Оглавление

[*Введение* 0](#_Toc455090648)

[*Обзор существующих технологий* 2](#_Toc455090649)

[Экспертная система 2](#_Toc455090650)

[База знаний ( *k*n*owledge base,KB*) 3](#_Toc455090651)

[СИЭС 4](#_Toc455090652)

[Схема Захмана 5](#_Toc455090653)

[*Постановка задачи* 8](#_Toc455090654)

[Цели 8](#_Toc455090655)

[Данные 11](#_Toc455090656)

[Функции 13](#_Toc455090657)

[Роли 17](#_Toc455090658)

[Размещение 19](#_Toc455090659)

[Процессы 21](#_Toc455090660)

[*Решение задачи* 24](#_Toc455090661)

[Визуализация базы знаний СИЭС 24](#_Toc455090662)

[Средства разработки. 29](#_Toc455090663)

[Выбор технологий 30](#_Toc455090664)

[Визуализация графа 30](#_Toc455090665)

[Работа с базой данных 31](#_Toc455090666)

[Примеры экранов 33](#_Toc455090667)

[*Заключение* 34](#_Toc455090668)

[*Литература* 35](#_Toc455090669)

[*Приложение* 36](#_Toc455090670)

[Index.html 36](#_Toc455090671)

[Main.js 37](#_Toc455090672)

# *Введение*

В настоящее время многие из производителей ориентируются на переход от монолитных приложений к отдельным сервисам, в соответствии с подходом SOA (**service**-**oriented** **architecture**). Существуют специальные платформы класса «промежуточного слоя» (middlware), позволяющие управлять данными сервисами. В итоге идеальная корпоративная информационная система (КИС) представляет собой набор правильно подобранных систем или модулей, каждый из которых наиболее эффективно решает свою задачу.

В общем случае, набор модулей может быть представлен разными производителями, использующими различную терминологию и разные средства автоматизации.

Дополнительная сложность заключается в том, что системы, предлагаемые потенциальными исполнителями, как правило, отличаются не только набором возможностей, но и терминологией, используемой при их описании. Различная терминология приводит к умышленному или неумышленному искажению возможностей предлагаемых систем, что вызывает необходимость использования единого стандартного представления параметров участков и систем.

В отличие от остальных этапов внедрения КИС (диагностика, анализ, дизайн, разработка и тестирование, развертывание, опытная эксплуатация) этап подготовки, на котором осуществляется выбор системы и поставщика, является наименее формализованным. Дело в том, что когда поставщик решения уже выбран, он предлагает свою методику создания КИС, ориентированную на свой продукт и отработанную на большем или меньшем количестве предприятий. На этапе же выбора системы и поставщика заказчик оказывается наедине перед многообразием предлагаемых решений.

Наиболее важными задачами на этапе выбора инструмента создания КИС в целом или ее элемента являются следующие:

- формализация целей проекта автоматизации;

- определение области (участка) автоматизации;

- формирование требований к участку автоматизации;

- выбор подходящей системы или модуля для участка автоматизации.

Наибольший эффект достигается, если все перечисленные задачи будут решены комплексно.

Помощь заказчику КИС могла бы оказать Экспертная Система (ЭС) в области Корпоративных информационных систем.

В данной работе была поставлена задача реализовать прототип одного из модулей СИЭС – модуль визуализации на примере базы знаний о системах автоматизации.

# *Обзор существующих технологий*

## Экспертная система

**Экспертная система** - это интеллектуальная информационная система (ИИС), предназначенная для решения слабо формализуемых задач на основе накапливаемого в базе знаний опыта работы экспертов в проблемной области.

Экспертная система включает базу знаний с набором правил и механизмом вывода и позволяет на основании предоставляемых пользователем фактов распознать ситуацию, поставить диагноз, сформулировать решение или дать рекомендацию для выбора действия.

**Ситуация** - совокупность вопроса, ответа на него и множества рекомендаций.  
Рекомендацией может быть как ссылка на нужный программный продукт, так и ссылка на другую Ситуацию. Схематично это выглядит так:

С = < В , О , {Р} > ,

Где С – Ситуация, В – вопрос, О – ответ, {Р} – множество рекомендаций.

ЭС также способствует систематизации знаний. При помощи ЭС можно установить связи между решениями и рекомендациями, которые обычно хранятся только в памяти человека.

* Экспертные системы предназначены для воссоздания опыта, знаний профессионалов высокого уровня и использования этих знаний, в процессе управления.
* В основе построения экспертных систем лежит база знаний, которая основывается на моделях представления знаний.
* В системах, основанных на знаниях, правила (или эвристики), по которым решаются проблемы в конкретной предметной области, хранятся в базе знаний.
* Проблемы ставятся перед системой в виде совокупности фактов, описывающих некоторую ситуацию, и система с помощью базы знаний пытается вывести заключение из этих фактов.

## База знаний ( *k*n*owledge base,KB*)

- в информатике и исследованиях искусственного интеллекта — это особого рода база данных, разработанная для оперирования знаниями. База знаний содержит структурированную информацию, покрывающую некоторую область знаний, для использования кибернетическим устройством (или человеком) с конкретной целью. Современные базы знаний работают совместно с системами поиска информации, имеют классификационную структуру и формат представления знаний.

Полноценные базы знаний содержат в себе не только фактическую информацию, но и правила вывода, допускающие автоматические умозаключения о вновь вводимых фактах и, как следствие, осмысленную обработку информации.

Область наук об искусственном интеллекте, изучающая базы знаний и методы работы со знаниями, называется инженерией знаний.

Иерархический способ представления в базе знаний набора понятий и их отношений называется онтологией. Онтологию некоторой области знаний вместе со сведениями о свойствах конкретных объектов также можно назвать базой знаний.

Для построения базы знаний требуется:

* + провести опрос специалистов, являющихся экспертами в конкретной предметной области;
  + затем систематизировать, организовать и снабдить эти знания указателями, чтобы впоследствии их можно было легко извлечь из базы знаний.

БЗ содержит элементы:

* Факты (данные) из предметной области ;
* Специальные правила (эвристики), которые управляют использованием фактов при генерации знаний.

## СИЭС

СИЭС представляет собой оболочку экспертных систем, работающих в областях прикладной статистики и проектирования организационно- технических структур. Все знания система хранит в базах знаний, каждая из которых может соответствовать своей проблемной области. БЗ состоит из ситуаций, каждая из которых содержит набор Рекомендаций.

В одной ситуации обычно хранятся взаимосвязанные рекомендации, которые отличаются одним или несколькими условиями использования. Эти условия задаются Вопросами к пользователю, которые вместе с допустимыми ответами могут храниться в ситуации. Вместо рекомендации может быть ссылка на другую ситуацию или же происходить замена ответа на вопрос в другой ситуации. СИЭС накладывает определенные ограничения, обусловленные, главным образом, психологическими возможностями человека.

При использовании (СИЭС) - эксперт сам может классифицировать свои знания и вводить их в систему без помощи инженера по знаниям. В этом заключается основное преимущество СИЭС для работы с областями знаний - эксперт может сам вводить их в систему, не пользуясь при этом услугами инженера по знаниям.

## Схема Захмана

Схема Захмана является наиболее полным архитектурным каркасом и определяет общие свойства информационных систем на том уровне, когда они еще не зависят от парадигмы проектирования, технологии и средств разработки. Она систематизирует знания об архитектуре информационной системы, охватывая все аспекты проектирования за счет использования системы шести универсальных вопросов «Что? Кто? Где? Когда? Как? Почему?».

[](http://reqcenter.pro/wp-content/uploads/2015/08/zachman-rows-01.jpg)

Рисунок 1. Столбцы схемы Захмана

Захман адаптировал эти вопросы к проектированию информационных систем и уточнил срезы проектирования.

Срезы **Что** и **Как** отдаются для описания **Данных** и **Функций** информационной системы — объективному ядру систем этого класса.

**Зачем** и **Кто** — **Мотивация** и **Люди** вынесены вперед, это концентрирует внимание на том, что прежде всего должны быть определены цели создания системы, а потом уже конкретные функции и способы их реализации. **Люди** — это и заказчики системы, чьи цели должны быть достигнуты, и пользователи, выполняющие функции. Именно для них создается система и под их цели проектируются структуры данных и функций.

