствование результатов, синтезированных LLM [26, 27]. В контексте данного исследования метод HITL реализуется как итеративный процесс взаимодействия эксперта с вычислительной системой в процессе синтеза модели ЭСИ.

Алгоритм синтеза верифицированной модели ЭСИ на основе требований экспертов и подхода HITL представлен на рис. 3.

Алгоритм синтеза верифицированной модели ЭСИ на основе требований экспертов и подхода HITL, показанный на рис. 3, начинает работу с получения входных данных в виде текстовых требований от n экспертов, обозначенных как $\{s_1, s_2, \dots, s_n\}$. Целью алгоритма является формирование модели ЭСИ, представленной в виде набора SQL-запросов PSQL и UML-диаграммы PUML.

Как и в алгоритме, показанном на рис. 1, в алгоритме синтеза верифицированной модели ЭСИ на первом шаге требования всех экспертов объединяются, а первым параметром указывается ограничивающая инструкция для LLM q, содержащая инструкции по синтезу ПП в формате, соответствующем КСГ.

Далее в цикле выполняется генерация графового промежуточного представления P с использованием LLM, которая обрабатывает инструкции и требования из строки с запросом S. После генерации представление P проходит проверку на соответствие стандарту ECMA-404 JSON и контекстно-свободной грамматике (1-7). Кроме того, проверяется связность графа. Если проверка проходит успешно, то P транслируется в формат PlantUML,

который затем проверяется на соответствие требованиям экспертом. Таким образом реализуется подход HITL – эксперт анализирует полученную модель в формате PlantUML (см. рис. 2) на предмет соответствия требованиям, и в случае необходимости предоставляет новое уточняющее требование h, которое соединяется со сформированным ранее запросом S к LLM функцией concat (см. рис. 3).

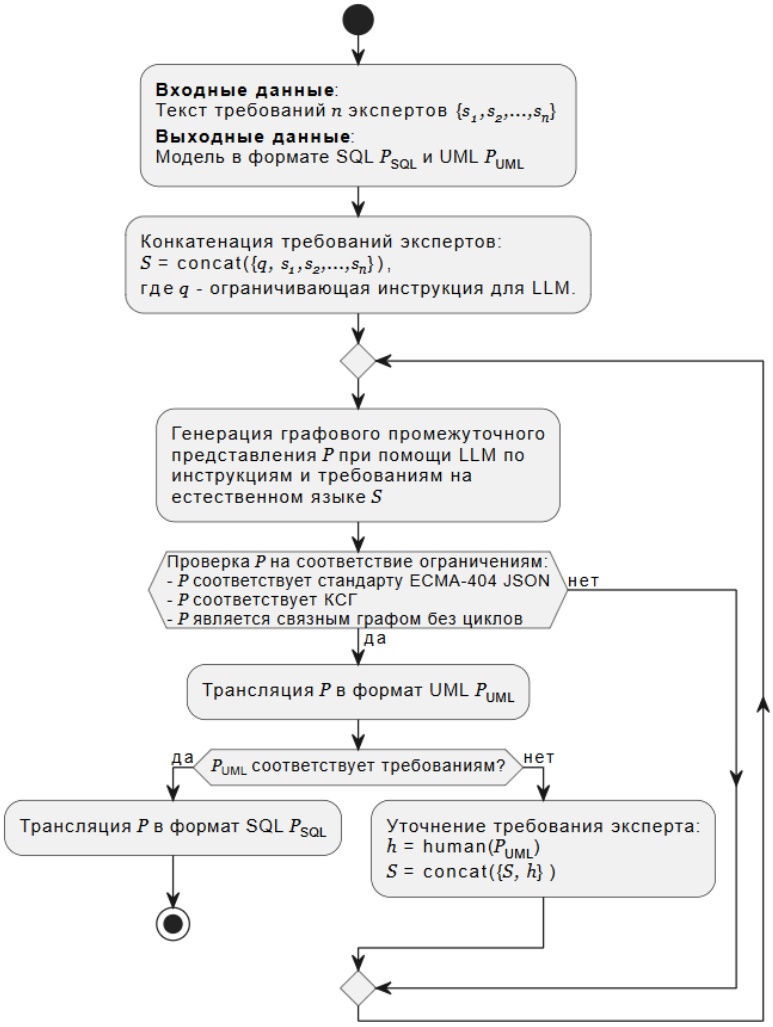

Рисунок 3 – Алгоритм синтеза верифицированной модели ЭСИ на основе требований экспертов и подхода human-in-the-loop