Следующие срезы — это **Место** и **Время,** соответствующие вопросам **Где** и **Когда**. Анализ информационной системы по этим срезам позволяет сформулировать [нефункциональные требования](http://nonfunctional-requirements/), без учета которых можно попасть в просак и не достигнуть поставленных целей.

Наши цели всегда привязаны к конкретному месту и времени и не имеют смысла без них. Чтобы это лучше запомнить, просто замкните схему Захмана вот так:

[](http://reqcenter.pro/wp-content/uploads/2015/08/zachman-circle2-01.jpg)

Рисунок 2. Свертка схемы Захмана

Также в схеме приводятся, но не навязываются, типы моделей для описания каждого среза системы.

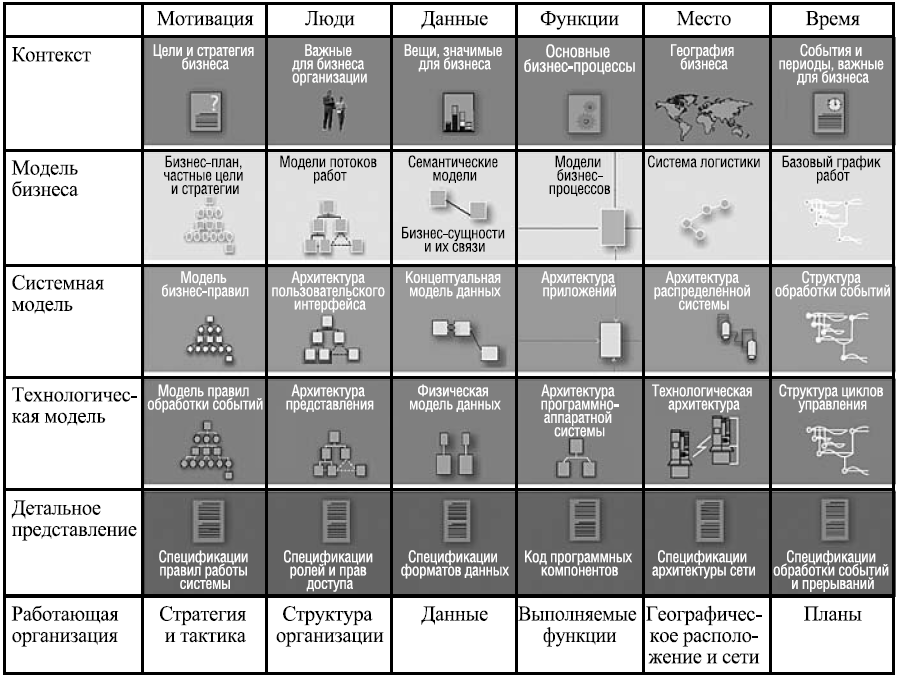
[](http://edu.reqcenter.pro/wp-content/uploads/2013/03/zachman.gif)

Рисунок 4. Расширенная схема Захмана в русском адаптированном переводе

# *Постановка задачи*

* Изучение структуры базы знаний СИЭС на основе текста диплома и дополнительных материалов.
* развитие модулей получения и извлечения знаний в части визуализации знаний на сайте.

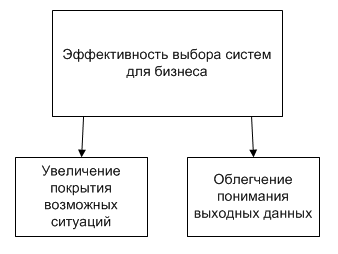
Опишем постановку задачи более подробно с точки зрения Целей, Данных, Функций, Ролей, Размещения, Процессов.

## Цели

1. ***План***

Реализовать сервис для предоставления экспертных мнений клиентам

1. ***Концепция***



1. ***Логическая модель***

Задачи:

* провести обзор средств моделирования знаний в предметной области вопросно-ответных экспертных систем;
* разработать модель представления знаний с учетом поставленных требований;
* разработать архитектуру программного комплекса, для работы с базой знаний и обеспечения возможности визуализации базы знаний;
* реализовать модули экспертной системы для получения знаний пользователем и извлечения знаний у эксперта;
* заполнить базу знаний СИЭС на основе материалов курса «Корпоративные информационные системы».

Цели:

Традиционно наиболее важная цель

– увеличение финансовых показателей, прибыли.

Данная цель является и наиболее легко измеримой целью благодаря наличию множества разработанных методик. В соответствии с методологией BalancedScoreCard в качестве основных целей предприятия выявлены еще три направления:

- улучшение внутренних бизнес-процессов;

- улучшение отношения сотрудников, их обучение, и рост организации;

-улучшение положения на рынке (маркетинговые показатели, отношение клиентов).



Рис. Пример дерева целей

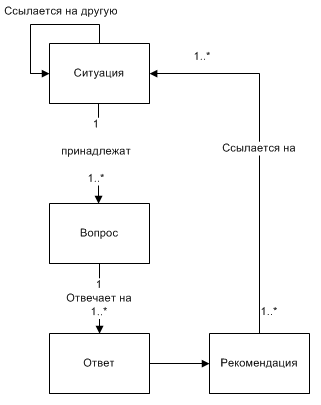
## Данные

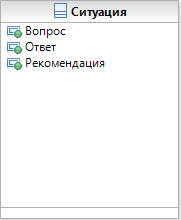
1. ***План***

* Ситуации выбора, содержащие вопросы, ответы и рекомендации по выбору систем автоматизации
* Модели целей, функций, данных участков автоматизации с описаниями требований к системам автоматизации

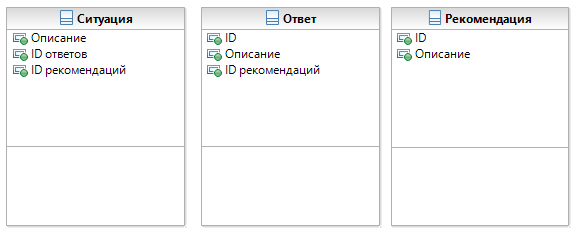
1. ***Концепция***

1)

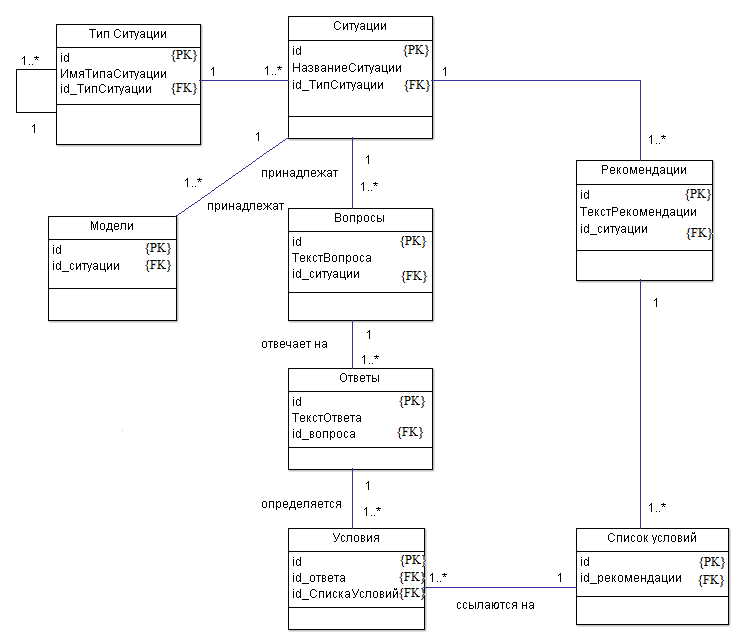
2) 