4. Апробация алгоритма синтеза верифицированной модели ЭСИ

В рамках апробации предложенного алгоритма, показанного на рис. 3, на вход его программной реализации на языке Python подавались те же требования к изделию, что и в разделе 2. Однако, после синтеза модели ЭСИ, на этапе уточнения требований эксперта запрос S дополнялся следующим текстом требований эксперта h на естественном языке:

Эксперт 3, параметр s_3: «*Усилитель мощности должен быть представлен отдельной сущностью и содержать численную характеристику мощности*».

В результате применения алгоритма синтеза верифицированной модели ЭСИ, показанного на рис. 3, была синтезирована ER-диаграмма, представленная на рис. 4.

Добавленная в результате уточнения требований сущность выделена на рис. 4 красным цветом.

Алгоритм синтеза модели ЭСИ на основе требований экспертов на естественном языке (см. рис. 1) рекомендуется применять в том случае, если участие экспертов в процессе проектирования прототипа электронной модели изделия невозможно или затруднено.

При возможности участия экспертов в итеративном процессе разработки для уточнения требований рекомендуется использовать алгоритм синтеза модели ЭСИ на основе подхода «человек в цикле» (human-in-the-loop) (см. рис. 3).

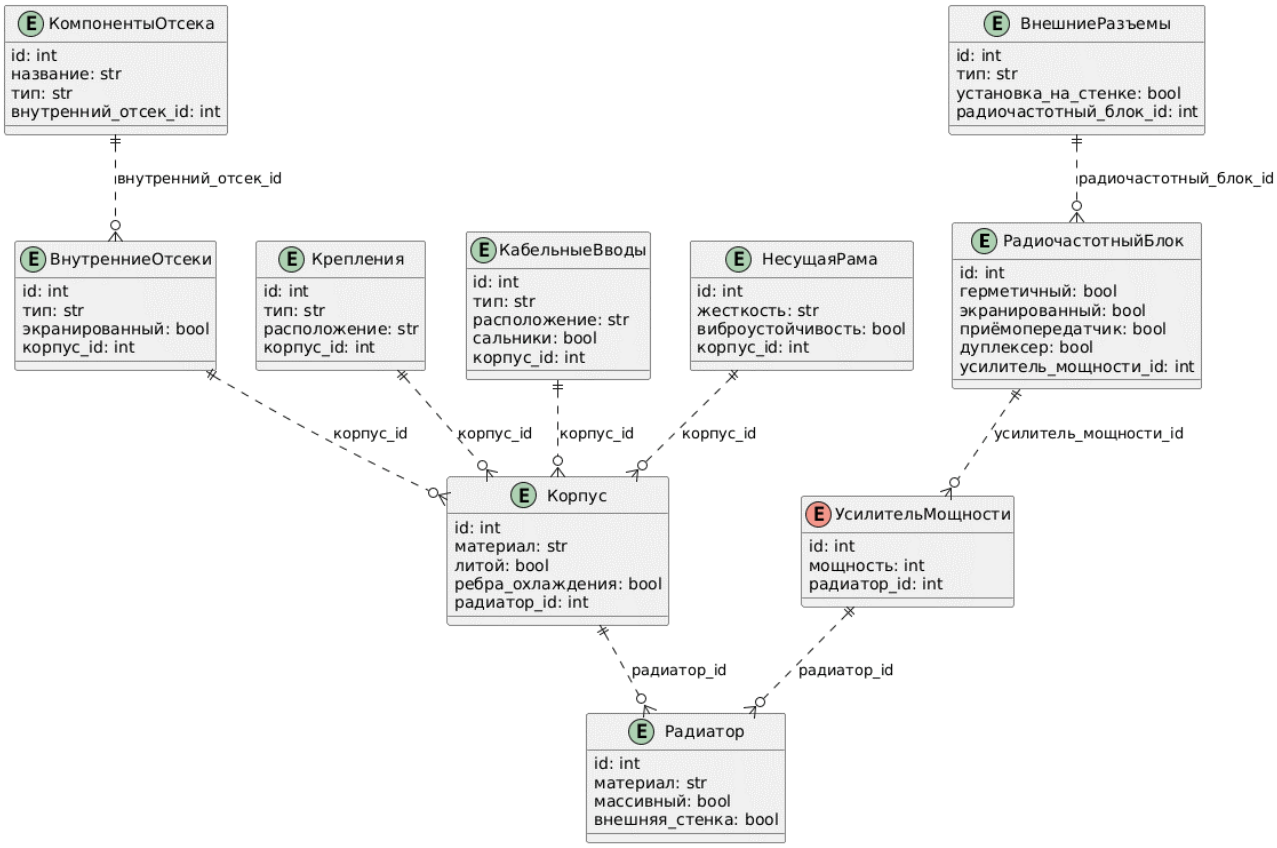

Рисунок 4 – ER-диаграмма, построенная по алгоритму синтеза верифицированной модели ЭСИ на основе требований экспертов на естественном языке и подхода human-in-the-loop, в формате PlantUML

5. Место разработанных алгоритмов в архитектуре системы синтеза электронного описания изделия

Предлагаемая система для синтеза электронной структуры изделия (ЭСИ) описывается модульной архитектурой, состоящей из трех основных уровней (см. рис. 5).

Уровень взаимодействия будет отвечать за интерфейсы системы. Прием и первичная обработка входных экспертных данных может быть реализована в виде отдельного компонента, позволяющего вводить текстовые требования, просматривать сгенерированные диаграммы и вводить уточнения в рамках цикла «human-in-the-loop». Может быть реализован программный интерфейс для прямой