1. ***Логическая модель***



структура БД

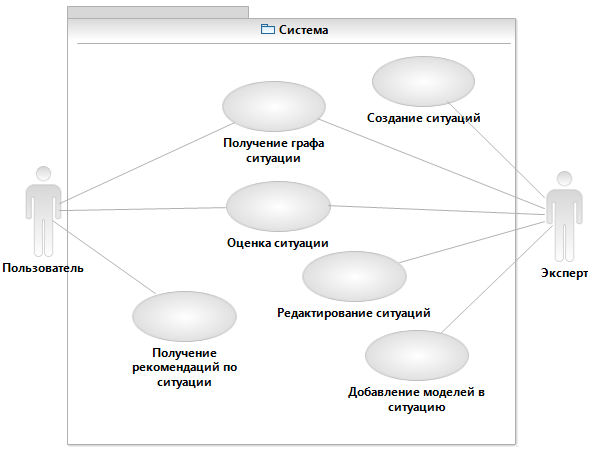


## Функции

1. ***План***

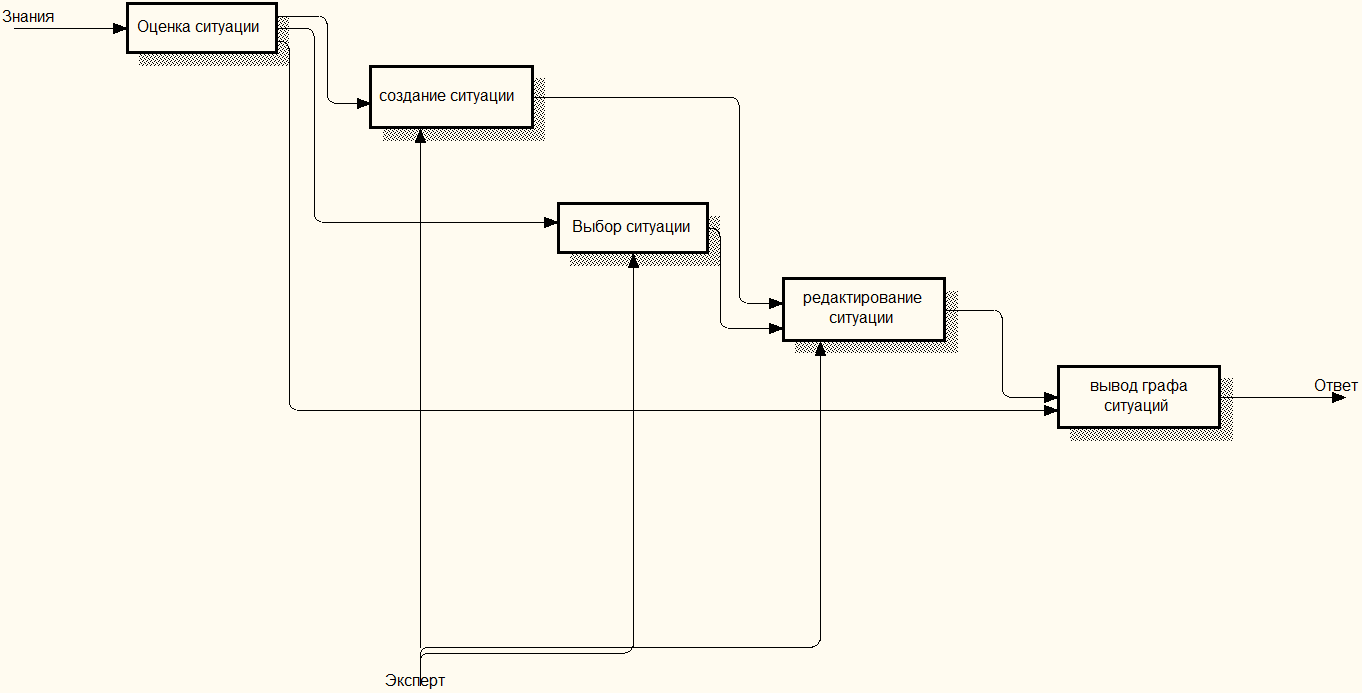
* создание ситуаций
* добавление правил выбора в ситуацию
* добавление моделей в ситуацию
* количественная оценка объема, полноты, противоречивости ситуации
* выдача информации на сайте в виде графа ситуаций
* получение рекомендаций о выборе системы автоматизации по конкретной ситуации

1. ***Концепция***

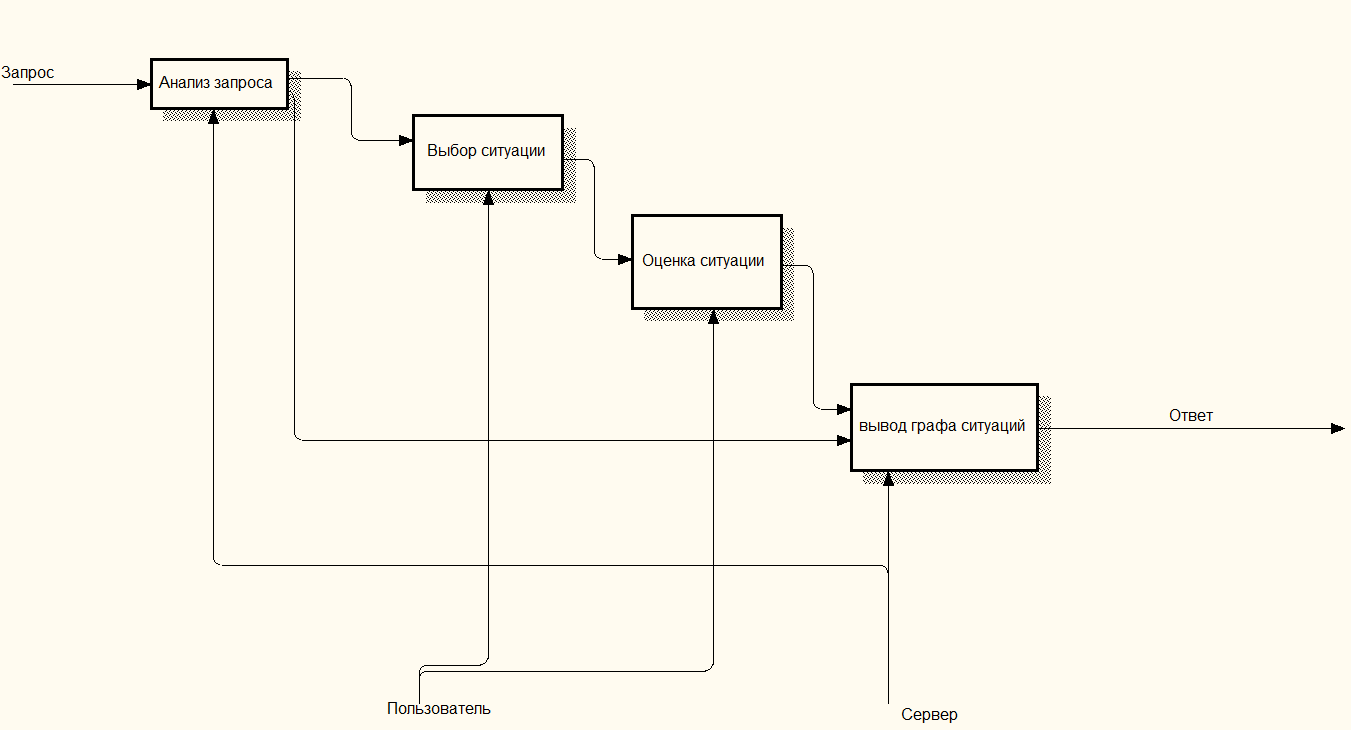


1. ***Логическая модель***

Эксперт



Пользователь



Список функций

* Навигация по разделам с помощью меню
* поиск необходимой ситуации графически
* навигация по дереву ситуаций
* поиск необходимой ситуации вопросно-ответной формой
* скачивание полного описания ситуации
* сравнение ситуаций
* получение графического коллажа самых часто используемых слов в описании ситуации
* авторизация
* добавление новых ситуаций
* пополнение описания существующих ситуаций

Разделы меню:

Выбор:

– БАЗА ЗНАНИЙ;

– СИТУАЦИЯ:

            - Список;

            - Графика;

Модификация:

– СИТУАЦИЯ:

– Создание;

– Удаление;

– Переименование;

– Содержание:

– ВОПРОСЫ;

– ОТВЕТЫ;

– ДЕЙСТВИЯ;

- МОДЕЛИ.

Выполнение;

Выявление:

– Дополнение:

– Действие;

– Вопрос;

– Оценка;

## Роли

1. ***План***

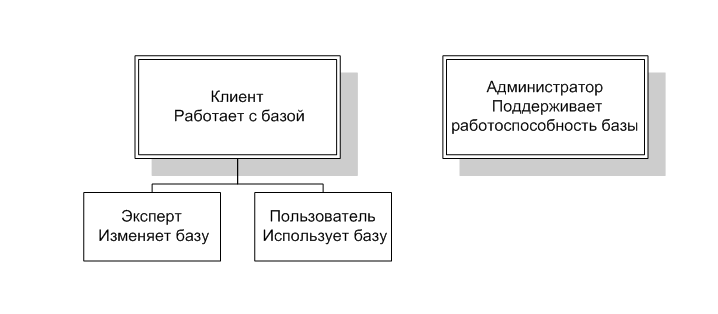
• Администраторы – люди, следящие за работоспособностью системы и занимающиеся её настройкой

• Клиенты – пользователи и эксперты системы;

• Эксперты – клиенты, изменяющие базу знаний СИЭС

• Пользователи – клиенты, получающие рекомендации от СИЭС

1. ***Концепция***



1. ***Логическая модель***

В роли Администратора должен выступать технический специалист, знакомыйсо структурой сайта.  
В роли Пользователей будут выступать предприниматели, желающие использовать необходимый в их случае набор программ для бизнеса.

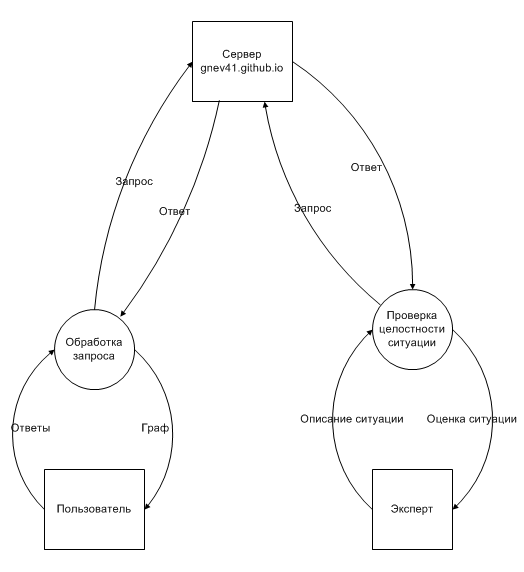
В роли Экспертов будут выступать люди, умеющие грамотно управлять бизнесом в разных ситуациях.

## Размещение

1. ***План***

* веб-сервер и привязанная к нему база знаний должна располагаться на выделенном сервере(нынешнее расположение - [gnev41.github.io](http://gnev41.github.io))
* Клиент (пользователь или эксперт) может подключаться с любого браузера из сети интернет

1. ***Концепция***



1. ***Логическая модель***

Клиенты будут использовать систему пользуясь своими браузерами дома,на работе и т.д., т.е. там, где им удобно.

Сама база знаний будет находится на выделенном сервере с личным доменом.

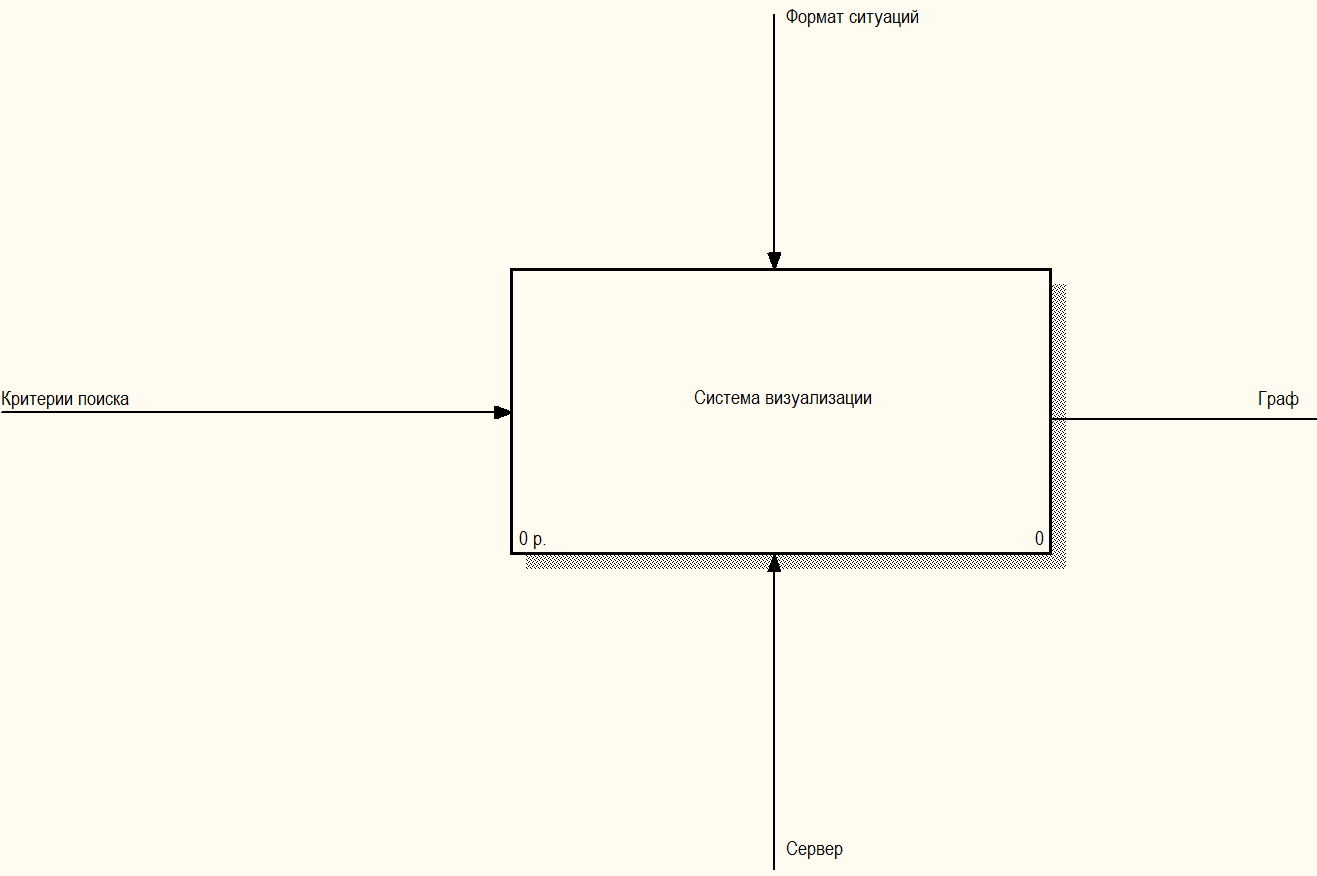
(нынешнее расположение - <gnev41.github.io>)