интеграции с используемыми в информационном пространстве предприятия информационными системами. На вход модуля «ввода экспертных требований» подается сформулированные n экспертами требования к изделию $\{s_1, s_2, \dots, s_n\}$, на выходе – стандартизированный текстовый запрос, включающий инструкции q и требования экспертов.

Уровень обработки и синтеза – ядро системы, где происходит преобразование требований в формальные модели. На данном уровнем происходит преобразование естественного языка в структурированное промежуточное представление (ПП). Компоненты модуля «синтеза»: LLM-Коннектор, обеспечивающий взаимодействие с внешними большими языковыми моделями, контекстный менеджер, формирующий промпты, включая системные инструкции, для обеспечения детерминированного формата вывода LLM, генератор промежуточного представления для реализации отображения $\Sigma^* \to \mathcal{IR}$.

Полученные ПП должны пройти проверку корректности: проверку связности графа, проверку соответствия заданной контекстно-свободной грамматике (формальные правила), проверку типов данных, уникальности имен сущностей и атрибутов. В случае ошибки модуль запускает цикл повторной генерации. При успехе передает валидированное ПП трансляторам.

Трансляторы преобразуют верифицированное ПП в пригодные для использования промышленные форматы. В статье описаны SQL-транслятор, реализующий отображение $\tau_{SQL}: \mathcal{IR} \to \Sigma^*_{SQL}$ и UML-транслятор, реализующий отображение $\tau_{UML}: \mathcal{IR} \to \Sigma^*_{PUML}$. Может быть реализован встроенный визуализатор PlantUML для преобразования текстового описания в графическую диаграмму, которая будет доступна эксперту в интерфейсе системы.

Электронное представление информации (данных) подразделяется на три вида: представление в виде БД, представление в виде файла, представление в визуальном виде, пригодном для восприятия человеком [5]. В связи

с этим уровень данных должен обеспечивать наличие хранилища моделей реляционной СУБД и хранилища документов для версионного хранения сгенерированных ПП, UML-диаграмм и истории запросов.