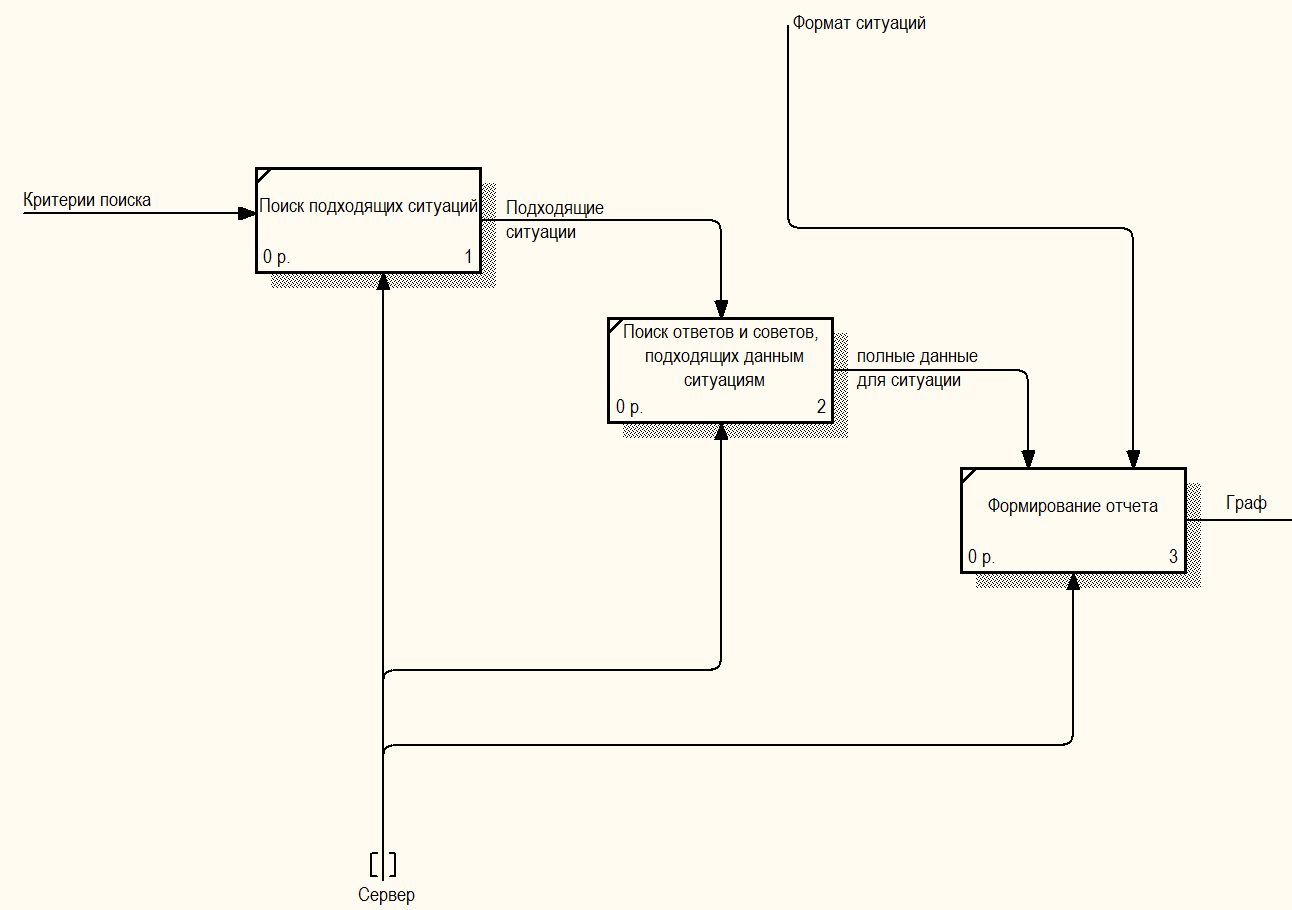
## Процессы

1. ***План***

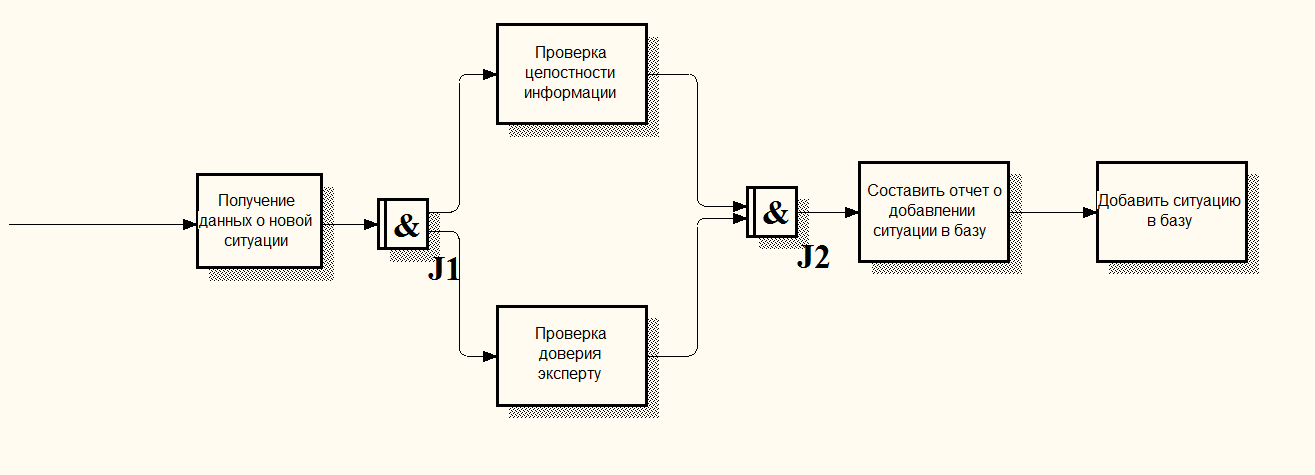
* Демонстрация графа ситуаций:
  + Показ графа целиком
  + Показ ситуаций, связанных с выбранной ситуацией
  + Показ моделей, связанных с ситуацией
* Получение рекомендации:
  + Выбрана ситуация пользователем
  + Система задала вопрос
  + Пользователь выбрал ответ
  + Система обработала ответ, в результате, выполнено одно из действий:
    - Задан новый вопрос
    - Выполнен переход к новой ситуации
    - Выдана рекомендация
* Пополнение ситуации:
  + Создана ситуация экспертом
  + Выбрана ситуация экспертом
  + Эксперт добавил рекомендацию
  + Эксперт добавил вопрос
  + Система проверила ситуацию на формальную целостность, выполнен запрос дополнительной информации

1. ***Концепция***

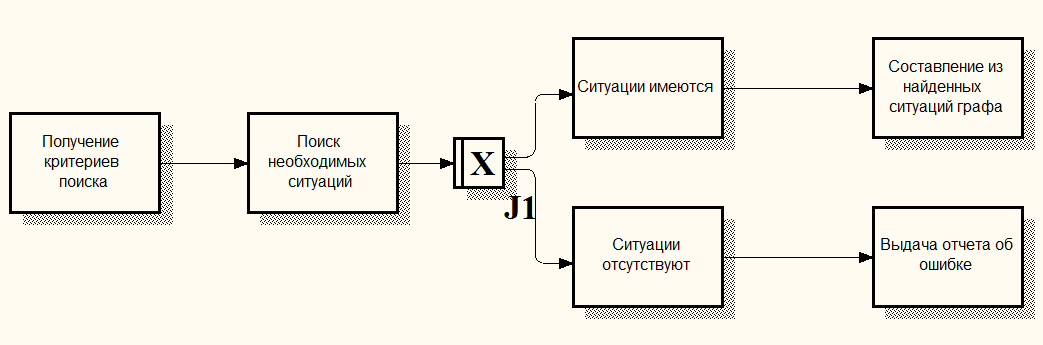




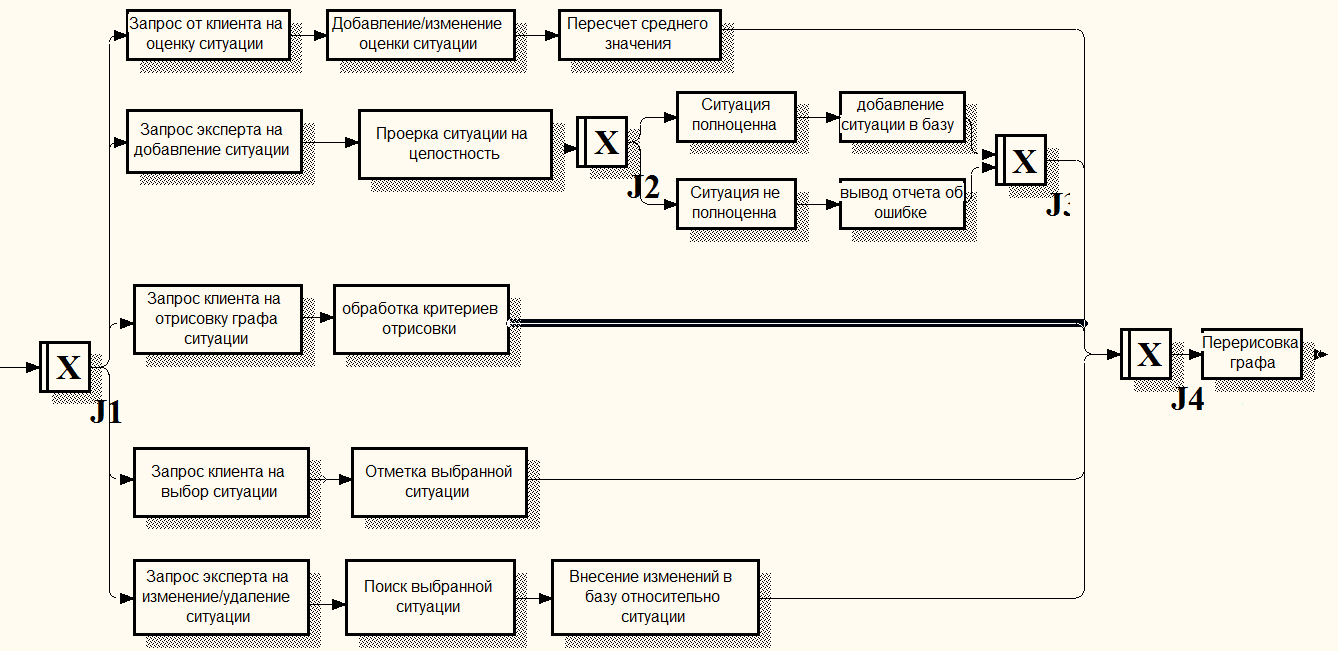
**Эксперт**



**Пользователь**



1. ***Логическая модель***



# *Решение задачи*

## Визуализация базы знаний СИЭС

Как следует из представленной модели базы знаний, в общем случае она представляет собой ориентированный граф, вершинами которого являются ситуации, а ребрами – операции, которые представляют собой ссылки на другие ситуации.