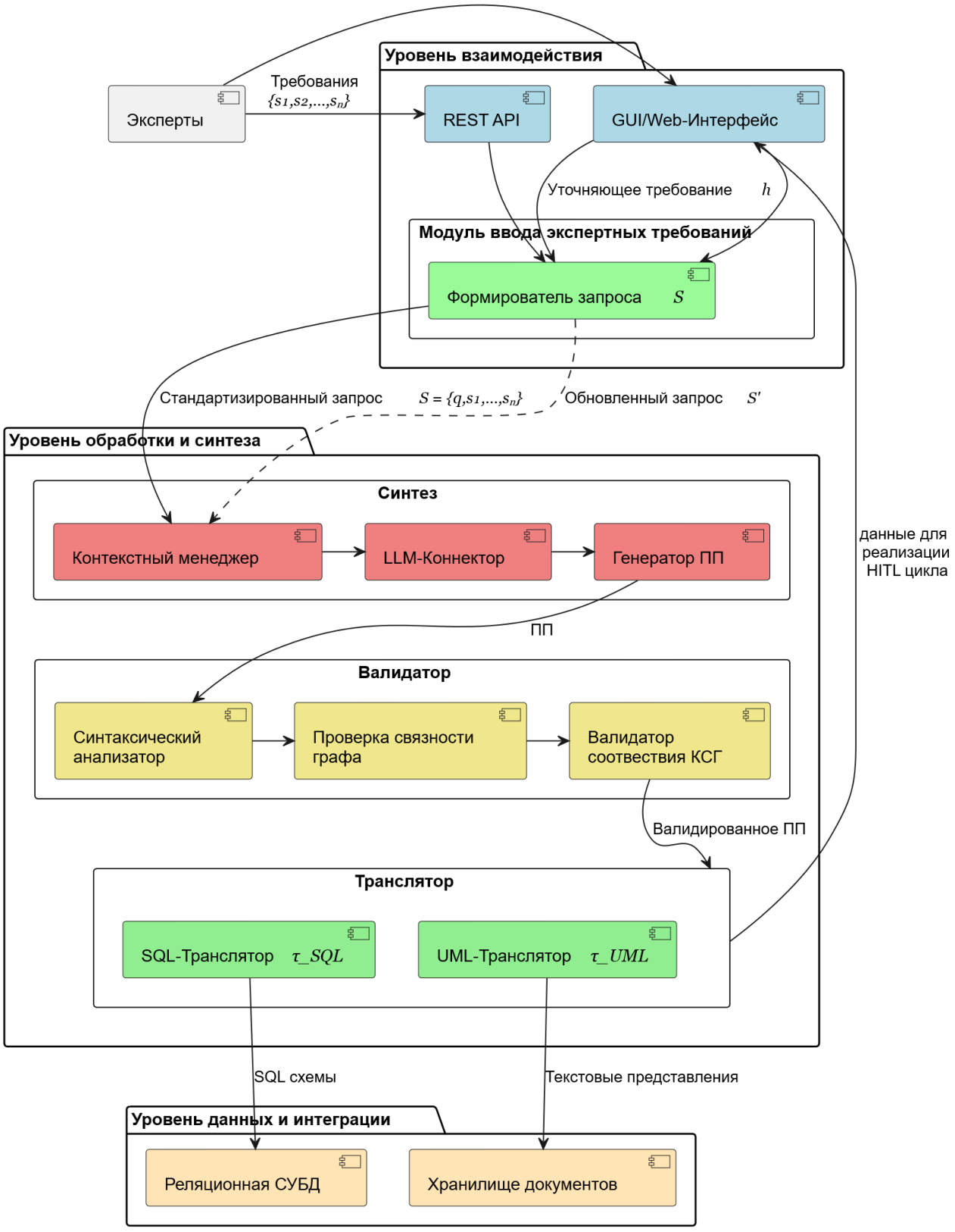

Рисунок 5 – Архитектура системы для синтеза электронной структуры изделия

Таким образом, эксперты взаимодействуют с предложенной программной системой при помощи пользовательского интерфейса (Graphical User Interface, GUI), указывая требования к изделию, после чего веб-приложение с GUI отправляет запрос к REST (Representational State Transfer) API (Application Programming

Interface) с перечнем требований, из которых формируется запрос по алгоритму 1 (см. рис. 1). Этот запрос поступает в модуль взаимодействия с LLM (LLM-коннектор на рис. 5), генерирующий ПП из естественно-языкового описания ЭСИ.

После чего сгенерированный ПП проверяется на соответствие формальной грамматике и другим ограничениям в соответствии с подходом «сгенерировать и проверить» [19, 20].

Два целевых формата, генерируемые на этапе трансляции, взаимно дополняют друг друга: SQL обеспечивает возможность сохранения и обработки модели ЭСИ в вычислительных системах, PlantUML позволяет представить синтезированную модель в виде, понятном для эксперта.

Предложенный в работе подход к синтезу электронной структуры изделия на основе языковых моделей, программирования в ограничениях и трансляторов ПП находит применение в различных областях промышленности и этапах жизненного цикла изделия.

Основные области применения предложенного подхода:

1) Автоматизация проектирования: формирование электронной структуры изделия и электронного описания изделия в соответствии с требованиями ГОСТ 2.054-2013 и ГОСТ Р 2.053-2023, прототипирование баз данных для систем управления жизненным циклом продукции, интеграция конструкторских данных в единую информационную среду производственного предприятия;

2) Технологическая подготовка производства: преобразование экспертных требований в формализованные представления, генерация реляционных схем данных для систем управления технологическими процессами, создание документации в виде UML-диаграмм для визуализации структуры изделия;

3) Поддержка жизненного цикла изделия: формирование электронного описания изделия как первоисточника информации, обеспечение согласованности данных на стадиях проектирования, производства и эксплуатации, поддержка процессов технического обслуживания и ремонта через структурированное представление данных;

4) Интеграция информационных систем: обеспечение взаимодействия между информационными системами и базами данных, трансляция промежуточных представлений в различные целевые форматы, поддержка миграции данных между различными системами управления.

Заключение

Предложенный в настоящей работе подход к синтезу электронного описания изделия на основе языковых моделей, программирования в ограничениях и трансляторов ПП предназначен для применения на различных стадиях жизненного цикла изделия.

ЭОИ, как основа информационной поддержки жизненного цикла изделия, должна обеспечивать:

– на начальных стадиях ЖЦИ (компьютерное представление результатов проведения научно-исследовательских и опытно-конструкторских работ);

– на стадиях разработки (представление формализованного описания конструкции изделия в форме неструктурированных электронных конструкторских документов или электронных конструкторских документов);

– на стадии технологической подготовки производства и самого производства (проектирование технологических процессов с использованием прикладных систем);

– на стадии сертификации (представление доказательной документации и данных, связанных с описанием конструкции изделия или его составных частей);

– на стадии эксплуатации (прямой доступ к описанию конструкции изделия, характеристик его составных частей для организации материально-технического обеспечения, гарантийного и послегарантийного обслуживания, а также ремонта);

– на стадиях ремонта и утилизации изделий (актуальные данные, связанные с конструктивными элементами изделия или изделием в целом).

В информационной основе ЭОИ лежит набор интегрированных информационных моделей – самого жизненного цикла изделия и выполняемых в его ходе процессов, касающихся непосредственно изделия (конечного продукта), производственной и эксплуатационной среды.

Основой реализации ЭОИ является база данных изделия (как правило, распределённая), непосредственно обеспечивающая информационную интеграцию всех информационных систем за счёт стандартизации формализованных описаний информационных объектов и рассмотрения совокупности данных и электронных документов как единого целого.