Рассмотрим возможные объекты визуализации СИЭС. Среди них:

- общая база ситуаций;

- набор ситуаций и связей между ними, объединенный общим разделом или головной ситуацией;

- отдельная ситуация, включающая в качестве своих элементов вопросы, ответы, рекомендации, модели;

В общем случае набор ситуаций СИЭС можно изобразить в виде обычной иерархии, раскрывая которую можно быстро попасть к нужной ситуации (Рисунок 2).



*Пример графического изображения набора ситуаций в виде графа*

Альтернативный способ представления дерева ситуаций - в виде вложенных кругов, каждый из которых соответствует одной ситуации. При этом если из одной ситуации вызывается несколько других, вызываемые ситуации могут быть изображены в виде кругов, вложенных в головную ситуацию (Рисунок 2).



*Пример графического изображения набора ситуаций в виде вложенных кругов*

Данный вариант представляется более компактным, поэтому далее будет рассматриваться в качестве основного. Воспользуемся возможностью дать наглядное представление об отдельной ситуации. Если использовать представление ситуаций в виде кругов, то размер круга, толщина стенок, его цвет могут следующим образом соответствовать качественным и количественным характеристикам ситуаций:

- качественные: формальная полнота и непротиворечивость, в соответствии с цветами спектра (Рисунок 3):

- синий: полная и непротиворечивая;

- зеленый: полная, но противоречивая;

- желтый: неполная, но непротиворечивая;

- красный: неполная и противоречивая;

- количественные: количество условий/ вопросов/ полей/ рекомендаций/ моделей/ ссылок на ситуации, с помощью следующих графических элементов:

- размер круга по сравнению с соседними;

- толщина стенок.



*Пример графического изображения набора ситуаций с использованием цвета*

В этом случае модель ситуации имеет вид:

С = < {В}, {O}, {Р} , {Пок}>

Здесь Пок - Показатели

{Пок} = < {Поккачества}, {Покколичества}>

Где Поккачества = {Полнота, Непротиворечивость},

Покколичесива = {Количество правил, Количество вопросов, Количество рекомендаций, Количество связей, …}

Поскольку графических возможностей существенно меньше, чем характеристик, предлагается дать пользователю или эксперту настроить возможность выбора наиболее важных характеристик для графического отображения.

Кроме того, в целях экономии пространства изображения, предлагается ограничиться размером круга в качестве количественного показателя (Рисунок 4).



*Пример графического изображения набора ситуаций с использованием цвета и размера*

## Средства разработки.

1)В реализации программы использовался язык программирования JavaScript.

[прототипно-ориентированный](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D1%82%D0%B8%D0%BF%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [сценарный](https://ru.wikipedia.org/wiki/%D0%A1%D1%86%D0%B5%D0%BD%D0%B0%D1%80%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). Является реализацией языка [ECMAScript](https://ru.wikipedia.org/wiki/ECMAScript) .

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам [приложений](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0). Наиболее широкое применение находит в [браузерах](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) как язык сценариев для придания [интерактивности](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D1%8C) [веб-страницам](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0).

Основные архитектурные черты: [динамическая типизация](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), [слабая типизация](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D0%B0%D0%B1%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), [автоматическое управление памятью](https://ru.wikipedia.org/wiki/%D0%A1%D0%B1%D0%BE%D1%80%D0%BA%D0%B0_%D0%BC%D1%83%D1%81%D0%BE%D1%80%D0%B0_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29), [прототипное программирование](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D1%82%D0%B8%D0%BF%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), функции как [объекты первого класса](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_%D0%BF%D0%B5%D1%80%D0%B2%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0).

Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

2)Среда разработки – Sublime Text.

[кроссплатформенный](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D1%81%D1%81%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) [проприетарный](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BF%D1%80%D0%B8%D0%B5%D1%82%D0%B0%D1%80%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) [текстовый редактор](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%80%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80). Поддерживает [плагины](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%B3%D0%B8%D0%BD) на языке программирования [Python](https://ru.wikipedia.org/wiki/Python).

Программа часто используется как [редактор](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80_%D0%B8%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BE%D0%B4%D0%B0) [исходного кода](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4).

*3)Model-view-controller*(MVC, «модель-представление-поведение», «модель-представление-контроллер», «модель-вид-контроллер») — схема использования нескольких [шаблонов проектирования](https://ru.wikipedia.org/wiki/%D0%A8%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), с помощью которых [модель данных](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) приложения, [пользовательский интерфейс](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Данная схема проектирования часто используется для построения [архитектурного каркаса](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%80%D0%BA%D0%B0%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), когда переходят от теории к реализации в конкретной предметной области

## Выбор технологий

### Визуализация графа

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Критерии*** | ***D3.js*** | | ***Arbor.js*** | |
|  | ***Оценка*** | ***Вес*** | ***Оценка*** | ***Вес*** |
| Простота установки/настройки | 10 | 5 | 10 | 5 |
| Количество функций | 10 | 10 | 6 | 10 |
| Качество документации | 7 | 7 | 7 | 7 |
| Скорость работы | 10 | 10 | 9 | 10 |
| Итого: | 299 |  | 249 |  |

Установка обеих библиотек происходила идентичными способами- либо указанием прямой ссылки на файл в интернете *(<script src=https://d3js.org/d3.v3.min.js ></script>*), либо скачиванием необходимых файлов с официального сайта и указанием ссылки на него (*<script src="js/arbor.js"></script>* ).

Арбор является удобной библиотекой для работы с графами, легким к пониманию и широким набором функций для работы с ними. Простейший пример можно сделать буквально за пару минут.

При сравнении c D3, Арбор имеет более узконаправленный направленность функций т.к. создан для работы только с графами, в то время как D3 призван для обработки и визуализации данных. Функций в D3 более чем достаточно: работа с 3D графикой, работа с графами, построение таблиц, различных динамических диаграмм, деревьев и даже карт. Высокая скорость работы в D3 достигается модульностью своих частей, т.е. при желании нарисовать простой кружочек, вы не будете подгружать также функции для рисовки 3D моделей и кривых Безье.

Выбран был D3.JS ввиду большего количества доступных функций.