При этом использование ЭОИ как первоисточника информации исключает всевозможные ошибки и неточности как для конструкторской проработки (в т.ч. для различных видов анализа), так и для технологической подготовки производства, для нужд самого производства, для технической эксплуатации.

Предложенный подход позволяет автоматизировать создание ЭСИ как часть электронного описания изделия путём преобразования неструктурированных требований экспертов на естественном языке в формализованные представления баз данных и диаграмм архитектурного моделирования. Это обеспечивает:

– снижение трудозатрат на разработку структур баз данных для PLM/PDM систем;
– повышение качества проектирования за счёт формализации и верификации требований;
– ускорение процессов интеграции конструкторских данных в единую информационную среду предприятия;
– обеспечение содержательности данных между различными автоматизированными системами на протяжении всего жизненного цикла изделия.

Список литературы

1. Лаврентьева М.В., Говорков А.С. Идентификация объектов структуры электронной модели изделия с помощью выявленных формализованных параметров из конструкторско-технологической среды // Труды МАИ. 2017. №. 96. С. 29.
2. Лаврентьева М.В., Чимитов П.Е. Выборка параметров электронной модели изделия для автоматизированного проектирования технологической оснастки // Современные технологии. Системный анализ. Моделирование. 2016. №. 3 (51). С. 55-60.
3. ГОСТ Р 2.005-2023. Единая система конструкторской документации. Термины и определения.
4. ГОСТ Р 2.053-2023. Единая система конструкторской документации. Электронная структура изделия. Основные положения.
5. ГОСТ 2.054-2013. Электронное описание изделия. Общие положения.
6. Габриелян Г.А. Подходы к построению интегрированной информационной среды производственного предприятия // Новые информационные технологии в научных исследованиях. Материалы XXIX Всероссийской научно-технической конференции студентов, молодых ученых и специалистов. Рязань, 2024. С. 218-220.
7. Габриелян Г.А. Подходы к структурированному описанию бизнес-процессов в задачах создания информационно-семантической среды предприятия // Новые информационные технологии в научных исследованиях. Материалы XXVIII Всероссийской научно-технической конференции студентов, молодых ученых и специалистов. Рязань, 2023. С. 72-73.
8. Мироненко А.А. 1С:Академия ERP. Конструкторско-технологическая подготовка производства с использованием решений фирмы «1С». – М.: 1С-Паблишинг, 2024. – 253 с.
9. Murphy W., Holzer N., Qiao F., Cui L., Rothkopf R., Koenig N., Santolucito M. Combining LLM Code Generation with Formal Specifications and Reactive Program Synthesis // arXiv preprint arXiv:2410.19736. 2024.
10. Резуник Л., Александров Д.В. Подход к созданию сервиса генерации программного кода мобильных приложений с использованием больших языковых моделей. Электронный научный журнал «ИТ-Стандарт», 2024, № 4, С. 34-41.
11. Sobo A., Mubarak A., Baimagambetov A., Polatidis N. Evaluating LLMs for code generation in HRI: A comparative study of ChatGPT, Gemini, and Claude // Applied Artificial Intelligence. Taylor & Francis, 2025. Vol. 39, no. 1. P. 2439610.
12. Gorchakov A. V., Demidova L. A. An Approach to Generating Recommendations for Improving the Performance of Software for Heterogeneous Computing Platforms //International Journal of Open Information Technologies. – 2024. – Т. 12. – №. 4. – С. 68-76.
13. Максименко А. Н., Костылева В. В., Разин И. Б. Подходы к преобразованию реляционных баз данных в базы знаний // International Journal of Professional Science. 2021. №. 3. С. 55-62.
14. Koc H., Erdoğan A.M., Barjakly Y., Peker S. UML Diagrams in Software Engineering Research: A Systematic Literature Review // Proceedings. MDPI, 2021. Vol. 74. No. 1. P. 13.
15. Пентус, А. Е. Математическая теория формальных языков : учебное пособие / А. Е. Пентус, М. Р. Пентус. — 2-е изд. — Москва : ИНТУИТ, 2016. — 218 с.
16. Жегалин М. А., Яндринский В. В. Программный модуль для работы с контекстно-свободными грамматиками // Компьютерные инструменты в образовании. 2024. №. 2. С. 72-84.
17. Bray T. RFC 8259: The JavaScript object notation (JSON) data interchange format. 2017.