### Работа с базой данных

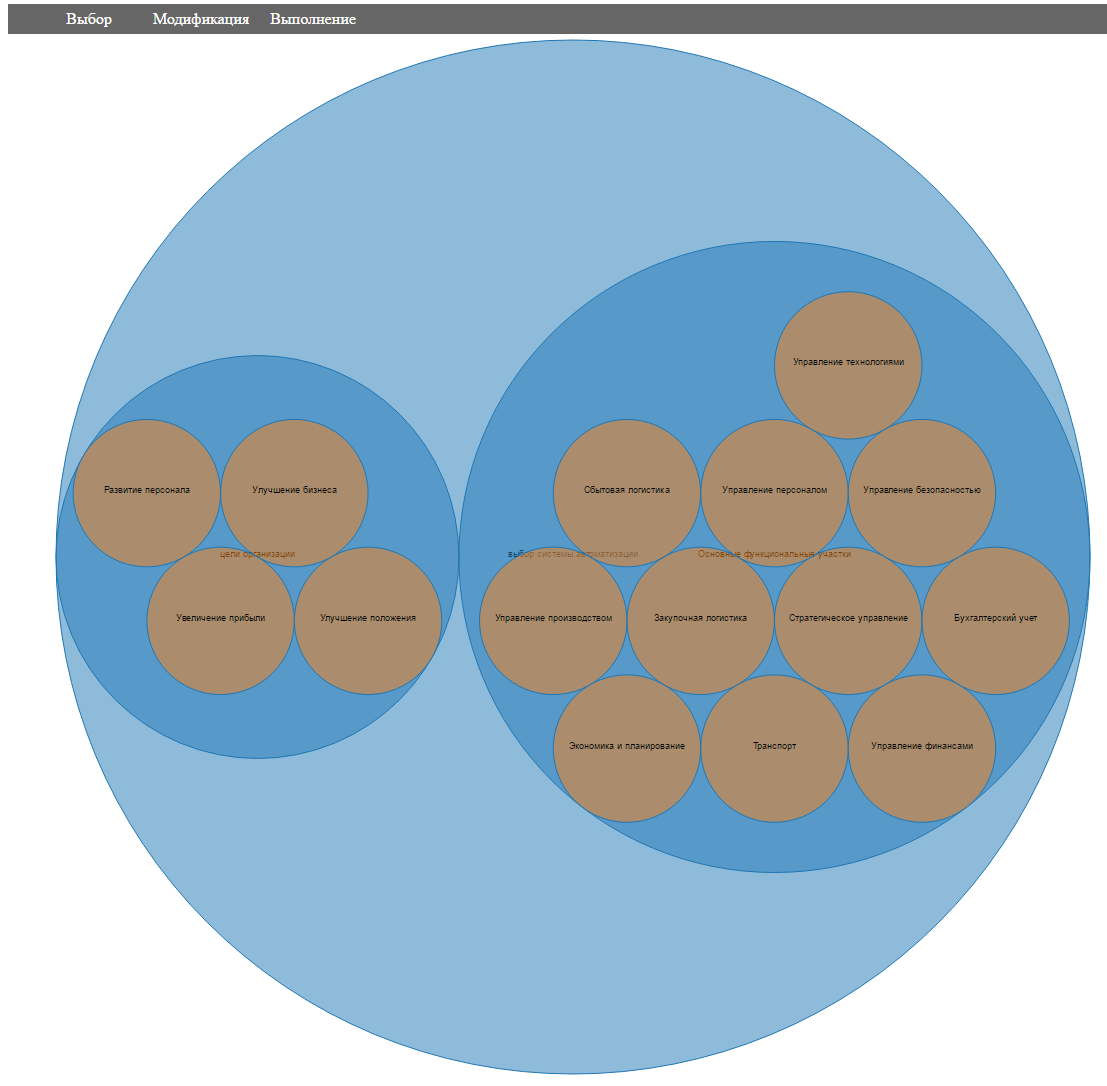
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Критерии*** | ***mongoose.js*** | | ***Ajax+php+MySql*** | |
|  | ***Оценка*** | ***Вес*** | ***Оценка*** | ***Вес*** |
| Простота установки/настройки | 3 | 7 | 10 | 7 |
| Количество функций | 10 | 10 | 10 | 10 |
| Качество документации | 7 | 7 | 6 | 7 |
| Скорость работы | 10 | 10 | 10 | 10 |
| Нагрузоустойчивость | 7 | 10 | 10 | 10 |
| Итого: | 340 |  | 412 |  |

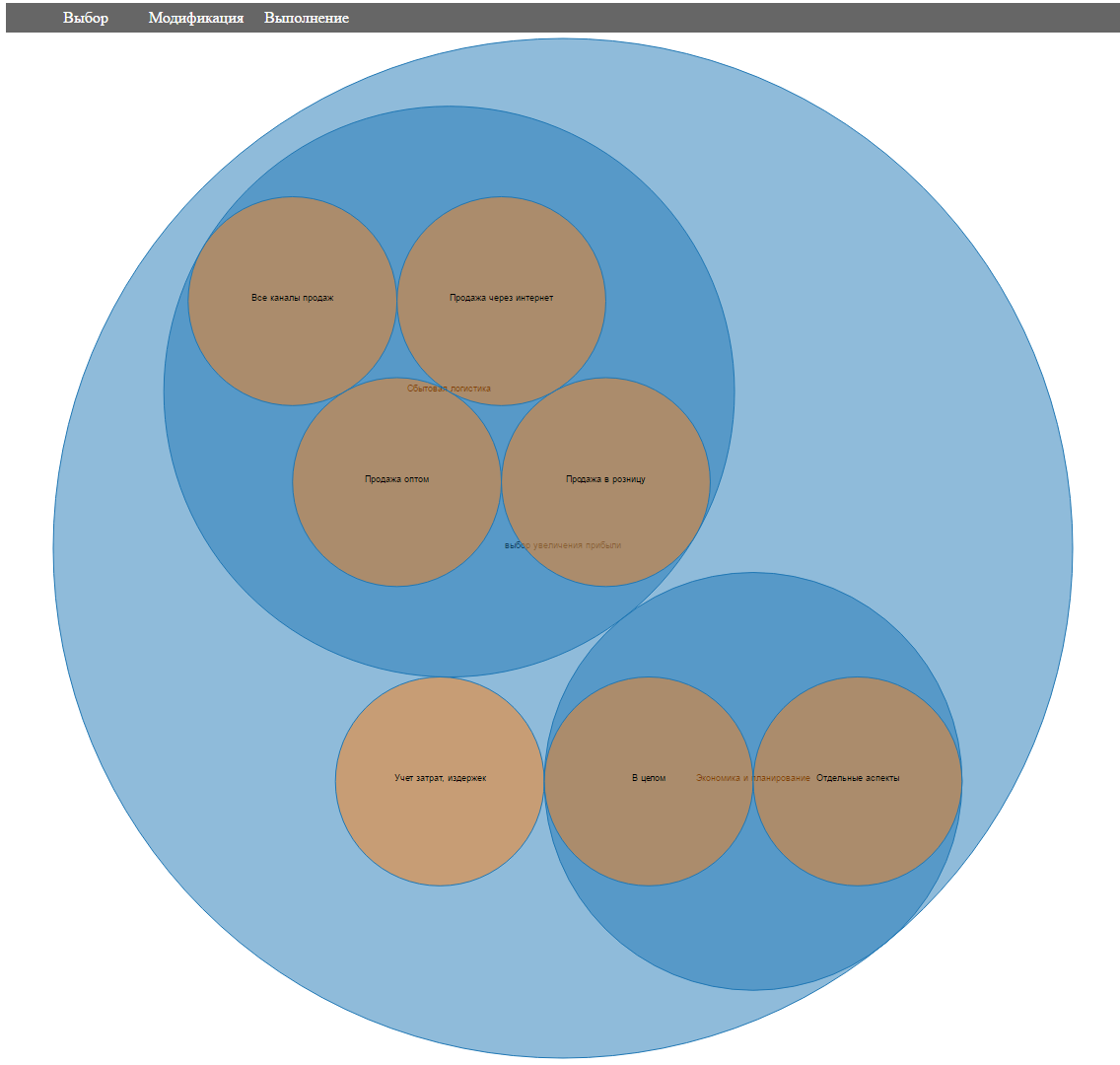
Установка Mongoose происходит через встроенный менеджер пакетов библиотеки Node.js, которую также необходимо установить на сервер установщиком, как обычное ПО. Для использования связки AJAX+PHP+MySQL необходимо развернуть лишь MySQL сервер, а PHP и AJAX будут поддерживаться Веб-сервером, который уже стоит.  
Mongoose – библиотека для работы с базой данных MongoDB через JavaScript. Безопасность обеспечивается плохая т.к. приходится в коде в открытую указывать логин/пароль для входа в базу, а код JavaScript может увидеть любой пользователь через свой браузер. MongoDB является NoSQL базой и потому теряет скорость работы при больших объёмах данных.

Эти недостатки решены при использовании связки – логин/пароль указывается лишь в коде php, которому пользователь не может так легко получить доступ, а MySQL является нагрузоустойчивым типом базы данных. Слабостью этого метода является то, что необходимо учитывать особенности всех 3 использованных технологий и есть возможность ошибиться в 3 разных местах.

Выбрана была схема Ajax+php+MySql ввиду повышенной нагрузоустойчивости и большей информационной безопасности.

## Примеры экранов





# *Заключение*

В результате проделанной работы были изучены вопросы создания ЭС, основные проблемы, возникающие при создании ЭС.

Были изучены основные принципы построения ЭС и основные выполняемые ЭС функции.

Был изучен принцип разбиения системы по схеме Захмана.

Была изучена с позиции пользователя система СИЭС, принципы ее работы.