18. Khan W., Kumar T., Zhang C., Raj K., Roy A.M., Luo B. SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review // Big Data and Cognitive Computing. 2023. Т. 7. №. 2. P. 97.

19. Нариньяни А.С., Телерман В.В., Ушаков Д.М., Швецов И.Е. Программирование в ограничениях и недоопределенные модели. Информационные технологии. 1998. Т. 7. С. 13–22.

20. Советов П.Н., Горчаков А.В. Модели и алгоритмы автоматической генерации учебных задач по работе с системой git и разрешению зависимостей между пакетами. Электронный научный журнал «ИТ-Стандарт», 2025, № 3, с. 47-57.

21. Vijayakumar P., Pyingkodi M., Devi S., Thulasidharan P., Santhosh S., Samyuktha K. Comparative Analysis of AI Chatbot for Assessing Gemini AI, DeepSeek AI, and Qwen AI via OpenRouter API Integration // Proceedings of the 2025 6th International Conference on Data Intelligence and Cognitive Informatics (ICDICI). IEEE, 2025. P. 01-08.

22. Liu A. et al. Deepseek-v3 Technical Report // arXiv preprint arXiv:2412.19437. 2024.

23. Бабанов А. М. Методика структуризации данных в семантических моделях типа «Сущность-Связь» // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2022. №. 60. С. 93-101.

24. Gaffney K. P. et al. SQLite: past, present, and future //Proceedings of the VLDB Endowment. 2022. Т. 15. №. 12.

25. Mosqueira-Rey E. et al. Human-in-the-loop machine learning: a state of the art // Artificial Intelligence Review. 2023. Т. 56. №. 4. С. 3005-3054.

26. Wu X. et al. A survey of human-in-the-loop for machine learning. Future Generation Computer Systems. 2022. Т. 135. С. 364-381.

27. Kumar S. et al. Applications, challenges, and future directions of human-in-the-loop learning IEEE Access. 2024. Т. 12. С. 75735-75760.

References

1. Lavrentieva M.V., Govorkov A.S. Identification of objects of the electronic product model structure using identified formalized parameters from the design and technological environment // Proceedings of MAI. 2017. No. 96. P. 29.

2. Lavrentieva M.V., Chimitov P.E. Selection of parameters of the electronic product model for computer-aided design of technological equipment // Modern technologies. System analysis. Modeling. 2016. No. 3 (51). pp. 55-60.

3. GOST R 2.005-2023. A unified system of design documentation. Terms and definitions.

4. GOST R 2.053-2023. A unified system of design documentation. The electronic structure of the product. The main provisions.

5. GOST 2.054-2013. Electronic description of the product. General provisions.

6. Gabrielyan G.A. Approaches to building an integrated information environment of a manufacturing enterprise // New information technologies in scientific research. Materials of the XXIX All-Russian Scientific and Technical Conference of Students, Young Scientists and specialists. Ryazan, 2024. pp. 218-220.

7. Gabrielyan G.A. Approaches to the structured description of business processes in the tasks of creating an information and semantic environment of an enterprise // New information technologies in scientific research. Proceedings of the XXVIII All-Russian Scientific and Technical Conference of Students, Young Scientists and Specialists. Ryazan, 2023. pp. 72-73.

8. Mironenko A.A. 1C:Academy of ERP. Design and technological preparation of production using 1C solutions. Moscow: 1C-Publishing, 2024. 253 p.

9. Murphy W., Holzer N., Qiao F., Cui L., Rothkopf R., Koenig N., Santolucito M. Combining LLM Code Generation with Formal Specifications and Reactive Program Synthesis // arXiv preprint arXiv:2410.19736. 2024.