Разработан проект по реализации системы СИЭС с использованием методики Захмана: построены уровни планирования, концепции, логический для доменов Цели, Данные, Функции, Роли, Размещение, Процессы.

Создан макет базы знаний из 22 ситуаций. Реализован макет модуля визуализации базы знаний, доступный на сайте [gnev41.github.io](http://gnev41.github.io)

# *Литература*

1. .Г.В. Рыбина “Основы построения интеллектуальных систем”.
2. Дэвид Флэнаган ” JavaScript: Подробное руководство”
3. Дзенгелевский А.Е. “Математическое и программное обеспечение ситуационной инструментальной экспертной системы”
4. Описание схемы Захмана ( <http://reqcenter.pro/zachman-framework/> )

# *Приложение*

## Index.html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta charset="utf-8">

</head>

<link rel="stylesheet" type="text/css" href="style.css" media="all" />

<body>

<ul id="navbar">

<li><a class="radius">Выполнение</a></li>

<li><a >Выбор</a>

<ul>

<li><a >База Знаний</a></li>

<li><a >Ситуация</a></li>

</ul>

</li>

<li><a >Модификация ситуации</a></li>

<li><a >Выполнение</a></li>

</ul>

<table id="question">

<div id="main"></div>

</table>

<script src="../libs/d3/d3.min.js"></script>

<!--эта часть здесь для смертного меня, когда у меня не будет выхода в интернет, а делать уир будет хотеться -->

<script src="https://d3js.org/d3.v3.min.js" charset="utf-8"></script>

<script src="data.json"> </script>

<script src="main.js"></script>

</body>

</html>

## Main.js

var diameter = (window.innerWidth < window.innerHeight ? window.innerWidth : window.innerHeight)-50,

format = d3.format(",d");//описание функции приведения числа в нужный вид 3008 -> 3,008

var pack = d3.layout.pack();

var svg = d3.select("#main").append("svg");

var node ;

var data = dat;

var lastX,lastY;

function draw3DTextCircleCTX(s, x, y, radius, iSAngle){

// Радиан на символ

var fRadPerLetter = (Math.PI / s.length)-0.07;

// Сохраняем контекст, переводим и вращаем его

ctx.save();

ctx.translate(x,y);

ctx.rotate(iSAngle);

// Количество дополнительных нижних слоев

var iDepth = 4;

// Устанавливаем темно-зеленый цвет для дополнительных слоев

ctx.fillStyle = '#168d1e';

// Обрабоатываем каждый символ строки

for (var i=0; i<s.length; i++) {

ctx.save();

ctx.rotate(i\*fRadPerLetter);

// Выводим дополнительные слои

for (var n = 0; n < iDepth; n++) {

ctx.fillText(s[i], n, n - radius);

}

// Параметры тени

ctx.fillStyle = '#00d50f';

ctx.shadowColor = 'black';

ctx.shadowBlur = 10;

ctx.shadowOffsetX = iDepth + 2;

ctx.shadowOffsetY = iDepth + 2;

// выводим символы

ctx.fillText(s[i], 0, -radius);

ctx.restore();

}

ctx.restore();

}

var circle,text,question;

function resize()

{

diameter =( window.innerWidth < window.innerHeight ? window.innerWidth : window.innerHeight)-50;

console.log(diameter);

render(data);

}

window.onresize = resize;

//обновление информации в

function get\_data\_update (name) {

d3.json(name+".json", function(error, json) {

if (error) return console.warn(error);

data=json;

render(json);

});

}

function render(json){

d3.select("#question")

.select("text")

.remove();

/\*

//вставляем текст

question = d3.select("#question")

.select("td")

.append("text")//если проходит фильтр(нет детей т.е. сам ребенок последнего поколения),

.attr("dy", ".3em")// то пишем текст

.style("text-anchor", "middle")

.style("font-size","17px")

.text(function(d) { return json.question })

;

\*/

d3.select("svg").remove();

svg=[];

pack = d3.layout.pack()

.size([diameter - 15, diameter - 15])

.value(function(d) { return d.size; });

svg = d3.select("#main").append("svg")

.attr("width", window.innerWidth)

.attr("height", window.innerHeight)

.append("g")

.attr("transform", "translate(1,1)");

d3.select("svg")

.append("defs")

;

/\*

d3.select("svg")

.append("path")

.attr("d" , getPathData1()//getPathData( d3.select("#main\_circle").datum().x, d3.select("#main\_circle").datum().y,diameter/2)

)

d3.select("svg")

.append("defs")

.append("path")

.attr("d" , getPathData1()//getPathData(d3.select("#main\_circle").datum().x, d3.select("#main\_circle").datum().y,diameter/2)

)

.attr("id", "curvedTextPath")

;

d3.select("svg")

.append("text")

.append("textPath")

.attr("startOffset", "46%")

.attr("xlink:href", "#curvedTextPath")

.text("Hello, world!");

\*/

node = svg.datum(json).selectAll(".node")//добавление всех кругов

.data(pack.nodes)

.enter().append("g")

.attr("class", function(d) { return d.children ? "node" : "leaf node"; })

.attr("transform", function(d) { return "translate(" + d.x + "," + d.y + ")"; })

;

node//добавление всплывающего текста

.filter(function(d) { return d.children; })

.append("title")

.text(function(d) { return d.name })

;

function getPathData(x,y,ra) {

// adjust the radius a little so our text's baseline isn't sitting directly on the circle

var r = ra \* 1;

return 'm' + (x-r) + ',' + (y) + ' ' + //точно верные точки с поправкой на r|ra

'a' + ra + ',' + ra + ' 0 1 1 ' + (2\*r) + ',1 Z';

}

//отрисовка самих кружочков

circle = node

//.filter(function(d) { return !d.children; })

.append("circle")

.attr("style","fill-opacity: 0.5")//прозрачность круга

.attr("r", function(d) { return d.r; })

;

node//отметка большого круга главным

.filter(function(d) { return d.r>(diameter/2 - 10); })

//.select("circle")

.attr("id", function(d) { return "main\_circle"; })

;

node//добавление текста над кругом

.filter(function(d) { return d.children; })

.append("text")

.append("textPath")

//.attr("startOffset", "40%")

.attr("xlink:href",function(d) {

d3.select("defs")

.append("path")

.attr("id",d.name)

.attr("d",getPathData(

d.x,

d.y,

d.r)

);

return "#"+d.name; })

.text(function(d) { return d.name })

;

//текст внутри пузырьков

text = node.filter(function(d) { return !d.children; })

.append("text")//если проходит фильтр(нет детей т.е. сам ребенок последнего поколения),

.style("text-anchor", "middle")

.style("font-size","8px")

.text(function(d) { return d.name})

// d.r / 3 < d.name.length ? d.name.substring(0,d.name.substring(0, d.r / 3).lastIndexOf(" ")) : d.name; })

;

node

.filter(function(d) { return !d.children; })

//////////////////////////////////////////////////////////

.on('click', function (d, i){

//console.log(d3.select("#main\_circle").datum().x+" "+d3.select("#main\_circle").datum().x);

lastX=d3.select(this).datum().x;

lastY=d3.select(this).datum().y;

d3.select(this) // Выберем элемент, на который наведена мышь

.select("circle")

.transition() // Начинаем анимацию

.duration(3000) // Длительность анимации

.attr("transform", function(d) {

return "translate(" + (d3.select("#main\_circle").datum().x-d.x) + "," + (d3.select("#main\_circle").datum().y-d.y) + ")"; })

.attr("r", function(d) { return d3.select("#main\_circle").datum().r ; })

;

console.log("click");

get\_data\_update(d.name);

;

})

//////////////////////////////////////////////////////////

.on('mouseenter', function(d) {

lastX=d3.select(this).datum().x;

lastY=d3.select(this).datum().y;

console.log("mouseenter");

d3.select(this) // Выберем элемент, на который наведена мышь

.select("circle")

.transition() // Начинаем анимацию

.duration(400) // Длительность анимации

.attr('r', function(d) { return d.r+3;})

;

})

//////////////////////////////////////////////////////////

.on('mouseleave', function(d) {

console.log("mouseleave");

d3.select(this)

.select("circle")

.transition()

.duration(100)

// Возвращаем в начальную позицию

.attr('r', function(d) { return d.r;})

.attr("transform", function(d) { return "translate(" + (lastX- d.x) + "," + (lastY-d.y) + ")"; })

});

}

render(data);