10. Rezunik L., Alexandrov D.V. Approach to Creating the Service for Code Generation of Mobile Applications Using the Large Language Models. Electronic Scientific Journal "IT-Standard", 2024, No. 4, pp. 34-41.

11. Sobo A., Mubarak A., Baimagambetov A., Polatidis N. Evaluating LLMs for code generation in HRI: A comparative study of ChatGPT, Gemini, and Claude // Applied Artificial Intelligence. Taylor & Francis, 2025. Vol. 39, No. 1. P. 2439610.

12. Gorchakov A. V., Demidova L. A. An Approach to Generating Recommendations for Improving the Performance of Software for Heterogeneous Computing Platforms //International Journal of Open Information Technologies. – 2024. – Vol. 12. – No. 4. – pp. 68-76.

13. Maksimenko A. N., Kostyleva V. V., Razin I. B. Approaches to the transformation of relational databases into

knowledge bases // International Journal of Professional Science. 2021. No. 3. pp. 55-62.

14. Koc H., Erdoğan A.M., Barjakly Y., Peker S. UML Diagrams in Software Engineering Research: A Systematic Literature Review // Proceedings. MDPI, 2021. Vol. 74. No. 1. P. 13.

15. Pentus, A. E. Mathematical theory of formal languages : a textbook / A. E. Pentus, M. R. Pentus. — 2nd ed. - Moscow : INTUIT, 2016. — 218 p.

16. Zhegalin M. A., Yandrinsky V. V. A software module for working with context-free grammars // Computer tools in education. 2024. No. 2. pp. 72-84.

17. Bray T. RFC 8259: The JavaScript object notation (JSON) data interchange format. 2017.

18. Khan W., Kumar T., Zhang C., Raj K., Roy A.M., Luo B. SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review // Big Data and Cognitive Computing. 2023. vol. 7. No. 2. P. 97.

19. Narinyani A.S., Telerman V.V., Ushakov D.M., Shvetsov I.E. Programmirovaniye v ogranicheniyakh i neoopredelennyye modeli. Informatsionnyye Tekhnologii. 1998. vol. 7. pp. 13–22.

20. Sovietov P.N., Gorchakov A.V. Models and Algorithms for Automatic Generation of Educational Problems on Git System Usage and Package Dependency Resolution. Electronic Scientific Journal "IT-Standard", 2025, No. 3, pp. 47-57.

21. Vijayakumar P., Pyingkodi M., Devi S., Thulasidharan P., Santhosh S., Samyuktha K. Comparative Analysis of AI Chatbot for Assessing Gemini AI, DeepSeek AI, and Qwen AI via OpenRouter API Integration // Proceedings of the 2025 6th International Conference on Data Intelligence and Cognitive Informatics (ICDICI). IEEE, 2025. P. 01-08.

22. Liu A. et al. Deepseek-v3 Technical Report // arXiv preprint arXiv:2412.19437. 2024.

23. Babanov A.M. Methodology of data structuring in semantic models of the "Essence-Connection" type // Bulletin of Tomsk State University. Management, computer engineering and computer science. 2022. No. 60. pp. 93-101.

24. Gaffney K.P., Prammer M., Brasfield L., Hipp D.R., Kennedy D., Patel J.M. SQLite: Past, Present, and Future // Proceedings of the VLDB Endowment. 2022. Vol. 15. No. 12.

25. Mosqueira-Rey E., Hernández-Pereira E., Alonso-Ríos D., Bobes-Bascarán J., Fernández-Leal Á. Human-in-the-loop Machine Learning: a State of the Art // Artificial Intelligence Review. 2023. Vol. 56. No. 4. pp. 3005-3054.

26. Wu X., Xiao L., Sun Y., Zhang J., Ma T., He L. A Survey of Human-in-the-loop for Machine Learning // Future Generation Computer Systems. 2022. Vol. 135. pp. 364-381.

27. Kumar S., Datta S., Singh V., Datta D., Singh S.K., Sharma R. Applications, Challenges, and Future Directions of Human-in-the-loop Learning // IEEE Access. 2024. Vol. 12. pp. 75735-75760.